**Title**
Some Contributions to High-dimensional Statistical Machine Learning



**Author**
Wei, Zhenyu

**Publication Date**
2021

Peer reviewed|Thesis/dissertation

**Some Contributions to High-dimensional Statistical Machine Learning**

By

ZHENYU WEI

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

---

Thomas C.M. Lee, Chair

---

Alexander Aue

---

Can Le

Committee in Charge

2021

To my parents and Xinyi for their endless love, support and encouragement.

# Contents

Some Contributions to High-dimensional Statistical Machine Learning

**Abstract**

This dissertation makes contributions to the broad area of high-dimensional statistical machine learning. When the number of features $p$ is much larger than the number of observations $n$, often written as $p \gg n$, the problems are known as high-dimensional problems. With the rapid growth of the dimensionality and complexity of the data, such problems have become increasingly common and important, for example, in genomics and computational biology. This dissertation focuses on two such high-dimensional problems and develops solutions for them. The first problem concerns uncertainty quantification for the multi-task learning problem. Using the generalized fiducial inference framework, a novel method termed GMTask is developed. This method is shown to enjoy desirable theoretical and empirical properties. The second problem studies variable selection, robust estimation, and nonparametric additive model fitting in the high-dimensional scenario. The minimum description length principle is employed as one unified approach to simultaneous solve these issues.

**Acknowledgments**

First and foremost, I would like to express my sincere appreciation to my advisor, Professor Thomas C.M. Lee, for his continuous support and guidance during my Ph.D. study over the past four years. With his help, I have completed the transition from a fresh graduate student who knew little about academic research to an independent researcher who can have a basic sense of conducting the research. Thomas gave me lots of valuable suggestions on both my research and life, so he is not only my academic advisor, but also my life mentor. I really enjoy working with Thomas, and we can communicate in both Cantonese and English. I could not imagine that I could finally speak fluent Cantonese when I am going to graduate, thanks to the professional bilingual working environment.

Besides my advisors, I would like to thank Professor Alexander Aue, Professor Can Le, Professor James Sharpnack, and Professor Miles Lopes for serving on my PhD dissertation defense committee and providing useful constructive comments on my thesis. I am also grateful to Professor Raymond Wong for his insightful input, collaboration and discussion on my research projects and paper revisions. I would also like to thank Professor Cho-Jui Hsieh, Professor Shiqian Ma, and Professor Rahman Azari for serving on my qualifying exam committee and their encouragement in the early stages of my research.

Many sincere thanks should be given to our department staff Pete Scully, Cristeta Rillera, Sara Driver, Olga Rodriguez, Kimberly McMullen and Nehad Ismail, for providing numerous resources from tea time snacks to research facilities. I really miss the annual new year party hosted by our department which made me feel at home.

I would like to thank my friends and fellow classmates at the Department of Statistics, it is my unforgettable memories to fight with you for many deadlines, work with you to discuss the course projects, and gather with you at leisure during the last five years.

Last but not the least, I would like to extend my deepest gratitude to my parents for their unconditional love and support. Thank Xinyi for consistently encouraging and accompanying me throughout these years. I would like to dedicate this dissertation to my family and make them proud.

CHAPTER 1

# Introduction

Nowadays, with the rapid development in information and data collection technology, the dimension and size of the data are increasing at a rapid rate, which brings many exciting and challenging problems in various areas such as finance, e-commerce, social network, and biotech. For example, gene expression or fMRI studies typically include tens of thousands of features for only a small number of observations. Traditional methods used for small data sets are not suited for analyzing such complex big data, which often requires new techniques to approach the problems. Therefore, this dissertation investigates some of these problems and proposes methods to discover the useful patterns and information behind the data. Topics considered by this dissertation include multi-task learning, variable selection, robust fitting, and statistical inference in the high-dimensional setting with extension to nonparametric additive models.

Over the past decades, the Multi-Task Learning (MTL) problem has attracted much attention in the artificial intelligence and machine learning communities. However, most published work in this area focuses on point estimation; i.e., estimating model parameters and/or making predictions. Chapter 2 in this dissertation studies the MTL problem from another important aspect: uncertainty quantification for model choices and predictions. To be more specific, this chapter develops a novel method for deriving a probability density function on the space of all potential models. With this density function, point estimates as well as confidence and prediction ellipsoids can be obtained for quantities of interest, such as future observations. The proposed method, termed GMTask, is based on the generalized fiducial inference (GFI) framework, a modern variant of Fisher's original fiducial inference idea, and is shown to enjoy desirable theoretical properties. Its promising empirical properties are illustrated via a sequence of numerical experiments and applications to two real data sets.

Chapter 3 considers the application of the minimum description length (MDL) principle, a popular tool for choosing model complexity in the signal processing and statistics communities,

to some high-dimensional statistical learning problems. Previous success applications of MDL include signal denoising and variable selection in linear regression, for which the corresponding MDL solutions often enjoy consistent properties and produce very promising empirical results. This chapter further extends the use of MDL to the high-dimensional setting, where the number of predictors $p$ is larger than the number of observations $n$. It first considers the case of multiple linear regression, then allows for outliers in the data, and lastly extends to the robust fitting of nonparametric additive models. To the best of our knowledge, this is the first time that outlier-resistant estimation and variable selection for high-dimensional nonparametric additive models is considered. Results from numerical experiments are presented to demonstrate the efficiency and effectiveness of the MDL approach.

CHAPTER 2

# Uncertainty Quantification for High-dimensional Multi-Task Learning

## 2.1. Introduction

Multi-task learning (MTL) (Caruana, 1997), a sub-field of machine learning, aims to jointly learn multiple related tasks with the hope to improve the learning for each individual task. In the past two decades, many approaches have been developed to exploit the common structure shared amongst the tasks. One popular approach is to model the problem with multivariate regression and estimate the parameters through regularization. For example, Liu *et al.* (2014) propose a calibrated multivariate regression method and Obozinski *et al.* (2006) extend the least absolute shrinkage and selection operator (LASSO) method of Tibshirani (1996) to the multi-task $L_1/L_2$ LASSO. Evgeniou and Pontil (2007) make a joint sparsity assumption that leads to a feature selection problem, and Yuan *et al.* (2007) assume a low rank structure for the parameter matrix and estimate it with a convex relaxation criterion. Lastly, Seneviratne and Solo (2012) develop a cyclic descent algorithm for vector $L_0$ penalized least squares.

Another approach is to use deep neural networks, where the goal is to learn feature representations by using linear or nonlinear transformations of the original features. For example, Zhang *et al.* (2014) learn common feature representations among different tasks by sharing the first multiple layers, and followed by several task specific layers. Misra *et al.* (2016) start out with two separate identical network architectures for two tasks, then use what they refer to as a cross-stitch operation to learn related feature representation for different tasks. In contrast to the multivariate regression approach, where the same data are shared by all the tasks, in deep learning the features of different tasks could come from different data sets.

This dissertation follows the multivariate regression framework, which can be described by the following model:

$$\boldsymbol{Y}_{n\times m} = \boldsymbol{X}_{n\times p}\boldsymbol{B}_{p\times m} + \boldsymbol{E}_{n\times m}, \tag{2.1}$$

where $\boldsymbol{Y}_{n\times m} = (\boldsymbol{Y}_{(1)}, \ldots, \boldsymbol{Y}_{(m)})$ is the response matrix, $\boldsymbol{X}_{n\times p} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ is the design matrix, $\boldsymbol{B}_{p\times m} = (\boldsymbol{\beta}_{(1)}, \ldots, \boldsymbol{\beta}_{(m)}) = (\beta_{ij})$ is the regression coefficient parameter matrix, and $\boldsymbol{E}_{n\times m} = (\boldsymbol{\varepsilon}_{(1)}, \ldots, \boldsymbol{\varepsilon}_{(m)}) = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_n)^T$ is the noise matrix. It is assumed that $\boldsymbol{\varepsilon}_i$ is iid from $N_m(\boldsymbol{0}, \boldsymbol{\Sigma})$ with unknown $\boldsymbol{\Sigma}$, and that $\boldsymbol{E}$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are independent. This dissertation will study the case when $p \gg n$, the so-called high-dimensional scenario. Under this situation, it often assumes that all the regression tasks share a common sparsity pattern; that is, many rows of $\boldsymbol{B}$ are zero vectors.

Up-to-date, most existing work for the MTL learning problem using multivariate regression focuses on the issues of selecting a model, estimating parameters of the selected model, and making predictions from the selected model. In other words, the major focus has been point estimation. This dissertation looks at the MTL problem from a different angle: it examines the issue of uncertainty quantification. More specifically, this dissertation applies the generalized fiducial inference methodology of Hannig *et al.* (2016) to perform statistical inference for the MTL problem. In addition to providing point estimates for quantities of interest, the new method also offers various uncertainty measures, such as prediction ellipsoids for future observations. To the best of our knowledge, this dissertation is one of the earliest that systematically addresses uncertainty quantification for the MTL problem when $p \gg n$.

Equation (2.1) can be represent as an ensemble of univariate linear regression model

$$\boldsymbol{Y}_{(t)} = \boldsymbol{X}\boldsymbol{\beta}_{(t)} + \boldsymbol{\varepsilon}_{(t)}, \quad t = 1, \ldots, m.$$

It is also called single-task learning problem, which trains a regression model for each task separately. However, it does not utilize the information that responses are related between different tasks. Therefore, compared with multi-task learning, multiple univariate tasks perform poorly for uncertainty quantification when the correlation between tasks or the number of tasks $m$ are relatively large. We provide more details in Section 2.5.4.

The rest of this chapter is organized as follows. Section 2.2 provides background on the generalized fiducial inference methodology. Then in Section 2.3 this methodology is applied to the MTL problem to develop the proposed method (GMTask) for uncertainty quantification, and Section 2.4 examines the theoretical properties of GMTask. Empirical performance of GMTask is illustrated via numerical experiments and applications to two real data examples in Section 2.5 and 2.6, respectively. Finally, concluding remarks are offered in Section 2.7 and technical details are delayed to the appendix.

## 2.2. Background of Generalized Fiducial Inference

The original fiducial inference idea was first introduced by Fisher in 1930's (Fisher, 1930) with the goal to assign an appropriate statistical distribution on the parameter space when there is no prior information and hence the classical Bayes' theorem is not applicable. For readers who are interested about the history of fiducial inference, please refer to Hannig and Lee (2009) and Hannig *et al.* (2016).

In the recent two decades, there have been lots of efforts devoted to reformulating the fiducial concepts. Some modern modifications include Dempster-Shafer theory (Dempster, 2008) and its related work (Zhang and Liu, 2011; Martin and Liu, 2013), confidence distributions (Xie and Singh, 2013), and generalized inference (Weerahandi, 1995; Gamage *et al.*, 2013). In particular, the *generalized fiducial inference* (GFI) is one of the successful modern formulations of Fisher's fiducial inference idea. The GFI framework has been successfully applied to various statistical problems, including wavelet regression (Hannig and Lee, 2009), linear mixed models (Cisewski *et al.*, 2012), extreme value estimation (Wandler and Hannig, 2012) and logistic regression (Liu and Hannig, 2016). In particular, Lai *et al.* (2015) successfully applied GFI framework to develop inferential procedure for the ultrahigh-dimensional regression problem, which shows the excellent theoretical and practical performance.

In the generalized fiducial inference framework, the relationship between the data $\boldsymbol{Y}$ and the parameters $\boldsymbol{\theta}$ can be expressed as

$$\boldsymbol{Y} = \boldsymbol{G}(\boldsymbol{\theta}, \boldsymbol{U}),$$

where $\boldsymbol{G}(\cdot,\cdot)$ is a deterministic function, $\boldsymbol{U}$ is a random component whose distribution is completely known (e.g. i.i.d. N(0,1)) and independent with $\boldsymbol{\theta}$.

The most essential idea behind the philosophy of GFI is the so-called switch principle, which is also the idea behind maximum likelihood estimation: assume $\boldsymbol{y}$ is the observed data of $\boldsymbol{Y}$, then the likelihood is a function of $\boldsymbol{\theta}$, so $\boldsymbol{y}$ is treated as fixed while $\boldsymbol{\theta}$ is random. With this thinking, for any given $\boldsymbol{y}$ if we assume the inverse mapping of $\boldsymbol{G}$ always exists, we could define the following set:

$$\boldsymbol{Q_y}(\boldsymbol{u}) = \{\boldsymbol{\theta} : \boldsymbol{y} = \boldsymbol{G}(\boldsymbol{\theta}, \boldsymbol{u})\}, \tag{2.2}$$

where $\boldsymbol{u}$ is a realization of $\boldsymbol{U}$. There are two scenarios that inverse mapping may not exist: no $\boldsymbol{\theta}$ or more than one $\boldsymbol{\theta}$ in the set of $\{\boldsymbol{\theta} : \boldsymbol{y} = \boldsymbol{G}(\boldsymbol{\theta}, \boldsymbol{u})\}$. For the first case, we could remove those values of $\boldsymbol{u}$ for which there is no solution and re-normalized the density function of $\boldsymbol{u}$. For the second case, Hannig (2009) suggests that any one of the solution would give satisfactory results, so just randomly pick one. These two strategies guarantee the inverse mapping exists.

Thus, since the distribution of of $\boldsymbol{U}$ is known, one can always generate a random sample $\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, \ldots$, and then obtain a sample of $\boldsymbol{\theta}$ by (2.2): $\tilde{\boldsymbol{\theta}}_1 = \boldsymbol{Q_y}(\tilde{\boldsymbol{u}}_1), \tilde{\boldsymbol{\theta}}_2 = \boldsymbol{Q_y}(\tilde{\boldsymbol{u}}_2), \ldots$ We call $\{\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, \ldots\}$ a fiducial sample of $\boldsymbol{\theta}$. Its function is similar to Bayesian posterior samples, so we could use it to perform statistical inference like constructing the confidence interval for $\boldsymbol{\theta}$.

The procedure above implicitly defines the density function of $\boldsymbol{\theta}$ which is termed as generalized fiducial density (GFD). Hannig *et al.* (2016) shows that, under some reasonable smoothness assumptions, the GFD denoted as $r(\boldsymbol{\theta}|\boldsymbol{y})$ is absolutely continuous and has density as

$$r(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}, \boldsymbol{\theta}) J(\boldsymbol{y}, \boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} f(\boldsymbol{y}, \boldsymbol{\theta}') J(\boldsymbol{y}, \boldsymbol{\theta}') d\boldsymbol{\theta}'}, \tag{2.3}$$

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

$$J(\boldsymbol{y}, \boldsymbol{\theta}) = D\left( \frac{\boldsymbol{d}}{\boldsymbol{d\theta}} \boldsymbol{G}(\boldsymbol{\theta}, \boldsymbol{u}) \Big|_{\boldsymbol{u}=\boldsymbol{G}^{-1}(\boldsymbol{y},\boldsymbol{\theta})} \right), \tag{2.4}$$

where $D(\boldsymbol{A}) = (det \boldsymbol{A}^T \boldsymbol{A})^{\frac{1}{2}}$ and $\boldsymbol{u} = \boldsymbol{G}^{-1}(\boldsymbol{y}, \boldsymbol{\theta})$ is the value of $\boldsymbol{u}$ such that $\boldsymbol{y} = \boldsymbol{G}(\boldsymbol{\theta}, \boldsymbol{u})$.

Equations (2.3) and (2.4) show the interesting connection between GFI and Bayesian methodology: $r(\boldsymbol{\theta}|\boldsymbol{y})$ in (2.3) behaves like a Bayesian posterior while $J(\boldsymbol{y}, \boldsymbol{\theta})$ as the "prior". However,

$J(\boldsymbol{y}, \boldsymbol{\theta})$ depend on $\boldsymbol{y}$, so technically it is not a prior density. Note that $J(\boldsymbol{y}, \boldsymbol{\theta})$ is very similar to the Jeffreys prior.

Sometimes it's impossible to calculate the $r(\boldsymbol{\theta}|\boldsymbol{y})$ in (2.3) analytically and $r(\boldsymbol{\theta}|\boldsymbol{y})$ is known up to a normalizing constant. In such case, one may resort to use the MCMC methods to generate a fiducial sample from $r(\boldsymbol{\theta}|\boldsymbol{y})$, so it's computationally demanding.

### 2.3. GFI for Multi-Task learning (GMTask)

When the model dimension is unknown, Equation (2.3) is not applicable, which is the situation for the current problem. To solve this problem, Hannig and Lee (2009) proposed a method for introducing penalty in the GFI framework. In particular, it can be shown that the fiducial probability of each candidate model $M$ is proportional to

$$r(M) \propto e^{-q(M)} \int_{\boldsymbol{\Theta}} f_M(\boldsymbol{y}, \boldsymbol{\theta}) \boldsymbol{J}_M(\boldsymbol{y}, \boldsymbol{\theta}) d\boldsymbol{\theta} \tag{2.5}$$

where $f_M(\boldsymbol{y}, \boldsymbol{\theta})$ is the likelihood, $\boldsymbol{J}_M(\boldsymbol{y}, \boldsymbol{\theta})$ is the Jacobian (2.4), and $q(M)$ is the penalty term associated with the model $M$. We will use the minimum description length (MDL) principle (Rissanen, 1989, 2007) to derive the penalty term $q(M)$, which shows excellent theoretical and empirical properties.

We will first calculate $\boldsymbol{J}_M(\boldsymbol{y}, \boldsymbol{\theta})$ in (2.4), then $f_M(\boldsymbol{y}, \boldsymbol{\theta})$, and finally $r(M)$ in (2.5).

For this multi-task learning problem, $\boldsymbol{\theta}$ contains three components: $\{M, \boldsymbol{B}_M, \boldsymbol{\Sigma}\}$, where $M$ denotes a candidate model, $\boldsymbol{B}_M$ is the parameters values for significant predictors coefficients, and $\boldsymbol{\Sigma}$ is the noise covariance matrix. The data generating function is now

$$\boldsymbol{Y} = \boldsymbol{G}(M, \boldsymbol{B}_M, \boldsymbol{\Sigma}, \boldsymbol{U}) = \boldsymbol{X}_M \boldsymbol{B}_M + \boldsymbol{U} \boldsymbol{\Sigma}^{\frac{1}{2}}, \quad M \in \mathcal{M}, \tag{2.6}$$

where $\boldsymbol{X}_M$ is the design matrix for $M$, and $\mathcal{M}$ is a collection of candidate models. We denotes $\hat{\boldsymbol{B}}_M = (\boldsymbol{X}_M^T \boldsymbol{X}_M)^{-1} \boldsymbol{X}_M^T \boldsymbol{y}$ as the least square estimates of $\boldsymbol{B}_M$, and $\boldsymbol{S}_M = (\boldsymbol{y} - \boldsymbol{X}_M \hat{\boldsymbol{B}}_M)^T (\boldsymbol{y} - \boldsymbol{X}_M \hat{\boldsymbol{B}}_M)$. If we use the vectorization of $\boldsymbol{B}_M$ and $\boldsymbol{Y}$ to reformat (2.6), we obtain

$$\boldsymbol{J}_M(\boldsymbol{y}, \boldsymbol{\theta}) = |\det(\boldsymbol{X}_M^T \boldsymbol{X}_M)|^{\frac{m}{2}} \operatorname{tr}(\boldsymbol{S}_M \boldsymbol{\Sigma}^{-1})^{\frac{1}{2}},$$

where $m$ is the number of tasks, and $\mathrm{tr}(\cdot)$ is the trace function of the matrix. The likelihood function can be formulated as

$$f_M(\boldsymbol{y}, \boldsymbol{\theta}) \propto |\Sigma|^{-\frac{n}{2}} \exp\left\{ \mathrm{tr}\left[ -\frac{1}{2}\left( \boldsymbol{S}_M + (\boldsymbol{B}_M - \hat{\boldsymbol{B}}_M)^T \boldsymbol{X}_M^T \boldsymbol{X}_M (\boldsymbol{B}_M - \hat{\boldsymbol{B}}_M) \right) \boldsymbol{\Sigma}^{-1} \right] \right\}.$$

If we approximate $\mathrm{tr}(\boldsymbol{S}_M \boldsymbol{\Sigma}^{-1})$ by its expectation, the generalized fiducial probability for any candidate model $M$ is

$$r(M) \propto R(M) = \Gamma_m\left(\frac{n - |M| - m}{2}\right) |\pi \boldsymbol{S}_M|^{-\frac{n-|M|-m}{2}} [(n - |M|)m]^{\frac{1}{2}} \times (n^{-\frac{m}{2}} p^{-1})^{|M|}, \tag{2.7}$$

where $\Gamma_m(\cdot)$ is multivariate gamma function, and $|M|$ is the number of significant predictors in $M$. In particular, the penalty term used is

$$q(M) = \frac{|M|m}{2} \log n + |M| \log p = |M| \left( \frac{m}{2} \log n + \log p \right). \tag{2.8}$$

We note that the second term of (2.8), which coincidentally shares the same asymptotic order as the corresponding penalty term in EBIC of Chen and Chen (2008). We note that, however, $q(M)$ is different from EBIC for finite samples when $m = 1$.

**2.3.1. Practical Generation of Fiducial Sample.** In this subsection, we propose a procedure to practically generate fiducial sample of $\{M, \boldsymbol{B}, \boldsymbol{\Sigma}\}$ for the multi-task learning problem. The original total number of models are $2^p$, and it's an tremendous amount when $p \gg n$. To reduce the candidate model space, we want to construct a collection of candidates models, denoted as $\mathcal{M}^*$. This $\mathcal{M}^*$ should only contain candidate models with non-negligible values of $r(M)$ including true model.

First, we need to apply a screening procedure to remove a large number of insignificant predictors, so we reduce the number of predictors from $p$ to $p'$, where $p' < n$. Inspired by sure independence screening (SIS) procedure in Fan and Lv (2008), which ranks the predictors according to the absolute values of marginal correlations with the response. We calculate the sum of marginal correlation across different tasks for each predictor since the current problem has multivariate responses; this is reasonable when we assume different tasks share the same sparsity pattern.

We call this procedure Multi-task SIS, and in practice we set $p' = n - 1$. Notice that Multi-task SIS can be skipped when $p = O(n)$, and save the computational time when $p \gg n$.

To further reduce the candidate model space, where the total number of model is $2^{p'}$, we apply multi-task $L_1/L_2$ Lasso (Obozinski *et al.*, 2006) to the remaining $p'$ predictors that survived Multi-task SIS. The $L_1/L_2$ block-regularization is commonly referred to as the group Lasso, and the objective function to minimize is

$$\frac{1}{2n}||\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{B}||_F^2 + \lambda \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{m} \beta_{ij}^2},$$

where $||\cdot||_F$ is the Frobenius norm. Therefore, one can perform row selection by solving the optimization problem above. Thus, we take the sequence of nested models that lie on the regularization path or so-called solution path to form $\mathcal{M}^*$. Note that the multi-task $L_1/L_2$ Lasso solution path can be derived quickly by coordinate descent method (Friedman *et al.*, 2010). For the purpose not to missing any candidate models with non-negligible values of $r(M)$, we repeat the $L_1/L_2$ Lasso procedure on the randomly selected subset of the data multiple times, and take all the models that lie on the solution path to form $\mathcal{M}^*$. By this way, we expect $\sum_{M\in\mathcal{M}^*} r(M)$ is very close to 1 and the size of $\mathcal{M}^*$ is substantially smaller than $2^p$.

Once $\mathcal{M}^*$ is obtained, for each $M \in \mathcal{M}^*$, we compute

$$R(M) = \Gamma_m(\frac{n - |M| - m}{2})|\pi \boldsymbol{S}_M|^{-\frac{n-|M|-m}{2}}[(n - |M|)m]^{\frac{1}{2}} \times (n^{-\frac{m}{2}}p^{-1})^{|M|}.$$

Thus, for each $M \in \mathcal{M}^*$, the generalized fiducial probability $r(M)$ can be well approximated by

$$\hat{r}(M) = \frac{R(M)}{\sum_{M'\in\mathcal{M}^*} R(M')}. \tag{2.9}$$

For any given $M$, according to the generalized inverse Wishart distribution in Triantafyllopoulos (2011), the generalized fiducial distribution of $\boldsymbol{\Sigma}$ conditional on $M$ is

$$\boldsymbol{\Sigma} \sim W_m^{-1}(n - |M|, \boldsymbol{S}_M), \tag{2.10}$$

where $W_m^{-1}(\cdot,\cdot)$ is the inverse Wishart distribution with the first parameter as the degrees of freedom and second one as scale matrix. Thus, the generalized fiducial distribution of $\boldsymbol{B}_M$ conditional on $M$ and $\boldsymbol{\Sigma}$ is

$$\text{vec}(\boldsymbol{B}_M) \sim N\big(\text{vec}(\hat{\boldsymbol{B}}_M), \boldsymbol{\Sigma} \otimes (\boldsymbol{X}_M^T \boldsymbol{X}_M)^{-1}\big), \tag{2.11}$$

where $\text{vec}(\cdot)$ is vectorization of the matrix, $\hat{\boldsymbol{B}}_M$ is the maximum likelihood estimates of $\boldsymbol{B}_M$, and $\otimes$ is Kronecker product. Equation (2.11) has a equivalent form in matrix normal distribution with a more efficient sampling algorithm as

$$\boldsymbol{B}_M \sim MN\big(\hat{\boldsymbol{B}}_M, (\boldsymbol{X}_M^T \boldsymbol{X}_M)^{-1}, \boldsymbol{\Sigma}\big), \tag{2.12}$$

where $MN(\cdot,\cdot,\cdot)$ is the matrix normal distribution with first parameter as mean, second one as the among-row covariance matrix, and last one as as the among-column covariance matrix.

To sum up, we can generate the fiducial sample $\{\tilde{M}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{\Sigma}}\}$ by the following steps:

(1) Draw a model $\tilde{M} \in \mathcal{M}^*$ from (2.9).
(2) Generate $\tilde{\boldsymbol{\Sigma}}$ from (2.10) given $\tilde{M}$.
(3) Sample $\tilde{\boldsymbol{B}}$ from (2.12) given $\tilde{M}$ and $\tilde{\boldsymbol{\Sigma}}$.

Note that the algorithm above is not computationally demanding like many MCMC methods, so the generation of a fiducial sample is relative fast.

**2.3.2. Point Estimates, Confidence Ellipsoids and Prediction Ellipsoids.** Repeating the procedure above multiple times, one can obtain multiple copies of the fiducial sample $\{\tilde{M}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{\Sigma}}\}$, which is similar to Bayesian posterior sample and can be used for inference. Note that the GFI framework draws the candidate model according to $\hat{r}(M)$ in (2.9), which results in not only a single model being selected in the fiducial sample. For the mean function $E(Y_i|\boldsymbol{x}_i) = \boldsymbol{B}^T\boldsymbol{x}_i$, we have a sample of $\tilde{\boldsymbol{B}}^T\boldsymbol{x}_i$'s, the sample mean of which is denoted as $\hat{\boldsymbol{\mu}}$ and the sample covariance matrix noted as $\hat{\boldsymbol{S}}$. Thus, one can use $\hat{\boldsymbol{\mu}}$ as a point estimates, and the $100(1-\alpha)\%$ confidence ellipsoid for $E(Y_i|\boldsymbol{x}_i) = \boldsymbol{B}^T\boldsymbol{x}_i$ is provided by the inequality

$$\left(\boldsymbol{B}^T\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}\right)^T \hat{\boldsymbol{S}}^{-1} \left(\boldsymbol{B}^T\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}\right) \leq \chi^2_m(\alpha), \tag{2.13}$$

where $\chi^2_m(\alpha)$ is the upper $(100\alpha)$th percentile of a $\chi^2$-distribution with $m$ d.f.

Similarly, the point estimates and prediction ellipsoid for $Y_i$ at $\boldsymbol{x}_i$ can be derived by using the sample mean and inequality (2.13) for $\tilde{\boldsymbol{B}}^T\boldsymbol{x}_i + \tilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}\boldsymbol{Z}$, where $\boldsymbol{Z} \sim N(\boldsymbol{0}, \boldsymbol{I}_m)$.

## 2.4. Theoretical Properties

This section presents some asymptotic properties of the above generalized fiducial based method. We assume that $p$ is diverging and and the size of true model is either fixed or diverging. Before we present our theorem, we first provide the following necessary notations and assumptions which are similar to those of Yanagihara *et al.* (2015).

**2.4.1. Preliminaries and Notations.** Let $M$ be any model and $M_0$ be the true model, while $|M|$ and $|M_0|$ be the number of predictors selected for models $M$ and $M_0$, respectively. To reduce the candidate model space, we only consider $M \in \mathcal{M}$ where $\mathcal{M} = \{M : |M| \leq c|M_0|\}$ for a fixed finite constant $c > 1$; i.e., the model whose size is comparable to the true model.

Denote

$$\mathcal{M}_+ = \{M : M_0 \subset M\}, \quad \mathcal{M}_- = \{M : M_0 \nsubseteq M\} \quad \text{and} \quad \mathcal{M} = \mathcal{M}_- \cup \{M_0\} \cup \mathcal{M}_+.$$

Define

$$\boldsymbol{S}_M = \boldsymbol{Y}^T(\boldsymbol{I}_n - \boldsymbol{H}_M)\boldsymbol{Y}, \hat{\boldsymbol{\Sigma}}_M = \frac{1}{n}\boldsymbol{S}_M,$$

where $\boldsymbol{H}_M = \boldsymbol{X}_M(\boldsymbol{X}_M^T\boldsymbol{X}_M)^{-1}\boldsymbol{X}_M^T$ is the projection matrix to the subspace spanned by the columns of $\boldsymbol{X}_M$. For simplicity, we write $\boldsymbol{X}_{M_0}$ as $\boldsymbol{X}_0$, same for $\boldsymbol{B}$ and $\boldsymbol{\Sigma}$. We assume that the data are generated from the following true model:

$$\boldsymbol{Y} \sim N_{n\times m}(\boldsymbol{X}_0\boldsymbol{B}_0, \boldsymbol{\Sigma}_0 \otimes \boldsymbol{I}_n).$$

In order to prove the consistency of our method, we need to describe non-centrality matrix. For a model $M \in \mathcal{M}$, let a $m \times m$ non-centrality matrix denoted by

$$\Delta(M) = \boldsymbol{\Sigma}_0^{-\frac{1}{2}}\boldsymbol{B}_0^T\boldsymbol{X}_0^T(\boldsymbol{I}_n - \boldsymbol{H}_M)\boldsymbol{X}_0\boldsymbol{B}_0\boldsymbol{\Sigma}_0^{-\frac{1}{2}}.$$

**2.4.2. Assumptions.** (A1) The true model $M_0 \in \mathcal{M}$, where $\mathcal{M} = \{M : |M| \leq c|M_0|\}$ for a fixed finite constant $c > 1$.

(A2) $\lim_{n\to\infty} \frac{1}{n} \boldsymbol{X}_M^T \boldsymbol{X}_M$ exists and is positive definite, and $\lim_{n\to\infty} \frac{1}{n} \Delta(M) = \boldsymbol{\Psi}_M$ exist and is not the zero matrix for all $M \in \mathcal{M}_-$.

(A3) When $p$ is too large, a variable screening procedure could be used to reduce the size of $\mathcal{M}$ in practice. That screening procedure should result in a class of candidate models $\mathcal{M}^*$, which satisfies:

$$P(M_0 \in \mathcal{M}^*) \to 1 \quad \text{and} \quad \log(|\mathcal{M}_j^*|) = o(j \log n), \tag{2.14}$$

where $\mathcal{M}_j^*$ denotes the set of all sub-models in $\mathcal{M}^*$ of size $j$. The first condition in (2.14) ensures that the true model is contained in $\mathcal{M}^*$; the second condition in (2.14) implies that the size of model space $\mathcal{M}^*$ is not too large.

**2.4.3. Main Result.** We establish the following theorem and its proof can be found in the appendix.

THEOREM 2.4.1. *Assume A1-A2 hold. As $n \to \infty$, $p \to \infty$, $\log p = o(n)$ and $|M_0| + m = o(\log n)$, we have*

$$\max_{M \neq M_0, M \in \mathcal{M}} \frac{r(M)}{r(M_0)} \xrightarrow{P} 0. \tag{2.15}$$

*Moreover if A3 also holds, we have*

$$\frac{r(M_0)}{\sum_{M \in \mathcal{M}^*} r(M)} \xrightarrow{P} 1. \tag{2.16}$$

Theorem 2.4.1 shows that the true model $M_0$ has the highest generalized fiducial probability among all the candidate models in $\mathcal{M}$. However, equation (2.15) does not imply (2.16) in general since the model candidate space can be very large as $p$ goes to infinity. If we can constrain candidate models in a class that satisfies (2.14), then the true model will be selected with probability tending to 1.

In practice, we use the multi-task $L_1/L_2$ Lasso to generate the candidate models as discussed in Section 2.3.1. The resulting model space satisfies (2.14), since the multi-task Lasso has consistent selection for some $\lambda$ as shown in Obozinski *et al.* (2008). By Theorem 2 and Theorem 3 of Hannig *et al.* (2016) , Theorem 2.4.1 also implies that the confidence ellipsoids and prediction ellipsoids constructed using the generalized fiducial density (2.7) will have correct asymptotic coverage rates, and the generalized fiducial distribution and the derived point estimates are consistent.

## 2.5. Simulation Results

**2.5.1. Uncertainty Quantification.** Simulation experiments were conducted to evaluate the practical performance of GMTask. Inspired by the setting in Fan *et al.* (2012) and Bedrick and Tsai (1994), we consider the following data-generating model

$$\boldsymbol{Y} = b(X_1 + \cdots + X_k)\mathbb{1}_m + \boldsymbol{E},$$

where $\boldsymbol{E}$ is iid as $N_m(\mathbf{0}, \boldsymbol{\Sigma})$, $\mathbb{1}_m$ is a $m \times 1$ vector of ones, $k$ is the number of significant predictors with non-zero coefficients, and $b$ is the coefficient value that controls the signal-to-noise ratio. The covariates $X_i$'s are generated from standard normal distribution with $Cor(X_i, X_j) = \rho_X^{|i-j|}$. We set the covariance matrix $\boldsymbol{\Sigma} = (1-\rho_\Sigma)\boldsymbol{I}_m + \rho_\Sigma \boldsymbol{J}_m$, where $\boldsymbol{J}_m$ is a $m \times m$ matrix of ones. Two combinations of $(n, p, m, k)$ were used: (200, 2000, 2, 3), (200, 2000, 3, 3). For each combination setting, we used three different $b$, two different $\rho_X$ and three different $\rho_\Sigma$: $b = 1/\sqrt{k}, 2/\sqrt{k}, 3/\sqrt{k}, \rho_X = 0, 0.5$ and $\rho_\Sigma = 0.3, 0.6, 0.8$. Therefore, a total of $2 \times 3 \times 2 \times 3 = 36$ experimental settings were considered. The number of repetitions for each setting was 1,000. For the data set generated in each repetition, we applied the proposed generalized fiducial procedure described in Section 2.3.1 to obtain the 10,000 fiducial sample of $\{M, \boldsymbol{B}, \boldsymbol{\Sigma}\}$.

For each simulated data set, we applied five methods to obtain the estimates $\hat{\boldsymbol{B}}$ and $\hat{\boldsymbol{\Sigma}}$ for the coefficient $\boldsymbol{B}$ and the covariance matrix $\boldsymbol{\Sigma}$, respectively. Then, we constructed the prediction ellipsoids for new $Y_i$'s and the confidence ellipsoids for the mean function $E(Y_i|\boldsymbol{x}_i)$ for 50 randomly selected new designed points $\boldsymbol{x}_i$'s. The five methods were

- Proposed generalized fiducial method (GMTask);
- V-L0LS-CD method of Seneviratne and Solo (2012);

- AICc of Bedrick and Tsai (1994);
- BIC;
- Oracle method that uses the true model

Of course, the last one, oracle method, is not applicable in practice, but we use it as a benchmark comparison. For all five methods, we constructed $90\%, 95\%, 99\%$ prediction ellipsoids and confidence ellipsoids from each simulation setting. The proposed fiducial method uses (2.13) in Section 2.3.2 to construct the ellipsoids. For AICc and BIC, we first used them to perform model selection in $\mathcal{M}^*$, then applied the classical linear model theory to the final selected model to build the ellipsoids.

To evaluate the performance of different methods on uncertainty quantification, we calculated the average empirical coverage rates for these prediction ellipsoids and confidence ellipsoids. The results are summarized in Tables 2.1 to 2.6. It can be seen that the proposed generalized fiducial method are nearly as good as the oracle method, and GMTask outperform other non-oracle methods significantly especially for the confidence ellipsoids of the mean function $E(Y_i|\boldsymbol{x}_i)$.

**2.5.2. Point Estimates.** Furthermore, we also calculated Mean Squared Prediction Error (MSPE) and Mean Squared Mahalanobis Distance (MSMD) to measure the performance of predicting new $Y_i$ at $\boldsymbol{x}_i$, defined respectively as

$$\text{MSPE}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^T(Y_i - \hat{Y}_i)$$

and

$$\text{MSMD}(Y, \hat{Y}) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^T D_i^{-1}(Y_i - \hat{Y}_i),$$

where $\hat{Y}_i$ is $m \times 1$ vector and the corresponding prediction for $Y_i$ by one of the five methods; $D_i$ is $Cov(Y_i - \hat{Y}_{i,\text{oracle}}) = (1 + \boldsymbol{x}_{i0}^T(\boldsymbol{X}_0^T\boldsymbol{X}_0)^{-1}\boldsymbol{x}_{i0})\boldsymbol{\Sigma}_0$, where $\boldsymbol{x}_{i0}$ is the true model $M_0$ subset of $\boldsymbol{x}_i$ and $\hat{Y}_{i,\text{oracle}} = \hat{\boldsymbol{B}}_{\text{oracle}}^T\boldsymbol{x}_{i0}$ is the oracle prediction for $Y_i$. Note that when $D_i$ is an identity matrix, MSMD reduces to MSPE.

Similarly, to measure the performance of predicting the mean function $E(Y_i|\boldsymbol{x}_i)$ at $\boldsymbol{x}_i$, we define Mean Squared Error (MSE) and Mean Squared Mahalanobis Distance (MSMD) respectively as

$$\text{MSE}\big(E(Y), \hat{Y}\big) = \frac{1}{n}\sum_{i=1}^{n}\big(E(Y_i) - \hat{Y}_i\big)^T\big(E(Y_i) - \hat{Y}_i\big)$$

and

$$\text{MSMD}\big(E(Y), \hat{Y}\big) = \frac{1}{n}\sum_{i=1}^{n}\big(E(Y_i) - \hat{Y}_i\big)^T \tilde{D}_i{}^{-1}\big(E(Y_i) - \hat{Y}_i\big),$$

where $E(Y_i) = \boldsymbol{B}_0^T\boldsymbol{x}_{i0}$ and $\tilde{D}_i = Cov(\hat{Y}_{i,\text{oracle}}) = \boldsymbol{x}_{i0}^T(\boldsymbol{X}_0^T\boldsymbol{X}_0)^{-1}\boldsymbol{x}_{i0}\boldsymbol{\Sigma}_0$. Again, MSMD reduces to MSE when $\tilde{D}_i$ is an identity matrix.

The results are summarized in Tables 2.7 to 2.12. From these tables, one can see that MSE, MSPE, MSMD of the fiducial estimates are usually not much larger than the those from the oracle estimates. For the mean function $E(Y_i|\boldsymbol{x}_i)$, the fiducial estimates outperformed the other non-oracle estimates significantly.

**2.5.3. True Model Not Included.** Previously in Section 2.5.1, we considered the case when the main effects are in the column space of the design matrix $\boldsymbol{X}$, which means the mean function $E(Y)$ can be represented by a linear combination of the covariates $X_i$'s. In this subsection, we apply GMTask to the scenarios where the main effects are not a linear combination of the candidate covariates $X_i$'s; i.e., the true model is not in the model candidate space. The following models were used to generate the noisy data

(i) $\boldsymbol{Y} = b(X_1 + X_2 + X_{2001})\mathbb{1}_m + \boldsymbol{E}$, where $X_{2001}$ is not one of the $p = 2000$ candidate covariates $X_i$'s;

(ii) $\boldsymbol{Y} = b(X_1 + X_2 + X_{1999}X_{2000})\mathbb{1}_m + \boldsymbol{E}$, where $X_{1999}X_{2000}$ is an interaction term;

(iii) $\boldsymbol{Y} = b(X_1 + X_2 + X_{2000}^2)\mathbb{1}_m + \boldsymbol{E}$, where $X_{2000}^2$ is the squared term of $X_{2000}$;

(iv) $\boldsymbol{Y} = b(X_1 + X_2 + X_1^2)\mathbb{1}_m + \boldsymbol{E}$, where $X_1^2$ is the squared term of $X_1$.

Specifically, for model (i), we used a predictor $X_{2001}$ that is outside of the candidate model space; for model (ii), we considered interaction term in the main effects; for model (iii) and (iv), we introduced the higher order term in the main effects. Except for the main effects, the remaining experimental parameters are similar to Section 2.5.1. The average empirical coverage rates of the prediction ellipsoids for predicting $Y_i$ under different scenarios are reported in Tables 2.13 to 2.24.

Compare with other methods, GMTask provide more reliable results when $E(Y)$ is not in the column space of the design matrix $\boldsymbol{X}$. All other non-oracle methods could only select one single model a time, and the selected model could not be the true model. However, GMTask was able to generate fiducial sample of $\{M, \boldsymbol{B}, \boldsymbol{\Sigma}\}$ with $M$ from different models. Therefore, the fiducial sample of $Y_i$ is an ensemble of different models' predictions, which result in higher empirical coverage rates for the prediction ellipsoids.

**2.5.4. Univariate GFI.** In this subsection, we compare the uncertainty quantification performance of GMTask and the univariate version GFI of Lai *et al.* (2015), which is a single-task learning approach to generate the fiducial sample for each task independently. In other words, there is no sharing of information across tasks for the univariate GFI.

The results are summarized in Tables 2.25 to 2.28. When $\rho_\Sigma = 0$, which means the tasks are independent, the performance of GMTask and the univariate GFI are almost identical. However, as the correlation across tasks $\rho_\Sigma$ and the number of tasks $m$ increase, GMTask gradually outperformed the univariate GFI in terms of the empirical coverage rate. Also, the volumes of the prediction ellipsoids of GMTask are much smaller than those of univariate GFI.

Figure 2.1 and Figure 2.2 show some results of applying these two methods to a typical simulated data set with $n = 200, p = 2000, m = 2, k = 3, \rho_X = 0$, while $\rho_\Sigma = 0$ and 0.8 respectively. Because the univariate GFI generates the fiducial sample for each task independently, so there is no obvious relationship between the fiducial sample of two tasks no matter if $\rho_\Sigma = 0$ or 0.8. When $\rho_\Sigma = 0$, two methods behave quite similar even almost the same. When $\rho_\Sigma = 0.8$, the fiducial sample generated by GMTask captured the correlation between two tasks, and the 90% prediciton ellipsoid covered the $Y_i$ near the top-right corner while the prediciton ellipsoid of univariate GFI did not.

**2.5.5. Case When $p < n$.** In this subsection, the setting is similar to Section 2.5.1 but with $(n, p) = (200, 100)$. The results are reported in Tables 2.29 to 2.31 for the empirical coverage rates of prediction ellipsoids. One can see that the performance of GMTask, BIC and V-L0LS-CD methods are all nearly as good as the oracle method except AICc.

When $p \gg n$, there is a higher chance to select the redundant insignificant predictors. However, the model candidate space of $p < n$ case is much smaller, so there is less chance to select the

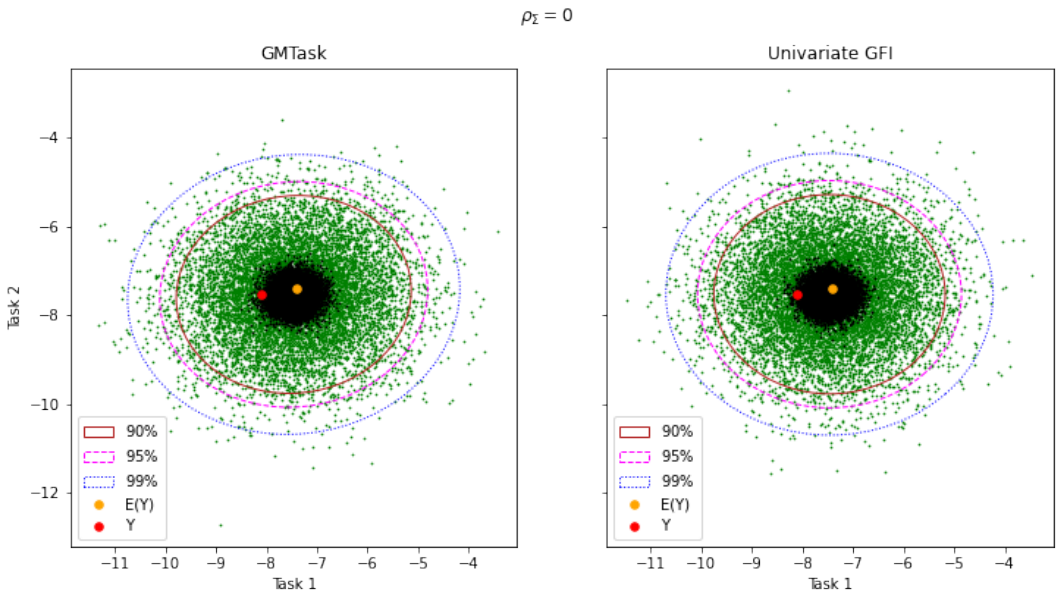


FIGURE 2.1. Prediction ellipsoids for $Y_i$ by GMTask and univariate GFI when $\rho_\Sigma = 0$. Green dots are fiducial sample of $Y_i$ while black dots are fiducial sample of $E(Y_i|\boldsymbol{x}_i)$.

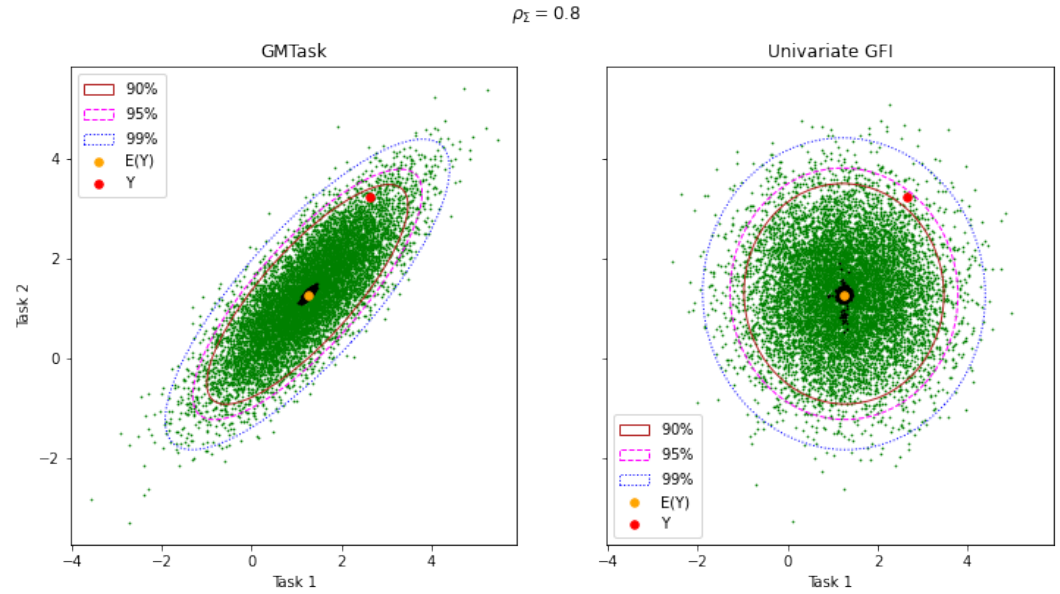


FIGURE 2.2. Similar to Figure 2.1 but for $\rho_\Sigma = 0.8$.

insignificant predictors, which could be the reason why these three methods' performance are all well. Therefore, in addition to the $p \gg n$ case, GMTask could also give reliable uncertainty quantification result when $p < n$.

## 2.6. Real Data Examples

**2.6.1. Polymerase Chain Reaction (PCR) Data.** An experiment was conducted by Lan *et al.* (2006) to examine the genetics of two inbred mouse populations (B6 and BTBR). The expression levels of 22,575 genes of 60 mice were measured. Some physiological phenotypes, including numbers of stearoyl-CoA desaturase 1 (SCD1) and phosphoenopyruvate carboxykinase (PEPCK), were also measured by quantitative real-time PCR. Since $n = 60, p = 22575$ and $m = 2$, it is tackling a high dimensional multi-task learning problem as $p \gg n$. Bondell and Reich (2012) used the credible approach to predict each of these two phenotypes independently based on the gene expression data, and hence it is a single task method for prediction. Here, we apply GMTask to study the uncertainty quantification performance.

The following process was conducted to evaluate the empirical coverage rates of V-L0LS-CD method of Seneviratne and Solo (2012), AICc, BIC and GMTask on this data set. Firstly, we left out the first observation as the test point with two response (SCD1 and PEPCK). Then we apply the proposed generalized fiducial procedure and other methods to the remaining 59 observations to construct the $90\%, 95\%$ and $99\%$ prediction ellipsoids for the first observation. We repeated this leave-one-out process to the remaining 59 observations and the resulting prediction ellipsoids are summarized in Table 2.32. It can be seen that GMTask is the preferred method as its coverage rates of the $90\%, 95\%$ and $99\%$ prediction ellipsoids are all close to the nominal levels. Furthermore, the volume of the 90% fiducial prediction ellipsoid is smaller than that of the 99% AICc prediction ellipsoid, while the empirical coverage rates of both methods are 0.900. Therefore, GMTask is more efficient for the uncertainty quantification on this data set.

**2.6.2. Alzheimer's Disease Cognitive Scores Data.** In this subsection, we present the uncertainty quantification results to demonstrate the effectiveness of GMTask on the progression of Alzheimer'sdisease (AD). We used a data set from the Alzheimer's Disease Neuroimaging Initiative

(ADNI), which is a multi-site study that aims to improve clinical trials for the treatment and prevention of AD.

In ADNI, all participants received 1.5 Tesla (T) structural MRI. A team from UCSF (University of California at San Francisco) performed cortical re-construction and volumetric segmentations with the FreeSurfer image analysis suite, and processed the imaging data from the ADNI database. The FreeSurfer software was employed to extract MRI features from cortical and subcortical regions of interests (ROIs), and the MRI features can be grouped into 5 categories: the cortical thickness average (TA), standard deviation of thickness (TS), surface area (SA), cortical volume (CV) and subcortical volume (SV).

In our work, we investigate the uncertainty quantification performance of our GMTask for inferring cognitive outcomes in a number of neuropsychological assessments at baseline time. After performed a similar preprocessing procedure in Liu *et al.* (2018) to the MRI data, there are a total of $n = 802$ subjects and $p = 319$ MRI features.

For the responses, we focus on 5 widely used cognitive test scores, which means there are $m = 5$ tasks. The cognitive tests is in form of questionnaire or activities to measure the attention, calculation, language, memory and other cognitive abilities which could reflect the symptoms of AD. In particular, the cognitive tests used in our analysis are: Alzheimer's Disease Assessment Scale cognitive total score (ADAS), Mini Mental State Exam score (MMSE), Rey Auditory Verbal Learning Test (RAVLT) total score (TOTAL), RAVLT 30 minutes delay score (T30), and RAVLT recognition score (RECOG). Therefore, the goal is to infer the cognitive scores based on the structural MRI features.

We split the data into 20 folds, and left 5% of the subjects as a test set every time, and the remaining 95% of the subjects were used to train the model which was then applied to construct the prediction ellipsoids for the test set. The resulting empirical coverage rates of the prediction ellipsoids are reported in Table 2.33. The results show that GMTask performs well, and performance of the other methods are similar to it, probably because there are strong linear effect in this data set. However, GMTask usually provides the same empirical coverage rate with a smaller volume of the ellipsoid, which suggests that GMTask gives a more efficient uncertainty quantification.

## 2.7. Conclusion

In this chapter, we investigated the issue of uncertainty quantification in the multi-task learning problem under the "large $p$ small $n$" setting. We adopted a generalized fiducial inference methodology to perform statistical inference for this problem. In particular we developed a procedure to generate fiducial samples based on the generalized fiducial distribution of a set of candidate models obtained from multi-task $L_1/L_2$ Lasso, and to construct various confidence ellipsoids and prediction ellipsoids by making use of these samples. Our theoretical results show that the estimates obtained by this procedure are consistent, while confidence ellipsoids and prediction ellipsoids constructed by this procedure have correct asymptotic frequentist property under some regularity assumptions. Numerical results from simulation experiments confirm with these theoretical findings. To the best of our knowledge, there are very few published articles that are devoted to quantify uncertainties in multi-task learning problem, therefore this dissertation is one of the first to provide a systematic method to this problem. We note that the current framework can be extended to other problem in high dimensional settings, such as the classification problem.

**Acknowledgements** The author wants to thank Prof. Victor Solo for providing the code of V-L0LS-CD method.

## 2.8. Proof of Theorem 2.4.1

Proof. Since

$$r(M) \propto \Gamma_m(\frac{n-|M|-m}{2})|\pi \boldsymbol{S}_M|^{-\frac{n-|M|-m}{2}}[(n-|M|)m]^{\frac{1}{2}} \times q^{|M|},$$

where $q = n^{-\frac{m}{2}}p^{-1}$ is derived by using the MDL principle. Let $k = |M|$ and $k_0 = |M_0|$ for simplicity, thus we have

$$\frac{r(M)}{r(M_0)} = \exp\{-T_1 - T_2\},$$

where

$$T_1 = \frac{n-k-m}{2}\log\left(\frac{|\hat{\boldsymbol{\Sigma}}_M|}{|\hat{\boldsymbol{\Sigma}}_{M_0}|}\right)$$

and

$$T_2 = \frac{k_0 - k}{2} \log\left(|\pi \boldsymbol{S}_{M_0}|\right) + \log\left\{\Gamma_m(\frac{n-k_0-m}{2})/\Gamma_m(\frac{n-k-m}{2})\right\} + (k_0 - k)\log(q) + \frac{1}{2}\log\frac{n-k_0}{n-k}.$$

As $n \to \infty$, we can see that $\hat{\boldsymbol{\Sigma}}_{M_0} \xrightarrow{P} \boldsymbol{\Sigma}_0$ when $M \notin \mathcal{M}_-$ and $\hat{\boldsymbol{\Sigma}}_M \xrightarrow{P} \boldsymbol{\Sigma}_0^{\frac{1}{2}} \boldsymbol{\Psi}_M \boldsymbol{\Sigma}_0^{\frac{1}{2}} + \boldsymbol{\Sigma}_0$ when $M \in \mathcal{M}_-$, where $\boldsymbol{\Psi}_M = \lim_{n\to\infty} \frac{1}{n}\Delta(M)$. By our assumption, $\boldsymbol{\Psi}_M$ is a positive semi-definite matrix.

**Case 1:** $\forall M \in \mathcal{M}_-$**.**

We have

$$\begin{aligned} \frac{T_1}{n} &\xrightarrow{P} \frac{1}{2}\log|\boldsymbol{\Sigma}_0^{\frac{1}{2}} \boldsymbol{\Psi}_M \boldsymbol{\Sigma}_0^{\frac{1}{2}} + \boldsymbol{\Sigma}_0| - \frac{1}{2}\log|\boldsymbol{\Sigma}_0| \\ &= \frac{1}{2}\log|\boldsymbol{\Psi}_M + \boldsymbol{I}_m| > 0 \end{aligned} \tag{2.17}$$

Since $\boldsymbol{S}_{M_0}$ is distributed as $W_m(n-k_0, \Sigma_0)$, we can derive the log-expectation and log-variance as

$$E\big[\log|\boldsymbol{S}_{M_0}|\big] = \psi_m(\frac{n-k_0}{2}) + m\log 2 + \log|\Sigma_0|$$

and

$$Var\big[\log|\boldsymbol{S}_{M_0}|\big] = \sum_{i=1}^{m} \psi_1\left(\frac{n-k_0+1-i}{2}\right)$$

where $\psi_m$ is the multivariate digamma function, i.e. the derivative of the log of the multivariate gamma function; and $\psi_1$ is the trigamma function. Then we have

$$\log|\boldsymbol{S}_{M_0}| = m\log(n-k_0)(1+o_p(1)) = m\log n(1+o_p(1))$$

By the definition of multivariate gamma function $\Gamma_p$, we have

$$\Gamma_p(a) = \pi^{\frac{p(p-1)}{4}} \prod_{j=1}^{p} \Gamma\left(a + \frac{1-j}{2}\right).$$

According to Stirling's approximation,

$$\log\left\{\Gamma_m(\frac{n-k_0-m}{2})/\Gamma_m(\frac{n-k-m}{2})\right\} = \frac{m(k-k_0)}{2}\log n(1+o(1)).$$

Therefore we have

$$T_2 = \frac{m(k-k_0)}{2}\left\{ \log n(o_p(1)) - \log \pi - \frac{\log(q^2)}{m} \right\} + \frac{1}{2}\log\frac{n-k_0}{n-k} \tag{2.18}$$

and

$$\lim_{n\to\infty} \frac{T_2}{n} = 0 \tag{2.19}$$

By (2.17) and (2.19), for case 1, we have

$$\min_{M\in\mathcal{M}_-} T_1 + T_2 \to \infty$$

**Case 2:** $\forall M \in \mathcal{M}_+$**.**

Let $\boldsymbol{V}$ and $\boldsymbol{Z}_M$ be the $m \times m$ and the $k \times m$ matrices defined by

$$\boldsymbol{V} = \frac{1}{\sqrt{n}}\big(\boldsymbol{U}^T\boldsymbol{U} - n\boldsymbol{I}_m\big), \boldsymbol{Z}_M = (\boldsymbol{X}_M^T\boldsymbol{X}_M)^{-\frac{1}{2}}\boldsymbol{X}_M^T\boldsymbol{U}$$

where

$$\boldsymbol{U} = (\boldsymbol{Y} - \boldsymbol{X}_0\boldsymbol{B}_0)\boldsymbol{\Sigma}_0^{-\frac{1}{2}}$$

We know that $\boldsymbol{V}$ has an asymptotic normality as $n \to \infty$ and $\boldsymbol{Z}_M \sim N_{k\times m}(\boldsymbol{0}_{k\times m}, \boldsymbol{I}_{km})$

Furthermore, since

$$\boldsymbol{\Sigma}_0^{-\frac{1}{2}}\hat{\boldsymbol{\Sigma}}_M\boldsymbol{\Sigma}_0^{-\frac{1}{2}} = \frac{1}{n}\boldsymbol{U}^T(\boldsymbol{I}_n - \boldsymbol{H}_M)\boldsymbol{U} = \frac{1}{n}(\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{Z}_M^T\boldsymbol{Z}_M),$$

we have

$$\boldsymbol{\Sigma}_0^{-\frac{1}{2}}\hat{\boldsymbol{\Sigma}}_M\boldsymbol{\Sigma}_0^{-\frac{1}{2}} = \boldsymbol{I}_m + \frac{1}{\sqrt{n}}\boldsymbol{V} - \frac{1}{n}\boldsymbol{Z}_M^T\boldsymbol{Z}_M \tag{2.20}$$

By using (2.20), $n\log|\hat{\boldsymbol{\Sigma}}_M|$ can be expanded as

$$n\log|\hat{\boldsymbol{\Sigma}}_M| = n\log|\boldsymbol{\Sigma}_0| + \sqrt{n}\mathrm{tr}(\boldsymbol{V}) - \left(\frac{1}{2}\mathrm{tr}(\boldsymbol{V}^2) + \mathrm{tr}(\boldsymbol{Z}_M^T\boldsymbol{Z}_M)\right) + O_p(n^{-\frac{1}{2}}).$$

Then, we derive

$$T_1 = -\frac{n-k-m}{2n}\big(\mathrm{tr}(\boldsymbol{Z}_M^T\boldsymbol{Z}_M) - \mathrm{tr}(\boldsymbol{Z}_{M_0}^T\boldsymbol{Z}_{M_0}) + O_p(n^{-\frac{1}{2}})\big)$$

By (2.18), for all $M \in \mathcal{M}_+$, $\lim_{n\to\infty} T_2 = \infty$. Then for case 2, we have

$$\min_{M\in\mathcal{M}_+} T_1 + T_2 \to \infty$$

Combing case 1 and case 2, we can show that

$$\max_{M\neq M_0, M\in\mathcal{M}} \frac{r(M)}{r(M_0)} \to 0.$$

Moreover, if condition (A3) holds, we have

$$\begin{aligned}\sum_{M\neq M_0, M\in\mathcal{M}^*} \frac{r(M)}{r(M_0)} &\leq \sum_{j=1}^{ck_0} \sum_{\mathcal{M}_j^*} \frac{r(M)}{r(M_0)} \\ &\leq \sum_{j=1}^{ck_0} |\mathcal{M}_j^*| \max_{M\neq M_0, M\in\mathcal{M}_j^*} \frac{r(M)}{r(M_0)} \to 0.\end{aligned}$$

This completes the proof for Theorem 2.4.1. □

## 2.9. Tables

TABLE 2.1. Empirical coverage rates of confidence ellipsoids for $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.906**(0.229) | **0.954**(0.298) | **0.989**(0.458) | **0.893**(0.144) | **0.943**(0.201) | **0.988**(0.352) |
| | AICc | 0.514(0.378) | 0.605(0.493) | 0.759(0.765) | 0.526(0.301) | 0.624(0.424) | 0.782(0.752) |
| | BIC | 0.584(0.304) | 0.665(0.397) | 0.800(0.615) | 0.746(0.167) | 0.815(0.235) | 0.903(0.416) |
| | Vector L0 | 0.386(0.661) | 0.462(0.863) | 0.602(1.339) | 0.706(0.187) | 0.776(0.263) | 0.873(0.467) |
| | Oracle | 0.896(0.213) | 0.949(0.278) | 0.988(0.431) | 0.887(0.138) | 0.940(0.195) | 0.988(0.345) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.908**(0.225) | **0.956**(0.292) | **0.990**(0.449) | **0.903**(0.141) | **0.950**(0.198) | **0.988**(0.346) |
| | AICc | 0.528(0.382) | 0.622(0.499) | 0.775(0.774) | 0.555(0.306) | 0.649(0.431) | 0.804(0.764) |
| | BIC | 0.632(0.286) | 0.713(0.374) | 0.841(0.579) | 0.805(0.159) | 0.864(0.223) | 0.934(0.395) |
| | Vector L0 | 0.587(0.403) | 0.661(0.526) | 0.773(0.815) | 0.802(0.163) | 0.860(0.229) | 0.930(0.405) |
| | Oracle | 0.900(0.215) | 0.952(0.280) | 0.989(0.434) | 0.900(0.139) | 0.949(0.196) | 0.988(0.347) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.902**(0.221) | **0.949**(0.288) | **0.988**(0.443) | **0.911**(0.143) | **0.955**(0.200) | **0.990**(0.350) |
| | AICc | 0.535(0.381) | 0.630(0.497) | 0.784(0.771) | 0.571(0.308) | 0.671(0.433) | 0.826(0.767) |
| | BIC | 0.657(0.275) | 0.737(0.359) | 0.856(0.557) | 0.839(0.154) | 0.892(0.216) | 0.951(0.383) |
| | Vector L0 | 0.635(0.346) | 0.710(0.452) | 0.818(0.701) | 0.846(0.154) | 0.899(0.216) | 0.955(0.383) |
| | Oracle | 0.894(0.213) | 0.945(0.277) | 0.988(0.430) | 0.908(0.140) | 0.953(0.197) | 0.991(0.350) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.907**(0.232) | **0.952**(0.302) | **0.990**(0.464) | **0.910**(0.145) | **0.955**(0.202) | **0.990**(0.354) |
| | AICc | 0.507(0.379) | 0.598(0.495) | 0.752(0.767) | 0.530(0.309) | 0.625(0.435) | 0.783(0.771) |
| | BIC | 0.561(0.306) | 0.648(0.399) | 0.790(0.619) | 0.773(0.166) | 0.835(0.234) | 0.915(0.414) |
| | Vector L0 | 0.497(0.483) | 0.574(0.631) | 0.701(0.978) | 0.771(0.176) | 0.832(0.247) | 0.908(0.438) |
| | Oracle | 0.892(0.213) | 0.943(0.277) | 0.989(0.430) | 0.904(0.140) | 0.953(0.197) | 0.991(0.349) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.905**(0.223) | **0.951**(0.290) | **0.989**(0.445) | **0.899**(0.139) | **0.948**(0.195) | **0.990**(0.341) |
| | AICc | 0.524(0.386) | 0.621(0.504) | 0.779(0.781) | 0.553(0.306) | 0.653(0.430) | 0.809(0.762) |
| | BIC | 0.636(0.282) | 0.718(0.369) | 0.846(0.572) | 0.812(0.153) | 0.871(0.215) | 0.942(0.381) |
| | Vector L0 | 0.636(0.354) | 0.706(0.462) | 0.813(0.717) | 0.820(0.153) | 0.879(0.215) | 0.947(0.382) |
| | Oracle | 0.897(0.213) | 0.947(0.278) | 0.988(0.430) | 0.896(0.137) | 0.946(0.192) | 0.990(0.341) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.904**(0.222) | **0.951**(0.289) | **0.991**(0.445) | **0.889**(0.140) | **0.941**(0.196) | **0.987**(0.343) |
| | AICc | 0.541(0.392) | 0.637(0.512) | 0.796(0.793) | 0.553(0.309) | 0.656(0.434) | 0.818(0.770) |
| | BIC | 0.682(0.271) | 0.759(0.353) | 0.874(0.548) | 0.818(0.152) | 0.878(0.213) | 0.947(0.378) |
| | Vector L0 | 0.683(0.324) | 0.753(0.423) | 0.852(0.656) | 0.827(0.151) | 0.887(0.213) | 0.952(0.377) |
| | Oracle | 0.898(0.215) | 0.947(0.280) | 0.991(0.435) | 0.885(0.138) | 0.939(0.194) | 0.987(0.345) |

TABLE 2.2. Empirical coverage rates of confidence ellipsoids for $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.916**(0.190) | **0.961**(0.247) | **0.993**(0.380) | **0.901**(0.094) | **0.950**(0.132) | **0.989**(0.230) |
| | AICc | 0.529(0.317) | 0.619(0.414) | 0.763(0.641) | 0.554(0.202) | 0.648(0.284) | 0.798(0.505) |
| | BIC | 0.603(0.251) | 0.683(0.327) | 0.812(0.507) | 0.784(0.109) | 0.846(0.153) | 0.921(0.271) |
| | Vector L0 | 0.454(0.515) | 0.529(0.672) | 0.657(1.043) | 0.808(0.116) | 0.866(0.163) | 0.928(0.289) |
| | Oracle | 0.906(0.179) | 0.954(0.234) | 0.992(0.362) | 0.899(0.093) | 0.950(0.130) | 0.990(0.231) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.901**(0.188) | **0.948**(0.244) | **0.990**(0.375) | **0.902**(0.095) | **0.951**(0.132) | **0.990**(0.232) |
| | AICc | 0.527(0.319) | 0.618(0.417) | 0.770(0.646) | 0.560(0.206) | 0.657(0.290) | 0.808(0.514) |
| | BIC | 0.628(0.241) | 0.709(0.315) | 0.834(0.488) | 0.818(0.104) | 0.878(0.146) | 0.943(0.259) |
| | Vector L0 | 0.601(0.304) | 0.676(0.396) | 0.796(0.614) | 0.842(0.104) | 0.898(0.147) | 0.954(0.260) |
| | Oracle | 0.894(0.179) | 0.945(0.234) | 0.990(0.363) | 0.898(0.093) | 0.949(0.131) | 0.990(0.233) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.905**(0.187) | **0.952**(0.244) | **0.991**(0.374) | **0.898**(0.095) | **0.951**(0.133) | **0.992**(0.233) |
| | AICc | 0.545(0.327) | 0.639(0.427) | 0.789(0.662) | 0.570(0.205) | 0.667(0.288) | 0.817(0.510) |
| | BIC | 0.675(0.231) | 0.752(0.302) | 0.867(0.467) | 0.814(0.104) | 0.876(0.146) | 0.944(0.259) |
| | Vector L0 | 0.690(0.253) | 0.763(0.331) | 0.863(0.512) | 0.836(0.103) | 0.895(0.145) | 0.954(0.258) |
| | Oracle | 0.899(0.180) | 0.948(0.235) | 0.990(0.365) | 0.895(0.094) | 0.950(0.132) | 0.991(0.234) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.911**(0.200) | **0.952**(0.260) | **0.986**(0.399) | **0.896**(0.099) | **0.942**(0.138) | **0.981**(0.242) |
| | AICc | 0.522(0.320) | 0.610(0.418) | 0.758(0.647) | 0.547(0.205) | 0.640(0.288) | 0.788(0.510) |
| | BIC | 0.577(0.256) | 0.661(0.334) | 0.798(0.517) | 0.778(0.109) | 0.842(0.153) | 0.919(0.271) |
| | Vector L0 | 0.531(0.384) | 0.608(0.501) | 0.734(0.777) | 0.817(0.111) | 0.874(0.156) | 0.935(0.276) |
| | Oracle | 0.900(0.180) | 0.947(0.234) | 0.988(0.363) | 0.898(0.093) | 0.949(0.131) | 0.990(0.232) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.907**(0.187) | **0.955**(0.243) | **0.990**(0.373) | **0.893**(0.094) | **0.945**(0.131) | **0.988**(0.230) |
| | AICc | 0.546(0.324) | 0.638(0.423) | 0.788(0.656) | 0.563(0.207) | 0.658(0.292) | 0.814(0.517) |
| | BIC | 0.656(0.235) | 0.735(0.306) | 0.854(0.475) | 0.809(0.104) | 0.869(0.146) | 0.941(0.259) |
| | Vector L0 | 0.685(0.256) | 0.758(0.334) | 0.856(0.518) | 0.839(0.103) | 0.896(0.145) | 0.956(0.257) |
| | Oracle | 0.901(0.179) | 0.951(0.234) | 0.990(0.363) | 0.890(0.093) | 0.944(0.131) | 0.988(0.232) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.902**(0.183) | **0.950**(0.238) | **0.988**(0.366) | **0.887**(0.094) | **0.942**(0.132) | **0.989**(0.230) |
| | AICc | 0.553(0.325) | 0.648(0.424) | 0.799(0.657) | 0.572(0.207) | 0.671(0.291) | 0.826(0.516) |
| | BIC | 0.700(0.221) | 0.776(0.289) | 0.884(0.448) | 0.828(0.101) | 0.890(0.142) | 0.955(0.253) |
| | Vector L0 | 0.742(0.228) | 0.810(0.298) | 0.897(0.461) | 0.846(0.101) | 0.905(0.141) | 0.966(0.251) |
| | Oracle | 0.896(0.178) | 0.946(0.233) | 0.988(0.361) | 0.886(0.094) | 0.941(0.132) | 0.989(0.233) |

TABLE 2.3. Empirical coverage rates of confidence ellipsoids for $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.910**(0.141) | **0.955**(0.183) | **0.988**(0.281) | **0.897**(0.051) | **0.948**(0.072) | **0.988**(0.126) |
| | AICc | 0.530(0.237) | 0.619(0.310) | 0.764(0.480) | 0.557(0.110) | 0.650(0.155) | 0.797(0.274) |
| | BIC | 0.615(0.185) | 0.695(0.242) | 0.818(0.375) | 0.789(0.058) | 0.854(0.082) | 0.928(0.145) |
| | Vector L0 | 0.465(0.359) | 0.538(0.469) | 0.666(0.727) | 0.790(0.062) | 0.851(0.087) | 0.920(0.154) |
| | Oracle | 0.903(0.134) | 0.951(0.175) | 0.989(0.271) | 0.894(0.050) | 0.947(0.071) | 0.989(0.126) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.909**(0.141) | **0.955**(0.184) | **0.991**(0.283) | **0.906**(0.051) | **0.956**(0.071) | **0.991**(0.124) |
| | AICc | 0.538(0.243) | 0.628(0.317) | 0.777(0.491) | 0.569(0.110) | 0.662(0.155) | 0.810(0.274) |
| | BIC | 0.630(0.185) | 0.710(0.241) | 0.833(0.374) | 0.810(0.057) | 0.870(0.080) | 0.936(0.141) |
| | Vector L0 | 0.593(0.222) | 0.672(0.290) | 0.791(0.450) | 0.832(0.056) | 0.890(0.079) | 0.946(0.140) |
| | Oracle | 0.904(0.136) | 0.952(0.178) | 0.991(0.276) | 0.904(0.050) | 0.955(0.071) | 0.991(0.125) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.900**(0.138) | **0.951**(0.180) | **0.989**(0.276) | **0.904**(0.050) | **0.952**(0.071) | **0.990**(0.123) |
| | AICc | 0.545(0.242) | 0.638(0.316) | 0.786(0.490) | 0.581(0.110) | 0.676(0.154) | 0.824(0.274) |
| | BIC | 0.667(0.173) | 0.746(0.226) | 0.863(0.350) | 0.838(0.055) | 0.893(0.077) | 0.954(0.136) |
| | Vector L0 | 0.666(0.186) | 0.741(0.243) | 0.849(0.376) | 0.859(0.054) | 0.912(0.076) | 0.964(0.135) |
| | Oracle | 0.894(0.134) | 0.948(0.175) | 0.989(0.271) | 0.903(0.050) | 0.952(0.071) | 0.991(0.125) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.893**(0.150) | **0.936**(0.195) | **0.974**(0.300) | **0.859**(0.055) | **0.907**(0.078) | **0.955**(0.136) |
| | AICc | 0.522(0.240) | 0.612(0.313) | 0.758(0.485) | 0.550(0.111) | 0.644(0.156) | 0.788(0.277) |
| | BIC | 0.580(0.190) | 0.664(0.249) | 0.798(0.385) | 0.764(0.060) | 0.828(0.084) | 0.911(0.149) |
| | Vector L0 | 0.516(0.280) | 0.592(0.365) | 0.723(0.567) | 0.794(0.059) | 0.854(0.083) | 0.924(0.147) |
| | Oracle | 0.898(0.135) | 0.947(0.176) | 0.988(0.273) | 0.893(0.051) | 0.944(0.071) | 0.988(0.126) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.912**(0.138) | **0.957**(0.180) | **0.991**(0.276) | **0.903**(0.050) | **0.951**(0.070) | **0.990**(0.123) |
| | AICc | 0.541(0.243) | 0.631(0.317) | 0.781(0.491) | 0.568(0.112) | 0.663(0.157) | 0.812(0.279) |
| | BIC | 0.653(0.175) | 0.732(0.228) | 0.852(0.354) | 0.818(0.056) | 0.878(0.078) | 0.944(0.139) |
| | Vector L0 | 0.662(0.189) | 0.737(0.246) | 0.849(0.382) | 0.851(0.055) | 0.906(0.077) | 0.961(0.136) |
| | Oracle | 0.906(0.134) | 0.954(0.174) | 0.990(0.270) | 0.902(0.050) | 0.951(0.070) | 0.990(0.125) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.892**(0.137) | **0.944**(0.179) | **0.990**(0.275) | **0.891**(0.050) | **0.943**(0.070) | **0.987**(0.123) |
| | AICc | 0.551(0.245) | 0.644(0.320) | 0.793(0.496) | 0.581(0.110) | 0.678(0.155) | 0.833(0.274) |
| | BIC | 0.683(0.168) | 0.761(0.219) | 0.875(0.340) | 0.835(0.054) | 0.895(0.076) | 0.955(0.134) |
| | Vector L0 | 0.715(0.170) | 0.787(0.223) | 0.888(0.345) | 0.856(0.053) | 0.914(0.075) | 0.968(0.132) |
| | Oracle | 0.887(0.134) | 0.941(0.175) | 0.990(0.271) | 0.890(0.050) | 0.943(0.070) | 0.987(0.125) |

TABLE 2.4. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(14.118) | **0.949**(18.368) | **0.990**(28.236) | **0.899**(60.037) | **0.949**(83.912) | **0.989**(146.775) |
| | AICc | 0.869(13.490) | 0.931(17.614) | 0.985(27.305) | 0.876(57.713) | 0.934(81.159) | 0.985(143.938) |
| | BIC | 0.877(13.651) | 0.936(17.824) | 0.986(27.629) | 0.892(59.412) | 0.946(83.543) | 0.988(148.144) |
| | Vector L0 | 0.789(12.029) | 0.866(15.708) | 0.952(24.356) | 0.888(58.945) | 0.942(82.887) | 0.987(146.985) |
| | Oracle | 0.899(14.146) | 0.950(18.470) | 0.990(28.629) | 0.900(60.342) | 0.950(84.849) | 0.990(150.456) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.900**(14.103) | **0.949**(18.349) | **0.989**(28.207) | **0.898**(60.071) | **0.949**(83.960) | **0.989**(146.859) |
| | AICc | 0.874(13.543) | 0.932(17.683) | 0.984(27.412) | 0.877(58.117) | 0.936(81.727) | 0.985(144.945) |
| | BIC | 0.885(13.759) | 0.939(17.965) | 0.986(27.847) | 0.895(59.778) | 0.948(84.057) | 0.989(149.054) |
| | Vector L0 | 0.859(13.329) | 0.920(17.404) | 0.979(26.979) | 0.894(59.727) | 0.947(83.985) | 0.989(148.927) |
| | Oracle | 0.900(14.126) | 0.950(18.444) | 0.990(28.588) | 0.900(60.370) | 0.950(84.888) | 0.990(150.524) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.898**(14.048) | **0.948**(18.277) | **0.989**(28.096) | **0.897**(59.936) | **0.947**(83.771) | **0.989**(146.528) |
| | AICc | 0.874(13.532) | 0.932(17.669) | 0.983(27.390) | 0.879(58.261) | 0.936(81.929) | 0.985(145.304) |
| | BIC | 0.885(13.758) | 0.940(17.964) | 0.986(27.845) | 0.895(59.840) | 0.946(84.144) | 0.989(149.207) |
| | Vector L0 | 0.872(13.518) | 0.929(17.650) | 0.982(27.360) | 0.895(59.866) | 0.946(84.180) | 0.989(149.271) |
| | Oracle | 0.898(14.071) | 0.949(18.372) | 0.990(28.477) | 0.898(60.229) | 0.948(84.691) | 0.990(150.175) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(14.023) | **0.949**(18.244) | **0.989**(28.045) | **0.896**(60.063) | **0.948**(83.948) | **0.989**(146.838) |
| | AICc | 0.869(13.366) | 0.930(17.453) | 0.984(27.055) | 0.874(57.741) | 0.933(81.198) | 0.984(144.008) |
| | BIC | 0.877(13.534) | 0.935(17.671) | 0.986(27.392) | 0.891(59.541) | 0.945(83.724) | 0.988(148.464) |
| | Vector L0 | 0.838(12.841) | 0.905(16.768) | 0.972(25.995) | 0.889(59.414) | 0.943(83.546) | 0.988(148.150) |
| | Oracle | 0.898(14.041) | 0.949(18.333) | 0.990(28.416) | 0.897(60.343) | 0.949(84.850) | 0.989(150.458) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(14.044) | **0.949**(18.272) | **0.989**(28.089) | **0.899**(59.895) | **0.948**(83.714) | **0.989**(146.428) |
| | AICc | 0.870(13.490) | 0.932(17.614) | 0.984(27.305) | 0.880(58.070) | 0.935(81.661) | 0.985(144.828) |
| | BIC | 0.881(13.715) | 0.940(17.908) | 0.987(27.759) | 0.896(59.694) | 0.947(83.939) | 0.989(148.844) |
| | Vector L0 | 0.868(13.489) | 0.928(17.613) | 0.982(27.302) | 0.896(59.741) | 0.947(84.005) | 0.989(148.962) |
| | Oracle | 0.899(14.068) | 0.950(18.368) | 0.990(28.470) | 0.900(60.187) | 0.949(84.632) | 0.989(150.070) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(14.096) | **0.949**(18.339) | **0.989**(28.192) | **0.896**(59.824) | **0.946**(83.614) | **0.989**(146.254) |
| | AICc | 0.875(13.601) | 0.933(17.759) | 0.985(27.530) | 0.876(58.189) | 0.934(81.829) | 0.985(145.126) |
| | BIC | 0.888(13.849) | 0.942(18.083) | 0.987(28.029) | 0.894(59.734) | 0.945(83.994) | 0.989(148.942) |
| | Vector L0 | 0.878(13.697) | 0.935(17.884) | 0.985(27.722) | 0.894(59.791) | 0.946(84.074) | 0.989(149.084) |
| | Oracle | 0.899(14.119) | 0.950(18.434) | 0.990(28.573) | 0.896(60.118) | 0.948(84.534) | 0.990(149.896) |

TABLE 2.5. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(11.822) | **0.949**(15.381) | **0.990**(23.644) | **0.898**(40.284) | **0.947**(56.304) | **0.988**(98.485) |
| | AICc | 0.871(11.306) | 0.930(14.762) | 0.984(22.884) | 0.876(38.891) | 0.934(54.691) | 0.984(96.996) |
| | BIC | 0.880(11.454) | 0.936(14.956) | 0.985(23.183) | 0.893(39.987) | 0.945(56.228) | 0.988(99.706) |
| | Vector L0 | 0.801(10.314) | 0.873(13.469) | 0.950(20.883) | 0.891(39.902) | 0.943(56.108) | 0.987(99.495) |
| | Oracle | 0.899(11.843) | 0.950(15.462) | 0.991(23.967) | 0.899(40.477) | 0.949(56.917) | 0.989(100.925) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.897**(11.787) | **0.947**(15.335) | **0.989**(23.573) | **0.900**(40.328) | **0.949**(56.365) | **0.989**(98.592) |
| | AICc | 0.872(11.308) | 0.929(14.765) | 0.983(22.888) | 0.881(39.080) | 0.936(54.957) | 0.985(97.468) |
| | BIC | 0.882(11.482) | 0.937(14.992) | 0.986(23.238) | 0.897(40.210) | 0.947(56.541) | 0.989(100.260) |
| | Vector L0 | 0.864(11.258) | 0.924(14.700) | 0.980(22.787) | 0.897(40.250) | 0.948(56.597) | 0.989(100.361) |
| | Oracle | 0.897(11.807) | 0.949(15.417) | 0.990(23.896) | 0.901(40.530) | 0.951(56.991) | 0.990(101.057) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.898**(11.843) | **0.949**(15.408) | **0.989**(23.686) | **0.898**(40.207) | **0.948**(56.196) | **0.988**(98.295) |
| | AICc | 0.873(11.415) | 0.931(14.905) | 0.984(23.105) | 0.878(39.070) | 0.936(54.942) | 0.985(97.441) |
| | BIC | 0.885(11.615) | 0.941(15.166) | 0.986(23.507) | 0.895(40.091) | 0.947(56.374) | 0.989(99.965) |
| | Vector L0 | 0.880(11.545) | 0.935(15.074) | 0.984(23.367) | 0.895(40.145) | 0.947(56.449) | 0.989(100.098) |
| | Oracle | 0.898(11.862) | 0.949(15.488) | 0.990(24.007) | 0.899(40.404) | 0.949(56.814) | 0.989(100.744) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(11.854) | **0.948**(15.422) | **0.990**(23.707) | **0.899**(40.393) | **0.949**(56.456) | **0.989**(98.750) |
| | AICc | 0.870(11.285) | 0.930(14.735) | 0.983(22.842) | 0.875(38.790) | 0.934(54.549) | 0.985(96.745) |
| | BIC | 0.879(11.428) | 0.935(14.921) | 0.985(23.129) | 0.895(39.961) | 0.946(56.192) | 0.989(99.643) |
| | Vector L0 | 0.844(10.935) | 0.907(14.279) | 0.971(22.136) | 0.894(40.019) | 0.946(56.273) | 0.988(99.786) |
| | Oracle | 0.898(11.847) | 0.949(15.468) | 0.990(23.976) | 0.900(40.451) | 0.951(56.879) | 0.990(100.859) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.902**(11.800) | **0.950**(15.353) | **0.989**(23.601) | **0.899**(40.195) | **0.948**(56.180) | **0.989**(98.268) |
| | AICc | 0.877(11.350) | 0.933(14.821) | 0.984(22.975) | 0.880(39.038) | 0.935(54.898) | 0.986(97.363) |
| | BIC | 0.888(11.539) | 0.941(15.066) | 0.987(23.353) | 0.895(40.074) | 0.947(56.350) | 0.989(99.923) |
| | Vector L0 | 0.883(11.487) | 0.937(14.999) | 0.985(23.250) | 0.896(40.162) | 0.947(56.474) | 0.989(100.141) |
| | Oracle | 0.903(11.819) | 0.950(15.432) | 0.989(23.919) | 0.899(40.392) | 0.949(56.797) | 0.990(100.713) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(11.804) | **0.947**(15.358) | **0.989**(23.609) | **0.898**(40.256) | **0.947**(56.265) | **0.989**(98.415) |
| | AICc | 0.873(11.407) | 0.932(14.895) | 0.984(23.090) | 0.879(39.246) | 0.937(55.190) | 0.986(97.881) |
| | BIC | 0.885(11.618) | 0.940(15.169) | 0.987(23.512) | 0.896(40.228) | 0.946(56.567) | 0.989(100.307) |
| | Vector L0 | 0.885(11.616) | 0.940(15.167) | 0.986(23.510) | 0.896(40.291) | 0.947(56.654) | 0.989(100.461) |
| | Oracle | 0.897(11.823) | 0.948(15.437) | 0.989(23.927) | 0.898(40.449) | 0.948(56.877) | 0.989(100.855) |

TABLE 2.6. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.896**(8.868) | **0.947**(11.538) | **0.988**(17.737) | **0.895**(21.949) | **0.947**(30.678) | **0.989**(53.661) |
| | AICc | 0.870(8.484) | 0.929(11.078) | 0.982(17.173) | 0.875(21.182) | 0.934(29.787) | 0.985(52.828) |
| | BIC | 0.880(8.605) | 0.935(11.235) | 0.985(17.415) | 0.892(21.806) | 0.944(30.662) | 0.988(54.372) |
| | Vector L0 | 0.811(7.844) | 0.878(10.242) | 0.955(15.880) | 0.889(21.743) | 0.943(30.574) | 0.987(54.215) |
| | Oracle | 0.896(8.880) | 0.948(11.594) | 0.989(17.971) | 0.897(22.039) | 0.948(30.991) | 0.989(54.953) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.901**(8.893) | **0.950**(11.570) | **0.990**(17.785) | **0.897**(21.808) | **0.948**(30.481) | **0.989**(53.316) |
| | AICc | 0.876(8.532) | 0.933(11.140) | 0.984(17.270) | 0.879(21.129) | 0.935(29.712) | 0.985(52.695) |
| | BIC | 0.885(8.656) | 0.939(11.302) | 0.986(17.520) | 0.894(21.712) | 0.946(30.531) | 0.989(54.139) |
| | Vector L0 | 0.870(8.506) | 0.929(11.106) | 0.982(17.216) | 0.894(21.746) | 0.946(30.579) | 0.989(54.223) |
| | Oracle | 0.901(8.908) | 0.951(11.631) | 0.990(18.028) | 0.898(21.915) | 0.949(30.815) | 0.990(54.641) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.895**(8.853) | **0.948**(11.518) | **0.989**(17.707) | **0.898**(21.815) | **0.949**(30.490) | **0.989**(53.332) |
| | AICc | 0.872(8.532) | 0.932(11.140) | 0.984(17.270) | 0.880(21.227) | 0.937(29.850) | 0.986(52.939) |
| | BIC | 0.883(8.675) | 0.940(11.327) | 0.987(17.557) | 0.896(21.788) | 0.948(30.638) | 0.989(54.328) |
| | Vector L0 | 0.879(8.633) | 0.937(11.272) | 0.986(17.472) | 0.897(21.823) | 0.948(30.686) | 0.989(54.413) |
| | Oracle | 0.896(8.867) | 0.949(11.577) | 0.990(17.945) | 0.899(21.921) | 0.950(30.824) | 0.990(54.658) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(8.905) | **0.948**(11.586) | **0.988**(17.811) | **0.898**(22.113) | **0.948**(30.907) | **0.988**(54.062) |
| | AICc | 0.869(8.455) | 0.928(11.040) | 0.982(17.113) | 0.877(21.093) | 0.934(29.662) | 0.984(52.607) |
| | BIC | 0.876(8.566) | 0.935(11.184) | 0.984(17.337) | 0.892(21.711) | 0.945(30.529) | 0.988(54.136) |
| | Vector L0 | 0.845(8.223) | 0.910(10.738) | 0.970(16.646) | 0.893(21.769) | 0.945(30.610) | 0.988(54.279) |
| | Oracle | 0.898(8.874) | 0.949(11.587) | 0.989(17.960) | 0.899(21.991) | 0.949(30.922) | 0.989(54.831) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(8.870) | **0.950**(11.540) | **0.989**(17.740) | **0.898**(21.918) | **0.947**(30.634) | **0.988**(53.583) |
| | AICc | 0.874(8.530) | 0.932(11.137) | 0.984(17.265) | 0.878(21.270) | 0.934(29.911) | 0.984(53.048) |
| | BIC | 0.885(8.676) | 0.941(11.329) | 0.987(17.560) | 0.895(21.852) | 0.947(30.727) | 0.988(54.486) |
| | Vector L0 | 0.880(8.639) | 0.938(11.279) | 0.986(17.484) | 0.896(21.908) | 0.947(30.806) | 0.989(54.625) |
| | Oracle | 0.900(8.885) | 0.950(11.601) | 0.990(17.981) | 0.898(22.024) | 0.949(30.969) | 0.989(54.914) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.901**(8.864) | **0.950**(11.532) | **0.990**(17.728) | **0.898**(21.761) | **0.948**(30.415) | **0.988**(53.200) |
| | AICc | 0.878(8.563) | 0.935(11.181) | 0.985(17.333) | 0.881(21.231) | 0.936(29.856) | 0.985(52.950) |
| | BIC | 0.890(8.716) | 0.943(11.381) | 0.988(17.640) | 0.897(21.751) | 0.948(30.585) | 0.988(54.234) |
| | Vector L0 | 0.890(8.716) | 0.943(11.381) | 0.987(17.641) | 0.897(21.794) | 0.948(30.645) | 0.989(54.341) |
| | Oracle | 0.902(8.879) | 0.951(11.593) | 0.991(17.969) | 0.898(21.867) | 0.949(30.748) | 0.989(54.522) |

TABLE 2.7. Bias of the various estimates of $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSPE | MSMD | MSPE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.031**(2.108) | **2.004**(1.997) | **3.058**(2.704) | **3.006**(2.452) |
| | AICc | 2.189(2.288) | 2.138(2.130) | 3.260(2.917) | 3.158(2.576) |
| | BIC | 2.140(2.239) | 2.097(2.093) | 3.113(2.766) | 3.048(2.487) |
| | Vector L0 | 2.561(2.758) | 2.462(2.495) | 3.156(2.820) | 3.078(2.515) |
| | Oracle | 2.029(2.106) | 2.002(1.995) | 3.057(2.702) | 3.006(2.451) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.018**(2.096) | **1.993**(1.988) | **3.038**(2.683) | **2.993**(2.433) |
| | AICc | 2.171(2.280) | 2.121(2.121) | 3.222(2.900) | 3.132(2.557) |
| | BIC | 2.107(2.204) | 2.067(2.065) | 3.073(2.722) | 3.019(2.454) |
| | Vector L0 | 2.239(2.379) | 2.181(2.194) | 3.080(2.732) | 3.023(2.459) |
| | Oracle | 2.017(2.096) | 1.992(1.987) | 3.037(2.682) | 2.993(2.432) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.028**(2.117) | **1.997**(1.999) | **3.046**(2.715) | **2.998**(2.462) |
| | AICc | 2.177(2.296) | 2.122(2.129) | 3.226(2.911) | 3.135(2.579) |
| | BIC | 2.105(2.212) | 2.062(2.069) | 3.070(2.741) | 3.016(2.477) |
| | Vector L0 | 2.172(2.314) | 2.121(2.146) | 3.071(2.745) | 3.016(2.478) |
| | Oracle | 2.028(2.116) | 1.997(1.998) | 3.045(2.715) | 2.998(2.461) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.021**(2.093) | **1.993**(1.985) | **3.056**(2.733) | **3.005**(2.458) |
| | AICc | 2.182(2.270) | 2.129(2.114) | 3.266(2.969) | 3.161(2.593) |
| | BIC | 2.136(2.219) | 2.090(2.077) | 3.106(2.793) | 3.043(2.493) |
| | Vector L0 | 2.337(2.478) | 2.260(2.266) | 3.123(2.817) | 3.053(2.503) |
| | Oracle | 2.018(2.088) | 1.991(1.982) | 3.054(2.729) | 3.004(2.457) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.024**(2.109) | **1.996**(1.990) | **3.030**(2.702) | **2.991**(2.456) |
| | AICc | 2.179(2.285) | 2.127(2.119) | 3.218(2.914) | 3.134(2.579) |
| | BIC | 2.107(2.204) | 2.067(2.061) | 3.060(2.734) | 3.014(2.475) |
| | Vector L0 | 2.188(2.319) | 2.137(2.150) | 3.061(2.737) | 3.013(2.474) |
| | Oracle | 2.023(2.109) | 1.995(1.990) | 3.029(2.702) | 2.990(2.456) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.032**(2.129) | **1.999**(2.004) | **3.058**(2.727) | **3.011**(2.466) |
| | AICc | 2.178(2.302) | 2.123(2.134) | 3.235(2.913) | 3.151(2.584) |
| | BIC | 2.094(2.203) | 2.052(2.061) | 3.083(2.755) | 3.031(2.483) |
| | Vector L0 | 2.146(2.264) | 2.098(2.109) | 3.085(2.754) | 3.030(2.481) |
| | Oracle | 2.031(2.127) | 1.998(2.003) | 3.057(2.726) | 3.010(2.465) |

TABLE 2.8. Bias of the various estimates of $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSPE | MSMD | MSPE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.036**(2.358) | **1.999**(1.978) | **3.048**(3.263) | **3.008**(2.458) |
| | AICc | 2.225(2.610) | 2.129(2.108) | 3.309(3.619) | 3.156(2.586) |
| | BIC | 2.163(2.530) | 2.085(2.067) | 3.109(3.355) | 3.044(2.491) |
| | Vector L0 | 2.595(3.186) | 2.393(2.438) | 3.122(3.369) | 3.048(2.493) |
| | Oracle | 2.035(2.356) | 1.997(1.977) | 3.045(3.260) | 3.007(2.456) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.037**(2.372) | **2.008**(2.002) | **3.021**(3.247) | **2.984**(2.449) |
| | AICc | 2.223(2.604) | 2.140(2.125) | 3.261(3.544) | 3.124(2.562) |
| | BIC | 2.142(2.509) | 2.082(2.074) | 3.060(3.298) | 3.008(2.469) |
| | Vector L0 | 2.249(2.684) | 2.154(2.165) | 3.057(3.303) | 3.005(2.468) |
| | Oracle | 2.036(2.370) | 2.007(2.001) | 3.020(3.246) | 2.983(2.447) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.028**(2.360) | **2.003**(2.013) | **3.056**(3.303) | **3.011**(2.461) |
| | AICc | 2.202(2.591) | 2.126(2.137) | 3.286(3.616) | 3.145(2.580) |
| | BIC | 2.110(2.473) | 2.061(2.075) | 3.091(3.348) | 3.032(2.479) |
| | Vector L0 | 2.150(2.542) | 2.088(2.110) | 3.089(3.349) | 3.029(2.477) |
| | Oracle | 2.028(2.359) | 2.002(2.011) | 3.056(3.303) | 3.011(2.461) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.051**(2.417) | **2.010**(2.003) | **3.065**(3.293) | **2.995**(2.441) |
| | AICc | 2.229(2.651) | 2.135(2.131) | 3.309(3.610) | 3.138(2.565) |
| | BIC | 2.173(2.580) | 2.095(2.092) | 3.110(3.350) | 3.025(2.466) |
| | Vector L0 | 2.393(2.936) | 2.247(2.283) | 3.110(3.362) | 3.024(2.468) |
| | Oracle | 2.041(2.400) | 2.004(1.996) | 3.046(3.268) | 2.987(2.434) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.011**(2.362) | **1.983**(1.989) | **3.026**(3.240) | **2.981**(2.434) |
| | AICc | 2.188(2.599) | 2.109(2.117) | 3.267(3.555) | 3.119(2.555) |
| | BIC | 2.108(2.491) | 2.052(2.058) | 3.069(3.295) | 3.005(2.455) |
| | Vector L0 | 2.136(2.526) | 2.070(2.075) | 3.063(3.283) | 3.001(2.451) |
| | Oracle | 2.010(2.360) | 1.982(1.987) | 3.026(3.240) | 2.980(2.434) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.035**(2.374) | **2.009**(2.012) | **3.047**(3.255) | **3.012**(2.465) |
| | AICc | 2.206(2.602) | 2.131(2.136) | 3.262(3.530) | 3.143(2.567) |
| | BIC | 2.105(2.468) | 2.059(2.065) | 3.077(3.297) | 3.030(2.480) |
| | Vector L0 | 2.122(2.505) | 2.069(2.083) | 3.072(3.290) | 3.026(2.478) |
| | Oracle | 2.035(2.373) | 2.008(2.012) | 3.046(3.254) | 3.012(2.465) |

TABLE 2.9. Bias of the various estimates of $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSPE | MSMD | MSPE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.053**(2.643) | **2.022**(2.033) | **3.103**(3.816) | **3.029**(2.463) |
| | AICc | 2.256(2.921) | 2.149(2.163) | 3.372(4.212) | 3.162(2.579) |
| | BIC | 2.179(2.817) | 2.102(2.113) | 3.159(3.902) | 3.058(2.486) |
| | Vector L0 | 2.594(3.441) | 2.358(2.400) | 3.190(3.953) | 3.068(2.497) |
| | Oracle | 2.050(2.636) | 2.020(2.029) | 3.098(3.808) | 3.027(2.461) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.030**(2.587) | **1.991**(1.986) | **3.029**(3.728) | **2.998**(2.446) |
| | AICc | 2.227(2.867) | 2.118(2.120) | 3.300(4.107) | 3.133(2.563) |
| | BIC | 2.149(2.760) | 2.068(2.070) | 3.081(3.806) | 3.024(2.472) |
| | Vector L0 | 2.250(2.917) | 2.127(2.140) | 3.082(3.815) | 3.022(2.470) |
| | Oracle | 2.029(2.587) | 1.991(1.985) | 3.029(3.728) | 2.998(2.446) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **2.026**(2.589) | **2.004**(1.994) | **3.039**(3.738) | **2.996**(2.432) |
| | AICc | 2.223(2.857) | 2.129(2.119) | 3.299(4.097) | 3.131(2.552) |
| | BIC | 2.115(2.723) | 2.061(2.056) | 3.075(3.791) | 3.016(2.450) |
| | Vector L0 | 2.159(2.797) | 2.085(2.088) | 3.072(3.789) | 3.012(2.447) |
| | Oracle | 2.025(2.587) | 2.003(1.993) | 3.039(3.739) | 2.996(2.432) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.058**(2.627) | **2.017**(2.021) | **3.109**(3.870) | **3.009**(2.468) |
| | AICc | 2.244(2.886) | 2.136(2.146) | 3.344(4.193) | 3.132(2.574) |
| | BIC | 2.178(2.793) | 2.095(2.101) | 3.124(3.882) | 3.025(2.484) |
| | Vector L0 | 2.404(3.148) | 2.231(2.262) | 3.133(3.912) | 3.024(2.486) |
| | Oracle | 2.034(2.591) | 2.005(2.009) | 3.047(3.768) | 2.987(2.449) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.020**(2.602) | **1.989**(1.989) | **3.020**(3.721) | **3.007**(2.461) |
| | AICc | 2.222(2.876) | 2.118(2.119) | 3.289(4.106) | 3.147(2.588) |
| | BIC | 2.126(2.751) | 2.056(2.058) | 3.062(3.772) | 3.028(2.477) |
| | Vector L0 | 2.159(2.807) | 2.073(2.082) | 3.053(3.762) | 3.022(2.472) |
| | Oracle | 2.018(2.600) | 1.988(1.987) | 3.019(3.721) | 3.006(2.461) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **2.002**(2.550) | **1.976**(1.971) | **3.041**(3.755) | **2.997**(2.454) |
| | AICc | 2.190(2.828) | 2.097(2.099) | 3.296(4.116) | 3.125(2.562) |
| | BIC | 2.083(2.676) | 2.029(2.030) | 3.072(3.800) | 3.012(2.466) |
| | Vector L0 | 2.092(2.693) | 2.031(2.033) | 3.064(3.791) | 3.007(2.464) |
| | Oracle | 2.001(2.548) | 1.975(1.970) | 3.041(3.754) | 2.996(2.454) |

TABLE 2.10. Bias of the various estimates of $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSE | MSMD | MSE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.032**(0.053) | **2.404**(18.043) | **0.049**(0.071) | **3.257**(10.050) |
| | AICc | 0.190(0.253) | 29.681(348.489) | 0.250(0.318) | 30.016(130.280) |
| | BIC | 0.141(0.221) | 23.217(393.831) | 0.104(0.194) | 10.531(55.054) |
| | Vector L0 | 0.565(0.798) | 98.338(1058.719) | 0.142(0.283) | 16.043(176.468) |
| | Oracle | 0.030(0.048) | 2.016(2.011) | 0.048(0.069) | 3.091(2.580) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.031**(0.048) | **2.069**(2.402) | **0.045**(0.066) | **3.004**(3.662) |
| | AICc | 0.185(0.248) | 26.352(144.121) | 0.239(0.306) | 30.966(188.171) |
| | BIC | 0.117(0.193) | 15.883(83.732) | 0.083(0.156) | 9.041(133.110) |
| | Vector L0 | 0.248(0.467) | 37.265(342.002) | 0.091(0.188) | 9.613(131.531) |
| | Oracle | 0.030(0.048) | 2.004(2.026) | 0.045(0.065) | 2.943(2.501) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.032**(0.052) | **2.142**(4.360) | **0.046**(0.064) | **3.008**(2.887) |
| | AICc | 0.180(0.240) | 25.328(119.710) | 0.224(0.289) | 28.790(155.721) |
| | BIC | 0.107(0.180) | 13.984(76.042) | 0.070(0.130) | 6.563(52.675) |
| | Vector L0 | 0.182(0.355) | 26.125(157.472) | 0.071(0.143) | 6.414(31.240) |
| | Oracle | 0.032(0.051) | 2.046(2.082) | 0.045(0.063) | 2.941(2.401) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.034**(0.062) | **2.308**(4.253) | **0.047**(0.089) | **3.058**(3.160) |
| | AICc | 0.196(0.262) | 31.034(507.979) | 0.253(0.323) | 32.058(150.864) |
| | BIC | 0.147(0.228) | 26.038(949.904) | 0.097(0.179) | 10.596(99.732) |
| | Vector L0 | 0.352(0.586) | 62.648(1543.818) | 0.114(0.240) | 11.610(65.063) |
| | Oracle | 0.031(0.051) | 2.040(2.063) | 0.046(0.068) | 2.951(2.410) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.031**(0.052) | **2.212**(18.045) | **0.046**(0.066) | **3.053**(3.022) |
| | AICc | 0.185(0.245) | 28.051(309.836) | 0.233(0.298) | 29.997(197.420) |
| | BIC | 0.114(0.188) | 15.923(131.636) | 0.076(0.137) | 7.294(43.527) |
| | Vector L0 | 0.188(0.374) | 28.900(346.941) | 0.077(0.148) | 7.005(32.826) |
| | Oracle | 0.030(0.049) | 1.987(2.025) | 0.046(0.065) | 3.000(2.456) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.031**(0.049) | **2.065**(4.983) | **0.049**(0.073) | **3.199**(4.304) |
| | AICc | 0.178(0.235) | 26.542(171.117) | 0.228(0.291) | 33.539(989.819) |
| | BIC | 0.095(0.162) | 13.313(134.134) | 0.073(0.133) | 6.874(45.766) |
| | Vector L0 | 0.147(0.302) | 21.642(189.043) | 0.073(0.144) | 6.395(31.247) |
| | Oracle | 0.030(0.048) | 1.969(1.986) | 0.048(0.072) | 3.115(2.553) |

TABLE 2.11. Bias of the various estimates of $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSE | MSMD | MSE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.031**(0.057) | **2.124**(5.733) | **0.047**(0.096) | **3.109**(3.344) |
| | AICc | 0.218(0.313) | 57.641(6919.174) | 0.296(0.422) | 28.919(114.599) |
| | BIC | 0.154(0.260) | 48.798(6917.275) | 0.106(0.227) | 9.334(57.205) |
| | Vector L0 | 0.584(1.008) | 87.937(3434.255) | 0.124(0.344) | 10.576(79.387) |
| | Oracle | 0.030(0.052) | 1.939(1.919) | 0.045(0.073) | 3.011(2.452) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.032**(0.054) | **2.156**(3.643) | **0.047**(0.076) | **3.087**(3.782) |
| | AICc | 0.216(0.310) | 26.482(122.579) | 0.291(0.413) | 29.436(125.336) |
| | BIC | 0.138(0.246) | 16.148(86.074) | 0.085(0.186) | 7.158(35.498) |
| | Vector L0 | 0.245(0.530) | 29.253(171.468) | 0.084(0.214) | 7.023(50.119) |
| | Oracle | 0.031(0.052) | 2.026(2.019) | 0.046(0.074) | 3.018(2.470) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.031**(0.056) | **2.161**(8.506) | **0.048**(0.079) | **3.075**(3.409) |
| | AICc | 0.207(0.294) | 25.380(146.628) | 0.281(0.401) | 27.724(146.376) |
| | BIC | 0.114(0.210) | 13.813(111.259) | 0.085(0.180) | 7.062(36.654) |
| | Vector L0 | 0.153(0.369) | 19.688(303.273) | 0.083(0.196) | 6.422(35.355) |
| | Oracle | 0.030(0.053) | 2.001(1.995) | 0.048(0.078) | 3.016(2.448) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.040**(0.096) | **2.477**(4.423) | **0.065**(0.190) | **3.533**(4.410) |
| | AICc | 0.224(0.321) | 29.566(466.661) | 0.308(0.436) | 32.441(217.630) |
| | BIC | 0.165(0.273) | 20.749(222.568) | 0.107(0.230) | 9.355(52.112) |
| | Vector L0 | 0.378(0.748) | 51.046(663.619) | 0.111(0.309) | 9.449(73.465) |
| | Oracle | 0.031(0.052) | 2.006(2.050) | 0.046(0.076) | 2.993(2.457) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.031**(0.055) | **2.080**(4.465) | **0.047**(0.074) | **3.064**(2.574) |
| | AICc | 0.209(0.300) | 26.972(359.639) | 0.285(0.400) | 30.611(285.241) |
| | BIC | 0.125(0.227) | 14.780(95.705) | 0.086(0.180) | 8.094(153.809) |
| | Vector L0 | 0.159(0.358) | 18.284(120.087) | 0.081(0.205) | 6.845(52.639) |
| | Oracle | 0.030(0.052) | 1.982(2.009) | 0.047(0.074) | 3.046(2.490) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.031**(0.054) | **2.054**(2.187) | **0.048**(0.075) | **3.111**(2.602) |
| | AICc | 0.202(0.291) | 24.395(135.799) | 0.272(0.396) | 28.593(169.078) |
| | BIC | 0.101(0.193) | 12.278(106.485) | 0.074(0.160) | 6.106(28.585) |
| | Vector L0 | 0.112(0.272) | 13.905(142.584) | 0.070(0.167) | 5.381(25.746) |
| | Oracle | 0.031(0.054) | 2.027(2.056) | 0.047(0.075) | 3.096(2.495) |

TABLE 2.12. Bias of the various estimates of $E(Y_i|\boldsymbol{x}_i)$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are standard errors. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | $m = 2$ | | $m = 3$ | |
|---|---|---|---|---|---|
| | | MSE | MSMD | MSE | MSMD |
| $b = 1/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.034**(0.095) | **2.177**(3.420) | **0.050**(0.121) | **3.177**(3.413) |
| | AICc | 0.239(0.348) | 27.768(243.401) | 0.335(0.496) | 31.358(252.820) |
| | BIC | 0.164(0.290) | 18.488(160.758) | 0.109(0.254) | 10.107(201.409) |
| | Vector L0 | 0.589(1.061) | 68.741(834.504) | 0.139(0.385) | 12.166(219.127) |
| | Oracle | 0.031(0.057) | 1.983(2.022) | 0.046(0.083) | 3.049(2.492) |
| $b = 2/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.032**(0.060) | **2.201**(29.418) | **0.045**(0.079) | **2.985**(2.612) |
| | AICc | 0.233(0.342) | 26.088(163.066) | 0.324(0.479) | 28.679(118.796) |
| | BIC | 0.153(0.274) | 16.382(103.300) | 0.099(0.237) | 8.071(49.576) |
| | Vector L0 | 0.255(0.540) | 27.177(192.115) | 0.098(0.265) | 7.471(52.465) |
| | Oracle | 0.031(0.056) | 1.964(1.969) | 0.045(0.079) | 2.959(2.393) |
| $b = 3/\sqrt{3}$ $\rho_X = 0$ | GMTask | **0.032**(0.061) | **2.084**(2.209) | **0.046**(0.080) | **2.982**(2.478) |
| | AICc | 0.229(0.343) | 26.803(225.713) | 0.307(0.459) | 27.775(180.262) |
| | BIC | 0.128(0.250) | 13.431(117.135) | 0.079(0.176) | 6.083(32.481) |
| | Vector L0 | 0.168(0.373) | 17.201(129.633) | 0.076(0.191) | 5.601(33.885) |
| | Oracle | 0.032(0.060) | 2.043(2.043) | 0.046(0.080) | 2.961(2.375) |
| $b = 1/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.053**(0.164) | **2.800**(4.564) | **0.108**(0.314) | **4.605**(21.800) |
| | AICc | 0.247(0.362) | 27.534(176.748) | 0.347(0.512) | 31.904(339.281) |
| | BIC | 0.178(0.305) | 19.105(105.692) | 0.123(0.277) | 10.112(94.750) |
| | Vector L0 | 0.400(0.776) | 42.118(221.185) | 0.130(0.367) | 9.245(63.801) |
| | Oracle | 0.031(0.059) | 2.001(2.026) | 0.047(0.084) | 3.082(2.508) |
| $b = 2/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.031**(0.060) | **2.052**(4.085) | **0.046**(0.083) | **2.963**(2.529) |
| | AICc | 0.230(0.337) | 26.147(179.569) | 0.322(0.476) | 29.752(158.191) |
| | BIC | 0.134(0.250) | 14.680(94.019) | 0.090(0.209) | 7.273(43.387) |
| | Vector L0 | 0.170(0.375) | 19.664(304.231) | 0.081(0.217) | 5.899(33.531) |
| | Oracle | 0.030(0.056) | 1.968(1.948) | 0.046(0.082) | 2.947(2.426) |
| $b = 3/\sqrt{3}$ $\rho_X = 0.5$ | GMTask | **0.032**(0.060) | **2.117**(2.300) | **0.048**(0.091) | **3.056**(2.611) |
| | AICc | 0.222(0.328) | 25.555(194.531) | 0.297(0.451) | 26.682(159.774) |
| | BIC | 0.115(0.223) | 12.585(101.191) | 0.075(0.179) | 6.253(93.653) |
| | Vector L0 | 0.125(0.300) | 14.054(195.894) | 0.070(0.175) | 5.770(179.730) |
| | Oracle | 0.032(0.058) | 2.078(2.073) | 0.047(0.091) | 3.045(2.549) |

TABLE 2.13. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{2001}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(17.258) | **0.947**(22.453) | **0.988**(34.516) | **0.897**(75.930) | **0.947**(106.126) | **0.988**(185.630) |
| | AICc | 0.861(16.232) | 0.923(21.195) | 0.980(32.856) | 0.868(72.178) | 0.929(101.499) | 0.982(180.009) |
| | BIC | 0.873(16.496) | 0.931(21.539) | 0.983(33.386) | 0.892(75.172) | 0.944(105.701) | 0.988(187.423) |
| | Vector L0 | 0.742(13.808) | 0.823(18.032) | 0.921(27.962) | 0.888(74.874) | 0.942(105.282) | 0.986(186.682) |
| | Oracle | 0.897(14.159) | 0.950(18.487) | 0.990(28.655) | 0.899(60.034) | 0.948(84.417) | 0.989(149.689) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.899**(24.468) | **0.948**(31.833) | **0.989**(48.935) | **0.896**(110.993) | **0.948**(155.132) | **0.990**(271.350) |
| | AICc | 0.863(23.080) | 0.924(30.137) | 0.980(46.717) | 0.868(105.864) | 0.928(148.870) | 0.983(264.022) |
| | BIC | 0.876(23.525) | 0.933(30.716) | 0.984(47.611) | 0.891(110.095) | 0.944(154.805) | 0.989(274.492) |
| | Vector L0 | 0.820(21.853) | 0.887(28.536) | 0.961(44.239) | 0.890(110.116) | 0.945(154.835) | 0.989(274.544) |
| | Oracle | 0.900(14.110) | 0.950(18.423) | 0.989(28.555) | 0.899(59.934) | 0.949(84.276) | 0.990(149.439) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.897**(33.166) | **0.948**(43.150) | **0.989**(66.332) | **0.894**(152.372) | **0.946**(212.966) | **0.988**(372.511) |
| | AICc | 0.863(31.311) | 0.924(40.883) | 0.981(63.375) | 0.867(145.383) | 0.927(204.444) | 0.982(362.586) |
| | BIC | 0.876(31.916) | 0.932(41.672) | 0.984(64.594) | 0.889(151.105) | 0.944(212.470) | 0.988(376.740) |
| | Vector L0 | 0.835(30.245) | 0.899(39.492) | 0.967(61.223) | 0.888(151.005) | 0.943(212.330) | 0.987(376.492) |
| | Oracle | 0.899(14.128) | 0.950(18.446) | 0.990(28.591) | 0.897(60.125) | 0.948(84.544) | 0.989(149.915) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.901**(17.049) | **0.949**(22.182) | **0.990**(34.099) | **0.898**(74.658) | **0.948**(104.348) | **0.989**(182.521) |
| | AICc | 0.871(15.889) | 0.930(20.747) | 0.983(32.161) | 0.874(70.144) | 0.932(98.639) | 0.984(174.939) |
| | BIC | 0.877(16.098) | 0.935(21.019) | 0.985(32.581) | 0.891(72.461) | 0.945(101.890) | 0.988(180.675) |
| | Vector L0 | 0.799(14.465) | 0.872(18.888) | 0.953(29.286) | 0.889(72.203) | 0.942(101.529) | 0.987(180.036) |
| | Oracle | 0.900(14.145) | 0.950(18.469) | 0.991(28.626) | 0.899(60.306) | 0.951(84.799) | 0.990(150.366) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(22.497) | **0.949**(29.269) | **0.990**(44.994) | **0.895**(102.114) | **0.947**(142.722) | **0.988**(249.644) |
| | AICc | 0.871(21.453) | 0.931(28.012) | 0.984(43.424) | 0.873(98.064) | 0.933(137.903) | 0.984(244.575) |
| | BIC | 0.880(21.780) | 0.938(28.438) | 0.986(44.081) | 0.891(101.010) | 0.944(142.036) | 0.988(251.864) |
| | Vector L0 | 0.852(20.978) | 0.915(27.392) | 0.975(42.463) | 0.890(101.037) | 0.944(142.073) | 0.988(251.932) |
| | Oracle | 0.897(14.031) | 0.949(18.320) | 0.990(28.396) | 0.895(59.898) | 0.946(84.225) | 0.989(149.349) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(29.998) | **0.949**(39.028) | **0.989**(59.996) | **0.897**(137.670) | **0.948**(192.418) | **0.989**(336.569) |
| | AICc | 0.873(28.749) | 0.932(37.538) | 0.984(58.191) | 0.877(132.879) | 0.934(186.862) | 0.984(331.406) |
| | BIC | 0.883(29.166) | 0.939(38.082) | 0.986(59.030) | 0.894(136.813) | 0.947(192.379) | 0.989(341.135) |
| | Vector L0 | 0.861(28.377) | 0.923(37.053) | 0.978(57.439) | 0.893(136.725) | 0.946(192.256) | 0.988(340.918) |
| | Oracle | 0.903(14.169) | 0.951(18.500) | 0.990(28.675) | 0.899(60.486) | 0.949(85.051) | 0.990(150.814) |

TABLE 2.14. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{2001}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.897**(13.970) | **0.948**(18.176) | **0.989**(27.941) | **0.896**(48.319) | **0.947**(67.534) | **0.989**(118.128) |
| | AICc | 0.862(13.144) | 0.923(17.162) | 0.980(26.604) | 0.867(45.922) | 0.928(64.578) | 0.983(114.530) |
| | BIC | 0.875(13.381) | 0.932(17.472) | 0.983(27.083) | 0.890(47.755) | 0.943(67.148) | 0.988(119.064) |
| | Vector L0 | 0.768(11.603) | 0.846(15.152) | 0.933(23.495) | 0.886(47.465) | 0.941(66.741) | 0.987(118.344) |
| | Oracle | 0.897(11.790) | 0.949(15.394) | 0.990(23.860) | 0.898(40.453) | 0.949(56.883) | 0.990(100.866) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.898**(19.144) | **0.949**(24.907) | **0.990**(38.289) | **0.897**(66.940) | **0.948**(93.560) | **0.988**(163.651) |
| | AICc | 0.862(18.063) | 0.923(23.585) | 0.981(36.560) | 0.868(63.875) | 0.929(89.824) | 0.983(159.303) |
| | BIC | 0.876(18.408) | 0.932(24.035) | 0.984(37.256) | 0.892(66.388) | 0.945(93.349) | 0.988(165.521) |
| | Vector L0 | 0.814(17.022) | 0.885(22.226) | 0.959(34.459) | 0.891(66.277) | 0.943(93.192) | 0.987(165.244) |
| | Oracle | 0.898(11.801) | 0.948(15.408) | 0.990(23.883) | 0.898(40.433) | 0.949(56.854) | 0.990(100.815) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.900**(25.661) | **0.950**(33.385) | **0.988**(51.321) | **0.893**(89.786) | **0.945**(125.492) | **0.988**(219.505) |
| | AICc | 0.866(24.232) | 0.925(31.640) | 0.980(49.048) | 0.866(85.691) | 0.926(120.502) | 0.982(213.711) |
| | BIC | 0.877(24.681) | 0.933(32.225) | 0.983(49.950) | 0.889(89.092) | 0.942(125.273) | 0.987(222.125) |
| | Vector L0 | 0.834(23.306) | 0.899(30.433) | 0.965(47.178) | 0.887(88.857) | 0.941(124.943) | 0.987(221.543) |
| | Oracle | 0.902(11.867) | 0.951(15.494) | 0.990(24.016) | 0.897(40.243) | 0.947(56.587) | 0.989(100.340) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.896**(13.918) | **0.947**(18.108) | **0.989**(27.837) | **0.897**(48.044) | **0.946**(67.150) | **0.988**(117.456) |
| | AICc | 0.866(12.932) | 0.926(16.886) | 0.981(26.177) | 0.871(44.791) | 0.930(62.987) | 0.983(111.710) |
| | BIC | 0.875(13.121) | 0.933(17.132) | 0.984(26.556) | 0.890(46.395) | 0.943(65.237) | 0.987(115.679) |
| | Vector L0 | 0.808(11.967) | 0.879(15.627) | 0.954(24.228) | 0.887(46.115) | 0.940(64.844) | 0.986(114.984) |
| | Oracle | 0.900(11.848) | 0.949(15.470) | 0.990(23.978) | 0.897(40.249) | 0.948(56.595) | 0.990(100.355) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(17.911) | **0.948**(23.303) | **0.990**(35.823) | **0.899**(62.619) | **0.948**(87.521) | **0.989**(153.089) |
| | AICc | 0.869(17.041) | 0.929(22.250) | 0.983(34.492) | 0.877(59.949) | 0.934(84.304) | 0.985(149.515) |
| | BIC | 0.878(17.281) | 0.936(22.564) | 0.986(34.976) | 0.894(61.722) | 0.946(86.791) | 0.989(153.902) |
| | Vector L0 | 0.852(16.695) | 0.915(21.799) | 0.975(33.793) | 0.894(61.767) | 0.946(86.853) | 0.989(154.012) |
| | Oracle | 0.898(11.828) | 0.949(15.443) | 0.990(23.936) | 0.898(40.540) | 0.950(57.005) | 0.990(101.083) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(23.091) | **0.948**(30.043) | **0.989**(46.183) | **0.900**(81.677) | **0.950**(114.158) | **0.989**(199.680) |
| | AICc | 0.872(22.094) | 0.931(28.849) | 0.983(44.720) | 0.879(78.769) | 0.935(110.769) | 0.985(196.453) |
| | BIC | 0.881(22.425) | 0.937(29.280) | 0.986(45.386) | 0.895(81.045) | 0.948(113.961) | 0.989(202.081) |
| | Vector L0 | 0.861(21.853) | 0.921(28.534) | 0.979(44.233) | 0.896(81.134) | 0.948(114.086) | 0.989(202.302) |
| | Oracle | 0.901(11.806) | 0.949(15.415) | 0.990(23.893) | 0.900(40.419) | 0.949(56.835) | 0.990(100.780) |

TABLE 2.15. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{2001}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(10.318) | **0.948**(13.424) | **0.989**(20.636) | **0.898**(25.537) | **0.947**(35.692) | **0.989**(62.431) |
| | AICc | 0.863(9.714) | 0.925(12.684) | 0.980(19.663) | 0.870(24.227) | 0.929(34.068) | 0.983(60.421) |
| | BIC | 0.874(9.888) | 0.932(12.911) | 0.983(20.013) | 0.894(25.221) | 0.945(35.464) | 0.989(62.883) |
| | Vector L0 | 0.771(8.623) | 0.849(11.260) | 0.935(17.459) | 0.890(25.109) | 0.943(35.307) | 0.988(62.604) |
| | Oracle | 0.898(8.845) | 0.948(11.549) | 0.989(17.901) | 0.899(21.851) | 0.950(30.726) | 0.990(54.483) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.899**(13.868) | **0.949**(18.043) | **0.990**(27.736) | **0.900**(34.607) | **0.950**(48.369) | **0.989**(84.604) |
| | AICc | 0.863(13.101) | 0.924(17.106) | 0.981(26.517) | 0.870(33.023) | 0.930(46.438) | 0.983(82.359) |
| | BIC | 0.876(13.344) | 0.932(17.423) | 0.984(27.007) | 0.894(34.315) | 0.947(48.250) | 0.989(85.554) |
| | Vector L0 | 0.819(12.373) | 0.889(16.156) | 0.960(25.048) | 0.892(34.175) | 0.945(48.054) | 0.988(85.207) |
| | Oracle | 0.899(8.879) | 0.950(11.593) | 0.989(17.969) | 0.899(22.018) | 0.950(30.961) | 0.990(54.900) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.897**(18.327) | **0.947**(23.844) | **0.989**(36.655) | **0.897**(45.772) | **0.947**(63.975) | **0.989**(111.902) |
| | AICc | 0.862(17.329) | 0.922(22.627) | 0.980(35.075) | 0.872(43.798) | 0.930(61.591) | 0.983(109.232) |
| | BIC | 0.874(17.657) | 0.932(23.054) | 0.984(35.735) | 0.894(45.501) | 0.945(63.979) | 0.988(113.443) |
| | Vector L0 | 0.830(16.677) | 0.895(21.776) | 0.965(33.758) | 0.891(45.255) | 0.943(63.634) | 0.988(112.833) |
| | Oracle | 0.898(8.872) | 0.949(11.584) | 0.989(17.955) | 0.898(21.922) | 0.949(30.826) | 0.989(54.660) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.896**(10.330) | **0.947**(13.440) | **0.989**(20.661) | **0.896**(25.966) | **0.948**(36.292) | **0.988**(63.480) |
| | AICc | 0.865(9.553) | 0.925(12.474) | 0.982(19.337) | 0.871(23.993) | 0.930(33.740) | 0.983(59.839) |
| | BIC | 0.875(9.703) | 0.932(12.669) | 0.984(19.638) | 0.890(24.867) | 0.943(34.966) | 0.988(62.002) |
| | Vector L0 | 0.808(8.841) | 0.878(11.544) | 0.955(17.899) | 0.883(24.627) | 0.939(34.630) | 0.986(61.407) |
| | Oracle | 0.898(8.842) | 0.949(11.545) | 0.990(17.894) | 0.897(21.976) | 0.948(30.902) | 0.989(54.795) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(13.031) | **0.947**(16.954) | **0.989**(26.062) | **0.897**(32.660) | **0.948**(45.647) | **0.988**(79.844) |
| | AICc | 0.869(12.379) | 0.927(16.164) | 0.982(25.057) | 0.877(31.132) | 0.934(43.780) | 0.984(77.645) |
| | BIC | 0.878(12.559) | 0.935(16.398) | 0.985(25.418) | 0.893(32.037) | 0.946(45.048) | 0.988(79.881) |
| | Vector L0 | 0.852(12.107) | 0.913(15.808) | 0.975(24.506) | 0.893(32.065) | 0.946(45.088) | 0.988(79.952) |
| | Oracle | 0.899(8.879) | 0.949(11.593) | 0.990(17.969) | 0.899(22.000) | 0.949(30.935) | 0.989(54.854) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.896**(16.600) | **0.947**(21.597) | **0.989**(33.199) | **0.897**(41.644) | **0.948**(58.205) | **0.988**(101.810) |
| | AICc | 0.870(15.875) | 0.929(20.728) | 0.983(32.133) | 0.877(40.050) | 0.934(56.321) | 0.985(99.887) |
| | BIC | 0.882(16.123) | 0.937(21.051) | 0.986(32.631) | 0.893(41.220) | 0.947(57.961) | 0.988(102.779) |
| | Vector L0 | 0.859(15.714) | 0.921(20.518) | 0.977(31.806) | 0.893(41.290) | 0.946(58.059) | 0.988(102.954) |
| | Oracle | 0.898(8.846) | 0.949(11.549) | 0.990(17.902) | 0.898(21.912) | 0.949(30.811) | 0.990(54.635) |

TABLE 2.16. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{1999}X_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.897**(17.237) | **0.945**(22.426) | **0.986**(34.474) | **0.894**(75.644) | **0.942**(105.726) | **0.985**(184.931) |
| | AICc | 0.863(16.226) | 0.921(21.187) | 0.977(32.843) | 0.868(71.843) | 0.926(101.029) | 0.979(179.175) |
| | BIC | 0.874(16.484) | 0.928(21.523) | 0.980(33.362) | 0.888(74.732) | 0.939(105.082) | 0.985(186.326) |
| | Vector L0 | 0.749(13.784) | 0.825(18.000) | 0.921(27.914) | 0.884(74.399) | 0.937(104.614) | 0.983(185.498) |
| | Oracle | 0.901(14.120) | 0.949(18.436) | 0.990(28.576) | 0.898(60.205) | 0.949(84.656) | 0.990(150.113) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.896**(24.449) | **0.939**(31.809) | **0.978**(48.899) | **0.891**(111.032) | **0.937**(155.186) | **0.979**(271.445) |
| | AICc | 0.869(23.067) | 0.920(30.119) | 0.971(46.690) | 0.870(105.822) | 0.922(148.811) | 0.972(263.917) |
| | BIC | 0.877(23.499) | 0.927(30.682) | 0.973(47.559) | 0.886(110.001) | 0.934(154.674) | 0.978(274.260) |
| | Vector L0 | 0.827(21.716) | 0.887(28.356) | 0.951(43.962) | 0.887(110.133) | 0.934(154.859) | 0.978(274.587) |
| | Oracle | 0.900(14.138) | 0.948(18.459) | 0.989(28.612) | 0.900(60.326) | 0.949(84.827) | 0.990(150.416) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.893**(33.035) | **0.935**(42.980) | **0.975**(66.071) | **0.891**(153.247) | **0.936**(214.189) | **0.977**(374.649) |
| | AICc | 0.869(31.194) | 0.918(40.730) | 0.968(63.138) | 0.870(146.293) | 0.921(205.723) | 0.971(364.852) |
| | BIC | 0.877(31.784) | 0.924(41.500) | 0.970(64.327) | 0.887(151.905) | 0.934(213.596) | 0.977(378.735) |
| | Vector L0 | 0.840(29.991) | 0.897(39.161) | 0.957(60.710) | 0.885(151.276) | 0.933(212.713) | 0.976(377.177) |
| | Oracle | 0.897(14.133) | 0.948(18.452) | 0.990(28.601) | 0.899(60.286) | 0.949(84.771) | 0.990(150.317) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.890**(18.056) | **0.937**(23.492) | **0.979**(36.113) | **0.890**(79.644) | **0.938**(111.317) | **0.980**(194.710) |
| | AICc | 0.856(16.882) | 0.914(22.043) | 0.970(34.170) | 0.863(75.312) | 0.920(105.907) | 0.973(187.828) |
| | BIC | 0.866(17.152) | 0.921(22.395) | 0.973(34.715) | 0.883(78.374) | 0.934(110.202) | 0.979(195.406) |
| | Vector L0 | 0.790(15.420) | 0.862(20.136) | 0.941(31.220) | 0.883(78.239) | 0.933(110.014) | 0.979(195.072) |
| | Oracle | 0.897(14.128) | 0.948(18.446) | 0.989(28.592) | 0.900(60.230) | 0.950(84.691) | 0.990(150.175) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.891**(26.347) | **0.930**(34.278) | **0.970**(52.694) | **0.884**(120.670) | **0.929**(168.657) | **0.971**(295.007) |
| | AICc | 0.864(24.838) | 0.913(32.431) | 0.962(50.274) | 0.864(115.031) | 0.915(161.762) | 0.964(286.887) |
| | BIC | 0.873(25.284) | 0.919(33.013) | 0.964(51.172) | 0.880(119.574) | 0.927(168.135) | 0.970(298.128) |
| | Vector L0 | 0.844(24.059) | 0.896(31.416) | 0.953(48.702) | 0.880(119.513) | 0.926(168.049) | 0.970(297.976) |
| | Oracle | 0.899(14.110) | 0.949(18.423) | 0.989(28.556) | 0.899(60.143) | 0.950(84.570) | 0.990(149.960) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.887**(36.318) | **0.926**(47.251) | **0.965**(72.637) | **0.880**(167.470) | **0.924**(234.068) | **0.966**(409.421) |
| | AICc | 0.866(34.306) | 0.910(44.794) | 0.957(69.438) | 0.862(159.758) | 0.911(224.659) | 0.960(398.438) |
| | BIC | 0.873(34.963) | 0.917(45.650) | 0.960(70.760) | 0.876(166.048) | 0.921(233.482) | 0.966(413.996) |
| | Vector L0 | 0.853(33.696) | 0.902(43.998) | 0.951(68.204) | 0.872(163.649) | 0.918(230.115) | 0.964(408.051) |
| | Oracle | 0.898(14.101) | 0.949(18.412) | 0.989(28.538) | 0.900(60.190) | 0.948(84.636) | 0.989(150.077) |

TABLE 2.17. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{1999}X_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.899**(14.002) | **0.945**(18.218) | **0.986**(28.005) | **0.897**(48.363) | **0.945**(67.596) | **0.986**(118.236) |
| | AICc | 0.866(13.194) | 0.924(17.228) | 0.979(26.706) | 0.868(45.959) | 0.927(64.630) | 0.980(114.622) |
| | BIC | 0.877(13.425) | 0.931(17.528) | 0.981(27.170) | 0.891(47.810) | 0.941(67.226) | 0.986(119.201) |
| | Vector L0 | 0.767(11.556) | 0.842(15.091) | 0.931(23.400) | 0.887(47.519) | 0.939(66.818) | 0.985(118.479) |
| | Oracle | 0.899(11.841) | 0.949(15.460) | 0.990(23.963) | 0.900(40.568) | 0.950(57.043) | 0.990(101.150) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.897**(19.206) | **0.941**(24.987) | **0.979**(38.411) | **0.891**(66.677) | **0.938**(93.192) | **0.981**(163.008) |
| | AICc | 0.869(18.139) | 0.922(23.685) | 0.972(36.716) | 0.869(63.635) | 0.923(89.486) | 0.974(158.705) |
| | BIC | 0.879(18.480) | 0.928(24.129) | 0.974(37.402) | 0.887(66.166) | 0.935(93.037) | 0.980(164.967) |
| | Vector L0 | 0.827(17.132) | 0.889(22.371) | 0.954(34.682) | 0.886(66.120) | 0.936(92.972) | 0.980(164.852) |
| | Oracle | 0.898(11.817) | 0.949(15.428) | 0.990(23.914) | 0.898(40.204) | 0.948(56.532) | 0.990(100.243) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.896**(25.534) | **0.936**(33.221) | **0.976**(51.068) | **0.891**(89.973) | **0.935**(125.753) | **0.977**(219.961) |
| | AICc | 0.872(24.114) | 0.919(31.486) | 0.967(48.809) | 0.870(85.825) | 0.921(120.691) | 0.971(214.046) |
| | BIC | 0.880(24.566) | 0.925(32.075) | 0.970(49.717) | 0.886(89.167) | 0.933(125.379) | 0.977(222.315) |
| | Vector L0 | 0.844(23.185) | 0.900(30.273) | 0.956(46.932) | 0.885(88.698) | 0.932(124.721) | 0.976(221.152) |
| | Oracle | 0.902(11.846) | 0.950(15.467) | 0.989(23.974) | 0.897(40.521) | 0.948(56.978) | 0.990(101.034) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.893**(14.645) | **0.943**(19.054) | **0.983**(29.291) | **0.892**(50.567) | **0.940**(70.677) | **0.982**(123.624) |
| | AICc | 0.861(13.647) | 0.921(17.820) | 0.973(27.623) | 0.864(47.431) | 0.921(66.699) | 0.975(118.292) |
| | BIC | 0.873(13.891) | 0.928(18.138) | 0.977(28.114) | 0.886(49.327) | 0.936(69.360) | 0.981(122.985) |
| | Vector L0 | 0.804(12.666) | 0.874(16.540) | 0.949(25.643) | 0.882(49.044) | 0.933(68.963) | 0.980(122.283) |
| | Oracle | 0.899(11.843) | 0.950(15.463) | 0.990(23.967) | 0.899(40.432) | 0.949(56.853) | 0.990(100.813) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.888**(20.560) | **0.930**(26.749) | **0.970**(41.120) | **0.886**(71.996) | **0.930**(100.626) | **0.972**(176.011) |
| | AICc | 0.862(19.375) | 0.913(25.298) | 0.961(39.216) | 0.863(68.598) | 0.916(96.465) | 0.965(171.082) |
| | BIC | 0.872(19.726) | 0.919(25.756) | 0.964(39.923) | 0.882(71.292) | 0.928(100.245) | 0.972(177.749) |
| | Vector L0 | 0.838(18.720) | 0.895(24.444) | 0.951(37.894) | 0.881(71.255) | 0.927(100.192) | 0.971(177.656) |
| | Oracle | 0.898(11.843) | 0.949(15.463) | 0.990(23.967) | 0.897(40.432) | 0.948(56.853) | 0.989(100.812) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.887**(27.890) | **0.928**(36.285) | **0.968**(55.779) | **0.882**(98.359) | **0.926**(137.473) | **0.968**(240.462) |
| | AICc | 0.866(26.324) | 0.913(34.371) | 0.960(53.281) | 0.864(93.865) | 0.914(131.997) | 0.962(234.098) |
| | BIC | 0.873(26.825) | 0.918(35.025) | 0.963(54.291) | 0.879(97.522) | 0.924(137.128) | 0.968(243.147) |
| | Vector L0 | 0.852(25.870) | 0.904(33.779) | 0.954(52.363) | 0.873(95.709) | 0.920(134.581) | 0.966(238.650) |
| | Oracle | 0.901(11.831) | 0.951(15.447) | 0.990(23.943) | 0.898(40.459) | 0.949(56.891) | 0.990(100.879) |

TABLE 2.18. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_{1999}X_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.901**(10.362) | **0.948**(13.482) | **0.987**(20.724) | **0.896**(25.720) | **0.945**(35.949) | **0.986**(62.880) |
| | AICc | 0.867(9.740) | 0.927(12.717) | 0.979(19.714) | 0.870(24.389) | 0.928(34.297) | 0.980(60.827) |
| | BIC | 0.878(9.920) | 0.933(12.952) | 0.982(20.077) | 0.892(25.392) | 0.943(35.704) | 0.986(63.309) |
| | Vector L0 | 0.771(8.567) | 0.845(11.187) | 0.932(17.347) | 0.889(25.246) | 0.940(35.499) | 0.985(62.945) |
| | Oracle | 0.901(8.895) | 0.950(11.614) | 0.991(18.002) | 0.898(22.053) | 0.949(31.010) | 0.990(54.987) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.895**(13.892) | **0.940**(18.074) | **0.980**(27.784) | **0.894**(34.593) | **0.940**(48.349) | **0.982**(84.571) |
| | AICc | 0.867(13.116) | 0.921(17.126) | 0.973(26.548) | 0.869(32.980) | 0.926(46.378) | 0.975(82.252) |
| | BIC | 0.878(13.367) | 0.928(17.454) | 0.975(27.054) | 0.889(34.245) | 0.938(48.152) | 0.981(85.381) |
| | Vector L0 | 0.826(12.407) | 0.889(16.201) | 0.956(25.116) | 0.888(34.174) | 0.937(48.052) | 0.981(85.205) |
| | Oracle | 0.898(8.877) | 0.949(11.590) | 0.989(17.965) | 0.896(22.008) | 0.948(30.947) | 0.990(54.875) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.897**(18.367) | **0.938**(23.896) | **0.977**(36.734) | **0.894**(45.738) | **0.937**(63.926) | **0.977**(111.817) |
| | AICc | 0.872(17.335) | 0.921(22.635) | 0.969(35.088) | 0.872(43.634) | 0.923(61.361) | 0.971(108.824) |
| | BIC | 0.881(17.665) | 0.928(23.064) | 0.972(35.751) | 0.889(45.342) | 0.934(63.756) | 0.977(113.048) |
| | Vector L0 | 0.844(16.673) | 0.900(21.771) | 0.957(33.751) | 0.887(44.990) | 0.932(63.262) | 0.976(112.175) |
| | Oracle | 0.900(8.899) | 0.950(11.619) | 0.990(18.010) | 0.899(21.914) | 0.949(30.813) | 0.990(54.639) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.893**(10.833) | **0.942**(14.093) | **0.982**(21.665) | **0.890**(26.847) | **0.940**(37.524) | **0.983**(65.635) |
| | AICc | 0.859(10.046) | 0.917(13.117) | 0.973(20.334) | 0.863(25.007) | 0.922(35.167) | 0.976(62.368) |
| | BIC | 0.870(10.222) | 0.925(13.347) | 0.976(20.688) | 0.884(25.995) | 0.936(36.552) | 0.982(64.813) |
| | Vector L0 | 0.799(9.297) | 0.870(12.140) | 0.947(18.822) | 0.880(25.875) | 0.933(36.383) | 0.981(64.514) |
| | Oracle | 0.898(8.896) | 0.948(11.616) | 0.990(18.004) | 0.898(21.920) | 0.949(30.823) | 0.990(54.656) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.889**(14.871) | **0.931**(19.348) | **0.971**(29.743) | **0.887**(36.945) | **0.932**(51.637) | **0.973**(90.320) |
| | AICc | 0.863(14.026) | 0.914(18.314) | 0.963(28.390) | 0.865(35.175) | 0.918(49.465) | 0.967(87.726) |
| | BIC | 0.872(14.287) | 0.920(18.654) | 0.966(28.915) | 0.883(36.543) | 0.929(51.384) | 0.972(91.111) |
| | Vector L0 | 0.838(13.567) | 0.896(17.715) | 0.954(27.462) | 0.882(36.555) | 0.929(51.401) | 0.972(91.141) |
| | Oracle | 0.899(8.900) | 0.949(11.621) | 0.989(18.012) | 0.899(21.889) | 0.949(30.779) | 0.989(54.577) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.887**(20.034) | **0.928**(26.064) | **0.967**(40.067) | **0.881**(49.716) | **0.925**(69.487) | **0.968**(121.544) |
| | AICc | 0.865(18.906) | 0.912(24.686) | 0.959(38.268) | 0.862(47.459) | 0.912(66.739) | 0.961(118.363) |
| | BIC | 0.873(19.262) | 0.917(25.150) | 0.961(38.984) | 0.878(49.302) | 0.923(69.325) | 0.968(122.922) |
| | Vector L0 | 0.851(18.537) | 0.902(24.205) | 0.954(37.522) | 0.871(48.349) | 0.918(67.986) | 0.965(120.559) |
| | Oracle | 0.898(8.870) | 0.949(11.581) | 0.989(17.951) | 0.900(21.946) | 0.948(30.859) | 0.989(54.720) |

TABLE 2.19. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X^2_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.879**(20.019) | **0.926**(26.046) | **0.970**(40.039) | **0.877**(89.484) | **0.925**(125.070) | **0.970**(218.767) |
| | AICc | 0.849(18.793) | 0.903(24.538) | 0.960(38.038) | 0.851(84.762) | 0.907(119.196) | 0.962(211.396) |
| | BIC | 0.858(19.104) | 0.910(24.943) | 0.963(38.664) | 0.871(88.152) | 0.921(123.951) | 0.969(219.785) |
| | Vector L0 | 0.734(15.842) | 0.808(20.688) | 0.900(32.083) | 0.866(87.464) | 0.918(122.985) | 0.967(218.076) |
| | Oracle | 0.901(14.127) | 0.951(18.446) | 0.990(28.591) | 0.899(60.356) | 0.950(84.869) | 0.989(150.491) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.870**(31.307) | **0.912**(40.731) | **0.956**(62.614) | **0.867**(145.411) | **0.913**(203.237) | **0.960**(355.493) |
| | AICc | 0.845(29.492) | 0.895(38.508) | 0.947(59.694) | 0.847(138.556) | 0.898(194.843) | 0.952(345.557) |
| | BIC | 0.853(30.015) | 0.901(39.190) | 0.950(60.746) | 0.863(143.990) | 0.910(202.466) | 0.959(359.002) |
| | Vector L0 | 0.784(26.693) | 0.845(34.857) | 0.917(54.045) | 0.861(142.991) | 0.909(201.064) | 0.958(356.524) |
| | Oracle | 0.899(14.081) | 0.949(18.385) | 0.989(28.497) | 0.899(60.528) | 0.949(85.110) | 0.989(150.918) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.863**(44.338) | **0.906**(57.685) | **0.951**(88.676) | **0.865**(207.016) | **0.910**(289.341) | **0.956**(506.102) |
| | AICc | 0.841(41.843) | 0.891(54.635) | 0.942(84.693) | 0.846(197.318) | 0.896(277.476) | 0.947(492.107) |
| | BIC | 0.848(42.599) | 0.896(55.620) | 0.945(86.214) | 0.860(204.996) | 0.907(288.248) | 0.955(511.106) |
| | Vector L0 | 0.793(38.513) | 0.851(50.291) | 0.919(77.973) | 0.824(201.894) | 0.880(291.197) | 0.939(573.660) |
| | Oracle | 0.898(14.102) | 0.949(18.413) | 0.990(28.540) | 0.899(60.187) | 0.949(84.631) | 0.990(150.070) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.877**(20.131) | **0.923**(26.191) | **0.971**(40.262) | **0.874**(90.191) | **0.925**(126.057) | **0.971**(220.493) |
| | AICc | 0.846(18.705) | 0.901(24.423) | 0.960(37.860) | 0.849(84.431) | 0.906(118.730) | 0.962(210.571) |
| | BIC | 0.855(19.011) | 0.907(24.822) | 0.963(38.477) | 0.867(87.898) | 0.920(123.594) | 0.969(219.151) |
| | Vector L0 | 0.776(16.952) | 0.845(22.136) | 0.926(34.322) | 0.864(87.480) | 0.917(123.008) | 0.968(218.117) |
| | Oracle | 0.898(14.116) | 0.948(18.431) | 0.990(28.568) | 0.898(60.373) | 0.950(84.893) | 0.990(150.533) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.864**(31.501) | **0.908**(40.984) | **0.955**(63.002) | **0.869**(145.216) | **0.914**(202.965) | **0.959**(355.017) |
| | AICc | 0.840(29.672) | 0.891(38.743) | 0.945(60.058) | 0.847(137.923) | 0.897(193.953) | 0.949(343.978) |
| | BIC | 0.848(30.220) | 0.896(39.457) | 0.949(61.161) | 0.863(143.296) | 0.910(201.491) | 0.957(357.274) |
| | Vector L0 | 0.810(28.358) | 0.866(37.029) | 0.931(57.405) | 0.863(143.092) | 0.909(201.204) | 0.957(356.769) |
| | Oracle | 0.898(14.112) | 0.949(18.426) | 0.990(28.560) | 0.900(60.255) | 0.949(84.727) | 0.990(150.240) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.865**(44.569) | **0.907**(57.985) | **0.952**(89.137) | **0.864**(206.961) | **0.910**(289.264) | **0.957**(505.967) |
| | AICc | 0.842(41.980) | 0.890(54.815) | 0.943(84.972) | 0.844(196.894) | 0.894(276.880) | 0.949(491.052) |
| | BIC | 0.849(42.765) | 0.896(55.838) | 0.947(86.551) | 0.860(204.541) | 0.906(287.609) | 0.956(509.975) |
| | Vector L0 | 0.820(40.381) | 0.873(52.728) | 0.932(81.742) | 0.831(191.491) | 0.884(269.294) | 0.943(477.644) |
| | Oracle | 0.897(14.041) | 0.949(18.333) | 0.989(28.416) | 0.897(60.461) | 0.949(85.016) | 0.990(150.752) |

TABLE 2.20. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X^2_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.877**(15.907) | **0.926**(20.695) | **0.972**(31.813) | **0.878**(55.940) | **0.926**(78.186) | **0.973**(136.759) |
| | AICc | 0.848(14.913) | 0.904(19.473) | 0.962(30.186) | 0.852(52.748) | 0.908(74.177) | 0.965(131.553) |
| | BIC | 0.857(15.175) | 0.911(19.814) | 0.965(30.712) | 0.872(54.857) | 0.923(77.135) | 0.972(136.771) |
| | Vector L0 | 0.738(12.817) | 0.814(16.738) | 0.904(25.956) | 0.869(54.585) | 0.920(76.753) | 0.970(136.096) |
| | Oracle | 0.896(11.799) | 0.948(15.406) | 0.989(23.879) | 0.896(40.527) | 0.948(56.987) | 0.989(101.050) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.869**(24.338) | **0.912**(31.664) | **0.958**(48.675) | **0.867**(85.347) | **0.915**(119.287) | **0.963**(208.652) |
| | AICc | 0.846(22.962) | 0.895(29.982) | 0.949(46.477) | 0.846(81.355) | 0.899(114.405) | 0.954(202.898) |
| | BIC | 0.853(23.372) | 0.901(30.517) | 0.951(47.302) | 0.863(84.513) | 0.912(118.835) | 0.962(210.711) |
| | Vector L0 | 0.787(20.996) | 0.848(27.417) | 0.920(42.508) | 0.859(83.770) | 0.910(117.792) | 0.960(208.868) |
| | Oracle | 0.899(11.888) | 0.949(15.522) | 0.989(24.059) | 0.897(40.463) | 0.949(56.897) | 0.989(100.890) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.864**(34.132) | **0.906**(44.407) | **0.952**(68.264) | **0.864**(120.068) | **0.909**(167.817) | **0.957**(293.537) |
| | AICc | 0.842(32.205) | 0.889(42.051) | 0.942(65.186) | 0.845(114.340) | 0.894(160.789) | 0.948(285.162) |
| | BIC | 0.848(32.782) | 0.895(42.802) | 0.946(66.346) | 0.860(118.867) | 0.906(167.141) | 0.955(296.365) |
| | Vector L0 | 0.794(29.721) | 0.851(38.811) | 0.919(60.173) | 0.823(109.145) | 0.878(153.495) | 0.938(272.270) |
| | Oracle | 0.895(11.825) | 0.946(15.439) | 0.989(23.931) | 0.896(40.262) | 0.947(56.614) | 0.989(100.388) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.880**(16.179) | **0.928**(21.049) | **0.972**(32.358) | **0.877**(56.435) | **0.926**(78.878) | **0.974**(137.969) |
| | AICc | 0.848(14.970) | 0.905(19.547) | 0.962(30.301) | 0.852(52.493) | 0.908(73.818) | 0.964(130.917) |
| | BIC | 0.858(15.228) | 0.913(19.883) | 0.965(30.820) | 0.871(54.679) | 0.923(76.884) | 0.972(136.326) |
| | Vector L0 | 0.784(13.692) | 0.852(17.879) | 0.931(27.721) | 0.867(54.482) | 0.919(76.608) | 0.971(135.839) |
| | Oracle | 0.899(11.868) | 0.948(15.496) | 0.990(24.018) | 0.897(40.614) | 0.948(57.109) | 0.990(101.267) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.870**(24.345) | **0.914**(31.674) | **0.958**(48.690) | **0.869**(85.780) | **0.915**(119.892) | **0.962**(209.710) |
| | AICc | 0.844(22.901) | 0.895(29.903) | 0.949(46.354) | 0.847(81.396) | 0.900(114.463) | 0.953(203.002) |
| | BIC | 0.853(23.334) | 0.901(30.467) | 0.952(47.225) | 0.865(84.606) | 0.912(118.966) | 0.961(210.944) |
| | Vector L0 | 0.811(21.786) | 0.868(28.447) | 0.933(44.102) | 0.863(84.212) | 0.912(118.413) | 0.960(209.968) |
| | Oracle | 0.898(11.870) | 0.949(15.498) | 0.989(24.022) | 0.898(40.498) | 0.949(56.946) | 0.990(100.978) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.867**(34.193) | **0.909**(44.486) | **0.954**(68.386) | **0.863**(119.907) | **0.910**(167.591) | **0.957**(293.143) |
| | AICc | 0.844(32.230) | 0.893(42.083) | 0.945(65.235) | 0.842(114.088) | 0.894(160.436) | 0.948(284.535) |
| | BIC | 0.851(32.819) | 0.898(42.851) | 0.948(66.421) | 0.859(118.530) | 0.907(166.667) | 0.956(295.525) |
| | Vector L0 | 0.819(30.951) | 0.872(40.415) | 0.933(62.655) | 0.828(110.637) | 0.885(155.591) | 0.942(275.973) |
| | Oracle | 0.898(11.842) | 0.949(15.462) | 0.990(23.966) | 0.897(40.443) | 0.948(56.868) | 0.990(100.839) |

TABLE 2.21. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X^2_{2000}$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.878**(11.717) | **0.928**(15.244) | **0.973**(23.434) | **0.878**(29.298) | **0.931**(40.949) | **0.975**(71.627) |
| | AICc | 0.847(10.944) | 0.905(14.290) | 0.963(22.152) | 0.853(27.550) | 0.912(38.741) | 0.967(68.708) |
| | BIC | 0.857(11.136) | 0.912(14.540) | 0.967(22.537) | 0.873(28.645) | 0.927(40.277) | 0.974(71.417) |
| | Vector L0 | 0.750(9.541) | 0.823(12.460) | 0.911(19.321) | 0.870(28.534) | 0.925(40.122) | 0.973(71.143) |
| | Oracle | 0.899(8.871) | 0.949(11.582) | 0.990(17.952) | 0.897(22.048) | 0.949(31.003) | 0.989(54.975) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.874**(17.605) | **0.916**(22.905) | **0.960**(35.210) | **0.870**(43.670) | **0.916**(61.036) | **0.964**(106.762) |
| | AICc | 0.848(16.605) | 0.898(21.682) | 0.951(33.610) | 0.849(41.596) | 0.901(58.493) | 0.955(103.738) |
| | BIC | 0.857(16.925) | 0.904(22.098) | 0.954(34.254) | 0.866(43.174) | 0.913(60.708) | 0.962(107.644) |
| | Vector L0 | 0.791(15.156) | 0.853(19.791) | 0.922(30.685) | 0.862(42.843) | 0.911(60.243) | 0.961(106.822) |
| | Oracle | 0.900(8.867) | 0.950(11.577) | 0.990(17.944) | 0.898(21.965) | 0.949(30.885) | 0.990(54.766) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.865**(24.307) | **0.907**(31.624) | **0.954**(48.614) | **0.861**(60.386) | **0.907**(84.401) | **0.954**(147.630) |
| | AICc | 0.843(22.915) | 0.889(29.921) | 0.944(46.382) | 0.840(57.493) | 0.892(80.849) | 0.946(143.386) |
| | BIC | 0.850(23.342) | 0.896(30.477) | 0.947(47.241) | 0.856(59.730) | 0.904(83.987) | 0.953(148.921) |
| | Vector L0 | 0.794(21.103) | 0.852(27.557) | 0.919(42.726) | 0.817(54.774) | 0.875(77.031) | 0.935(136.639) |
| | Oracle | 0.898(8.887) | 0.947(11.603) | 0.990(17.985) | 0.897(21.913) | 0.947(30.813) | 0.990(54.637) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.878**(11.813) | **0.927**(15.369) | **0.973**(23.627) | **0.881**(29.738) | **0.929**(41.564) | **0.976**(72.702) |
| | AICc | 0.846(10.880) | 0.904(14.207) | 0.962(22.023) | 0.855(27.397) | 0.912(38.527) | 0.966(68.329) |
| | BIC | 0.857(11.080) | 0.911(14.467) | 0.965(22.425) | 0.875(28.541) | 0.926(40.132) | 0.974(71.160) |
| | Vector L0 | 0.783(10.000) | 0.853(13.058) | 0.932(20.245) | 0.871(28.552) | 0.923(40.147) | 0.972(71.187) |
| | Oracle | 0.896(8.846) | 0.947(11.550) | 0.989(17.902) | 0.898(22.008) | 0.948(30.947) | 0.989(54.875) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.869**(17.435) | **0.912**(22.684) | **0.959**(34.871) | **0.872**(43.965) | **0.916**(61.449) | **0.963**(107.484) |
| | AICc | 0.843(16.386) | 0.895(21.395) | 0.949(33.166) | 0.852(41.662) | 0.902(58.587) | 0.955(103.905) |
| | BIC | 0.851(16.675) | 0.900(21.773) | 0.952(33.749) | 0.868(43.294) | 0.914(60.877) | 0.963(107.944) |
| | Vector L0 | 0.811(15.650) | 0.870(20.435) | 0.934(31.680) | 0.866(43.105) | 0.913(60.611) | 0.961(107.474) |
| | Oracle | 0.899(8.840) | 0.949(11.543) | 0.990(17.891) | 0.899(22.107) | 0.949(31.085) | 0.990(55.120) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.867**(24.296) | **0.909**(31.610) | **0.955**(48.593) | **0.867**(61.071) | **0.911**(85.357) | **0.957**(149.302) |
| | AICc | 0.844(22.896) | 0.894(29.896) | 0.944(46.344) | 0.846(58.025) | 0.896(81.597) | 0.949(144.713) |
| | BIC | 0.851(23.330) | 0.899(30.461) | 0.949(47.216) | 0.862(60.266) | 0.908(84.741) | 0.957(150.259) |
| | Vector L0 | 0.819(21.958) | 0.873(28.672) | 0.934(44.450) | 0.827(55.975) | 0.883(78.719) | 0.943(139.627) |
| | Oracle | 0.899(8.913) | 0.949(11.637) | 0.990(18.038) | 0.899(22.015) | 0.949(30.956) | 0.990(54.891) |

TABLE 2.22. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_1^2$ in main effects.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.879**(19.914) | **0.925**(25.909) | **0.970**(39.828) | **0.873**(89.085) | **0.922**(124.512) | **0.970**(217.791) |
| | AICc | 0.847(18.665) | 0.904(24.371) | 0.959(37.780) | 0.849(84.391) | 0.906(118.674) | 0.962(210.468) |
| | BIC | 0.858(19.002) | 0.911(24.810) | 0.964(38.457) | 0.868(87.844) | 0.919(123.518) | 0.969(219.014) |
| | Vector L0 | 0.733(15.760) | 0.809(20.582) | 0.900(31.918) | 0.863(87.100) | 0.916(122.473) | 0.967(217.167) |
| | Oracle | 0.898(14.097) | 0.949(18.406) | 0.990(28.530) | 0.898(60.429) | 0.948(84.971) | 0.989(150.672) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.867**(31.467) | **0.910**(40.940) | **0.956**(62.935) | **0.865**(143.906) | **0.910**(201.133) | **0.958**(351.813) |
| | AICc | 0.844(29.663) | 0.893(38.731) | 0.947(60.040) | 0.844(137.029) | 0.895(192.696) | 0.949(341.750) |
| | BIC | 0.852(30.221) | 0.900(39.459) | 0.950(61.163) | 0.861(142.467) | 0.908(200.324) | 0.957(355.203) |
| | Vector L0 | 0.787(27.013) | 0.847(35.274) | 0.920(54.692) | 0.858(141.586) | 0.906(199.087) | 0.956(353.017) |
| | Oracle | 0.897(14.117) | 0.948(18.432) | 0.990(28.570) | 0.896(60.367) | 0.948(84.885) | 0.989(150.519) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.866**(44.142) | **0.908**(57.430) | **0.953**(88.284) | **0.861**(204.872) | **0.907**(286.345) | **0.955**(500.862) |
| | AICc | 0.845(41.668) | 0.891(54.407) | 0.943(84.340) | 0.841(195.315) | 0.892(274.660) | 0.946(487.114) |
| | BIC | 0.852(42.490) | 0.898(55.478) | 0.947(85.994) | 0.858(203.279) | 0.906(285.833) | 0.954(506.820) |
| | Vector L0 | 0.792(38.136) | 0.851(49.799) | 0.918(77.211) | 0.824(188.403) | 0.880(264.953) | 0.940(469.950) |
| | Oracle | 0.900(14.052) | 0.950(18.348) | 0.991(28.439) | 0.898(60.487) | 0.949(85.054) | 0.990(150.818) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.878**(20.035) | **0.924**(26.067) | **0.970**(40.071) | **0.875**(90.356) | **0.923**(126.288) | **0.970**(220.898) |
| | AICc | 0.846(18.603) | 0.903(24.290) | 0.960(37.653) | 0.849(84.792) | 0.906(119.238) | 0.962(211.468) |
| | BIC | 0.855(18.912) | 0.909(24.694) | 0.962(38.277) | 0.869(88.252) | 0.920(124.092) | 0.969(220.033) |
| | Vector L0 | 0.777(16.880) | 0.846(22.042) | 0.927(34.176) | 0.864(87.691) | 0.916(123.305) | 0.967(218.641) |
| | Oracle | 0.898(14.094) | 0.947(18.402) | 0.989(28.523) | 0.899(60.701) | 0.949(85.353) | 0.990(151.350) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.867**(31.275) | **0.910**(40.689) | **0.956**(62.549) | **0.864**(145.291) | **0.910**(203.070) | **0.957**(355.200) |
| | AICc | 0.843(29.424) | 0.893(38.419) | 0.947(59.556) | 0.845(137.879) | 0.895(193.892) | 0.950(343.870) |
| | BIC | 0.851(29.982) | 0.899(39.146) | 0.950(60.679) | 0.860(143.345) | 0.909(201.559) | 0.956(357.392) |
| | Vector L0 | 0.812(28.096) | 0.869(36.687) | 0.933(56.876) | 0.860(142.968) | 0.907(201.030) | 0.955(356.459) |
| | Oracle | 0.897(14.099) | 0.948(18.408) | 0.989(28.533) | 0.896(60.550) | 0.948(85.142) | 0.990(150.975) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.864**(44.274) | **0.907**(57.602) | **0.952**(88.548) | **0.864**(207.074) | **0.909**(289.422) | **0.957**(506.244) |
| | AICc | 0.841(41.694) | 0.888(54.440) | 0.941(84.391) | 0.845(197.022) | 0.894(277.061) | 0.948(491.372) |
| | BIC | 0.849(42.508) | 0.896(55.501) | 0.945(86.030) | 0.860(205.077) | 0.907(288.361) | 0.956(511.302) |
| | Vector L0 | 0.817(40.064) | 0.871(52.314) | 0.930(81.101) | 0.833(191.758) | 0.886(269.669) | 0.944(478.305) |
| | Oracle | 0.895(14.130) | 0.948(18.449) | 0.990(28.596) | 0.898(60.354) | 0.948(84.866) | 0.990(150.486) |

TABLE 2.23. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_1^2$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.877**(15.980) | **0.926**(20.791) | **0.971**(31.960) | **0.876**(55.576) | **0.926**(77.678) | **0.973**(135.870) |
| | AICc | 0.845(14.961) | 0.904(19.535) | 0.962(30.282) | 0.851(52.457) | 0.908(73.767) | 0.965(130.827) |
| | BIC | 0.856(15.241) | 0.911(19.900) | 0.965(30.845) | 0.871(54.536) | 0.923(76.684) | 0.972(135.971) |
| | Vector L0 | 0.744(12.951) | 0.817(16.913) | 0.908(26.226) | 0.868(54.342) | 0.922(76.411) | 0.971(135.488) |
| | Oracle | 0.899(11.837) | 0.949(15.455) | 0.990(23.955) | 0.897(40.370) | 0.948(56.766) | 0.990(100.658) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.871**(24.263) | **0.915**(31.567) | **0.959**(48.526) | **0.867**(85.171) | **0.913**(119.042) | **0.960**(208.223) |
| | AICc | 0.845(22.859) | 0.895(29.847) | 0.949(46.268) | 0.846(81.072) | 0.898(114.006) | 0.952(202.192) |
| | BIC | 0.854(23.284) | 0.902(30.402) | 0.952(47.124) | 0.863(84.335) | 0.911(118.584) | 0.959(210.266) |
| | Vector L0 | 0.785(20.895) | 0.848(27.285) | 0.919(42.304) | 0.861(83.831) | 0.909(117.877) | 0.959(209.017) |
| | Oracle | 0.899(11.849) | 0.949(15.471) | 0.990(23.981) | 0.899(40.410) | 0.950(56.822) | 0.991(100.758) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.866**(33.970) | **0.908**(44.196) | **0.954**(67.939) | **0.865**(120.515) | **0.911**(168.441) | **0.958**(294.629) |
| | AICc | 0.844(32.071) | 0.891(41.876) | 0.944(64.914) | 0.846(114.814) | 0.896(161.456) | 0.949(286.345) |
| | BIC | 0.851(32.693) | 0.897(42.687) | 0.948(66.166) | 0.861(119.453) | 0.909(167.965) | 0.957(297.824) |
| | Vector L0 | 0.792(29.422) | 0.850(38.420) | 0.918(59.569) | 0.824(109.771) | 0.880(154.375) | 0.941(273.828) |
| | Oracle | 0.900(11.864) | 0.950(15.491) | 0.989(24.011) | 0.897(40.525) | 0.949(56.984) | 0.990(101.044) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.878**(16.057) | **0.927**(20.891) | **0.973**(32.115) | **0.875**(55.923) | **0.925**(78.162) | **0.972**(136.718) |
| | AICc | 0.846(14.870) | 0.903(19.416) | 0.962(30.098) | 0.849(52.076) | 0.906(73.232) | 0.963(129.878) |
| | BIC | 0.855(15.138) | 0.911(19.766) | 0.966(30.638) | 0.868(54.216) | 0.921(76.233) | 0.971(135.172) |
| | Vector L0 | 0.780(13.580) | 0.850(17.733) | 0.930(27.494) | 0.864(54.093) | 0.917(76.061) | 0.969(134.868) |
| | Oracle | 0.896(11.827) | 0.948(15.442) | 0.990(23.935) | 0.895(40.331) | 0.947(56.712) | 0.989(100.562) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.866**(24.129) | **0.911**(31.393) | **0.957**(48.258) | **0.868**(85.594) | **0.915**(119.633) | **0.962**(209.256) |
| | AICc | 0.841(22.656) | 0.893(29.583) | 0.947(45.858) | 0.846(81.065) | 0.900(113.997) | 0.953(202.176) |
| | BIC | 0.849(23.072) | 0.899(30.125) | 0.951(46.695) | 0.864(84.385) | 0.913(118.654) | 0.961(210.391) |
| | Vector L0 | 0.814(21.722) | 0.871(28.363) | 0.934(43.971) | 0.862(84.065) | 0.911(118.205) | 0.960(209.598) |
| | Oracle | 0.897(11.843) | 0.948(15.462) | 0.989(23.967) | 0.898(40.284) | 0.949(56.645) | 0.990(100.444) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.867**(34.029) | **0.909**(44.273) | **0.953**(68.059) | **0.866**(119.389) | **0.911**(166.867) | **0.958**(291.876) |
| | AICc | 0.843(32.010) | 0.892(41.796) | 0.944(64.790) | 0.845(113.461) | 0.896(159.554) | 0.949(282.971) |
| | BIC | 0.851(32.595) | 0.898(42.558) | 0.947(65.967) | 0.861(117.913) | 0.908(165.799) | 0.957(293.986) |
| | Vector L0 | 0.824(30.880) | 0.876(40.321) | 0.933(62.509) | 0.832(115.545) | 0.886(165.584) | 0.943(318.020) |
| | Oracle | 0.897(11.825) | 0.947(15.440) | 0.989(23.931) | 0.899(40.427) | 0.949(56.845) | 0.990(100.799) |

TABLE 2.24. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table for including $X_1^2$ in main effects.

| | | $m=2$ | | | $m=3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.878**(11.650) | **0.927**(15.157) | **0.973**(23.300) | **0.879**(29.231) | **0.929**(40.855) | **0.975**(71.462) |
| | AICc | 0.845(10.871) | 0.904(14.194) | 0.963(22.003) | 0.855(27.453) | 0.912(38.606) | 0.967(68.467) |
| | BIC | 0.855(11.069) | 0.911(14.453) | 0.966(22.403) | 0.873(28.565) | 0.926(40.166) | 0.974(71.220) |
| | Vector L0 | 0.746(9.471) | 0.819(12.369) | 0.909(19.180) | 0.873(28.504) | 0.925(40.080) | 0.973(71.068) |
| | Oracle | 0.897(8.836) | 0.948(11.537) | 0.990(17.882) | 0.899(21.983) | 0.950(30.910) | 0.990(54.811) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.869**(17.300) | **0.913**(22.508) | **0.959**(34.600) | **0.867**(43.452) | **0.914**(60.732) | **0.961**(106.229) |
| | AICc | 0.845(16.325) | 0.896(21.316) | 0.950(33.043) | 0.846(41.395) | 0.900(58.212) | 0.954(103.239) |
| | BIC | 0.853(16.629) | 0.902(21.712) | 0.953(33.655) | 0.863(42.986) | 0.912(60.444) | 0.961(107.175) |
| | Vector L0 | 0.788(14.943) | 0.850(19.513) | 0.922(30.253) | 0.859(42.730) | 0.911(60.084) | 0.959(106.539) |
| | Oracle | 0.898(8.875) | 0.949(11.587) | 0.990(17.960) | 0.895(21.982) | 0.948(30.910) | 0.989(54.810) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.863**(24.075) | **0.908**(31.322) | **0.954**(48.150) | **0.862**(60.277) | **0.907**(84.248) | **0.955**(147.363) |
| | AICc | 0.842(22.711) | 0.890(29.654) | 0.943(45.968) | 0.843(57.400) | 0.893(80.718) | 0.947(143.155) |
| | BIC | 0.849(23.148) | 0.896(30.223) | 0.947(46.848) | 0.859(59.658) | 0.905(83.886) | 0.954(148.742) |
| | Vector L0 | 0.795(21.012) | 0.854(27.438) | 0.921(42.540) | 0.823(54.930) | 0.878(77.250) | 0.937(137.025) |
| | Oracle | 0.898(8.891) | 0.948(11.609) | 0.990(17.994) | 0.898(21.981) | 0.949(30.909) | 0.990(54.808) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.878**(11.835) | **0.927**(15.398) | **0.973**(23.670) | **0.881**(29.613) | **0.931**(41.389) | **0.975**(72.395) |
| | AICc | 0.844(10.916) | 0.903(14.253) | 0.963(22.094) | 0.855(27.389) | 0.911(38.516) | 0.967(68.309) |
| | BIC | 0.855(11.114) | 0.911(14.512) | 0.966(22.494) | 0.874(28.561) | 0.927(40.160) | 0.974(71.209) |
| | Vector L0 | 0.782(10.014) | 0.852(13.076) | 0.931(20.274) | 0.872(28.602) | 0.923(40.217) | 0.973(71.310) |
| | Oracle | 0.899(8.895) | 0.949(11.614) | 0.990(18.002) | 0.897(21.997) | 0.948(30.931) | 0.989(54.847) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.868**(17.539) | **0.913**(22.819) | **0.959**(35.078) | **0.867**(43.752) | **0.914**(61.151) | **0.961**(106.963) |
| | AICc | 0.842(16.456) | 0.895(21.487) | 0.949(33.308) | 0.848(41.367) | 0.900(58.171) | 0.954(103.168) |
| | BIC | 0.851(16.758) | 0.902(21.881) | 0.953(33.917) | 0.864(42.995) | 0.912(60.456) | 0.960(107.196) |
| | Vector L0 | 0.813(15.713) | 0.869(20.517) | 0.934(31.807) | 0.862(42.826) | 0.910(60.218) | 0.959(106.778) |
| | Oracle | 0.899(8.904) | 0.949(11.625) | 0.989(18.019) | 0.897(22.002) | 0.947(30.937) | 0.990(54.858) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.865**(24.247) | **0.907**(31.547) | **0.953**(48.495) | **0.861**(60.397) | **0.907**(84.415) | **0.955**(147.655) |
| | AICc | 0.841(22.801) | 0.890(29.771) | 0.943(46.151) | 0.841(57.363) | 0.893(80.667) | 0.947(143.064) |
| | BIC | 0.850(23.241) | 0.897(30.345) | 0.947(47.036) | 0.858(59.649) | 0.905(83.873) | 0.955(148.719) |
| | Vector L0 | 0.817(21.905) | 0.871(28.602) | 0.932(44.342) | 0.826(55.609) | 0.880(78.204) | 0.939(138.712) |
| | Oracle | 0.899(8.872) | 0.950(11.584) | 0.990(17.955) | 0.895(21.969) | 0.947(30.891) | 0.989(54.776) |

TABLE 2.25. Empirical coverage rates for prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | | 90% | 95% | 99% |
|---|---|---|---|---|---|
| $m = 2$ | $b = 1/\sqrt{3}$ | GMTask | 0.899(14.778) | 0.948(19.227) | 0.988(29.557) |
| | $\rho_X = 0$ | Univariate GFI | 0.898(14.745) | 0.948(19.183) | 0.989(29.489) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(14.737) | 0.948(19.173) | 0.990(29.474) |
| | $\rho_X = 0$ | Univariate GFI | 0.898(14.689) | 0.949(19.110) | 0.990(29.377) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.897(14.737) | 0.947(19.174) | 0.989(29.475) |
| | $\rho_X = 0$ | Univariate GFI | 0.897(14.693) | 0.947(19.116) | 0.989(29.386) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.899(14.709) | 0.949(19.137) | 0.989(29.418) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.898(14.792) | 0.949(19.245) | 0.989(29.585) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(14.750) | 0.948(19.190) | 0.989(29.500) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(14.701) | 0.948(19.127) | 0.989(29.402) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.896(14.736) | 0.947(19.172) | 0.988(29.472) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(14.694) | 0.947(19.117) | 0.988(29.388) |
| $m = 3$ | $b = 1/\sqrt{3}$ | GMTask | 0.899(67.794) | 0.948(94.754) | 0.989(165.739) |
| | $\rho_X = 0$ | Univariate GFI | 0.898(67.197) | 0.948(93.920) | 0.989(164.280) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.900(67.880) | 0.949(94.875) | 0.989(165.951) |
| | $\rho_X = 0$ | Univariate GFI | 0.899(67.227) | 0.949(93.961) | 0.989(164.352) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.898(67.514) | 0.948(94.363) | 0.989(165.055) |
| | $\rho_X = 0$ | Univariate GFI | 0.898(66.942) | 0.949(93.563) | 0.989(163.656) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.898(67.846) | 0.947(94.826) | 0.989(165.865) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(68.079) | 0.947(95.152) | 0.989(166.435) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(67.703) | 0.948(94.627) | 0.989(165.517) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.898(67.147) | 0.949(93.849) | 0.989(164.157) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.897(67.421) | 0.948(94.233) | 0.989(164.829) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(66.856) | 0.948(93.443) | 0.990(163.446) |

TABLE 2.26. Empirical coverage rates for prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | | 90% | 95% | 99% |
|---|---|---|---|---|---|
| $m = 2$ | $b = 1/\sqrt{3}$ | GMTask | 0.899(14.058) | 0.949(18.290) | 0.990(28.117) |
| | $\rho_X = 0$ | Univariate GFI | 0.897(14.684) | 0.945(19.104) | 0.987(29.368) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.899(14.091) | 0.950(18.333) | 0.990(28.182) |
| | $\rho_X = 0$ | Univariate GFI | 0.897(14.724) | 0.945(19.156) | 0.987(29.448) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.900(14.068) | 0.949(18.303) | 0.989(28.137) |
| | $\rho_X = 0$ | Univariate GFI | 0.896(14.706) | 0.947(19.133) | 0.988(29.412) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.900(14.163) | 0.949(18.427) | 0.989(28.326) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(14.930) | 0.946(19.425) | 0.987(29.861) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.899(14.114) | 0.948(18.363) | 0.990(28.228) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.896(14.751) | 0.946(19.192) | 0.988(29.503) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.900(14.111) | 0.949(18.358) | 0.990(28.221) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.897(14.756) | 0.945(19.198) | 0.986(29.512) |
| $m = 3$ | $b = 1/\sqrt{3}$ | GMTask | 0.897(59.835) | 0.947(83.630) | 0.989(146.282) |
| | $\rho_X = 0$ | Univariate GFI | 0.889(67.150) | 0.938(93.854) | 0.984(164.165) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.896(60.018) | 0.946(83.885) | 0.988(146.729) |
| | $\rho_X = 0$ | Univariate GFI | 0.891(67.171) | 0.938(93.883) | 0.983(164.216) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.898(59.961) | 0.949(83.805) | 0.989(146.589) |
| | $\rho_X = 0$ | Univariate GFI | 0.893(67.173) | 0.940(93.885) | 0.984(164.220) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.900(60.171) | 0.948(84.100) | 0.989(147.103) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.895(68.212) | 0.942(95.338) | 0.983(166.761) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.896(59.944) | 0.948(83.783) | 0.989(146.549) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.890(67.087) | 0.939(93.765) | 0.983(164.010) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.900(60.268) | 0.949(84.235) | 0.989(147.341) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.896(67.663) | 0.943(94.571) | 0.985(165.420) |

TABLE 2.27. Empirical coverage rates for prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | | 90% | 95% | 99% |
|---|---|---|---|---|---|
| $m = 2$ | $b = 1/\sqrt{3}$ | GMTask | 0.899(11.860) | 0.949(15.431) | 0.989(23.721) |
| | $\rho_X = 0$ | Univariate GFI | 0.888(14.788) | 0.935(19.239) | 0.980(29.575) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.896(11.789) | 0.946(15.338) | 0.989(23.578) |
| | $\rho_X = 0$ | Univariate GFI | 0.889(14.721) | 0.934(19.153) | 0.979(29.443) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.898(11.782) | 0.950(15.328) | 0.989(23.564) |
| | $\rho_X = 0$ | Univariate GFI | 0.887(14.699) | 0.935(19.124) | 0.980(29.398) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.900(11.879) | 0.950(15.455) | 0.989(23.758) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.890(14.871) | 0.935(19.347) | 0.980(29.742) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(11.789) | 0.949(15.337) | 0.989(23.577) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.889(14.703) | 0.935(19.129) | 0.980(29.405) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.897(11.796) | 0.949(15.347) | 0.989(23.592) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.891(14.735) | 0.937(19.171) | 0.980(29.470) |
| $m = 3$ | $b = 1/\sqrt{3}$ | GMTask | 0.897(40.114) | 0.947(56.066) | 0.989(98.068) |
| | $\rho_X = 0$ | Univariate GFI | 0.876(67.077) | 0.921(93.751) | 0.969(163.986) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(40.189) | 0.948(56.171) | 0.989(98.251) |
| | $\rho_X = 0$ | Univariate GFI | 0.880(67.319) | 0.923(94.091) | 0.970(164.579) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.897(40.219) | 0.948(56.213) | 0.988(98.326) |
| | $\rho_X = 0$ | Univariate GFI | 0.876(67.330) | 0.920(94.105) | 0.970(164.605) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.898(40.579) | 0.948(56.716) | 0.989(99.205) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.878(68.503) | 0.923(95.745) | 0.970(167.474) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.898(40.241) | 0.947(56.244) | 0.989(98.380) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.878(67.337) | 0.924(94.115) | 0.970(164.621) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.901(40.143) | 0.951(56.106) | 0.989(98.138) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.879(67.310) | 0.922(94.077) | 0.971(164.556) |

TABLE 2.28. Empirical coverage rates for prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 2000, 3)$.

| | | | 90% | 95% | 99% |
|---|---|---|---|---|---|
| $m = 2$ | $b = 1/\sqrt{3}$ | GMTask | 0.897(8.863) | 0.947(11.531) | 0.989(17.726) |
| | $\rho_X = 0$ | Univariate GFI | 0.879(14.695) | 0.925(19.118) | 0.974(29.389) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.899(8.857) | 0.949(11.523) | 0.989(17.714) |
| | $\rho_X = 0$ | Univariate GFI | 0.880(14.735) | 0.925(19.171) | 0.973(29.470) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.897(8.852) | 0.948(11.516) | 0.988(17.704) |
| | $\rho_X = 0$ | Univariate GFI | 0.880(14.692) | 0.924(19.115) | 0.973(29.385) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.900(8.939) | 0.950(11.630) | 0.990(17.878) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.882(14.967) | 0.928(19.473) | 0.975(29.935) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.899(8.846) | 0.948(11.509) | 0.989(17.691) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.881(14.743) | 0.926(19.181) | 0.973(29.486) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.898(8.871) | 0.949(11.542) | 0.989(17.742) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.879(14.745) | 0.927(19.184) | 0.975(29.490) |
| $m = 3$ | $b = 1/\sqrt{3}$ | GMTask | 0.897(21.935) | 0.946(30.658) | 0.989(53.626) |
| | $\rho_X = 0$ | Univariate GFI | 0.865(67.600) | 0.908(94.483) | 0.960(165.265) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.897(21.852) | 0.949(30.542) | 0.989(53.422) |
| | $\rho_X = 0$ | Univariate GFI | 0.866(67.838) | 0.909(94.815) | 0.959(165.846) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.899(21.895) | 0.950(30.602) | 0.990(53.527) |
| | $\rho_X = 0$ | Univariate GFI | 0.867(67.557) | 0.910(94.423) | 0.960(165.161) |
| | $b = 1/\sqrt{3}$ | GMTask | 0.898(22.059) | 0.947(30.831) | 0.989(53.928) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.864(68.011) | 0.906(95.057) | 0.958(166.270) |
| | $b = 2/\sqrt{3}$ | GMTask | 0.896(21.851) | 0.947(30.541) | 0.988(53.421) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.863(67.459) | 0.908(94.286) | 0.959(164.921) |
| | $b = 3/\sqrt{3}$ | GMTask | 0.899(21.812) | 0.949(30.486) | 0.989(53.324) |
| | $\rho_X = 0.5$ | Univariate GFI | 0.866(67.329) | 0.907(94.103) | 0.958(164.601) |

TABLE 2.29. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.3$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 100, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.897**(14.060) | **0.947**(18.292) | **0.989**(28.120) | **0.897**(59.986) | **0.949**(83.841) | **0.989**(146.651) |
| | AICc | 0.881(13.705) | 0.937(17.894) | 0.986(27.739) | 0.882(58.830) | 0.940(82.729) | 0.986(146.721) |
| | BIC | 0.893(13.956) | 0.944(18.222) | 0.989(28.245) | 0.897(60.083) | 0.949(84.485) | 0.989(149.812) |
| | Vector L0 | 0.893(13.973) | 0.945(18.245) | 0.989(28.280) | 0.897(60.156) | 0.950(84.588) | 0.989(149.993) |
| | Oracle | 0.898(14.084) | 0.948(18.389) | 0.990(28.503) | 0.898(60.281) | 0.950(84.764) | 0.990(150.304) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.899**(14.074) | **0.948**(18.310) | **0.989**(28.148) | **0.897**(60.024) | **0.948**(83.894) | **0.988**(146.743) |
| | AICc | 0.882(13.719) | 0.937(17.913) | 0.985(27.768) | 0.882(58.882) | 0.939(82.803) | 0.986(146.851) |
| | BIC | 0.894(13.968) | 0.946(18.237) | 0.988(28.268) | 0.896(60.153) | 0.949(84.584) | 0.989(149.987) |
| | Vector L0 | 0.894(13.976) | 0.946(18.248) | 0.988(28.285) | 0.897(60.197) | 0.949(84.646) | 0.989(150.096) |
| | Oracle | 0.899(14.099) | 0.950(18.409) | 0.989(28.534) | 0.898(60.320) | 0.950(84.818) | 0.989(150.400) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.898**(14.128) | **0.949**(18.381) | **0.989**(28.256) | **0.896**(59.801) | **0.946**(83.582) | **0.988**(146.198) |
| | AICc | 0.881(13.773) | 0.938(17.983) | 0.986(27.877) | 0.882(58.650) | 0.938(82.476) | 0.986(146.272) |
| | BIC | 0.893(14.017) | 0.947(18.301) | 0.988(28.368) | 0.896(59.936) | 0.947(84.278) | 0.989(149.444) |
| | Vector L0 | 0.893(14.028) | 0.946(18.316) | 0.988(28.390) | 0.896(59.972) | 0.947(84.328) | 0.989(149.533) |
| | Oracle | 0.898(14.152) | 0.950(18.478) | 0.990(28.641) | 0.897(60.097) | 0.947(84.505) | 0.989(149.845) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(14.096) | **0.948**(18.339) | **0.990**(28.192) | **0.900**(60.114) | **0.949**(84.020) | **0.989**(146.965) |
| | AICc | 0.881(13.760) | 0.937(17.966) | 0.987(27.851) | 0.884(58.962) | 0.941(82.914) | 0.986(147.049) |
| | BIC | 0.894(13.994) | 0.945(18.272) | 0.989(28.322) | 0.900(60.193) | 0.949(84.639) | 0.990(150.085) |
| | Vector L0 | 0.894(14.015) | 0.946(18.299) | 0.989(28.364) | 0.900(60.217) | 0.950(84.674) | 0.990(150.147) |
| | Oracle | 0.899(14.119) | 0.949(18.435) | 0.990(28.574) | 0.901(60.410) | 0.950(84.944) | 0.990(150.624) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.896**(14.091) | **0.947**(18.333) | **0.989**(28.182) | **0.899**(59.997) | **0.949**(83.856) | **0.988**(146.677) |
| | AICc | 0.879(13.757) | 0.936(17.963) | 0.986(27.846) | 0.885(58.891) | 0.940(82.815) | 0.986(146.872) |
| | BIC | 0.892(13.998) | 0.944(18.276) | 0.989(28.329) | 0.899(60.130) | 0.949(84.552) | 0.989(149.929) |
| | Vector L0 | 0.893(14.004) | 0.945(18.285) | 0.989(28.342) | 0.899(60.143) | 0.950(84.570) | 0.989(149.962) |
| | Oracle | 0.897(14.114) | 0.948(18.428) | 0.990(28.564) | 0.900(60.296) | 0.950(84.784) | 0.989(150.340) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.897**(14.075) | **0.948**(18.312) | **0.989**(28.150) | **0.899**(59.851) | **0.947**(83.652) | **0.989**(146.321) |
| | AICc | 0.881(13.741) | 0.937(17.942) | 0.986(27.813) | 0.884(58.799) | 0.939(82.685) | 0.987(146.643) |
| | BIC | 0.893(13.977) | 0.945(18.249) | 0.988(28.287) | 0.898(60.006) | 0.948(84.377) | 0.989(149.619) |
| | Vector L0 | 0.893(13.994) | 0.945(18.271) | 0.989(28.321) | 0.898(60.027) | 0.948(84.406) | 0.989(149.671) |
| | Oracle | 0.898(14.099) | 0.949(18.408) | 0.989(28.533) | 0.900(60.137) | 0.948(84.561) | 0.990(149.945) |

TABLE 2.30. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.6$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 100, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(11.820) | **0.949**(15.378) | **0.990**(23.639) | **0.900**(40.418) | **0.950**(56.491) | **0.990**(98.812) |
| | AICc | 0.883(11.561) | 0.939(15.095) | 0.987(23.400) | 0.887(39.830) | 0.943(56.011) | 0.988(99.334) |
| | BIC | 0.895(11.751) | 0.948(15.343) | 0.990(23.782) | 0.900(40.539) | 0.951(57.004) | 0.991(101.081) |
| | Vector L0 | 0.896(11.765) | 0.948(15.361) | 0.990(23.810) | 0.900(40.557) | 0.951(57.029) | 0.991(101.125) |
| | Oracle | 0.899(11.838) | 0.950(15.457) | 0.991(23.958) | 0.900(40.612) | 0.951(57.106) | 0.991(101.260) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.900**(11.822) | **0.949**(15.381) | **0.989**(23.645) | **0.897**(40.170) | **0.947**(56.144) | **0.988**(98.205) |
| | AICc | 0.883(11.561) | 0.940(15.096) | 0.985(23.401) | 0.884(39.578) | 0.939(55.656) | 0.986(98.706) |
| | BIC | 0.896(11.755) | 0.947(15.348) | 0.988(23.789) | 0.897(40.290) | 0.948(56.654) | 0.989(100.460) |
| | Vector L0 | 0.897(11.775) | 0.948(15.375) | 0.988(23.831) | 0.897(40.310) | 0.948(56.682) | 0.989(100.510) |
| | Oracle | 0.901(11.843) | 0.950(15.462) | 0.989(23.967) | 0.898(40.370) | 0.949(56.766) | 0.989(100.658) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.901**(11.803) | **0.949**(15.356) | **0.989**(23.606) | **0.896**(40.192) | **0.946**(56.175) | **0.989**(98.259) |
| | AICc | 0.883(11.546) | 0.939(15.076) | 0.987(23.371) | 0.884(39.629) | 0.939(55.727) | 0.987(98.831) |
| | BIC | 0.896(11.737) | 0.947(15.325) | 0.989(23.754) | 0.896(40.308) | 0.947(56.679) | 0.989(100.504) |
| | Vector L0 | 0.897(11.755) | 0.948(15.348) | 0.989(23.790) | 0.896(40.326) | 0.947(56.704) | 0.990(100.549) |
| | Oracle | 0.900(11.823) | 0.950(15.437) | 0.990(23.928) | 0.897(40.392) | 0.948(56.797) | 0.990(100.712) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(11.864) | **0.949**(15.435) | **0.990**(23.728) | **0.898**(40.163) | **0.948**(56.135) | **0.989**(98.188) |
| | AICc | 0.883(11.611) | 0.939(15.161) | 0.986(23.501) | 0.886(39.559) | 0.941(55.629) | 0.987(98.657) |
| | BIC | 0.894(11.794) | 0.947(15.399) | 0.989(23.869) | 0.899(40.247) | 0.948(56.593) | 0.990(100.351) |
| | Vector L0 | 0.894(11.803) | 0.947(15.411) | 0.989(23.887) | 0.899(40.287) | 0.948(56.650) | 0.990(100.452) |
| | Oracle | 0.899(11.880) | 0.950(15.511) | 0.990(24.042) | 0.900(40.342) | 0.949(56.726) | 0.990(100.588) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.896**(11.777) | **0.947**(15.322) | **0.989**(23.554) | **0.897**(40.319) | **0.950**(56.352) | **0.990**(98.569) |
| | AICc | 0.880(11.530) | 0.938(15.055) | 0.986(23.337) | 0.886(39.778) | 0.942(55.937) | 0.987(99.202) |
| | BIC | 0.893(11.709) | 0.946(15.288) | 0.989(23.696) | 0.897(40.434) | 0.950(56.856) | 0.990(100.819) |
| | Vector L0 | 0.894(11.726) | 0.946(15.311) | 0.989(23.732) | 0.897(40.446) | 0.950(56.872) | 0.990(100.847) |
| | Oracle | 0.897(11.796) | 0.948(15.401) | 0.990(23.872) | 0.898(40.513) | 0.951(56.967) | 0.990(101.015) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.898**(11.795) | **0.948**(15.346) | **0.989**(23.590) | **0.897**(40.421) | **0.948**(56.495) | **0.989**(98.818) |
| | AICc | 0.880(11.544) | 0.938(15.073) | 0.986(23.366) | 0.885(39.864) | 0.941(56.058) | 0.988(99.418) |
| | BIC | 0.894(11.725) | 0.946(15.309) | 0.989(23.729) | 0.897(40.528) | 0.949(56.987) | 0.990(101.051) |
| | Vector L0 | 0.894(11.740) | 0.947(15.328) | 0.989(23.759) | 0.897(40.556) | 0.950(57.027) | 0.990(101.122) |
| | Oracle | 0.898(11.812) | 0.950(15.423) | 0.990(23.905) | 0.898(40.616) | 0.950(57.112) | 0.990(101.271) |

TABLE 2.31. Empirical coverage rates of prediction ellipsoids for $Y_i$ and $\rho_\Sigma = 0.8$. Numbers in parentheses are averaged volumes of the ellipsoids. This table is for the case when $(n, p, k) = (200, 100, 3)$.

| | | $m = 2$ | | | $m = 3$ | | |
|---|---|---|---|---|---|---|---|
| $(b, \rho_X)$ | method | 90% | 95% | 99% | 90% | 95% | 99% |
| $(\frac{1}{\sqrt{3}}, 0)$ | GMTask | **0.898**(8.859) | **0.948**(11.526) | **0.989**(17.718) | **0.894**(21.978) | **0.946**(30.718) | **0.989**(53.731) |
| | AICc | 0.882(8.674) | 0.939(11.326) | 0.986(17.558) | 0.882(21.673) | 0.937(30.477) | 0.987(54.050) |
| | BIC | 0.896(8.818) | 0.947(11.514) | 0.989(17.846) | 0.895(22.041) | 0.947(30.992) | 0.990(54.956) |
| | Vector L0 | 0.896(8.825) | 0.947(11.523) | 0.989(17.861) | 0.895(22.059) | 0.947(31.018) | 0.990(55.001) |
| | Oracle | 0.899(8.876) | 0.949(11.589) | 0.990(17.962) | 0.895(22.084) | 0.948(31.052) | 0.990(55.063) |
| $(\frac{2}{\sqrt{3}}, 0)$ | GMTask | **0.897**(8.853) | **0.947**(11.517) | **0.988**(17.705) | **0.900**(21.846) | **0.949**(30.533) | **0.990**(53.407) |
| | AICc | 0.883(8.674) | 0.937(11.326) | 0.986(17.557) | 0.887(21.548) | 0.942(30.302) | 0.988(53.740) |
| | BIC | 0.895(8.812) | 0.945(11.505) | 0.989(17.833) | 0.899(21.905) | 0.949(30.801) | 0.990(54.617) |
| | Vector L0 | 0.895(8.820) | 0.945(11.516) | 0.989(17.850) | 0.900(21.927) | 0.950(30.833) | 0.990(54.673) |
| | Oracle | 0.898(8.868) | 0.948(11.578) | 0.989(17.947) | 0.901(21.955) | 0.950(30.872) | 0.991(54.742) |
| $(\frac{3}{\sqrt{3}}, 0)$ | GMTask | **0.897**(8.855) | **0.947**(11.521) | **0.989**(17.711) | **0.898**(21.901) | **0.948**(30.610) | **0.989**(53.542) |
| | AICc | 0.881(8.667) | 0.937(11.317) | 0.986(17.543) | 0.884(21.606) | 0.940(30.383) | 0.987(53.884) |
| | BIC | 0.894(8.810) | 0.946(11.502) | 0.989(17.829) | 0.898(21.962) | 0.949(30.881) | 0.989(54.759) |
| | Vector L0 | 0.894(8.817) | 0.946(11.512) | 0.989(17.844) | 0.898(21.976) | 0.949(30.901) | 0.989(54.795) |
| | Oracle | 0.898(8.869) | 0.948(11.580) | 0.990(17.949) | 0.899(22.009) | 0.949(30.948) | 0.989(54.877) |
| $(\frac{1}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(8.897) | **0.948**(11.576) | **0.990**(17.795) | **0.896**(21.840) | **0.947**(30.525) | **0.989**(53.394) |
| | AICc | 0.883(8.708) | 0.938(11.370) | 0.986(17.625) | 0.884(21.511) | 0.940(30.249) | 0.987(53.646) |
| | BIC | 0.896(8.839) | 0.946(11.540) | 0.989(17.888) | 0.896(21.856) | 0.948(30.733) | 0.989(54.496) |
| | Vector L0 | 0.896(8.843) | 0.946(11.546) | 0.989(17.897) | 0.896(21.858) | 0.948(30.735) | 0.989(54.500) |
| | Oracle | 0.900(8.899) | 0.949(11.619) | 0.990(18.010) | 0.897(21.890) | 0.949(30.781) | 0.990(54.581) |
| $(\frac{2}{\sqrt{3}}, 0.5)$ | GMTask | **0.895**(8.827) | **0.946**(11.484) | **0.989**(17.654) | **0.897**(21.931) | **0.948**(30.653) | **0.988**(53.616) |
| | AICc | 0.880(8.653) | 0.936(11.298) | 0.986(17.514) | 0.885(21.656) | 0.941(30.454) | 0.987(54.008) |
| | BIC | 0.892(8.783) | 0.943(11.467) | 0.989(17.774) | 0.897(22.000) | 0.949(30.935) | 0.989(54.855) |
| | Vector L0 | 0.893(8.791) | 0.944(11.478) | 0.989(17.792) | 0.897(22.009) | 0.949(30.948) | 0.989(54.878) |
| | Oracle | 0.896(8.841) | 0.947(11.544) | 0.990(17.893) | 0.898(22.039) | 0.950(30.990) | 0.989(54.951) |
| $(\frac{3}{\sqrt{3}}, 0.5)$ | GMTask | **0.899**(8.849) | **0.949**(11.513) | **0.990**(17.698) | **0.897**(21.797) | **0.949**(30.466) | **0.989**(53.289) |
| | AICc | 0.883(8.677) | 0.939(11.330) | 0.987(17.563) | 0.885(21.530) | 0.941(30.277) | 0.987(53.694) |
| | BIC | 0.896(8.807) | 0.947(11.499) | 0.990(17.824) | 0.898(21.865) | 0.949(30.745) | 0.990(54.517) |
| | Vector L0 | 0.895(8.809) | 0.947(11.501) | 0.990(17.827) | 0.897(21.869) | 0.949(30.750) | 0.990(54.527) |
| | Oracle | 0.900(8.864) | 0.950(11.574) | 0.991(17.940) | 0.899(21.906) | 0.950(30.803) | 0.990(54.620) |

TABLE 2.32. Empirical coverage rates of leave-one-out prediction ellipsoids for $Y_i$ on PCR Data. Numbers in parentheses are averaged volumes of the ellipsoids.

| | 90% | 95% | 99% |
|---|---|---|---|
| GMTask | 0.900(8.087) | 0.950(10.521) | 0.983(16.174) |
| AICc | 0.800(4.894) | 0.850(6.462) | 0.900(10.281) |
| BIC | 0.817(5.109) | 0.833(6.740) | 0.933(10.704) |
| Vector L0 | 0.683(4.545) | 0.783(6.023) | 0.833(9.670) |

TABLE 2.33. Empirical coverage rates of prediction ellipsoids for $Y_i$ on AD cognitive scores Data. Numbers in parentheses are averaged volumes of the ellipsoids.

| | 90% | 95% | 99% |
|---|---|---|---|
| GMTask | 0.890(905851) | 0.938(1424698) | 0.983(3088587) |
| AICc | 0.887(889203) | 0.935(1402863) | 0.980(3062007) |
| BIC | 0.894(920738) | 0.934(1452579) | 0.983(3170354) |
| Vector L0 | 0.853(979592) | 0.925(1545990) | 0.973(3376913) |

CHAPTER 3

# Extending the Use of MDL for High-Dimensional Problems: Variable Selection, Robust Fitting, and Additive Modeling

## 3.1. Introduction

The minimum description length (MDL) principle (Rissanen, 1989, 2007) has long been applied to perform signal denoising (Cohen *et al.*, 1999; Lee, 2000; Morsheddost *et al.*, 2014; Rissanen, 2000; Roos *et al.*, 2009) and model selection in regression and time series problems (Aue *et al.*, 2014; Cheung *et al.*, 2017; Gao *et al.*, 2017; Giurcăneanu *et al.*, 2011; Kallummil and Kalyani, 2016; Schmidt and Makalic, 2011; Wong *et al.*, 2010). It has been shown that for such problems the MDL solutions often enjoy consistent properties and produce very promising empirical results. The main goal of this chapter is to further extend the use of MDL to the so-called high-dimensional setting, where the number of predictors $p$ is larger than the number of observations $n$; that is, the "large $p$ small $n$" setting. We will first consider the case of linear regression, then allow for outliers in the data, and lastly extend to nonparametric additive models.

A typical description of the high-dimensional linear regression problem is as follows. Let $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ be a vector of $n$ responses and $\boldsymbol{x}_i$ be a $p$-variate predictor variable for $y_i$. Write $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$ as the design matrix of size $n \times p$. The observed responses and the predictors are related by the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3.1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is a vector of unknown parameters and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$ is a vector of i.i.d. random Gaussian errors with zero mean and unknown variance $\sigma^2$. It is assumed that $p \gg n$, making this high-dimensional regression problem different from the classical multiple regression regression problem for which $p < n$.

When $p \gg n$, one needs to assume that the number of significant predictors in the true model is small; i.e., the true model is sparse. The problem is then to identify which $\beta_j$'s are non-zero. This is sometimes known as the variable selection problem, and this dissertation applies the MDL principle to derive a solution. Although there are existing methods for addressing this problem (Fan and Lv, 2010), this dissertation seems to be one of the earliest attempts that MDL is being applied and carefully studied in high-dimensional settings.

This dissertation also considers robust fitting, by relaxing the Gaussianity assumption on $\boldsymbol{\varepsilon}$ and allowing the presence of heavy-tailed errors or outliers in the response. It achieves this goal by modeling the error component with the Laplace distribution.

Lastly this dissertation extends the variable selection problem for (3.1) to high-dimensional nonparametric additive models, defined as

$$y_i = \mu + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{3.2}$$

where $\mu$ is an intercept term, the $f_j$'s are unknown nonparametric functions, and $x_{ij}$ is the $j$th covariate of $\boldsymbol{x}_i$. Again, we consider $p \gg n$ and impose the sparsity assumption. We select the significant predictors, as well as allowing the possibility of outliers. To the best of our knowledge, this is the first time that outlier-resistant estimation and variable selection for high-dimensional nonparametric additive models is considered.

Below we first provide some background on MDL. We then present our new MDL solutions to the problems of high-dimensional linear regression, nonparametric additive models, and robust fitting. Both theoretical and empirical properties of our proposed methods will also be reported.

### 3.2. A Brief Description of the Minimum Description Length (MDL) Principle

In model selection problems the MDL principle defines the best fitting model as the one that produces the shortest code length of the data (Rissanen, 1989, 2007). In this context the code length of an object can be treated as the amount of memory space that is required to store the object. Of course comparing code lengths is neither the only nor the best approach for defining a best fitting model, but it is still a sensible one. It is because a common feature of a good encoding

(or compression) scheme and a good statistical model is the ability to capture the regularities, or patterns, hidden in the data.

There are different versions of MDL, and this dissertation focuses on the so-called two-part codes. When applying this, it is common to split the code length for a set of data into two parts: (i) a fitted model plus (ii) the data "conditioned on" the fitted model; i.e., the residuals. If we denote the data as $\boldsymbol{y}$, any fitted model as $\hat{\boldsymbol{\theta}}$, and the residuals as $\hat{\boldsymbol{e}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$, where $\hat{\boldsymbol{y}}$ is the fitted value of $\boldsymbol{y}$, we split $\boldsymbol{y}$ into $\hat{\boldsymbol{\theta}}$ plus $\hat{\boldsymbol{e}}$. Notice that knowing $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{e}}$ can completely retrieve $\boldsymbol{y}$.

If $\mathrm{CL}(z)$ denotes the code length of an object $z$, we have

$$\mathrm{CL}(\boldsymbol{y}) = \mathrm{CL}(\hat{\boldsymbol{\theta}}) + \mathrm{CL}(\hat{\boldsymbol{e}}|\hat{\boldsymbol{\theta}}).$$

Note that in this expression it is stressed that $\hat{\boldsymbol{e}}$ is conditional on $\hat{\boldsymbol{\theta}}$; i.e., different $\hat{\boldsymbol{\theta}}$'s would give different $\hat{\boldsymbol{e}}$'s. Now the task is to find an expression for $\mathrm{CL}(\boldsymbol{y})$ so that the best MDL $\hat{\boldsymbol{\theta}}$ can be defined and obtained as its minimizer.

### 3.3. High-Dimensional Linear Regression

We first consider variable selection for model (3.1). Let $S$ be a subset of $\{1, \ldots, p\}$. If $j \in S$, it means $\beta_j$ is significant. Hence $S$ can be used to represent any candidate model. Denote the corresponding design matrix as $\boldsymbol{X}_S$, and the maximum likelihood estimate of the corresponding coefficients $\boldsymbol{\beta}_S$ as $\hat{\boldsymbol{\beta}}_S$. Also, let $L(\cdot)$ be the likelihood function and $|S|$ be the number of elements in $S$, i.e., the number of significant $\beta_j$'s. It is shown in Section 3.3.1 that a MDL criterion for the model specified by $S$ is

$$\begin{aligned}
\mathrm{MDL}(S) &= -\log L(\boldsymbol{y}, \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S) + \frac{|S|}{2}\log(n) + |S|\log(p) \\
&= \frac{n}{2}\log\left\{\frac{(\boldsymbol{y} - \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S)^T(\boldsymbol{y} - \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S)}{n}\right\} + \frac{|S|}{2}\log(n) + |S|\log(p) \\
&= \frac{n}{2}\log\left(\frac{\mathrm{RSS}}{n}\right) + \frac{|S|}{2}\log(n) + |S|\log(p), \qquad (3.3)
\end{aligned}$$

where

$$\mathrm{RSS} = (\boldsymbol{y} - \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S)^T(\boldsymbol{y} - \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S)$$

is the residual sum of squares. When comparing to the classical MDL criterion for $p < n$, this new MDL($S$) has an additional penalty term $|S| \log(p)$, which coincidentally shares the same asymptotic order as the corresponding penalty term in EBIC of Chen and Chen (2008). We note that, however, MDL($S$) is different from EBIC for finite samples.

**3.3.1. Derivation of MDL.** This section outlines the derivation of (3.3). Here the parameter vector estimate $\hat{\boldsymbol{\theta}}$ is $\hat{\boldsymbol{\beta}}_S$, so we begin with

$$\mathrm{MDL}(S) = \mathrm{CL}(\boldsymbol{y}) = \mathrm{CL}(\hat{\boldsymbol{\beta}}_S) + \mathrm{CL}(\hat{\boldsymbol{e}}|\hat{\boldsymbol{\beta}}_S).$$

According to Rissanen (1989), the code length for encoding an integer $N$ is approximately $\log_2 N$ bits. To encode $\hat{\boldsymbol{\beta}}_S$, one first needs to identify which of the $|S|$ predictors are selected. Since each of the $|S|$ predictors can be uniquely identified by an index in $\{1, \ldots, p\}$, it takes a total of $|S| \log_2 p$ bits to encode this information. Next, the corresponding parameter estimates need to be encoded. In Rissanen (1989) it is demonstrated that if a maximum likelihood estimate of a real-valued parameter is computed from $N$ data points, then it can be effectively encoded with $\frac{1}{2} \log_2 N$ bits. This gives the total code length for the $|S|$ parameter estimates as $\frac{|S|}{2} \log_2 n$, and hence

$$\mathrm{CL}(\hat{\boldsymbol{\beta}}_S) = |S| \log_2 p + \frac{|S|}{2} \log_2 n. \tag{3.4}$$

Notice that in classical applications of MDL for problems with $p \ll n$, the term $|S| \log_2 p$ is often omitted as it is relatively small compared with $\frac{|S|}{2} \log_2 n$. However, when $p$ is comparable to $n$ or even $p \gg n$ , this term cannot be omitted as otherwise it will give erratic results.

Now it remains to calculate $\mathrm{CL}(\hat{\boldsymbol{e}}|\hat{\boldsymbol{\beta}}_S)$, and it is shown in Rissanen (1989) that this is equal to the negative of the log of the likelihood of $\hat{\boldsymbol{e}}$ conditioned on $\hat{\boldsymbol{\beta}}_S$. For the present problem, it simplifies to

$$\mathrm{CL}(\hat{\boldsymbol{e}}|\hat{\boldsymbol{\beta}}_S) = \frac{n}{2} \log_2 \left( \frac{\mathrm{RSS}}{n} \right). \tag{3.5}$$

Now by changing $\log_2$ to log and combining (3.4) and (3.5), one obtains MDL($S$) in (3.3).

**3.3.2. Practical Minimization of (3.3).** In practice minimizing (3.3) is not a trivial task, especially when $p \gg n$. This subsection presents a three-stage procedure that aims to locate a good approximated minimizer of (3.3). The first stage is to apply a screening procedure to remove a large number of insignificant predictors, so that we will only have to consider the remaining $m$ predictors, where $m < n$.

Then in the second stage the lasso (Tibshirani, 1996) method is applied to obtain a nested sequence of $m$ candidate models. Lastly, the MDL($S$) values for these $m$ candidate models are calculated and the one with the smallest value is taken as the final, best fitting model.

*Stage 1: Screening.* The goal here is to remove a lot of non-significant predictors quickly with high confidence. We propose using the sure independence screening (SIS) procedure of Fan and Lv (2008). The idea is to rank the predictors according to the magnitudes of their sample correlations with the response variable, and keep the $m$ largest ones. More precisely, let $\boldsymbol{\omega} = (\omega_1, ..., \omega_p)^T = \boldsymbol{X}^T\boldsymbol{y}$, where we assume that each column of the $n \times p$ design matrix $\boldsymbol{X}$ has been standardized with mean zero and variance one. We keep the $m$ predictors that have the largest $m$ values of $|w_j|$, and collect them in $S^*$; i.e.,

$$S^* = \{1 \leq j \leq p : |\omega_j| \text{ is among the first } m \text{ largest of all}\}.$$

This reduces the number of possible predictors $p \gg n$ to a more manageable number $m$. This SIS procedure will remove those predictors that have weak marginal correlations with the response, and has been shown to possess excellent theoretical and empirical properties (e.g., see Fan and Lv (2008)). In practice we set $m = n - 1$.

*Stage 2: Lasso fitting.* Lasso was proposed by Tibshirani (1996) to perform variable selection and shrinkage estimation for linear models. It produces a so-called solution path from which a sequence of nested models can be obtained. In the original lasso, cross-validation was suggested to choose a final model from these nested models. Here, however, we simply apply lasso to $S^*$ and obtain $m$ nested models. Given the LARS algorithm (Efron *et al.*, 2004), this step can be performed very efficiently.

*Stage 3: MDL($S$) Calculation.* Here we use (3.3) to choose a final best fitting model from those nested models obtained above. However, given the shrinkage nature of lasso, the parameter

estimates of these nested models obtained from above are shrunk towards zero. Therefore, these estimates (and other quantities derived from them such as RSS) should not be used for the calculation of (3.3). Thus, for each of the nested models, we use maximum likelihood to estimate the unknown parameters, and use these estimates to calculate (3.3). The model that gives the smallest value of (3.3) is taken as the final model.

### 3.4. Theoretical Properties

This section presents some theoretical backup for the above MDL criterion for high-dimensional regression. Let $S_0$ be the index set of the true model, and

$$\boldsymbol{\mu} = E(\boldsymbol{y}) = \boldsymbol{X}_{S_0}\boldsymbol{\beta}_{S_0}.$$

Define the projection matrix for any $S \subset \{1, \ldots, p\}$ as

$$\boldsymbol{P}_S = \boldsymbol{X}_S(\boldsymbol{X}_S^T\boldsymbol{X}_S)^{-1}\boldsymbol{X}_S^T,$$

and write $\delta(S) = \|\boldsymbol{\mu} - \boldsymbol{P}_S\boldsymbol{\mu}\|^2$, with $\|\cdot\|$ being the Euclidean norm. Clearly, if $S_0 \subset S$, we have $\delta(S) = 0$. In our theoretical analysis, we need the following identifiability condition, which is similar to the condition stated in Chen and Chen (2008).

CONDITION 1. *(Asymptotic identifiability) The true model $S_0$ is asymptotically identifiable if*

$$\lim_{n\to\infty} \min\left\{\frac{\delta(S)}{\log n} : S \neq S_0, |S| \leq k|S_0|\right\} = \infty$$

*for some fixed $k > 1$.*

Roughly speaking, a true model is asymptotically identifiable if no other model of finite size can predict the response as well as the true model. We now have the following theorem.

THEOREM 3.4.1. *Consider a data set $\{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$ from model (3.1). Suppose Condition 1 holds and $p = O(n^\gamma)$ for some fixed $\gamma$. Also assume $\varepsilon_1, \ldots, \varepsilon_n \overset{i.i.d.}{\sim} N(0, \sigma^2)$. Then we have*

$$P\left(\min_{S\neq S_0, |S|\leq k|S_0|} MDL(S) > MDL(S_0)\right) \to 1$$

*as $n \to \infty$.*

We note that the penalties in (3.3), derived from the MDL framework, share the same order as the one in EBIC. Indeed, the proof of Theorem 3.4.1 can be constructed by using similar idea in Chen and Chen (2008), and can be found in the appendix. Theorem 3.4.1 indicates that, as $n \to \infty$, the probability that MDL($S$) wrongly selects a model of a similar size other than the true model goes to zero.

### 3.5. Robust Fitting for High-Dimensional Linear Regression

This section demonstrates how robust estimation for high-dimensional linear regression can be handled by MDL. We first review some existing work in this area.

A robust version of lasso was proposed by Wang *et al.* (2007). It replaces the $L_2$ norm for measuring data fidelity with a least absolute deviation type of norm, which will make the method less sensitive to the presence of outliers. However, this method was not originally designed for the high-dimensional setting. Another method termed RLARS, short for robust least angle regression, was developed by Khan *et al.* (2007). It is a robust version of the LARS algorithm (Efron *et al.*, 2004), and it essentially uses a robust correlation to rank and select the most important variables. More recently, a sparse and regularized version of the least trimmed squares (LTS) was proposed by Alfons *et al.* (2013), which introduces an $L_1$ penalty to the LTS estimator. This sparse LTS estimator can also be interpreted as a trimmed version of the lasso. However, this method is computationally expensive.

Robust fitting can be embedded into the MDL framework. A natural approach is to adopt a heavy tail distribution for the errors $\varepsilon_i$'s to allow for outliers. For this we suggest using the zero mean Laplace distribution; i.e.,

$$\varepsilon_i \overset{\text{i.i.d.}}{\sim} \text{Laplace}(0, b),$$

where $b$ is a scale parameter.

Similar to Section 3.3.1, it can be shown that the MDL criterion for robust fitting with a model specified by $S$ is

$$\begin{aligned} \mathrm{MDL}_{\mathrm{robust}}(S) &= -\log L(\boldsymbol{y}, \boldsymbol{X}_S\hat{\boldsymbol{\beta}}_S) + \frac{|S|}{2}\log(n) + |S|\log(p) \\ &= n\log\left(\frac{\sum_{i=1}^n |y_i - \boldsymbol{x}_{S,i}\hat{\boldsymbol{\beta}}_S|}{n}\right) + \frac{|S|}{2}\log(n) + |S|\log(p) \\ &= n\log\left(\frac{\mathrm{SAE}}{n}\right) + \frac{|S|}{2}\log(n) + |S|\log(p). \end{aligned} \tag{3.6}$$

In the above $\mathrm{SAE} = \sum_{i=1}^n |y_i - \boldsymbol{x}_{S,i}\hat{\boldsymbol{\beta}}_S|$ is the sum of absolute errors, with $\boldsymbol{x}_{S,i}$ denoting the $i$-th row of $\boldsymbol{X}_S$. Note that the maximum likelihood estimate for the scale parameter $b$ is $\hat{b} = \mathrm{SAE}/n$.

Practical minimization of (3.6) can be achieved in a similar fashion as the 3-stage procedure described in Section 3.3.2. To be more specific, Stage 1 remains the same, while in Stage 2 the robust LARS method of Khan *et al.* (2007) is used in place of the original lasso, and in Stage 3 the MDL criterion (3.6) is used instead of (3.3) when calculating the MDL values.

## 3.6. High-Dimensional Nonparametric Additive Models

This section extends our work to the high-dimensional nonparametric additive models (3.2). The goal is to select those significant ones from the functions $f_1, \ldots, f_p$, as well as to estimate them nonparametrically. We first discuss the use of splines for modeling the $f_j$'s.

**3.6.1. Spline Modeling for Additive Functions.** Briefly, a spline function is a piecewise polynomial function. The locations at which two adjacent pieces join are called knots. Here we state their standard conditions and definition.

Suppose that $x \in [a, b]$ for some finite numbers $a < b$ and that $E(y^2) < \infty$.

To ensure identifiability, it is assumed $E\{f_j(x)\} = 0$ for $j = 1, \ldots, p$. Let $K$ be the number of knots for a partition of $[a, b]$ that satisfy specific conditions stated for example in Lai *et al.* (2012). Let $\mathcal{S}_n$ be the collection of functions $s$ with domain $[a, b]$ satisfying the following two conditions: (i) $s$ is a polynomial of degree $l$ (or less) on each subinterval, and (ii) for any two integers $l$ and $l'$ satisfying $l \geq 2$ and $0 \leq l' < l - 1$, $s$ is $l'$-times continuously differentiable on $[a, b]$. Then there exists a normalized B-spline basis $\{\varphi_k(\cdot), k = 1, \ldots, d_n\}$ such that for any $s \in \mathcal{S}_n$, we have

$$s(x) = \sum_{k=1}^{d_n} \alpha_k \varphi_k(x), \tag{3.7}$$

where $\alpha_k$ is the coefficient of the basis function $\varphi_k(x)$ for $k = 1, \ldots, d_n$ with $d_n = K + l$. Since $\mathcal{S}_n$ is a relatively rich class of smooth functions, in this dissertation, for the reason of speeding up technical calculations, we shall assume that the spline representation (3.7) is exact for the functions $f_j$'s. In other words, for $j = 1, \ldots, p$, it is assumed that

$$f_j(x) = \sum_{k=1}^{d_n} \alpha_{jk} \varphi_k(x), \tag{3.8}$$

where $\alpha_{jk}$'s are the corresponding coefficients of the bases $\varphi_k(x)$'s.

**3.6.2. MDL Criteria.** Recall that for the fitting of the high-dimensional nonparametric additive models (3.2), we aim to select those significant functions from $f_1, \ldots, f_p$, as well as to estimate them nonparametrically. For any candidate model, denote the number of significant $f_j$'s as $q$, and the number of basis functions used for each $f_j$ as $d_n$. Using similar steps as in Section 3.3.1, it can be shown that an MDL criterion for fitting (3.2) is:

$$\text{MDL}_{\text{additive}}(S) \quad = \quad \frac{n}{2} \log\left(\frac{\text{RSS}}{n}\right) + \frac{q d_n}{2} \log(n) + q \log(p). \tag{3.9}$$

One can also perform robust fitting as in Section 3.5 above, and the resulting MDL criterion is

$$\text{MDL}^{\text{robust}}_{\text{additive}}(S) \quad = \quad n \log\left(\frac{\text{SAE}}{n}\right) + \frac{q d_n}{2} \log(n) + q \log(p). \tag{3.10}$$

**3.6.3. Practical Minimization.** MDL criteria (3.9) and (3.10) can be minimized in a similar manner as in Section 3.3.2. In Stage 1 we screen out most of the non-significant function $f_j$'s. However, instead of using SIS which was designed for linear regression problem, we use the nonparametric independence screening (NIS) procedure of Fan *et al.* (2011).

In Stage 2 we apply the group lasso of Yuan and Lin (2006) to obtain a nested sequence of models. The reason the original lasso is not applicable here is that, all the coefficients $\alpha_{jk}$'s belonging to the same function $f_j$ should either be kept or removed together (see (3.8)), and the

original lasso will not guarantee this. On the other hand, the group lasso was designed for this purpose.

In the last stage we first re-fit all the nested models obtained from Stage 2 using maximum likelihood, and then calculate their corresponding MDL values using (3.9) or (3.10). The model that gives the smallest MDL value is taken as the best fitting model.

## 3.7. Empirical Properties

This section investigates the empirical properties of the proposed work via numerical experiments and a real data example.

**3.7.1. Simulation: Linear Regression.** Following the settings in Fan *et al.* (2012), the data were generated with the model

$$y_i = b(x_{i1} + \ldots + x_{id}) + \varepsilon_i$$

for $i = 1, \ldots, n$, where the coefficient $b$ controls the signal-to-noise ratio. The $\boldsymbol{x}_i$'s are standard normal variables with the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ set to be $\rho^{|i-j|}$ with $\rho = 0.5$. The number of observations was $n$, and the number of predictors was $p$, where only the first $d$ are significant. Five combinations of $(n, p, d)$ were used: $(100, 1000, 3)$, $(200, 3000, 5)$, $(300, 10000, 8)$, $(200, 10000, 5)$, and $(300, 20000, 8)$. For each of these four combinations, 3 values of $b$ were used: $b = 2/\sqrt{d}, 3/\sqrt{d}$ and $5/\sqrt{d}$. For the error term $\boldsymbol{\varepsilon}$, 4 distributions were used: $N(0, 1)$, Laplace$(0, 1)$, $t_3$ and a Gaussian mixture with two components: $95\%N(0, 1)$ and $5\%N(0, 7^2)$. The last one represents the situation where roughly 5% of the observations are outliers. Therefore, a total of $5 \times 3 \times 4 = 60$ experimental configurations were considered. The number of repetitions for each experimental configuration was 500.

For each generated data set, six methods were applied to select a best fitting model:

(1) MDL: the MDL method proposed in Section 3.3,

(2) RobustMDL: the robust version proposed in Section 3.5,

(3) RLARS: the robust LARS method of Khan *et al.* (2007),

(4) LAD-LASSO: the least absolute deviation lasso of Wang *et al.* (2007)

(5) SparseLTS: the sparse least trimmed squares method of Alfons *et al.* (2013), and

(6) WELSH: the adaptive welsh estimators of Amato *et al.* (2021). Since this method is quite computationally expensive, we only applied it in the cases with $n = 100$ and $p = 1000$.

To evaluate the performances of different methods on variable selection, we calculated the Precision of selection and the Recall of selection, defined respectively as

$$\text{Precision} = \frac{\#\text{ of } \{i : \beta_i \neq 0 \ \&\ \hat{\beta}_i \neq 0\}}{\#\text{ of } \{i : \hat{\beta}_i \neq 0\}} \quad \text{and} \quad \text{Recall} = \frac{\#\text{ of } \{i : \beta_i \neq 0 \ \&\ \hat{\beta}_i \neq 0\}}{\#\text{ of } \{i : \beta_i \neq 0\}}.$$

We also calculated F1 score defined as

$$\text{F1 score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Note that Precision can be interpreted as fraction of true significant variables among selected ones while Recall as the ability of detecting true significant variables. F1 score is a measure that combines precision and recall, to be more specific, it is the harmonic mean of precision and recall. We also calculated mean squared error (MSE) to measure the accuracy of the estimators.

The Precision, Recall, F1 score and MSE values for the 60 different experimental configurations obtained by the five methods are summarized in Tables 3.1 to 3.20. When considering computational speeds and performances, it seems that RobustMDL is the preferred method.

**3.7.2. Simulation: Nonparametric Additive Models.** For nonparametric additive models defined as (3.2), we set $n = 400$ and $p = 1000$. Only the first four $f_j$'s are significant:

$$\begin{aligned}
f_1(x) &= 5x, \\
f_2(x) &= 3(2x-1)^2, \\
f_3(x) &= 4\sin(2\pi x)/\{2 - \sin(2\pi x)\}, \\
f_4(x) &= 6\{0.1\sin(2\pi x) + 0.2\cos(2\pi x) + 0.3\sin^2(2\pi x) + 0.4\cos^3(2\pi x) + 0.5\sin^3(2\pi x)\}, \\
f_j(x) &= 0 \quad \text{for} \quad 5 \leq j \leq p.
\end{aligned}$$

The errors $\varepsilon_i$ were generated from four distributions: $N(0,1)$, Laplace$(0,1)$, $t_5$ and a Gaussian mixture with $95\%N(0,1)$ and $5\%N(0,5^2)$. For each $i$, the $x_{ij}$'s were generated from

$$x_{ij} = \begin{cases} (\omega_{ij} + tu_i)/(1+t) & \text{for } j = 1, \ldots, 4 \\ (\omega_{ij} + tk_i)/(1+t) & \text{for } j = 5, \ldots, p, \end{cases}$$

where $\omega_{i1}, \ldots, \omega_{ip}, u_i, k_i$ were i.i.d Uniform(0,1). The parameter $t$ controls the correlation among the predictors, and we used $t = (0, 1)$ in our simulation. Also, we used the cubic B-spline with six evenly spaced knots for all the function $f_j$'s; that is, we used $d_n = 9$ basis functions to approximate each $f_j$. Therefore, in total there were 8 experimental configurations, and the number of replications in each configuration was 500. The results are summarized in Table 3.21.

From the simulation results, RobustMDL gave better performances in terms of Recall, F1 score and MSE, and provided similar results in Precision as with MDL.

**3.7.3. Real Data Example.** This subsection presents a real data analysis on a riboflavin (vitamin $B_2$) production data set which is available in Supplementary Section A.1 of Bühlmann (2013). The response variable is the logarithm of the riboflavin production rate in Bacillus subtilis for $n = 71$ samples while there are $p = 4,088$ covariates measuring the logarithm of the expression level of the $p = 4,088$ genes. Linear models were used in Bühlmann (2013) and Javanmard and Montanari (2014) to detect significant genes that potentially affect riboflavin production. The gene YXLD-at was located by Bühlmann (2013) while the two genes YXLD-at and YXLE-at were identified by Javanmard and Montanari (2014) as significant. Here in addition to fitting a linear model, we also fit a nonparametric additive model to select the significant genes.

Following Bühlmann (2013), we first adopted a screening procedure and only used the 100 genes with the largest empirical variances. We then applied the proposed RobustMDL method to fit a linear model to the screened data set. The resulting model identified five genes as significant: YCKE-at, YXLD-at, YDAR-at, XHLA-at and YOAB-at.

We also fitted a nonparametric additive model using RobustMDL to the screened data set, which suggested that YXLD-at and PHRI-r-at are significant. In other words both the fitted RobustMDL linear model and nonparametric additive model were capable of detecting YXLD-at, which was considered significant in most previous analyses on this data set.

## 3.8. Conclusion

The MDL principle has long been adopted by researchers in different fields to perform various estimation tasks. In this chapter we extended its use to some "large $p$ small $n$" problems, including high-dimensional linear regression, nonparametric additive models, as well as their robust counterparts. As can be seen from above, one attractiveness of the MDL principle is that it can be applied to handle such problems in a natural manner; that is, by incorporating the code length of the additional parameters that are needed to specify the models.

The work presented above focused on the so-called two-part code version of MDL. It will be interesting to develop similar methods using the newer versions of MDL, such as normalized maximum likelihood (Grünwald, 2007; Rissanen, 2007), which might lead to further improved performances.

## 3.9. Proof

This appendix provides technical details, including the proof for Theorem 3.4.1.

### 3.9.1. Lemmas.

LEMMA 1. *Let $\chi^2_j$ denote a $\chi^2$ random variable with degrees of freedom $j$. If $c \to \infty$ and $\frac{J}{c} \to 0$, then*

$$P(\chi^2_j > c) = \frac{1}{\Gamma(j/2)}(c/2)^{j/2-1}\exp(-c/2)(1+o(1))$$

*uniformly for all $j \leq J$.*

The proof can be found in Luo and Chen (2013) by using integration by parts.

LEMMA 2. *Let $\chi^2_j$ be a chi-square random variable with degrees of freedom $j$ and $c_j = 2j[\log p + \log(j \log p)]$. If $p \to \infty$, then for any $J \leq p$,*

$$\sum_{j=1}^{J} \binom{p}{j} P(\chi^2_j > c_j) \to 0.$$

The proof is similar to the one for Lemma 1 and hence is omitted.

### 3.9.2. Proof of the Theorem 3.4.1.

PROOF. Since

$$\mathrm{MDL}(S) = \frac{n}{2}\log\left(\frac{\mathrm{RSS}}{n}\right) + \frac{|S|}{2}\log(n) + |S|\log(p),$$

we have

$$\mathrm{MDL}(S) - \mathrm{MDL}(S_0) = T_1 + T_2,$$

where

$$T_1 = \frac{n}{2}\log\left(\frac{\mathrm{RSS}_S}{\mathrm{RSS}_{S_0}}\right)$$

and

$$T_2 = (|S| - |S_0|)\log(p\sqrt{n}).$$

Without loss of generality, we assume that $\sigma^2 = 1$.

**Case 1:** $S_0 \not\subset S$**.**

Denote $\mathcal{S}$ as the collection of models which hold asymptotic identifiability condition 1, that is, $\mathcal{S} = \{S : |S| \leq k|S_0|\}$ for some fixed $k > 1$. In practice we only consider models with size comparable with the true model, so the restriction $|S| \leq k|S_0|$ is imposed. Let $\mathcal{S}_j = \{S : |S| = j, S \in \mathcal{S}\}$. Recall that $\boldsymbol{P}_S$ is the projection matrix for model $S$ and $\boldsymbol{P}_{S_0}$ is the projection matrix for the true model $S_0$. Note that

$$\begin{aligned}
\mathrm{RSS}_{S_0} &= (\boldsymbol{y} - \boldsymbol{X}_{S_0}\boldsymbol{\beta}_{S_0})^T(\boldsymbol{I} - \boldsymbol{P}_{S_0})(\boldsymbol{y} - \boldsymbol{X}_{S_0}\boldsymbol{\beta}_{S_0}) \\
&= \boldsymbol{\varepsilon}^T(\boldsymbol{I} - \boldsymbol{P}_{S_0})\boldsymbol{\varepsilon} = \sum_{i=1}^{n-|S_0|} Z_i^2 \\
&= (n - |S_0|)(1 + o_p(1)) = n(1 + o_p(1)),
\end{aligned}$$

where $Z_i$'s are i.i.d. standard normal variables.

Recall $\delta(S) = \|\boldsymbol{\mu} - \boldsymbol{P}_S\boldsymbol{\mu}\|^2$ with $\boldsymbol{\mu} = \boldsymbol{X}_{S_0}\boldsymbol{\beta}_{S_0}$. Then

$$\begin{aligned} \mathrm{RSS}_S - \mathrm{RSS}_{S_0} &= (\boldsymbol{\mu}+\boldsymbol{\varepsilon})^T(\boldsymbol{I}-\boldsymbol{P}_S)(\boldsymbol{\mu}+\boldsymbol{\varepsilon}) \\ &\quad -\boldsymbol{\varepsilon}^T(\boldsymbol{I}-\boldsymbol{P}_{S_0})\boldsymbol{\varepsilon} \\ &= \delta(S) + 2\boldsymbol{\mu}^T(\boldsymbol{I}-\boldsymbol{P}_S)\boldsymbol{\varepsilon} \\ &\quad -\boldsymbol{\varepsilon}^T\boldsymbol{P}_S\boldsymbol{\varepsilon} + \boldsymbol{\varepsilon}^T\boldsymbol{P}_{S_0}\boldsymbol{\varepsilon}, \end{aligned} \tag{3.11}$$

and $\boldsymbol{\varepsilon}^T\boldsymbol{P}_{S_0}\boldsymbol{\varepsilon} = |S_0|(1+o_p(1))$.

Write the second term in (3.11) as

$$\boldsymbol{\mu}^T(\boldsymbol{I}-\boldsymbol{P}_S)\boldsymbol{\varepsilon} = \sqrt{\delta(S)}Z_S,$$

where $Z_S \sim N(0,1)$. Then for any $S \in \mathcal{S}$,

$$\boldsymbol{\mu}^T(\boldsymbol{I}-\boldsymbol{P}_S)\boldsymbol{\varepsilon} \le \sqrt{\delta(S)}\max_{S\in\mathcal{S}}|Z_S|.$$

Let $c_j = 2j\{\log p + \log(j\log p)\}$ and $c = \max\{c_j : 1 \le j \le k|S_0|\}$, according to Lemma 2 we hence have

$$\begin{aligned} P(\max_{S\in\mathcal{S}}|Z_S| \ge \sqrt{c}) &= P(\max_{S\in\mathcal{S}_j, 1\le j\le k|S_0|}|Z_S| \ge \sqrt{c}) \\ &\le \sum_{j=1}^{k|S_0|}\binom{p}{j}P(\chi_1^2 \ge c) \\ &\le \sum_{j=1}^{k|S_0|}\binom{p}{j}P(\chi_j^2 \ge c) \\ &\le \sum_{j=1}^{k|S_0|}\binom{p}{j}P(\chi_j^2 \ge c_j) \to 0. \end{aligned}$$

By the identifiability condition 1, $\log n = o_p(\delta(S))$, we have $|\boldsymbol{\mu}^T(\boldsymbol{I}-\boldsymbol{P}_S)\boldsymbol{\varepsilon}| = \sqrt{\delta(S)O_p(k|S_0|\log p)} = o_p(\delta(S))$ uniformly over $\mathcal{S}$.

Similarly, for the third term in (3.11), as $\boldsymbol{\varepsilon}^T \boldsymbol{P}_S \boldsymbol{\varepsilon} = \chi^2_{|S|}$ we have

$$P(\max_{S \in \mathcal{S}} \boldsymbol{\varepsilon}^T \boldsymbol{P}_S \boldsymbol{\varepsilon} \geq c) = P(\max_{S \in \mathcal{S}_j, 1 \leq j \leq k|S_0|} \chi^2_j \geq c)$$
$$\leq \sum_{j=1}^{k|S_0|} \binom{p}{j} P(\chi^2_j \geq c_j) \to 0.$$

Thus we have

$$\max_{\mathcal{S}} \{\boldsymbol{\varepsilon}^T \boldsymbol{P}_S \boldsymbol{\varepsilon}\} = O_p(k|S_0| \log p) = o_p(\delta(S)).$$

In a word, $\delta(S)$ is the dominant term in (3.11). Therefore,

$$\mathrm{RSS}_S - \mathrm{RSS}_{S_0} = \delta(S)(1 + o_p(1))$$

and

$$\begin{aligned} T_1 &= \frac{n}{2} \log \left( 1 + \frac{\mathrm{RSS}_S - \mathrm{RSS}_{S_0}}{\mathrm{RSS}_{S_0}} \right) \\ &= \frac{n}{2} \log \left( 1 + \frac{\delta(S)(1 + o_p(1))}{n} \right) \\ &= \frac{\delta(S)(1 + o_p(1))}{2}. \end{aligned} \tag{3.12}$$

We also have

$$\begin{aligned} T_2 &= (|S| - |S_0|) \log(p\sqrt{n}) = \frac{|S| - |S_0|}{2} \log(p^2 n) \\ &\geq -\frac{|S_0|}{2} \log(p^2 n) \end{aligned} \tag{3.13}$$

By (3.12) and (3.13),

$$\begin{aligned} T_1 + T_2 &\geq \frac{\delta(S)(1 + o_p(1))}{2} - \frac{|S_0|}{2} \log(p^2 n) \\ &= \frac{\log n}{2} \left\{ \frac{\delta(S)(1 + o_p(1))}{\log n} - \frac{\log(p^2 n)}{\log n} \right\} \end{aligned}$$

Since $p = O(n^\gamma)$ as $n \to \infty$ for some fixed $\gamma$, which means

$$-\frac{\log(p^2 n)}{\log n} = O(1).$$

Thus, we have

$$\min_{S_0 \not\subset S} \mathrm{MDL}(S) - \mathrm{MDL}(S_0) \to \infty.$$

**Case 2:** $S_0 \subset S$**.**

Let $\mathcal{S}^* = \{S : S \in \mathcal{S}, S_0 \subset S, S \neq S_0\}$ and $\mathcal{S}_j^* = \{S : |S| = j, S \in \mathcal{S}^*\}$.

When $S_0 \subset S$, we have $(\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{\mu} = (\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{X}_{S_0}\boldsymbol{\beta}_{S_0} = 0$. Therefore, $\boldsymbol{y}^T(\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{y} = \boldsymbol{\varepsilon}^T(\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{\varepsilon}$. Also

$$\begin{aligned}
\mathrm{RSS}_{S_0} - \mathrm{RSS}_S &= \boldsymbol{\varepsilon}^T(\boldsymbol{I} - \boldsymbol{P}_{S_0})\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^T(\boldsymbol{I} - \boldsymbol{P}_S)\boldsymbol{\varepsilon} \\
&= \boldsymbol{\varepsilon}^T(\boldsymbol{P}_S - \boldsymbol{P}_{S_0})\boldsymbol{\varepsilon} \\
&= \chi^2_{|S|-|S_0|}(S),
\end{aligned}$$

where $\chi^2_{|S|-|S_0|}(S)$ follows chi-square distribution with degrees of freedom $|S| - |S_0|$.

Let $c_j = 2j\{\log p + \log(j \log p)\}$. According to Lemma 2, when $1 \leq j \leq k|S_0| - |S_0|$, we have

$$\begin{aligned}
P(\max_{S \in \mathcal{S}_j^*} \chi_j^2(S) \geq c_j) &\leq \sum_{i=1}^{j} P(\max_{S \in \mathcal{S}_i^*} \chi_i^2(S) \geq c_j) \\
&\leq \sum_{i=1}^{j} \binom{p - |S_0|}{i} P(\chi_i^2(S) \geq c_j) \\
&\leq \sum_{i=1}^{j} \binom{p}{i} P(\chi_i^2(S) \geq c_i) \to 0.
\end{aligned}$$

Therefore, $\chi^2_{|S|-|S_0|}(S) \leq c_{|S|-|S_0|}(1 + o_p(1))$ and

$$\begin{aligned}
T_1 &= \frac{n}{2} \log\left(\frac{\mathrm{RSS}_S}{\mathrm{RSS}_{S_0}}\right) \\
&= -\frac{n}{2} \log\left\{1 + \frac{\chi^2_{|S|-|S_0|}(S)}{\mathrm{RSS}_{S_0} - \chi^2_{|S|-|S_0|}(S)}\right\} \\
&\geq -\frac{n}{2}\left\{\frac{\chi^2_{|S|-|S_0|}(S)}{\mathrm{RSS}_{S_0} - \chi^2_{|S|-|S_0|}(S)}\right\}.
\end{aligned}$$

Since $n^{-1}\mathrm{RSS}_{S_0} \to \sigma^2 = 1$ as $n \to \infty$, we have $\mathrm{RSS}_{S_0} = n(1 + o(1))$. Note that $c_{|S|-|S_0|} = o(n)$,

$$\begin{aligned} T_1 &\geq -\frac{c_{|S|-|S_0|}}{2}(1 + o_p(1)) \\ &= -(|S| - |S_0|)\log p(1 + o_p(1)) \end{aligned} \tag{3.14}$$

uniformly over $\mathcal{S}^*$, and

$$T_2 = \frac{|S| - |S_0|}{2}\log(p^2 n). \tag{3.15}$$

For case 2, by (3.14) and (3.15),

$$\begin{aligned} T_1 + T_2 &\geq \frac{(|S| - |S_0|)\log n}{2}\left\{\frac{\log(p^2 n)}{\log n}\right. \\ &\quad \left. -\frac{2\log p}{\log n}(1 + o_p(1))\right\} \\ &= \frac{(|S| - |S_0|)\log n}{2} \to \infty. \end{aligned}$$

Therefore, we have

$$\min_{S_0 \subset S} \mathrm{MDL}(S) - \mathrm{MDL}(S_0) \to \infty.$$

Combing case 1 and case 2, it completes the proof for Theorem 3.4.1.

□

## 3.10. Tables

TABLE 3.1. The Precision, Recall, F1 score and MSE values for the methods compared in Section 3.7.1 for those experimental settings with $(n, p) = (100,1000)$ and $\varepsilon_i \sim N(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(3, \frac{2}{\sqrt{3}})$ | MDL | 99.65 | 100.00 | 99.80 | 0.04 | 0.05 |
| | RobustMDL | 97.89 | 100.00 | 98.79 | 0.05 | 0.04 |
| | RLARS | 50.77 | 99.93 | 64.87 | 0.25 | 0.14 |
| | LAD-lasso | 84.61 | 100.00 | 90.42 | 0.35 | 1.06 |
| | SparseLTS | 43.69 | 100.00 | 54.88 | 0.40 | 2.93 |
| | Welsh | 94.04 | 70.13 | 94.85 | 2.15 | 780.46 |
| $(3, \frac{3}{\sqrt{3}})$ | MDL | 99.45 | 100.00 | 99.69 | 0.04 | 0.05 |
| | RobustMDL | 97.89 | 100.00 | 98.79 | 0.05 | 0.04 |
| | RLARS | 56.73 | 100.00 | 69.50 | 0.20 | 0.15 |
| | LAD-lasso | 83.75 | 100.00 | 90.09 | 0.31 | 1.14 |
| | SparseLTS | 63.58 | 100.00 | 73.13 | 0.37 | 2.60 |
| | Welsh | 96.41 | 80.53 | 96.82 | 3.22 | 792.34 |
| $(3, \frac{5}{\sqrt{3}})$ | MDL | 99.65 | 100.00 | 99.80 | 0.04 | 0.05 |
| | RobustMDL | 98.90 | 100.00 | 99.37 | 0.04 | 0.04 |
| | RLARS | 66.96 | 100.00 | 77.35 | 0.14 | 0.15 |
| | LAD-lasso | 85.12 | 100.00 | 91.00 | 0.31 | 1.26 |
| | SparseLTS | 64.11 | 100.00 | 76.17 | 0.34 | 2.37 |
| | Welsh | 93.20 | 82.93 | 94.48 | 7.73 | 786.17 |

TABLE 3.2. Similar to Table 3.1 but for settings with $(n, p) = (100, 1000)$ and $\varepsilon_i \sim \text{Laplace}(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(3, \frac{2}{\sqrt{3}})$ | MDL | 98.80 | 98.87 | 98.57 | 0.12 | 0.05 |
| | RobustMDL | 98.97 | 100.00 | 99.39 | 0.09 | 0.04 |
| | RLARS | 71.85 | 100.00 | 80.62 | 0.24 | 0.14 |
| | LAD-lasso | 89.96 | 100.00 | 93.94 | 0.45 | 1.10 |
| | SparseLTS | 23.88 | 100.00 | 35.05 | 0.45 | 3.28 |
| | Welsh | 96.39 | 78.87 | 96.50 | 1.56 | 1045.80 |
| $(3, \frac{3}{\sqrt{3}})$ | MDL | 99.50 | 100.00 | 99.71 | 0.09 | 0.05 |
| | RobustMDL | 99.52 | 100.00 | 99.72 | 0.09 | 0.04 |
| | RLARS | 76.42 | 100.00 | 83.87 | 0.20 | 0.14 |
| | LAD-lasso | 89.62 | 100.00 | 93.77 | 0.45 | 1.15 |
| | SparseLTS | 40.20 | 100.00 | 52.84 | 0.45 | 2.74 |
| | Welsh | 98.49 | 87.00 | 98.17 | 2.12 | 1034.46 |
| $(3, \frac{5}{\sqrt{3}})$ | MDL | 99.80 | 100.00 | 99.89 | 0.09 | 0.05 |
| | RobustMDL | 99.70 | 100.00 | 99.83 | 0.09 | 0.04 |
| | RLARS | 81.20 | 100.00 | 87.38 | 0.16 | 0.14 |
| | LAD-lasso | 89.53 | 100.00 | 93.73 | 0.43 | 1.26 |
| | SparseLTS | 49.66 | 100.00 | 63.87 | 0.49 | 2.40 |
| | Welsh | 95.78 | 87.80 | 96.02 | 5.26 | 1031.58 |

TABLE 3.3. Similar to Table 3.1 but for settings with $(n, p) = (100, 1000)$ and $\varepsilon_i \sim t_3$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(3, \frac{2}{\sqrt{3}})$ | MDL | 98.38 | 92.47 | 94.12 | 0.35 | 0.05 |
| | RobustMDL | 98.64 | 99.33 | 99.05 | 0.17 | 0.04 |
| | RLARS | 65.75 | 99.60 | 76.19 | 0.32 | 0.15 |
| | LAD-lasso | 91.51 | 98.80 | 94.56 | 0.63 | 1.18 |
| | SparseLTS | 24.51 | 99.73 | 35.33 | 0.55 | 3.56 |
| | Welsh | 93.66 | 76.73 | 95.05 | 1.70 | 1206.73 |
| $(3, \frac{3}{\sqrt{3}})$ | MDL | 99.00 | 100.00 | 99.43 | 0.13 | 0.05 |
| | RobustMDL | 99.50 | 100.00 | 99.71 | 0.11 | 0.04 |
| | RLARS | 70.36 | 100.00 | 79.62 | 0.24 | 0.14 |
| | LAD-lasso | 91.73 | 100.00 | 95.09 | 0.49 | 1.22 |
| | SparseLTS | 42.36 | 100.00 | 54.53 | 0.44 | 2.68 |
| | Welsh | 97.41 | 90.00 | 97.22 | 1.45 | 1047.04 |
| $(3, \frac{5}{\sqrt{3}})$ | MDL | 99.30 | 99.87 | 99.50 | 0.17 | 0.05 |
| | RobustMDL | 99.70 | 100.00 | 99.83 | 0.13 | 0.04 |
| | RLARS | 78.19 | 100.00 | 85.31 | 0.18 | 0.14 |
| | LAD-lasso | 91.23 | 100.00 | 94.76 | 0.53 | 1.27 |
| | SparseLTS | 47.86 | 100.00 | 62.20 | 0.48 | 2.46 |
| | Welsh | 96.33 | 87.07 | 96.82 | 5.61 | 1123.60 |

TABLE 3.4. Similar to Table 3.1 but for settings with $(n, p) = (100, 1000)$ and with outliers $95\%N(0, 1)$ & $5\%N(0, 7^2)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(3, \frac{2}{\sqrt{3}})$ | MDL | 99.19 | 86.93 | 90.33 | 0.49 | 0.05 |
| | RobustMDL | 99.27 | 99.93 | 99.54 | 0.15 | 0.04 |
| | RLARS | 60.80 | 99.93 | 72.57 | 0.21 | 0.15 |
| | LAD-lasso | 94.05 | 99.80 | 96.38 | 0.49 | 1.11 |
| | SparseLTS | 39.49 | 100.00 | 50.88 | 0.39 | 3.37 |
| | Welsh | 95.57 | 83.67 | 96.37 | 1.23 | 1165.00 |
| $(3, \frac{3}{\sqrt{3}})$ | MDL | 99.39 | 99.53 | 99.32 | 0.17 | 0.04 |
| | RobustMDL | 99.52 | 100.00 | 99.72 | 0.13 | 0.04 |
| | RLARS | 68.97 | 100.00 | 78.78 | 0.16 | 0.14 |
| | LAD-lasso | 94.77 | 100.00 | 96.87 | 0.47 | 1.08 |
| | SparseLTS | 53.06 | 100.00 | 64.84 | 0.35 | 2.63 |
| | Welsh | 96.72 | 85.13 | 97.15 | 2.42 | 1069.91 |
| $(3, \frac{5}{\sqrt{3}})$ | MDL | 99.65 | 100.00 | 99.80 | 0.14 | 0.05 |
| | RobustMDL | 99.80 | 100.00 | 99.89 | 0.14 | 0.04 |
| | RLARS | 74.31 | 100.00 | 82.54 | 0.13 | 0.14 |
| | LAD-lasso | 93.53 | 100.00 | 96.16 | 0.45 | 1.20 |
| | SparseLTS | 57.84 | 100.00 | 71.23 | 0.33 | 2.41 |
| | Welsh | 94.06 | 85.93 | 95.12 | 6.29 | 1139.08 |

TABLE 3.5. Similar to Table 3.1 but for settings with $(n, p) = (200, 3000)$ and $\varepsilon_i \sim N(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.54 | 100.00 | 99.75 | 0.03 | 0.11 |
| | RobustMDL | 98.10 | 100.00 | 98.94 | 0.04 | 0.10 |
| | RLARS | 67.02 | 100.00 | 79.16 | 0.14 | 0.28 |
| | LAD-lasso | 88.45 | 100.00 | 93.29 | 0.27 | 4.79 |
| | SparseLTS | 67.63 | 100.00 | 78.56 | 0.28 | 18.86 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.80 | 100.00 | 99.89 | 0.03 | 0.11 |
| | RobustMDL | 99.02 | 100.00 | 99.46 | 0.03 | 0.10 |
| | RLARS | 71.89 | 100.00 | 82.46 | 0.11 | 0.28 |
| | LAD-lasso | 87.33 | 100.00 | 92.66 | 0.26 | 5.03 |
| | SparseLTS | 57.41 | 100.00 | 69.63 | 0.27 | 17.79 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.93 | 100.00 | 99.96 | 0.03 | 0.11 |
| | RobustMDL | 99.63 | 100.00 | 99.80 | 0.03 | 0.11 |
| | RLARS | 79.54 | 100.00 | 87.48 | 0.07 | 0.28 |
| | LAD-lasso | 87.35 | 100.00 | 92.67 | 0.25 | 5.89 |
| | SparseLTS | 91.51 | 100.00 | 95.10 | 0.35 | 17.57 |

TABLE 3.6. Similar to Table 3.1 but for settings with $(n, p) = (200, 3000)$ and $\varepsilon_i \sim \text{Laplace}(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.50 | 99.60 | 99.48 | 0.07 | 0.11 |
| | RobustMDL | 99.56 | 100.00 | 99.75 | 0.06 | 0.10 |
| | RLARS | 85.17 | 100.00 | 90.89 | 0.12 | 0.27 |
| | LAD-lasso | 92.43 | 100.00 | 95.70 | 0.35 | 5.08 |
| | SparseLTS | 46.54 | 100.00 | 58.94 | 0.33 | 19.91 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.80 | 100.00 | 99.89 | 0.06 | 0.11 |
| | RobustMDL | 99.80 | 100.00 | 99.89 | 0.06 | 0.10 |
| | RLARS | 88.49 | 100.00 | 93.01 | 0.10 | 0.27 |
| | LAD-lasso | 92.42 | 100.00 | 95.71 | 0.34 | 5.30 |
| | SparseLTS | 50.57 | 100.00 | 62.80 | 0.34 | 18.44 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.93 | 100.00 | 99.96 | 0.06 | 0.11 |
| | RobustMDL | 99.93 | 100.00 | 99.96 | 0.06 | 0.10 |
| | RLARS | 91.47 | 100.00 | 94.91 | 0.08 | 0.27 |
| | LAD-lasso | 92.24 | 100.00 | 95.61 | 0.31 | 6.14 |
| | SparseLTS | 74.34 | 100.00 | 83.83 | 0.41 | 17.62 |

TABLE 3.7. Similar to Table 3.1 but for settings with $(n, p) = (200, 3000)$ and $\varepsilon_i \sim t_3$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.24 | 96.60 | 97.43 | 0.19 | 0.11 |
| | RobustMDL | 99.24 | 99.96 | 99.56 | 0.10 | 0.10 |
| | RLARS | 80.07 | 100.00 | 87.60 | 0.15 | 0.27 |
| | LAD-lasso | 93.60 | 100.00 | 96.40 | 0.44 | 5.30 |
| | SparseLTS | 47.93 | 100.00 | 60.36 | 0.36 | 20.06 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.21 | 99.92 | 99.52 | 0.11 | 0.11 |
| | RobustMDL | 99.50 | 100.00 | 99.73 | 0.09 | 0.11 |
| | RLARS | 85.40 | 100.00 | 91.12 | 0.11 | 0.28 |
| | LAD-lasso | 93.34 | 100.00 | 96.24 | 0.42 | 5.68 |
| | SparseLTS | 51.73 | 100.00 | 63.45 | 0.36 | 18.82 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.57 | 100.00 | 99.76 | 0.10 | 0.11 |
| | RobustMDL | 99.67 | 100.00 | 99.82 | 0.09 | 0.10 |
| | RLARS | 89.81 | 100.00 | 93.91 | 0.08 | 0.27 |
| | LAD-lasso | 93.58 | 100.00 | 96.38 | 0.40 | 6.27 |
| | SparseLTS | 71.58 | 100.00 | 82.09 | 0.41 | 17.83 |

TABLE 3.8. Similar to Table 3.1 but for settings with $(n, p) = (200, 3000)$ and with outliers $95\%N(0, 1)$ & $5\%N(0, 7^2)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.49 | 94.08 | 96.03 | 0.27 | 0.11 |
| | RobustMDL | 99.64 | 100.00 | 99.80 | 0.11 | 0.11 |
| | RLARS | 76.36 | 100.00 | 85.33 | 0.10 | 0.29 |
| | LAD-lasso | 96.55 | 100.00 | 98.08 | 0.37 | 5.17 |
| | SparseLTS | 63.16 | 100.00 | 74.84 | 0.27 | 20.22 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.68 | 99.96 | 99.80 | 0.12 | 0.11 |
| | RobustMDL | 99.87 | 100.00 | 99.93 | 0.11 | 0.10 |
| | RLARS | 81.91 | 100.00 | 88.85 | 0.08 | 0.27 |
| | LAD-lasso | 96.76 | 100.00 | 98.19 | 0.36 | 5.12 |
| | SparseLTS | 54.24 | 100.00 | 66.69 | 0.26 | 18.36 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.80 | 100.00 | 99.89 | 0.11 | 0.11 |
| | RobustMDL | 99.97 | 100.00 | 99.98 | 0.11 | 0.11 |
| | RLARS | 85.81 | 100.00 | 91.30 | 0.06 | 0.28 |
| | LAD-lasso | 96.71 | 100.00 | 98.17 | 0.34 | 5.95 |
| | SparseLTS | 87.80 | 100.00 | 92.88 | 0.34 | 17.86 |

TABLE 3.9. Similar to Table 3.1 but for settings with $(n, p) = (300, 10000)$ and $\varepsilon_i \sim N(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.84 | 100.00 | 99.92 | 0.03 | 0.29 |
| | RobustMDL | 99.12 | 100.00 | 99.53 | 0.03 | 0.29 |
| | RLARS | 88.64 | 100.00 | 93.77 | 0.07 | 0.54 |
| | LAD-lasso | 87.32 | 100.00 | 92.81 | 0.30 | 15.16 |
| | SparseLTS | 73.89 | 100.00 | 83.73 | 0.31 | 64.98 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.93 | 100.00 | 99.96 | 0.03 | 0.29 |
| | RobustMDL | 99.58 | 100.00 | 99.78 | 0.03 | 0.29 |
| | RLARS | 91.24 | 100.00 | 95.22 | 0.05 | 0.53 |
| | LAD-lasso | 87.39 | 100.00 | 92.84 | 0.29 | 15.84 |
| | SparseLTS | 53.18 | 100.00 | 67.55 | 0.27 | 62.71 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.98 | 100.00 | 99.99 | 0.03 | 0.32 |
| | RobustMDL | 99.78 | 100.00 | 99.88 | 0.03 | 0.32 |
| | RLARS | 93.65 | 100.00 | 96.54 | 0.04 | 0.58 |
| | LAD-lasso | 87.78 | 100.00 | 93.10 | 0.27 | 19.25 |
| | SparseLTS | 96.09 | 100.00 | 97.87 | 0.41 | 63.62 |

TABLE 3.10. Similar to Table 3.1 but for settings with $(n, p) = (300, 10000)$ and $\varepsilon_i \sim \text{Laplace}(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.57 | 99.48 | 99.48 | 0.08 | 0.29 |
| | RobustMDL | 99.87 | 99.95 | 99.90 | 0.06 | 0.29 |
| | RLARS | 94.04 | 100.00 | 96.76 | 0.07 | 0.52 |
| | LAD-lasso | 91.72 | 100.00 | 95.43 | 0.40 | 16.27 |
| | SparseLTS | 48.96 | 100.00 | 62.16 | 0.36 | 67.98 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.80 | 100.00 | 99.89 | 0.06 | 0.30 |
| | RobustMDL | 99.96 | 100.00 | 99.98 | 0.06 | 0.30 |
| | RLARS | 95.32 | 100.00 | 97.46 | 0.06 | 0.54 |
| | LAD-lasso | 91.29 | 100.00 | 95.16 | 0.38 | 17.45 |
| | SparseLTS | 46.84 | 100.00 | 60.60 | 0.35 | 64.56 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.96 | 100.00 | 99.98 | 0.06 | 0.30 |
| | RobustMDL | 99.98 | 100.00 | 99.99 | 0.06 | 0.30 |
| | RLARS | 96.40 | 100.00 | 98.05 | 0.06 | 0.53 |
| | LAD-lasso | 91.58 | 100.00 | 95.36 | 0.35 | 19.64 |
| | SparseLTS | 79.90 | 100.00 | 88.13 | 0.49 | 62.71 |

TABLE 3.11. Similar to Table 3.1 but for settings with $(n, p) = (300, 10000)$ and $\varepsilon_i \sim t_3$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.74 | 94.80 | 96.63 | 0.23 | 0.30 |
| | RobustMDL | 99.76 | 99.80 | 99.75 | 0.10 | 0.30 |
| | RLARS | 92.61 | 99.88 | 95.92 | 0.09 | 0.54 |
| | LAD-lasso | 93.76 | 99.83 | 96.42 | 0.50 | 16.98 |
| | SparseLTS | 51.29 | 99.90 | 64.85 | 0.38 | 69.30 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.76 | 99.55 | 99.52 | 0.14 | 0.31 |
| | RobustMDL | 99.93 | 99.90 | 99.90 | 0.10 | 0.31 |
| | RLARS | 93.92 | 99.90 | 96.64 | 0.08 | 0.54 |
| | LAD-lasso | 93.21 | 99.83 | 96.07 | 0.50 | 18.01 |
| | SparseLTS | 52.11 | 99.90 | 64.41 | 0.39 | 65.30 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.87 | 99.83 | 99.77 | 0.16 | 0.30 |
| | RobustMDL | 99.98 | 100.00 | 99.99 | 0.11 | 0.30 |
| | RLARS | 95.40 | 100.00 | 97.50 | 0.06 | 0.54 |
| | LAD-lasso | 93.10 | 99.80 | 96.20 | 0.56 | 20.15 |
| | SparseLTS | 78.65 | 100.00 | 87.27 | 0.47 | 62.98 |

TABLE 3.12. Similar to Table 3.1 but for settings with $(n, p) = (300, 10000)$ and with outliers $95\%N(0, 1)$ & $5\%N(0, 7^2)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.84 | 91.17 | 94.63 | 0.33 | 0.30 |
| | RobustMDL | 99.87 | 99.92 | 99.89 | 0.11 | 0.30 |
| | RLARS | 91.76 | 100.00 | 95.49 | 0.06 | 0.54 |
| | LAD-lasso | 95.94 | 100.00 | 97.82 | 0.43 | 15.71 |
| | SparseLTS | 67.29 | 100.00 | 78.78 | 0.29 | 68.05 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.87 | 99.90 | 99.87 | 0.11 | 0.31 |
| | RobustMDL | 99.96 | 100.00 | 99.98 | 0.11 | 0.31 |
| | RLARS | 93.39 | 100.00 | 96.38 | 0.05 | 0.55 |
| | LAD-lasso | 96.15 | 100.00 | 97.92 | 0.41 | 17.31 |
| | SparseLTS | 49.33 | 100.00 | 63.90 | 0.25 | 65.13 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.96 | 100.00 | 99.98 | 0.11 | 0.30 |
| | RobustMDL | 99.98 | 100.00 | 99.99 | 0.11 | 0.30 |
| | RLARS | 95.60 | 100.00 | 97.60 | 0.04 | 0.54 |
| | LAD-lasso | 96.28 | 100.00 | 97.99 | 0.40 | 19.18 |
| | SparseLTS | 92.39 | 100.00 | 95.78 | 0.40 | 62.94 |

TABLE 3.13. Similar to Table 3.1 but for settings with $(n, p) = (200, 10000)$ and $\varepsilon_i \sim N(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.83 | 100.00 | 99.91 | 0.03 | 0.24 |
| | RobustMDL | 98.84 | 100.00 | 99.37 | 0.04 | 0.24 |
| | RLARS | 64.89 | 100.00 | 77.68 | 0.16 | 0.41 |
| | LAD-lasso | 85.59 | 100.00 | 91.52 | 0.31 | 5.83 |
| | SparseLTS | 62.34 | 100.00 | 73.87 | 0.31 | 19.10 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.93 | 100.00 | 99.96 | 0.03 | 0.24 |
| | RobustMDL | 99.31 | 100.00 | 99.62 | 0.03 | 0.24 |
| | RLARS | 70.90 | 100.00 | 81.68 | 0.12 | 0.41 |
| | LAD-lasso | 85.74 | 100.00 | 91.69 | 0.30 | 5.85 |
| | SparseLTS | 60.21 | 100.00 | 70.79 | 0.31 | 18.00 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 100.00 | 100.00 | 100.00 | 0.03 | 0.24 |
| | RobustMDL | 99.80 | 100.00 | 99.89 | 0.03 | 0.23 |
| | RLARS | 78.89 | 100.00 | 86.97 | 0.08 | 0.40 |
| | LAD-lasso | 85.32 | 100.00 | 91.37 | 0.28 | 6.30 |
| | SparseLTS | 87.85 | 100.00 | 92.86 | 0.35 | 17.49 |

TABLE 3.14. Similar to Table 3.1 but for settings with $(n, p) = (200, 10000)$ and $\varepsilon_i \sim \text{Laplace}(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.77 | 99.52 | 99.59 | 0.07 | 0.24 |
| | RobustMDL | 99.90 | 100.00 | 99.95 | 0.06 | 0.24 |
| | RLARS | 83.32 | 100.00 | 89.68 | 0.13 | 0.40 |
| | LAD-lasso | 91.11 | 100.00 | 94.91 | 0.42 | 6.41 |
| | SparseLTS | 37.36 | 100.00 | 49.80 | 0.36 | 20.47 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.83 | 100.00 | 99.91 | 0.06 | 0.24 |
| | RobustMDL | 99.97 | 100.00 | 99.98 | 0.06 | 0.24 |
| | RLARS | 87.39 | 100.00 | 92.27 | 0.10 | 0.40 |
| | LAD-lasso | 91.15 | 100.00 | 94.93 | 0.39 | 6.25 |
| | SparseLTS | 47.60 | 100.00 | 59.84 | 0.39 | 18.60 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.97 | 100.00 | 99.98 | 0.06 | 0.23 |
| | RobustMDL | 99.97 | 100.00 | 99.98 | 0.06 | 0.23 |
| | RLARS | 90.30 | 100.00 | 94.12 | 0.08 | 0.39 |
| | LAD-lasso | 91.06 | 100.00 | 94.89 | 0.36 | 6.71 |
| | SparseLTS | 65.42 | 100.00 | 77.54 | 0.43 | 17.64 |

TABLE 3.15. Similar to Table 3.1 but for settings with $(n, p) = (200, 10000)$ and $\varepsilon_i \sim t_3$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.43 | 93.84 | 95.77 | 0.31 | 0.23 |
| | RobustMDL | 99.57 | 99.76 | 99.85 | 0.13 | 0.23 |
| | RLARS | 78.13 | 99.80 | 86.42 | 0.18 | 0.40 |
| | LAD-lasso | 92.89 | 99.64 | 95.92 | 0.55 | 6.61 |
| | SparseLTS | 37.85 | 99.80 | 49.95 | 0.40 | 20.86 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.63 | 99.04 | 99.39 | 0.21 | 0.24 |
| | RobustMDL | 99.70 | 99.80 | 99.95 | 0.15 | 0.23 |
| | RLARS | 83.57 | 99.80 | 90.00 | 0.15 | 0.40 |
| | LAD-lasso | 93.15 | 99.80 | 96.23 | 0.53 | 6.45 |
| | SparseLTS | 51.25 | 99.80 | 63.07 | 0.43 | 18.90 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.97 | 99.72 | 99.76 | 0.17 | 0.24 |
| | RobustMDL | 100.00 | 99.92 | 99.95 | 0.12 | 0.24 |
| | RLARS | 88.02 | 99.92 | 92.69 | 0.11 | 0.41 |
| | LAD-lasso | 92.50 | 99.84 | 95.64 | 0.55 | 6.91 |
| | SparseLTS | 61.46 | 99.92 | 74.61 | 0.45 | 17.88 |

TABLE 3.16. Similar to Table 3.1 but for settings with $(n, p) = (200, 10000)$ and with outliers $95\%N(0, 1)$ & $5\%N(0, 7^2)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(5, \frac{2}{\sqrt{5}})$ | MDL | 99.72 | 91.00 | 94.14 | 0.35 | 0.24 |
| | RobustMDL | 99.77 | 100.00 | 99.87 | 0.11 | 0.23 |
| | RLARS | 75.46 | 100.00 | 84.67 | 0.11 | 0.41 |
| | LAD-lasso | 95.91 | 100.00 | 97.74 | 0.44 | 6.14 |
| | SparseLTS | 55.80 | 100.00 | 68.31 | 0.29 | 20.15 |
| $(5, \frac{3}{\sqrt{5}})$ | MDL | 99.71 | 99.84 | 99.74 | 0.12 | 0.23 |
| | RobustMDL | 99.90 | 100.00 | 99.95 | 0.11 | 0.23 |
| | RLARS | 80.29 | 100.00 | 87.80 | 0.09 | 0.40 |
| | LAD-lasso | 95.77 | 100.00 | 97.66 | 0.42 | 6.16 |
| | SparseLTS | 57.02 | 100.00 | 68.27 | 0.31 | 18.67 |
| $(5, \frac{5}{\sqrt{5}})$ | MDL | 99.81 | 100.00 | 99.89 | 0.11 | 0.24 |
| | RobustMDL | 99.93 | 100.00 | 99.96 | 0.11 | 0.24 |
| | RLARS | 84.63 | 100.00 | 90.60 | 0.07 | 0.41 |
| | LAD-lasso | 95.95 | 100.00 | 97.75 | 0.40 | 6.84 |
| | SparseLTS | 81.41 | 100.00 | 88.70 | 0.34 | 17.87 |

TABLE 3.17. Similar to Table 3.1 but for settings with $(n, p) = (300, 20000)$ and $\varepsilon_i \sim N(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.87 | 100.00 | 99.93 | 0.03 | 0.67 |
| | RobustMDL | 99.19 | 100.00 | 99.57 | 0.03 | 0.67 |
| | RLARS | 87.12 | 100.00 | 92.93 | 0.07 | 0.98 |
| | LAD-lasso | 86.37 | 100.00 | 92.23 | 0.33 | 22.71 |
| | SparseLTS | 68.65 | 100.00 | 79.85 | 0.32 | 78.90 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.98 | 100.00 | 99.99 | 0.03 | 0.68 |
| | RobustMDL | 99.60 | 100.00 | 99.79 | 0.03 | 0.68 |
| | RLARS | 90.01 | 100.00 | 94.53 | 0.06 | 1.00 |
| | LAD-lasso | 86.85 | 100.00 | 92.52 | 0.31 | 22.94 |
| | SparseLTS | 55.53 | 100.00 | 68.48 | 0.30 | 75.60 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.98 | 100.00 | 99.99 | 0.03 | 0.67 |
| | RobustMDL | 99.84 | 100.00 | 99.92 | 0.03 | 0.67 |
| | RLARS | 92.84 | 100.00 | 96.09 | 0.05 | 0.98 |
| | LAD-lasso | 86.21 | 100.00 | 92.16 | 0.29 | 24.46 |
| | SparseLTS | 93.52 | 100.00 | 96.40 | 0.41 | 74.36 |

TABLE 3.18. Similar to Table 3.1 but for settings with $(n, p) = (300, 20000)$ and $\varepsilon_i \sim \text{Laplace}(0, 1)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.72 | 99.35 | 99.48 | 0.08 | 0.64 |
| | RobustMDL | 99.72 | 99.90 | 99.80 | 0.06 | 0.64 |
| | RLARS | 94.05 | 100.00 | 96.77 | 0.08 | 0.93 |
| | LAD-lasso | 90.78 | 100.00 | 94.88 | 0.42 | 24.54 |
| | SparseLTS | 43.94 | 100.00 | 57.12 | 0.37 | 81.99 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.84 | 100.00 | 99.92 | 0.06 | 0.66 |
| | RobustMDL | 99.89 | 100.00 | 99.94 | 0.06 | 0.66 |
| | RLARS | 95.52 | 100.00 | 97.56 | 0.07 | 0.95 |
| | LAD-lasso | 91.51 | 100.00 | 95.30 | 0.40 | 24.52 |
| | SparseLTS | 47.28 | 100.00 | 60.33 | 0.38 | 77.14 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.96 | 100.00 | 99.98 | 0.06 | 0.66 |
| | RobustMDL | 99.98 | 100.00 | 99.99 | 0.06 | 0.66 |
| | RLARS | 97.01 | 100.00 | 98.37 | 0.06 | 0.95 |
| | LAD-lasso | 91.60 | 100.00 | 95.32 | 0.37 | 26.72 |
| | SparseLTS | 75.84 | 100.00 | 85.39 | 0.49 | 74.17 |

TABLE 3.19. Similar to Table 3.1 but for settings with $(n, p) = (300, 20000)$ and $\varepsilon_i \sim t_3$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.73 | 91.67 | 94.57 | 0.33 | 0.64 |
| | RobustMDL | 99.82 | 99.62 | 99.66 | 0.12 | 0.64 |
| | RLARS | 92.32 | 99.83 | 95.67 | 0.10 | 0.93 |
| | LAD-lasso | 93.21 | 99.52 | 96.04 | 0.57 | 25.24 |
| | SparseLTS | 48.23 | 99.85 | 61.25 | 0.41 | 83.76 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.82 | 99.20 | 99.30 | 0.18 | 0.67 |
| | RobustMDL | 99.93 | 100.00 | 99.96 | 0.10 | 0.67 |
| | RLARS | 93.64 | 100.00 | 96.53 | 0.08 | 0.96 |
| | LAD-lasso | 93.08 | 99.83 | 96.02 | 0.54 | 25.60 |
| | SparseLTS | 54.79 | 100.00 | 66.36 | 0.42 | 77.41 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.96 | 99.78 | 99.79 | 0.17 | 0.69 |
| | RobustMDL | 100.00 | 100.00 | 100.00 | 0.10 | 0.69 |
| | RLARS | 95.24 | 100.00 | 97.40 | 0.07 | 0.99 |
| | LAD-lasso | 93.32 | 100.00 | 96.31 | 0.48 | 27.85 |
| | SparseLTS | 73.04 | 100.00 | 83.31 | 0.47 | 75.01 |

TABLE 3.20. Similar to Table 3.1 but for settings with $(n, p) = (300, 20000)$ and with outliers $95\%N(0, 1)$ & $5\%N(0, 7^2)$.

| $(d, b)$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE | time(sec) |
|---|---|---|---|---|---|---|
| $(8, \frac{2}{\sqrt{8}})$ | MDL | 99.89 | 90.22 | 93.94 | 0.36 | 0.66 |
| | RobustMDL | 99.87 | 99.98 | 99.92 | 0.11 | 0.65 |
| | RLARS | 90.64 | 100.00 | 94.88 | 0.06 | 0.95 |
| | LAD-lasso | 95.17 | 100.00 | 97.36 | 0.45 | 22.87 |
| | SparseLTS | 63.36 | 100.00 | 75.78 | 0.30 | 81.55 |
| $(8, \frac{3}{\sqrt{8}})$ | MDL | 99.89 | 99.92 | 99.90 | 0.11 | 0.66 |
| | RobustMDL | 99.96 | 100.00 | 99.98 | 0.10 | 0.66 |
| | RLARS | 92.87 | 100.00 | 96.12 | 0.05 | 0.96 |
| | LAD-lasso | 95.34 | 100.00 | 97.48 | 0.44 | 23.64 |
| | SparseLTS | 50.88 | 100.00 | 64.39 | 0.28 | 77.52 |
| $(8, \frac{5}{\sqrt{8}})$ | MDL | 99.93 | 100.00 | 99.96 | 0.11 | 0.68 |
| | RobustMDL | 100.00 | 100.00 | 100.00 | 0.10 | 0.67 |
| | RLARS | 94.42 | 100.00 | 96.96 | 0.04 | 0.98 |
| | LAD-lasso | 95.42 | 100.00 | 97.51 | 0.41 | 25.76 |
| | SparseLTS | 89.88 | 100.00 | 94.29 | 0.39 | 74.94 |

TABLE 3.21. The Precision, Recall, F1 score and MSE values for the methods compared in the nonparametric additive model experiments.

| error distribution | $t$ | method | Precision(%) | Recall(%) | F1 score(%) | MSE |
|---|---|---|---|---|---|---|
| $N(0,1)$ | 0 | MDL | 100.00 | 99.85 | 99.92 | 0.93 |
| | | RobustMDL | 100.00 | 99.85 | 99.92 | 0.93 |
| | 1 | MDL | 99.55 | 98.65 | 99.10 | 0.94 |
| | | RobustMDL | 99.55 | 98.65 | 99.10 | 0.94 |
| Laplace(0,1) | 0 | MDL | 100.00 | 99.65 | 99.82 | 1.84 |
| | | RobustMDL | 98.08 | 99.65 | 98.86 | 1.83 |
| | 1 | MDL | 99.49 | 88.35 | 93.59 | 1.99 |
| | | RobustMDL | 97.56 | 94.15 | 95.83 | 1.89 |
| $t_5$ | 0 | MDL | 100.00 | 99.85 | 99.92 | 1.56 |
| | | RobustMDL | 100.00 | 99.85 | 99.92 | 1.56 |
| | 1 | MDL | 98.98 | 92.20 | 95.47 | 1.65 |
| | | RobustMDL | 97.86 | 96.10 | 96.97 | 1.59 |
| Outliers | 0 | MDL | 100.00 | 96.60 | 98.27 | 3.17 |
| | | RobustMDL | 100.00 | 99.45 | 99.72 | 3.10 |
| | 1 | MDL | 99.84 | 62.80 | 77.10 | 3.83 |
| | | RobustMDL | 95.17 | 81.70 | 87.92 | 3.40 |

CHAPTER 4

# Conclusions

In this dissertation, we studied and developed novel solutions to different problems in the area of high-dimensional statistical machine learning: uncertainty quantification and inference for multi-task learning, variable selection in the regression setting with the existence of outliers in the data, and the robust fitting of sparse nonparametric additive models. A sequence of simulation results and various real data applications such as gene expression and fMRI data are presented to demonstrate the efficiency and effectiveness of our proposed methods.

It is worth noticing that some components of our methods can be updated to more advanced and new techniques. For example, we use multi-task SIS and $L_1/L_2$ Lasso in Chapter 2 to construct a class of candidate models with reasonable size. However, one can use other screening procedures to reduce the number of candidate models if these procedures have better theoretical properties and performances on a specific type of applications. Moreover, our work in Chapter 3 focuses on the so-called two-part code version of MDL. It will be exciting to develop new versions of MDL that can provide better empirical performances.

Besides point estimation under normal assumptions, uncertainty quantification and robust fitting are also important in research and real world applications. We hope our contributions can stimulate more interest in applying generalized fiducial inference and minimum description length principle to solve other challenging problems in the statistical machine learning community.

# Bibliography


Alfons, A., Croux, C. and Gelper, S. (2013) Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 226–248.

Amato, U., Antoniadis, A., De Feis, I. and Gijbels, I. (2021) Penalised robust estimators for sparse and high-dimensional linear models. *Statistical Methods & Applications*, **30**, 1–48.

Aue, A., Cheung, R. C. Y., Lee, T. C. M. and Zhong, M. (2014) Segmented model selection in quantile regression using the minimum description length principle. *Journal of the American Statistical Association*, **109**, 1241–1256.

Bedrick, E. J. and Tsai, C.-L. (1994) Model selection for multivariate regression in small samples. *Biometrics*, 226–231.

Bondell, H. D. and Reich, B. J. (2012) Consistent high-dimensional bayesian variable selection via penalized credible regions. *Journal of the American Statistical Association*, **107**, 1610–1624.

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

Caruana, R. (1997) Multitask learning. *Machine learning*, **28**, 41–75.

Chen, J. and Chen, Z. (2008) Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, **95**, 759–771.

Cheung, R. C. Y., Aue, A. and Lee, T. C. M. (2017) Consistent estimation for partition-wise regression and classification models. *IEEE Transactions on Signal Processing*, **65**, 3662–3674.

Cisewski, J., Hannig, J. *et al.* (2012) Generalized fiducial inference for normal linear mixed models. *The Annals of Statistics*, **40**, 2102–2127.

Cohen, I., Raz, S. and Malah, D. (1999) Translation-invariant denoising using the minimum description length criterion. *Signal Processing*, **75**, 201–223.

Dempster, A. P. (2008) The dempster–shafer calculus for statisticians. *International Journal of approximate reasoning*, **48**, 365–377.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. *et al.* (2004) Least angle regression. *The Annals of Statistics*, **32**, 407–499.

Evgeniou, A. and Pontil, M. (2007) Multi-task feature learning. *Advances in neural information processing systems*, **19**, 41.

Fan, J., Feng, Y. and Song, R. (2011) Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, **106**, 544–557.

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

Fan, J. and Lv, J. (2008) Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B*, **70**, 849–911.

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

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

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

Gamage, J., Mathew, T. and Weerahandi, S. (2013) Generalized prediction intervals for blups in mixed models. *Journal of Multivariate Analysis*, **120**, 226–233.

Gao, Q., Lee, T. C. M. and Yau, C. Y. (2017) Nonparametric modeling and break point detection for time series signal of counts. *Signal Processing*, **138**, 307–312.

Giurcăneanu, C. D., Razavi, S. A. and Liski, A. (2011) Variable selection in linear regression: Several approaches based on normalized maximum likelihood. *Signal Processing*, **91**, 1671–1692.

Grünwald, P. D. (2007) *The minimum description length principle*. MIT press.

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

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

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

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

Kallummil, S. and Kalyani, S. (2016) High SNR consistent linear model order selection and subset selection. *IEEE Transactions on Signal Processing*, **64**, 4307–4322.

Khan, J. A., Van Aelst, S. and Zamar, R. H. (2007) Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, **102**, 1289–1299.

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

Lai, R. C. S., Huang, H.-C. and Lee, T. C. M. (2012) Fixed and random effects selection in nonparametric additive mixed models. *Electronic Journal of Statistics*, **6**, 810–842.

Lan, H., Chen, M., Flowers, J. B., Yandell, B. S., Stapleton, D. S., Mata, C. M., Mui, E. T.-K., Flowers, M. T., Schueler, K. L., Manly, K. F. *et al.* (2006) Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genet*, **2**, e6.

Lee, T. C. M. (2000) Regression spline smoothing using the minimum description length principle. *Statistics and Probability Letters*, **48**, 71–82.

Liu, H., Wang, L. and Zhao, T. (2014) Multivariate regression with calibration. *Advances in neural information processing systems*, **27**, 5630.

Liu, X., Goncalves, A. R., Cao, P., Zhao, D., Banerjee, A., Initiative, A. D. N. *et al.* (2018) Modeling alzheimer's disease cognitive scores using multi-task sparse group lasso. *Computerized Medical Imaging and Graphics*, **66**, 100–114.

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

Luo, S. and Chen, Z. (2013) Extended BIC for linear regression models with diverging number of relavant features and high or ultra-high feature spaces. *Journal of Statistical Planning and Inference*, **143**, 494–504.

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

Misra, I., Shrivastava, A., Gupta, A. and Hebert, M. (2016) Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*,

3994–4003.

Morsheddost, H., Asemani, D. and Mirahadi, N. (2014) Optimization of MDL-based wavelet denoising for fMRI data analysis. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, 33–36. IEEE.

Obozinski, G., Taskar, B. and Jordan, M. (2006) Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, **2**, 2.

Obozinski, G., Wainwright, M. J. and Jordan, M. I. (2008) Union support recovery in high-dimensional multivariate regression. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, 21–26. IEEE.

Rissanen, J. (1989) *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore.

Rissanen, J. (2000) MDL denoising. *IEEE Transactions on Information Theory*, **46**, 2537–2543.

Rissanen, J. (2007) *Information and Complexity in Statistical Modeling*. Springer Science & Business Media.

Roos, T., Myllymaki, P. and Rissanen, J. (2009) MDL denoising revisited. *IEEE Transactions on Signal Processing*, **57**, 3347–3360.

Schmidt, D. F. and Makalic, E. (2011) The consistency of MDL for linear regression models with increasing signal-to-noise ratio. *IEEE Transactions on Signal Processing*, **60**, 1508–1510.

Seneviratne, A. J. and Solo, V. (2012) On vector l0 penalized multivariate regression. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3613–3616.

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

Triantafyllopoulos, K. (2011) Real-time covariance estimation for the local level model. *Journal of Time Series Analysis*, **32**, 93–107.

Wandler, D. V. and Hannig, J. (2012) Generalized fiducial confidence intervals for extremes. *Extremes*, **15**, 67–87.

Wang, H., Li, G. and Jiang, G. (2007) Robust regression shrinkage and consistent variable selection through the lad-lasso. *Journal of Business & Economic Statistics*, **25**, 347–355.

Weerahandi, S. (1995) Generalized confidence intervals. In *Exact statistical methods for data analysis*, 143–168. Springer.

Wong, R. K. W., Lai, R. C. S. and Lee, T. C. M. (2010) Structural break estimation of noisy sinusoidal signals. *Signal processing*, **90**, 303–312.

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

Yanagihara, H., Wakaki, H., Fujikoshi, Y. *et al.* (2015) A consistency property of the aic for multivariate linear models when the dimension and the sample size are large. *Electronic Journal of Statistics*, **9**, 869–897.

Yuan, M., Ekici, A., Lu, Z. and Monteiro, R. (2007) Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **69**, 329–346.

Yuan, M. and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, **68**, 49–67.

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

Zhang, Z., Luo, P., Loy, C. C. and Tang, X. (2014) Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, 94–108. Springer.