

ABSTRACT

WHITE, KYLE ROSS. Model-Agnostic Variable Selection Through Measurement Error Model Selection Likelihoods. (Under the direction of Leonard A. Stefanski and Yichao Wu.)

A new wrapper method of variable selection is developed that can be applied to any “black-box” prediction model without the requirement of a likelihood. We first establish a connection between measurement error attenuation and L_1 - and L_2 -penalized linear regression by showing that weak model features may be viewed as contaminated with measurement error. Then, we force “false” measurement error into a fitting procedure and optimize the distribution of the errors across the predictors such that in-sample black-box predictions are impacted the least. The least important variables will absorb the most error. Using an inverse measurement error variance parameterization, we achieve variable selection by allowing features to absorb an infinite amount of error. We call this approach Measurement Error Model Selection Likelihoods, MEMSEL.

We define a four-step outline to MEMSEL and demonstrate it in the familiar context of linear regression. We prove the equivalence between MEMSEL and LASSO in the linear model. Thus, MEMSEL is a penalty-free generalization of LASSO. We apply it to the problem of variable selection in nonparametric classification, resulting in a new kernel-based classifier with LASSO-like variable shrinkage and selection properties. Finite-sample performance of the new classification method is studied via simulation and real-data examples, and consistency of the method is studied theoretically.

The MEMSEL approach is extended to density-based nonparametric regression. The Measurement Error Kernel Regression Operator (MEKRO) has the same form as the Nadaraya-Watson kernel estimator, but optimizes a Selection Likelihood to estimate the kernel bandwidths. Much like LASSO or COSSO solution paths, MEKRO results in solution paths depending on a tuning parameter that controls shrinkage and selection via a bound on the harmonic mean of the false measurement error standard deviations. We use small-sample-corrected AIC to select the tuning parameter. Large-sample properties of MEKRO are studied and small-sample properties are

explored via Monte Carlo experiments and applications to data.

Measurement Error Model Selection Likelihoods are generalized to allow any black-box prediction model. First, a new tuning method (Selection Information Criterion, SIC) is defined that penalizes overfitting more aggressively than BIC. Simulations suggest using SIC when the true model is sparse. Then we discuss a variety of ways to contaminate the black-box predictors. Variants of general MEMSEL are explored in the linear model. Two MEMSEL variants with favorable linear model performance are applied to a random forest black box using SIC for tuning. This outperforms competing random forest variable selection procedures on both selection and prediction. Finally a real-data example is given.

© Copyright 2017 by Kyle Ross White

All Rights Reserved

Model-Agnostic Variable Selection Through Measurement Error Model Selection Likelihoods

by
Kyle Ross White

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina

2017

APPROVED BY:

Dennis Boos

Eric Laber

Leonard A. Stefanski
Co-chair of Advisory Committee

Yichao Wu
Co-chair of Advisory Committee

DEDICATION

To Fletch, for being by my side through everything. I love you.

BIOGRAPHY

Kyle was raised in Waterford, Pennsylvania by his parents Bill and Lori and next to his older brother, Jerad. In 2008, he earned his B.S. in Mathematics from Penn State Erie, The Behrend College where he began developing an interest in Statistics. He chose to pursue a graduate degree in Statistics at North Carolina State University. From 2009 to 2012, Kyle served as an intern at Duke Clinical Research Institute performing secondary analyses of clinical trials. He earned his M.Stat. in 2010 and began Ph.D. research under the direction of Dr. Len Stefanski and Dr. Yichao Wu. He met his now wife in 2012 and they married two years later. While still working on his Ph.D., Kyle took an internship at a local startup, Republic Wireless. He was accepted as a full-time employee in 2014 and works there as a Data Scientist as of 2017.

ACKNOWLEDGEMENTS

First and foremost, thank you Dr. Stefanski. I will never forget the day in January 2012 that you convinced me to cancel my job search and stick with research. Thank you for the opportunity to work on this project with you. Thank you for believing in me and always being a source of encouragement and motivation.

To both of my advisors, Dr. Stefanski and Dr. Wu, thank you for your wisdom, guidance, and endless patience. I truly appreciate all of time you spent with me on this research and as mentors. Thank you for all the thoughtful discussions and comments that greatly improved my work. And I cannot thank you enough for the flexibility to allow me to balance research with my career and personal life.

To Dr. Boos, Dr. Laber, and Dr. Healey, thank you for serving on my committee and providing feedback on this dissertation.

To Dr. Davidian and Dr. Wu, thank you for funding me through portions of your grants (NIH T32HL079896; NIH P01CA142538; NSF DMS-1055210).

To the DCRI, thank you for the internship opportunity and invaluable experience. Special thanks to Karen Pieper for her incredibly selfless mentorship.

To Republic Wireless, thank you for giving me the freedom to finish this dissertation.

To The Fifth Moment, thank you for all of the memorable performances (and practices).

To my first-year cohort, thank you for making my transition to Raleigh easy and for the lifelong friendships. I was lucky to start at the same time as the rest of you.

To my closest friends—Joe and Jesse—thank you for your companionship and support, and for providing many welcome distractions.

To mom, dad, and bro, thank you for being there for me since day one.

To my wife, thank you for showing me nothing but love and patience. Thank you for every sacrifice you've made for me while I work on this dissertation. Thank you for being my rock.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	ix
Chapter 1 INTRODUCTION	1
Chapter 2 MEASUREMENT ERROR MODEL SELECTION LIKELIHOODS	4
2.1 Introduction	4
2.2 Variable Selection and Measurement Error	6
2.2.1 Attenuation, Shrinkage, and Selection	6
2.2.2 Oracle Heuristics	8
2.2.3 MEM Selection Likelihoods	9
2.3 Linear Model Selection Likelihoods	11
2.3.1 Linear Regression	11
2.3.2 Relationship to LASSO	13
2.4 Nonparametric Classification	15
2.4.1 Selection Likelihoods for Classification	15
2.4.2 Multicategory SKDA	18
2.4.3 Simulation Studies	18
2.4.4 Illustrations with Real Data	29
2.4.5 Consistency of SKDA	32
2.5 Summary	35
2.6 Appendix (Supplemental Files)	36
2.6.1 Equivalence of LASSO and \hat{L}_{SEL_1} -MESSO	36
2.6.2 Generating Sparse Classification Data	45
2.6.3 Proofs of Asymptotic Results	45
Chapter 3 MEKRO	50
3.1 Introduction	50
3.2 Measurement Error Kernel Regression Operator	52
3.2.1 Example	54
3.2.2 Tuning and Solution Paths	55
3.3 Extension to Categorical Predictors	57
3.4 Method Comparison and Numerical Results	58
3.4.1 Simulation Preliminaries	59
3.4.2 Simulation Results	61
3.5 Asymptotic Results	72
3.6 Discussion	72
3.7 Appendix	74
3.7.1 MEKRO Selection Likelihood Derivation	74
3.7.2 Asymptotic Selection Consistency	76
3.7.3 Numerical Study with Gaussian \mathbf{X}	80

Chapter 4 GENERAL MEASUREMENT ERROR MODEL SELECTION LIKE-LIHOODS	82
4.1 Introduction	82
4.2 Selection Information Criterion	83
4.2.1 Comparing Tuning Criteria Penalties	84
4.2.2 Tuning Method Simulation Study	86
4.2.3 SIC Sensitivity Analysis to p	93
4.3 MEMSEL	96
4.3.1 Contamination Via Pseudo-Measured Predictors	97
4.3.2 MEMSEL Objective, Optimization, and Tuning	100
4.4 MEMSEL in Linear Models	102
4.4.1 Simulation Study Setup	106
4.4.2 Simulation Study Results	107
4.5 MEMSEL in Random Forests	113
4.5.1 Motivation	113
4.5.2 Selection Using Variable Importance	113
4.5.3 Simulation Study Setup	115
4.5.4 Software Considerations	116
4.5.5 Simulation Study Results	116
4.5.6 Real Data Example – Concrete Workability	119
4.6 Summary	121
4.7 Appendix	122
4.7.1 Centering and Scaling \mathbf{Z}	122
4.7.2 Derivation of MEMSEL in Linear Models	123
BIBLIOGRAPHY	127

LIST OF TABLES

Table 2.1	Predictor selection frequencies for SKDA and SLDA for the study in Section 2.4.3.1.	20
Table 2.2	Predictor selection frequencies for SKDA and SLDA for the study in Section 2.4.3.2.	22
Table 2.3	Frequency of X_1 and X_2 and the average frequency of X_3, \dots, X_{50} being selected by SKDA and SLDA.	25
Table 2.4	Predictor selection frequencies for MSKDA for the study in Section 2.4.3.3.	25
Table 2.5	Predictor selection frequencies for MSKDA for the study in Section 2.4.3.4.	28
Table 2.6	Test error summary for 40 random splits of the WBCD data.	29
Table 2.7	Predictor selection frequencies for SLDA and SKDA for the WBCD data example. Asterisks indicate variables selected using the entire data set. . .	30
Table 2.8	Test error summary for 40 random splits of the chemical signature data. .	31
Table 2.9	Predictor selection frequencies for SLDA and SKDA for the chemical signature data. Asterisks indicate variables selected using the entire data set.	33
Table 3.1	Selection error rates for Model 1. MC standard errors for all cells ≤ 0.03 . .	62
Table 3.2	Selection error rates for Model 2. MC standard errors for all cells ≤ 0.03 . .	64
Table 3.3	Selection error rates for Model 3. MC standard errors for all cells ≤ 0.03 . .	66
Table 3.4	Selection error rates for Model 4, with X_1 in $Z(\cdot)$ and X_4 classified irrelevant. MC standard errors for all cells ≤ 0.04	67
Table 3.5	Main effect selection rates (not errors) for Model 5, averaged over the four simulation settings. The IRR row is the average selection rate for the four irrelevant predictors that were independently generated. MC standard errors for all cells ≤ 0.03	69
Table 3.6	Main effect selection rates (not errors) for the prostate data. ‘Avg Model’ is the method’s average model size; ‘Corr’ is the selection rate correlation of each method with LAS. MC standard errors for all selection rates ≤ 0.07 and all average model sizes ≤ 0.55	70
Table 3.7	Selection error rates for the model in Appendix C. MC standard errors for all cells ≤ 0.03	81
Table 4.1	Simulation setup to study effects of n , p , p_1 on SIC. Both p and n are explicitly chosen as simulation factors. The number of non-zero coefficients p_1 is implied by the choice of β and r . There are 84 total simulations created from full crosses of r , p , and n in each row.	93
Table 4.2	Selection and prediction errors for random forest MEMSEL (XM1, XM2), HAP, and VSURF. Simulation factor tweaks are shown in the ‘Setup’ column, e.g., $R^2 : 0.75 \searrow 0.50$ indicates decreasing R^2 from 0.75 to 0.50 in the model. Values in bold are statistically no different from those with the lowest error rate across the row using a paired t -test at $\alpha = 0.20$	118

Table 4.3 Selection results from concrete data. Values are the fraction of times a variable is chosen out of 100 redrawn samples. AVG = average selection rate over all variables. 121

LIST OF FIGURES

Figure 2.1	Plot of relative deviations from the average total variance versus total squared bias. LASSO (red-dashed); MESSO estimator from $\widehat{L}_{\text{SEL}_1}$ (solid blue); full-conditional MESSO estimator from $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}$ (black dashed); and the full-unconditional MESSO estimator from $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}\widehat{L}_{\text{SEL}_3}$ (green dot-dash). Because of their equivalency the $\widehat{L}_{\text{SEL}_1}$ -MESSO and LASSO lines overlap and appear as a single alternating red/blue dashed line. . . .	13
Figure 2.2	Plots of typical training samples and boxplots of test errors for data generated with $\rho = 0$ (top), 0.3 (middle), and 0.6 (bottom), from Section 2.4.3.1.	21
Figure 2.3	Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.2 with $p = 10$ and three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom).	23
Figure 2.4	Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.2 with $p = 50$ and three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom).	24
Figure 2.5	Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.3.	26
Figure 2.6	Plot of a training sample (left) and boxplots of test errors (right) for the discrete predictor simulation study in Section 2.4.3.3.	27
Figure 2.7	Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.4.	28
Figure 2.8	Boxplots of test errors for the WBCD example	30
Figure 2.9	Chemical data heat map. First 50 columns (left to right) are the “Y = 0” group. Columns within groups randomly ordered. Rows identify compounds X_1, \dots, X_{38}	32
Figure 3.1	$\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ contours and gradient vector fields of example model (3.5) for $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3 = \tau - \lambda_1 - \lambda_2)$ and $\tau \in \{1, 2, 3\}$; global maxima are denoted with solid circles and the neutral starting values $\boldsymbol{\lambda}_{\text{start}} = (\tau/p)\mathbf{1}_p$ are denoted with solid diamonds.	55
Figure 3.2	Solution paths of $\widehat{\boldsymbol{\lambda}}_\tau$ versus τ for Section 3.2.1 example with two active (solid) and three irrelevant (open) predictors. Dashed line: scaled $\text{AIC}_c(\tau)$, $\tau \in \boldsymbol{\tau}^* = \{0, 0.5, \dots, 7\}$	56
Figure 3.3	AISEs for Model 1. Note the scale differences. Out of the 400 MC samples, 3 AISE outliers are omitted from M3.	62
Figure 3.4	Study of AMKR estimates for Model 1, $n = 400$	63
Figure 3.5	AISEs for Model 2. Note the scale differences. Out of the 400 MC samples, 19 and 6 AISE outliers are omitted from M3 and M2, respectively.	64
Figure 3.6	AISEs for Model 3. Note the scale differences. Out of the 400 MC samples, 2 and 1 AISE outlier(s) are omitted from M3 and M2, respectively.	66

Figure 3.7	AISEs for Model 4; units for $Z(\cdot)$ plots are 10^3 and units for $\phi(\cdot)$ plots are 10^{-3} . Note the scale differences. Dashed lines indicate methods with AISEs too large to display.	67
Figure 3.8	ASPEs (average squared prediction errors) for Model 5. Note the scale differences. Out of the 400 MC samples, 3 ASPE outliers are omitted from both M3 and M2.	69
Figure 3.9	ASPEs (average squared prediction errors) for the prostate data. Out of the 100 MC samples, 2 and 4 ASPE outliers are omitted from M3 and M2, respectively.	71
Figure 3.10	AISEs (average integrated squared errors) for Appendix C Model. Note the scale differences. Out of the 400 MC samples, 2 AISE outliers are omitted from both M2 and M3.	81
Figure 4.1	Comparing overfitting penalties in AIC (dashed line), BIC (solid line), GCV (dash-dot-dash line), and SIC (dotted lines) as model degrees of freedom (k) increase. Left pane: $n = 50$. Right pane: $n = 100$. Larger penalties for the same model degrees of freedom result in sparser models.	85
Figure 4.2	Comparison of perfection selection rates (higher is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$	90
Figure 4.3	Comparison of perfection rates with increased n (higher is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$	91
Figure 4.4	Comparison of test errors in SIC simulation study (lower is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$	92
Figure 4.5	Results of sensitivity analysis of SIC. Each of the 84 points coincides with one simulation described in Table 4.1. Results are the tuning method that had the best perfect selection rate over 100 MC replicates. The results “best,” “tied,” and “not best” are judged with a paired t -test at $\alpha = 0.05$	95
Figure 4.6	Comparing AIC curves against coefficient size between LASSO (gray solid line) and the equivalent MEMSEL variant, (\mathbf{W}, \mathbf{W}) with $m = 1$ (red dashed line). Left: a grid size of 10 is used in MEMSEL modified coordinate descent. Right: a grid size of 100 is used in MEMSEL modified coordinate descent. Identical data are used for both plots. Discrepancies occur when optimized $\hat{\lambda}_\tau$ values are not global minimizers. The gray (red) circle denotes the minimum AIC value for LASSO (MEMSEL).	108
Figure 4.7	Results from linear Model 1. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the test loss axis is truncated.	110

Figure 4.8	Results from linear Model 2. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the test loss axis is truncated.	111
Figure 4.9	Results from linear Model 3. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the active selection error and test loss axes are truncated.	112
Figure 4.10	Distribution of RMSEs from concrete data. The thin solid vertical line represents the middle 95% of RMSEs, the thick solid vertical line represents the middle 50% of RMSEs, and the horizontal line is the median RMSE over 100 samples. The distribution of RMSEs from ANN is approximated from the range of test-set results in [85].	120

CHAPTER

1

INTRODUCTION

We study an algorithm to achieve variable selection generally, agnostic to the choice of model describing observed data. A common approach to variable selection adds an L_1 -penalty term to the likelihood to encourage coefficient sparsity. This is not straightforward or even sensible on complex models or models without coefficients or a likelihood. We propose a “wrapper method” variable selection approach to apply to any black-box method that produces a prediction function from input data. The idea is closely related to the coefficient attenuation observed when predictors are contaminated with measurement error, coining this approach broadly as Measurement Error Model Selection Likelihoods (MEMSEL). The acronym MEMSEL may be used to describe either the proposed variable selection method or the selection likelihood itself.

This dissertation contains two previously-published papers as chapters, and a chapter that is not yet submitted for publication. This introduction is then intentionally kept brief. The chapter layout is as follows:

- **Chapter 2.** Taken verbatim (with minor stylistic edits) from [68],

Stefanski, L. A. et al. "Variable Selection in Nonparametric Classification Via Measurement Error Model Selection Likelihoods." Journal of the American Statistical Association 109.506 (2014), pp. 574-589.

This chapter describes the relationship between measurement error and variable selection. It outlines a four-step approach for building measurement error selection likelihoods and proves the equivalence between (one variant of) MEMSEL in the linear model and LASSO. MEMSEL is applied to kernel-based discriminant analysis to produce Sparse Kernel Discriminant Analysis (SKDA) that demonstrates favorable performance over other sparse classifiers. Two real-data examples regarding breast cancer and gas well classification are given. Selection consistency for the MEMSEL classifier is proved.

- **Chapter 3.** Taken verbatim (with minor stylistic edits) from [79],

White, K. R. et al. "Variable Selection in Kernel Regression Using Measurement Error Selection Likelihoods." Journal of the American Statistical Association (accepted, to appear).

This chapter extends the work from Chapter 2 and applies MEMSEL to kernel density-based regression. The result is the Measurement Error Kernel Regression Operator (MEKRO). It has the same form as the Nadaraya-Watson estimator [53, 78] but with smoothing parameters determined by the MEMSEL variable selection algorithm. A method of handling categorical input variables is defined. Simulations demonstrate that MEKRO has superior selection and predictive performance to other selection-capable nonparametric regression approaches on models with many interactions. Real-data examples on robot arm movement and prostate cancer are provided. An argument for selection consistency is given.

- **Chapter 4.** This chapter formalizes a generalization of MEMSEL to be applied to any black-box model. It introduces and studies a new tuning criterion, Selection Information Criterion (SIC), that outperforms AIC, BIC, cross validation, and others in many cases

for selection. The four-step method from Chapter 2 is then generalized to include different forms of predictor contamination. As an aid to understanding how the new approach works, general MEMSEL variants are applied to linear models in detail via theory and simulation. A subset of general MEMSEL variants coupled with a random forest black-box estimator are then compared to other random forest selection estimators. MEMSEL shows favorable selection and prediction results. Random forest MEMSEL is illustrated on a real data set regarding wet concrete workability.

Appendices are provided immediately following each chapter. The appendix following Chapter 2 (Section 2.6) contains the supplemental files from [68] with additional details to support the key equivalence proof between LASSO and MEMSEL. The appendix following Chapter 3 (Section 3.7) is copied verbatim from the supplemental files in [79].

CHAPTER

2

MEASUREMENT ERROR MODEL SELECTION LIKELIHOODS

2.1 Introduction

Adopting a measurement-error-model-based approach to variable selection, we propose a nonparametric, kernel-based classifier with LASSO-like shrinkage and variable-selection properties. The use of measurement error model (MEM) ideas to implement variable selection is new and provides a different way of thinking about variable selection, with potential for applications in other nonparametric variable selection problems. Thus we also describe what we call *measurement error selection likelihoods* in addition to our main methodological results on variable selection in nonparametric classification.

Compared to parametric classification, variable selection for nonparametric classification

methods is in its infancy. Our research helps fill that gap with a sparsity-seeking kernel method, *sparse kernel discriminant analysis* (SKDA), obtained by implementing the MEM-based approach to variable selection. SKDA is kernel-based with a familiar form, but with a bandwidth parameterization and selection strategy that results in variable selection. We provide additional background and introductory material at the start of the main methodological section on classification, Section 2.4.

In response to a suggestion by the Associate Editor we end this section with remarks relating to the potential benefits of approaching variable selection problems via measurement error models. The fact that a version of our MEM-based approach to variable selection applied to linear regression results in LASSO estimation is of independent interest. Knowing different paths to the same result and the relationships among them, enhances understanding even when it does not lead to new methods. However, our approach is more than just another LASSO computational algorithm. It is a useful conceptualization and generalization of the LASSO that has potential to suggest new variable selection methods.

Penalizing parameters is not always intuitive simply because it is not always the case that variables enter a model through easily intuited parameters; as with nonparametric models and algorithmic fitting methods. However, it is always possible to intuit the case that a variable has (a lot of) measurement error in it. Admittedly, turning the idea that a variable contains measurement error into a variable selection method may require additional, creative modeling, and perhaps extensive computing power to simulate the measurement error process when analytical expressions are not possible. However, the key point is that the MEM-based approach provides another way for researchers to think about variable selection in nonstandard problems. Although variable selection in nonparametric classification is the primary methodological contribution of the paper, the idea of approaching variable selection via measurement error modeling may have broader impact because of the possibility of adapting the strategy to other problems not readily handled by traditional penalty approaches.

The connections between measurement error attenuation, shrinkage, and selection underlying

the new approach to variable selection are discussed in Section 2.2. The new approach is illustrated in the context of linear regression in Section 2.3. The main results on variable selection in nonparametric classification are in Section 2.4, which includes performance assessment via simulation studies, applications to two data sets, and asymptotic results. Concluding remarks appear in Section 2.5 and technical details in the online supplemental materials.

2.2 Variable Selection and Measurement Error

We now describe the connection between measurement error and variable selection that is used to derive the nonparametric classification selection method studied in Section 2.4.

2.2.1 Attenuation, Shrinkage, and Selection

We begin with the connections between measurement error attenuation, ridge regression, and LASSO estimation in linear models $\mathbf{Y}_{n \times 1} = \mathbf{X}\boldsymbol{\beta}_{p \times 1} + \boldsymbol{\epsilon}$.

Measurement error attenuation. Measurement error attenuation is usually introduced in the case of simple linear regression in terms of ‘true’ $\{Y_i, X_i\}$, and measured $\{Y_i, W_i\}$ data, $i = 1, \dots, n$, where $W_i = X_i + \sigma_U Z_i$, and $Z_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1)$ with the Z_i independent of (X_i, Y_i) [8, 11, 24]. The least squares slope of Y on W , $\hat{\beta}_{\text{ATTEN}} = s_{yw}/s_w^2 \xrightarrow{p} \sigma_{yw}/\sigma_w^2 = (\sigma_x^2 + \sigma_U^2)^{-1} \sigma_{yx}$. Attenuation results from the inequality $\sigma_x^2 + \sigma_U^2 > \sigma_x^2$. The multi-predictor version of this result for the error model $\mathbf{W} = \mathbf{X} + \mathcal{D}_{\{\sigma_U\}}\mathbf{Z}$, with $\mathcal{D}_{\{\sigma_U\}} = \text{diag}(\sigma_{U,1}, \dots, \sigma_{U,p})$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is

$$\hat{\beta}_{\text{ATTEN}} = \hat{\mathbf{V}}_{\mathbf{W}}^{-1} \hat{\mathbf{V}}_{\mathbf{W}Y} \approx \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_U^2\}} \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}Y} = \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{1/\sigma_U^2\}}^{-1} \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}Y}. \quad (2.1)$$

Notes. **i)** The approximation “ \approx ” in (2.1) is valid in large samples because $\hat{\mathbf{V}}_{\mathbf{W}}^{-1} \hat{\mathbf{V}}_{\mathbf{W}Y} - \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_U^2\}} \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}Y}$ converges in probability to $\mathbf{0}$. **ii)** We use $1/\sigma_U^2$ to denote componentwise division, i.e., $1/\boldsymbol{\sigma}_U^2 = (1/\sigma_{U,1}^2, \dots, 1/\sigma_{U,p}^2)^T$. **iii)** We include the right-hand-side expression because many of our results are better understood in terms of measurement error *precisions*

(or square-root precisions). **iv)** We use $\mathcal{D}_{\{\mathbf{v}\}}$ to denote a diagonal matrix with vector \mathbf{v} on its diagonal. **v)** For any vector \mathbf{z} , $|\mathbf{z}|$ denotes the vector of componentwise absolute values. Similarly, \mathbf{z}^m denotes the vector of componentwise m^{th} powers. **vi)** Finally, $\widehat{\mathbf{V}}$ denotes a sample variance/covariance matrix of the subscripted variables.

Ridge shrinkage. Ridge regression [38] is much studied and well known. Thus we simply note that after scaling the ridge parameters $\boldsymbol{\nu}$ it has the form

$$\widehat{\boldsymbol{\beta}}_{\text{RIDGE}} = \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\nu}\}} \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} = \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{1/\boldsymbol{\nu}\}}^{-1} \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}. \quad (2.2)$$

LASSO shrinkage and selection. The LASSO has an attenuation-like representation that follows from the iterated-ridge computation algorithm; see [74] and Section 2.3.2. The iterated ridge solution to the penalty form of the LASSO optimization is

$$\widehat{\boldsymbol{\beta}}_{\text{LASSO}} = \underset{\beta_1, \dots, \beta_p}{\operatorname{argmin}} \left\{ \frac{1}{n-1} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \eta \sum_{j=1}^p |\beta_j| \right\} \quad (2.3)$$

$$= \left(\mathbf{I} + \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_{\text{LASSO}}|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}} \right)^{-1} \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_{\text{LASSO}}|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \quad (2.4)$$

$$= \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_{\text{LASSO}}|/\eta\}}^{-1} \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}. \quad (2.5)$$

Notes: **i)** Expressions (2.4) and (2.5) are not computational solutions to the LASSO optimization problem, but rather equations satisfied by the solution. **ii)** Both are included to show that the inversion of a singular matrix in (2.5) when some components of $\widehat{\boldsymbol{\beta}}_{\text{LASSO}}$ are zero is not necessary; however, we use expressions like that in (2.5) for their compactness.

Comparing equations (2.1), (2.2), and (2.5) shows that $\widehat{\boldsymbol{\beta}}_{\text{ATTEN}}$, $\widehat{\boldsymbol{\beta}}_{\text{RIDGE}}$, and $\widehat{\boldsymbol{\beta}}_{\text{LASSO}}$ exhibit

the same *form* of attenuation, with equality in the case that

$$\frac{1}{\sigma_{U,j}^2} = \frac{1}{\nu_j} = \frac{|\widehat{\beta}_{\text{LASSO},j}|}{\eta}.$$

Measurement error attenuation results from uncontrollable and undesirable measurement error in the predictors, whereas ridge and LASSO attenuation is analyst-controlled to manipulate sampling properties of the estimators. In Section 2.2.3 we show that measurement error attenuation can also be controlled to accomplish shrinkage and selection. For another interesting example of creatively using measurement error to impart favorable sampling properties on estimators in a different context, see [62].

2.2.2 Oracle Heuristics

We motivate our measurement error approach to variable selection using a toy example, imagining an oracle’s solution to a simple challenge. Suppose that the regression of Y on predictors X_1, \dots, X_5 satisfies

$$E(Y | X_1, X_2, X_3, X_4, X_5) = E(Y | X_2, X_5).$$

The nullity of X_1 , X_3 , and X_4 is unknown to a statistician, but not to the oracle. However, suppose the oracle agrees to play a game wherein s/he must add a nonzero, total amount of measurement error, $(\sigma_{U,1}Z_1, \sigma_{U,2}Z_2, \sigma_{U,3}Z_3, \sigma_{U,4}Z_4, \sigma_{U,5}Z_5)$, to the predictors $(X_1, X_2, X_3, X_4, X_5)$ in a manner that minimizes the loss of their collective predictive power (“a nonzero, total amount of measurement error” means that $\sum_j \sigma_{U,j}^2 > 0$).

Because the oracle knows that

$$\begin{aligned} E(Y | X_1, X_2, X_3, X_4, X_5) = E(Y | X_2, X_5) &\implies \\ E(Y | X_1 + \sigma_{U,1}Z_1, X_2, X_3 + \sigma_{U,3}Z_3, X_4 + \sigma_{U,4}Z_4, X_5) &= E(Y | X_2, X_5), \end{aligned}$$

the oracle’s solution is to set $\sigma_{U,2} = \sigma_{U,5} = 0$ and let $\sigma_{U,1}$, $\sigma_{U,3}$, and $\sigma_{U,4}$ be nonzero. A statistician seeing the oracle’s solution would know the identity of the null and informative predictors. In the next section we show that this oracle game can be mimicked using a (pseudo-profile) likelihood to replace the omniscience of the oracle, and constraints to ‘force’ the addition of measurement error to the data.

2.2.3 MEM Selection Likelihoods

We now give a general description of the construction and use of measurement error model (MEM) selection likelihoods. As an illustration of the approach, we then implement it for linear models in Section 2.3, where it is shown to lead to the LASSO for linear models. This equivalency lends credibility to the new approach in the sense that it can be viewed as a generalization of the LASSO that reduces to the LASSO in the case of linear regression. In Section 2.4 we implement the approach in the context of nonparametric classification obtaining the new method, *sparse kernel discriminant analysis* (SKDA). The latter is the main methodological contribution of the paper. However, Section 2.4 also demonstrates the utility of the measurement error approach to variable selection for identifying and deriving variable selection strategies in nonparametric/nonstandard models.

Denote the data as (Y_i, \mathbf{X}_i) , $i = 1, \dots, n$, and let (Y, \mathbf{X}) represent a generic observation. The MEM selection likelihood construction proceeds in four basic steps:

1. Start with an assumed ‘true’ likelihood for (\mathbf{X}_i, Y_i) , $i = 1, \dots, n$, denoted $L_{\text{TRUE}}(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ could be finite (parametric) or infinite dimensional (nonparametric).
2. Construct the associated MEM likelihood under the ‘false’ assumption that the components of \mathbf{X} are measured with independent error. That is, assume that \mathbf{W} is observed in place of \mathbf{X} where $\mathbf{W} | \mathbf{X} \sim \mathcal{N}(\mathbf{X}, \mathcal{D}_{\{\sigma_U^2\}})$ with $\sigma_U^2 = (\sigma_{U,1}^2, \dots, \sigma_{U,p}^2)$. The resulting likelihood depends on $\boldsymbol{\theta}$ and σ_U^2 and is denoted $L_{\text{MEM}}(\boldsymbol{\theta}, \sigma_U^2)$. Note that even though $L_{\text{MEM}}(\boldsymbol{\theta}, \sigma_U^2)$ is derived under a measurement error model assumption, it is calculated from the error-free data (\mathbf{X}_i, Y_i) , $i = 1, \dots, n$.

3. Replace $\boldsymbol{\theta}$ in $L_{\text{MEM}}(\boldsymbol{\theta}, \boldsymbol{\sigma}_{\text{U}}^2)$ with an estimate $\widehat{\boldsymbol{\theta}}$, resulting in the pseudo-profile likelihood $\widehat{L}_{\text{pMEM}}(\boldsymbol{\sigma}_{\text{U}}^2) = L_{\text{MEM}}(\widehat{\boldsymbol{\theta}}, \boldsymbol{\sigma}_{\text{U}}^2)$. Note that $\widehat{\boldsymbol{\theta}}$ is an estimator for $\boldsymbol{\theta}$ calculated from the observed data without regard to the ‘false’ measurement error assumption, e.g., $\widehat{\boldsymbol{\theta}}$ could be the maximum likelihood estimator from $L_{\text{TRUE}}(\boldsymbol{\theta})$.
4. Reexpress the pseudo-profile likelihood $\widehat{L}_{\text{pMEM}}(\boldsymbol{\sigma}_{\text{U}}^2)$ in terms of precision (or square-root precision) $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ where $\lambda_j = 1/\sigma_{\text{U},j}^2$ (or $\lambda_j = 1/\sigma_{\text{U},j}$), resulting in the MEM selection likelihood $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$.

$\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ is maximized subject to: $\lambda_j \geq 0$, $j = 1, \dots, p$; and $\sum_j \lambda_j \leq \tau$. Setting the tuning parameter $\tau < \infty$ in the latter constraint ensures that the harmonic mean of the measurement error variances (or standard deviations) is $\geq p/\tau > 0$. This is how the approach ‘forces’ measurement error into the likelihood. Maximizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ subject to the constraint ensures that the measurement error is distributed to predictors optimally in the sense of diminishing the likelihood the least. The idea is that null predictors tend to be assigned large measurement error variances, whereas informative predictors tend to be assigned relatively small ones. Thus maximizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ plays the role of the oracle. The constrained maximum can have some $\widehat{\lambda}_j = 0$ for small τ , indicating that the data are consistent with predictor X_j containing infinite measurement error, and therefore is uninformative. Note that $\widehat{\lambda}_j = 0 \implies \widehat{\sigma}_{\text{U},j}^2 = 1/0 = \infty$, see (2.6).

Standardization was not mentioned previously. Doing so facilitates interpretation, numerical calculation, and comparison with other estimators, thus we assume henceforth that predictor variables are centered and scaled to unit variance. Viewed from the measurement error perspective, standardization ensures that the measurement error variances referred to in the development in Section 3, $\sigma_{\text{U},1}^2, \dots, \sigma_{\text{U},p}^2$, are all on the same scale.

The four steps broadly define the approach. In particular applications it may make sense to restrict to a partial/pseudo likelihood as illustrated in the next section. Regardless, the resulting ‘likelihood’ will generally not be a true likelihood; however, we use the term accepting the abuse of terminology in exchange for brevity. In the next section we illustrate the approach in the

familiar context of linear regression, which also serves to justify the approach since we show it leads to the LASSO. In Section 2.4 we show that the approach leads to a useful new variable selection method for kernel-based classification rules.

2.3 Linear Model Selection Likelihoods

2.3.1 Linear Regression

We start with the assumption that (Y, \mathbf{X}) is multivariate normal with mean $(\mu_Y, \boldsymbol{\mu}_X^T)^T$ and partitioned variance matrix $(V_Y, \mathbf{V}_{XY}^T, \mathbf{V}_{XY}, \mathbf{V}_X)$. Under the ‘false’ assumption that the predictor is measured with error, the distribution is multivariate normal with the same mean, but with partitioned variance matrix $(V_Y, \mathbf{V}_{XY}^T, \mathbf{V}_{XY}, \mathbf{V}_X + \mathcal{D}_{\{\sigma_U^2\}})$. Thus

$$\begin{aligned} L_{\text{TRUE}}(\boldsymbol{\theta}) &= \prod_i \Psi \left(Y_i, \mathbf{X}_i; (\mu_Y, \boldsymbol{\mu}_X^T)^T, (V_Y, \mathbf{V}_{XY}^T, \mathbf{V}_{XY}, \mathbf{V}_X) \right), \\ L_{\text{MEM}}(\boldsymbol{\theta}, \sigma_U^2) &= \prod_i \Psi \left(Y_i, \mathbf{X}_i; (\mu_Y, \boldsymbol{\mu}_X^T)^T, (V_Y, \mathbf{V}_{XY}^T, \mathbf{V}_{XY}, \mathbf{V}_X + \mathcal{D}_{\{\sigma_U^2\}}) \right), \end{aligned}$$

where $\Psi(y, \mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Omega})$ is the multivariate normal density with mean $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Omega}$, and $\boldsymbol{\theta} = (\mu_Y, \boldsymbol{\mu}_X, V_Y, \mathbf{V}_{XY}, \text{vech}(\mathbf{V}_X))$. We set $\hat{\boldsymbol{\theta}}$ equal to the sample moments of (Y_i, \mathbf{X}_i) , $i = 1, \dots, n$, i.e., $\hat{\boldsymbol{\theta}} = (\hat{\mu}_Y, \hat{\boldsymbol{\mu}}_X, \mathbf{V}_{XY}, \hat{\mathbf{V}}_{XY}, \text{vech}(\hat{\mathbf{V}}_X))$.

Then with $\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) = (\hat{\mathbf{V}}_X + \mathcal{D}_{\{\lambda^m\}}^{-1})^{-1} \hat{\mathbf{V}}_{XY}$, where $\boldsymbol{\lambda}^m = (\lambda_1^m, \dots, \lambda_p^m)$, and $m = 1$ for $\lambda_j = 1/\sigma_{U,j}^2$ and $m = 2$ for $\lambda_j = 1/\sigma_{U,j}$, it follows that

$$\hat{L}_{\text{SEL}}(\boldsymbol{\lambda}) \propto \hat{L}_{\text{SEL}_1} \hat{L}_{\text{SEL}_2} \hat{L}_{\text{SEL}_3},$$

where

$$\begin{aligned}
\widehat{L}_{\text{SEL1}} &= \left\{ \widehat{V}_Y - \widehat{V}_{\text{XY}}^T \widehat{\beta}(\boldsymbol{\lambda}) \right\}^{-n/2}, \\
\widehat{L}_{\text{SEL2}} &= \exp \left(-\frac{n}{2} \left[\frac{\widehat{V}_Y - \widehat{V}_{\text{XY}}^T \widehat{\beta}(\boldsymbol{\lambda}) + \left\{ \widehat{\beta}(\boldsymbol{\lambda})^T \widehat{V}_X \widehat{\beta}(\boldsymbol{\lambda}) - \widehat{V}_{\text{XY}}^T \widehat{\beta}(\boldsymbol{\lambda}) \right\}}{\widehat{V}_Y - \widehat{V}_{\text{XY}}^T \widehat{\beta}(\boldsymbol{\lambda})} \right] \right), \\
\widehat{L}_{\text{SEL3}} &= \left[\det \left\{ \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \right\} \right]^{n/2} \exp \left[-\frac{n}{2} \text{tr} \left\{ \widehat{V}_X \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \right\} \right]. \quad (2.6)
\end{aligned}$$

Factoring \widehat{L}_{SEL} as in (2.6) facilitates discussion of the linear model in Section 2.3.2.

Although $\left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1}$ is more succinct, when a component of $\boldsymbol{\lambda}$ is 0, we use either of the equivalent expressions, $\left(\mathbf{I} + \mathcal{D}_{\{\lambda^m\}} \widehat{V}_X \right)^{-1} \mathcal{D}_{\{\lambda^m\}}$ or $\mathcal{D}_{\{\lambda^m\}} \left(\mathbf{I} + \widehat{V}_X \mathcal{D}_{\{\lambda^m\}} \right)^{-1}$ (compare to (2.4) and (2.5)). These identities imply that

$$\begin{aligned}
\widehat{\beta}(\boldsymbol{\lambda}) &= \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \widehat{V}_{\text{XY}} = \mathcal{D}_{\{\lambda^m\}} \left(\mathbf{I} + \widehat{V}_X \mathcal{D}_{\{\lambda^m\}} \right)^{-1} \widehat{V}_{\text{XY}} \\
&= \left(\mathbf{I} + \mathcal{D}_{\{\lambda^m\}} \widehat{V}_X \right)^{-1} \mathcal{D}_{\{\lambda^m\}} \widehat{V}_{\text{XY}}, \quad (2.7)
\end{aligned}$$

from which it is apparent that $\lambda_j = 0$ implies that the j^{th} component of $\widehat{\beta}(\boldsymbol{\lambda}) = 0$. The simple identity $\mathcal{D}_{\{\mathbf{v}_1\}} \mathbf{v}_2 = \mathcal{D}_{\{\mathbf{v}_2\}} \mathbf{v}_1$ shows that $\widehat{\beta}(\boldsymbol{\lambda}) = \mathcal{D}_{\{\boldsymbol{\Delta}\}} \boldsymbol{\lambda}^m$ where $\boldsymbol{\Delta} = \boldsymbol{\Delta}(\boldsymbol{\lambda}) = \left(\mathbf{I} + \widehat{V}_X \mathcal{D}_{\{\lambda^m\}} \right)^{-1} \widehat{V}_{\text{XY}}$. Under common distributional assumptions on Y_1, \dots, Y_n , the components of $\boldsymbol{\Delta}$ are nonzero almost surely. In this case it follows that $\boldsymbol{\lambda}^m = \mathcal{D}_{\{\boldsymbol{\Delta}\}}^{-1} \widehat{\beta}(\boldsymbol{\lambda})$ almost surely, and thus the j^{th} component of $\widehat{\beta}(\boldsymbol{\lambda}) = 0$ if and only if $\lambda_j = 0$.

We now study MEM selection likelihood method for linear regression in the case that λ_j is parameterized in terms of precisions ($m = 1$). The main results in the following section are: i) maximizing $\widehat{L}_{\text{SEL1}}$ is a convex optimization problem that yields an estimator equivalent to the LASSO; ii) maximizing the full-conditional likelihood $\widehat{L}_{\text{SEL1}} \widehat{L}_{\text{SEL2}}$ with $m = 1$ is not equivalent to the LASSO, and in terms of variance per shrinking bias, can be less variable than the LASSO; and iii) maximizing the full likelihood $\widehat{L}_{\text{SEL1}} \widehat{L}_{\text{SEL2}} \widehat{L}_{\text{SEL3}}$ can never result in $\widehat{\lambda}_j = 0$ and thus is not an option for variable selection (but is an option for shrinkage only). We use the acronym

MESSO (Measurement Error Shrinkage and Selection Operator) to identify estimators derived from the selection likelihoods in Section 2.3.1.

2.3.2 Relationship to LASSO

In the supplemental Appendix Section 2.6.1 we prove that constrained maximization of $\widehat{L}_{\text{SEL}_1}$ in (2.6) results in LASSO solution paths as functions of τ . The proof establishes convexity of $\widehat{\sigma}^2(\boldsymbol{\lambda}) = \left(\widehat{L}_{\text{SEL}_1}\right)^{-2/n}$ and equivalence of KKT conditions. Note that although the MESSO selection likelihood in Section 2.3 was derived starting with an assumption of multivariate normality, the equivalency to the LASSO does not depend on that assumption.

We now consider the relationship of LASSO to other MESSO versions obtained by maximizing the full-conditional likelihood, $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}$, and the full likelihood $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}\widehat{L}_{\text{SEL}_3}$ for $\lambda_j = 1/\sigma_j^2$ and $\lambda_j = 1/\sigma_j$. A detailed comparison is beyond the scope of this paper. However, we make a

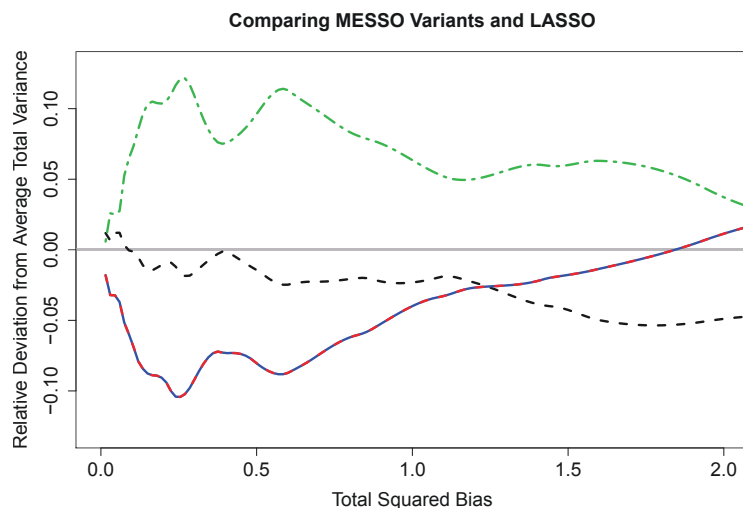


Figure 2.1 Plot of relative deviations from the average total variance versus total squared bias. LASSO (red-dashed); MESSO estimator from $\widehat{L}_{\text{SEL}_1}$ (solid blue); full-conditional MESSO estimator from $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}$ (black dashed); and the full-unconditional MESSO estimator from $\widehat{L}_{\text{SEL}_1}\widehat{L}_{\text{SEL}_2}\widehat{L}_{\text{SEL}_3}$ (green dot-dash). Because of their equivalency the $\widehat{L}_{\text{SEL}_1}$ -MESSO and LASSO lines overlap and appear as a single alternating red/blue dashed line.

few brief, relevant observations. Figure 2.1 displays plots of relative deviations from average total prediction error variance versus total squared prediction error bias from a small simulation study comparing LASSO and the three MESSO estimators when $\lambda_j = 1/\sigma_j^2$. That is, the plot displays $\{\text{TV-Ave}(\text{TV})\}/\text{Ave}(\text{TV})$ versus TB, where: $\text{TB}=\{\widehat{E}(\widehat{\beta}) - \beta\}^T \Omega \{\widehat{E}(\widehat{\beta}) - \beta\}$; $\text{TV}=\text{tr}\{\widehat{\text{Var}}(\widehat{\beta})\Omega\}$; Ω is the predictor covariance matrix; Ave(TV) is TV averaged over the three unique estimators; and \widehat{E} and $\widehat{\text{Var}}$ denote Monte Carlo expectation and variance. The plot allows differentiation of the estimators based on total variance for equivalent total squared bias. Data were simulated according to a design in [74]: $Y_i = \mathbf{X}_i^T \beta + \sigma \epsilon_i$, $i = 1, \dots, n$, where \mathbf{X}_i are iid $\mathcal{N}(\mathbf{0}_{p \times 1}, \text{AR1}(\rho))$, ϵ_i are iid $\mathcal{N}(0, 1)$, with $n = 20$, $p = 8$, $\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$, $\rho = .5$, and $\sigma = 3$. This model calls for a mix of selection and shrinkage. Figure 2.1 shows that the LASSO is less variable (as measured by $\text{tr}\{\widehat{\text{Var}}(\widehat{\beta})\Omega\}$) for small total squared bias (as measured by $\{\widehat{E}(\widehat{\beta}) - \beta\}^T \Omega \{\widehat{E}(\widehat{\beta}) - \beta\}$), but not for larger values of TB. For reference, we obtained $\text{TB}=1.27$ for the LASSO estimator tuned using 10-fold cross validation, which corresponds very closely to the value of TB for which the black-dashed and red-blue curves intersect. Thus in a region of total squared bias of known practical importance, the LASSO and the full-conditional MESSO estimator have similar total variance.

Examination of $\widehat{L}_{\text{SEL}_3}$ in (2.6) reveals that if any $\lambda_j = 0$, then the determinant term is zero and the maximizer of the full likelihood $\widehat{L}_{\text{SEL}_1} \widehat{L}_{\text{SEL}_2} \widehat{L}_{\text{SEL}_3}$ never contains zero elements, i.e., selection is not possible. Finally, when λ is parameterized as square-root precisions, $\lambda_j = 1/\sigma_j$, the MESSO optimization is no longer convex, and results in greater selection.

Our comparison to the LASSO is not because we consider MESSO to be a competitor to the LASSO for linear models, but rather to establish the viability of the new approach in a familiar context. In the next section we show that it leads

to a promising method of variable selection for nonparametric classification.

2.4 Nonparametric Classification

Classification has a long and important history in statistics. Several methods are widely used: Fisher’s linear discriminant analysis [LDA, 19]; logistic regression [52]; support vector machine [13, 76]; and boosting [20]. LDA is the estimated Bayes rule under within-class normality and equal covariance matrices [3], whereas with unequal covariances the estimated Bayes rule is quadratic discriminant analysis (QDA).

A nonparametric Bayes rule obtained by replacing conditional densities in the theoretical Bayes rule with kernel density estimates [77] is known as kernel density-based discriminant analysis (KDA) [12, 27]. See [14] and [36] for recent reviews of KDA.

With multiple predictors, variable selection is important. Some predictors may not carry useful information and their inclusion can deteriorate performance of an estimated classification rule. [6] studied how dimension affects performance and showed that LDA behaves like random guessing when the ratio of p to n is large. Thus motivated, they proposed an independence rule by using a diagonal variance/covariance matrix. [15] selected a set of important predictors via thresholding, and applied the independence rule to the selected predictors, calling the method an *annealed independence rule*. [51] proposed variable selection for LDA by formulating it as a regression problem. [10] studied a direct estimation approach for variable selection in LDA. For other methods of variable selection for classification, see [33], [81], [63] and references therein.

2.4.1 Selection Likelihoods for Classification

The true model is defined via the conditional distributions $\mathbf{X} \mid (Y = 0) \sim f_0(\mathbf{x})$, $\mathbf{X} \mid (Y = 1) \sim f_1(\mathbf{x})$, and the marginal probability $P(Y = 1) = \pi_1 = 1 - \pi_0$. It follows that

$$\begin{aligned} P(Y = 1 \mid \mathbf{X} = \mathbf{x}) = P_{\text{TRUE}}(\mathbf{x}; \pi_0, \pi_1, f_0, f_1) &= \pi_1 f_1(\mathbf{x}) / (\pi_0 f_0(\mathbf{x}) + \pi_1 f_1(\mathbf{x})) \\ &= \left\{ 1 + \frac{\pi_0 f_0(\mathbf{x})}{\pi_1 f_1(\mathbf{x})} \right\}^{-1}. \end{aligned}$$

Under the ‘false’ measurement error model assumption $\mathbf{W} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{X}, \mathbf{D}\sigma_{\mathbf{U}}^2)$, the conditional distributions are, using $\phi(\cdot)$ to denote the standard normal density,

$$\mathbf{W} \mid (Y = k) \sim f_k(\mathbf{x}, \boldsymbol{\lambda}) = \int \dots \int \prod_{j=1}^p \frac{1}{\sigma_{\mathbf{U},j}} \phi\left(\frac{x_j - t_j}{\sigma_{\mathbf{U},j}}\right) f_k(\mathbf{t}) dt_1 \dots dt_p.$$

It follows that the MEM selection likelihood is determined by

$$P(\mathbf{x}; \pi_0, \pi_1, f_0, f_1, \boldsymbol{\lambda}) = \left\{ 1 + \frac{\pi_0}{\pi_1} R(\mathbf{x}; f_0, f_1, \boldsymbol{\lambda}) \right\}^{-1}, \quad (2.8)$$

where:

$$\begin{aligned} R(\mathbf{x}; f_0, f_1, \boldsymbol{\lambda}) &= \frac{\int \dots \int \prod_{j=1}^p \lambda_j^c \phi\left(\lambda_j^c(x_j - t_j)\right) f_0(\mathbf{t}) dt_1 \dots dt_p}{\int \dots \int \prod_{j=1}^p \lambda_j^c \phi\left(\lambda_j^c(x_j - t_j)\right) f_1(\mathbf{t}) dt_1 \dots dt_p} \\ &= \frac{\int \dots \int \prod_{j=1}^p \phi\left(\lambda_j^c(x_j - t_j)\right) dF_0(\mathbf{t})}{\int \dots \int \prod_{j=1}^p \phi\left(\lambda_j^c(x_j - t_j)\right) dF_1(\mathbf{t})}; \end{aligned}$$

$c = 1/2$ when $\lambda_j = 1/\sigma_{\mathbf{U},j}^2$, and $c = 1$ when $\lambda_j = 1/\sigma_{\mathbf{U},j}$ respectively; $F_k(\cdot)$ is the distribution corresponding to $f_k(\cdot)$; and the common factor $\prod_{j=1}^p \lambda_j^c$ is deleted in the final ratio. For our work on classification we take $c = 1$.

Replacing π_1 with $\hat{\pi}_1 = \bar{Y}$, π_0 with $\hat{\pi}_0 = 1 - \hat{\pi}_1$, and F_0 and F_1 with their nonparametric MLEs (empirical distribution functions) in $P(\mathbf{x}; \pi_0, \pi_1, f_0, f_1, \boldsymbol{\lambda})$ in (2.8) results in

$$\hat{P}(\mathbf{x}; \boldsymbol{\lambda}) = \left\{ 1 + \frac{\hat{\pi}_0}{\hat{\pi}_1} \hat{R}(\mathbf{x}; \boldsymbol{\lambda}) \right\}^{-1}, \quad (2.9)$$

where

$$\begin{aligned}
\widehat{R}(\mathbf{x}; \boldsymbol{\lambda}) &= \frac{n_0^{-1} \sum_{i:Y_i=0}^n \prod_{j=1}^p \phi \left(\lambda_j^c(x_j - X_{j,i}) \right)}{n_1^{-1} \sum_{i:Y_i=1}^n \prod_{j=1}^p \phi \left(\lambda_j^c(x_j - X_{j,i}) \right)} \\
&= \frac{n_0^{-1} \sum_{i:Y_i=0}^n \prod_{j:\lambda_j \neq 0}^p \phi \left(\lambda_j^c(x_j - X_{j,i}) \right)}{n_1^{-1} \sum_{i:Y_i=1}^n \prod_{j:\lambda_j \neq 0}^p \phi \left(\lambda_j^c(x_j - X_{j,i}) \right)}, \tag{2.10}
\end{aligned}$$

with $n_k = n\widehat{\pi}_k$. The second expression in (2.10) is displayed to emphasize the fact that for every $\lambda_j = 0$ there is a common factor $\phi(0)$ in the numerator and denominator of $\widehat{R}(\mathbf{x}; \boldsymbol{\lambda})$ that cancels, in which case $\widehat{R}(\mathbf{x}; \boldsymbol{\lambda})$ *does not depend on* $X_{j,1}, \dots, X_{j,n}$.

Because Y_i is binary a suitable MEM selection likelihood for this model is

$$\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda}) = \sum_{i=1}^n Y_i \log(\widehat{P}(\mathbf{X}_i; \boldsymbol{\lambda})) + (1 - Y_i) \log(1 - \widehat{P}(\mathbf{X}_i; \boldsymbol{\lambda})),$$

resulting in the optimization problem

$$\begin{aligned}
&\max_{\lambda_1, \lambda_2, \dots, \lambda_p} \sum_{i=1}^n Y_i \log(\widehat{P}(\mathbf{X}_i; \boldsymbol{\lambda})) + (1 - Y_i) \log(1 - \widehat{P}(\mathbf{X}_i; \boldsymbol{\lambda})) \tag{2.11} \\
&\text{subject to} \quad \sum_{j=1}^p \lambda_j = \tau; \quad \lambda_j \geq 0, \quad j = 1, 2, \dots, p;
\end{aligned}$$

Maximizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ subject to the constraints $\lambda_j \geq 0$, $j = 1, \dots, p$, and $\sum_j \lambda_j \leq \tau$ results in some $\widehat{\lambda}_j = 0$ when τ is small enough. In this case $\widehat{R}(\mathbf{x}; \boldsymbol{\lambda})$ is a ratio of kernel density estimates of only those components of \mathbf{X} corresponding to nonzero λ_j . That is, $\widehat{\lambda}_j = 0$ implies that the j^{th} predictor is selected out of the model, just as in the linear model. Finally we note that (2.9) corresponds to the estimated Bayes rule classifier. For an estimated maximum likelihood classifier set $\widehat{\pi}_0 = \widehat{\pi}_1 = 1/2$. The tuning parameter τ can be selected by minimizing classification error, or the likelihood of an independent tuning set, or via cross validation. In our numerical work we used cross validation.

2.4.2 Multicategory SKDA

SKDA is readily extended to multicategory classification. In multicategory classification with K classes, we label the response as $1, 2, \dots, K$. The training data set is $\{(\mathbf{X}_i, Y_i) : i = 1, 2, \dots, n\}$ with $\mathbf{X}_i \in \mathbb{R}^p$ and $Y_i \in \{1, 2, \dots, K\}$.

With a slight abuse of notation define $\tilde{f}_k(\mathbf{x}; \boldsymbol{\lambda}) = \sum_{i: y_i=k} \prod_{j=1}^p K(\lambda_j(x_j - X_{ij}))/n_k$ and $\hat{\pi}_k = n_k/n$, $k = 1, 2, \dots, K$ with n_k being the number of observations in class k and $\hat{P}_k(\mathbf{x}; \boldsymbol{\lambda}) = \hat{\pi}_k \tilde{f}_k(\mathbf{x}; \boldsymbol{\lambda}) / (\sum_{m=1}^K \hat{\pi}_m \tilde{f}_m(\mathbf{x}; \boldsymbol{\lambda}))$. The log-likelihood is $\sum_{i=1}^n \log(\hat{P}_{y_i}(\mathbf{X}_i; \boldsymbol{\lambda}))$ and the multicategory SKDA (MSKDA) requires solving

$$\begin{aligned} & \max_{\lambda_1, \lambda_2, \dots, \lambda_p} \sum_{i=1}^n \log(\hat{P}_{y_i}(\mathbf{X}_i; \boldsymbol{\lambda})) & (2.12) \\ \text{subject to} & \sum_{j=1}^p \lambda_j = \tau, \quad \lambda_j \geq 0, \quad j = 1, 2, \dots, p. \end{aligned}$$

The MSKDA rule for $\mathbf{X} = \mathbf{x}$ is given by $\operatorname{argmax}_k \hat{P}_k(\mathbf{x}; \hat{\boldsymbol{\lambda}})$. As defined $\hat{P}_k(\mathbf{x}; \boldsymbol{\lambda})$ yields the estimated Bayes rule classifier. For the maximum likelihood classifier set all $\hat{\pi}_k = 1/K$.

2.4.3 Simulation Studies

In this section, we evaluate the performance of SKDA and MSKDA via simulation, comparing them with LDA, sparse LDA (SLDA) [51], and KDA (resp. MKDA, the multicategory extension of KDA). For the binary case, we also compare with the SCAD penalized logistic regression [16] and SCAD penalized support vector machine (SVM) [86]. Note that only LDA does not depend on a tuning parameter. The other methods require the determination of a single tuning parameter. In their most general form, KDA and MKDA each require the estimation of p tuning parameters (bandwidths). In our numerical work we replace the p bandwidths with a single common bandwidth. This simplification is necessary given the large dimensions studied.

We determine tuning parameters using 10-fold cross validation coupled with a grid search. We use classification error as the selection criterion for SLDA as done in [51] and SCAD SVM,

and the log-likelihood as the selection criterion for the kernel-based methods and SCAD logistic regression.

We included π_k , the prior probability of class k , and its estimator $\hat{\pi}_k$ in the proposed methods and their theoretical development for generality. However in the simulation examples, we consider simulation settings with equal prior probabilities for all classes and thus we set all $\hat{\pi}_k$ to be $1/K$. The same simplification is used in the real data examples, thus we study the maximum likelihood classification rule rather than the Bayes rule.

Our simulation study design requires generating multivariate observations of which only a subset of the (correlated) variables appear in the Bayes classifier. In the supplemental Appendix Section 2.6.2 we give a proposition explaining how to accomplish the data generation.

2.4.3.1 Two-group, Normal Linear Discriminant Model

Our first simulation study is two-group classification for which LDA and SLDA are specifically designed: $\mathbf{X}_{p \times 1} | (Y = k) \sim \text{MVN}_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ for $k = 0, 1$, $p = 10$ predictors, and $Y \sim \text{Bernoulli}(0.5)$. The means were set as $\mu_0 = (-1, 1, \mathbf{0}_{8 \times 1}^T)^T$ and $\mu_1 = (1, -1, \mathbf{0}_{8 \times 1}^T)^T$, and $\boldsymbol{\Sigma}$ was the equicorrelation matrix with parameter ρ , for levels of ρ given by 0, 0.3, and 0.6. A test set of size 20,000 was fixed to evaluate the performance of the classification methods. Training sets of size $n = 200$ are used in the simulation, and 10-fold cross validation was used to select the tuning parameters for KDA, SKDA, SLDA, SCAD-Logistic, and SCAD-SVM. Results for 100 simulated data sets are in Table 2.1 and Figure 2.2.

Table 2.1 gives predictor selection frequencies by SKDA, SLDA, SCAD logistic regression, and SCAD SVM over 100 repetitions. Note that, only X_1 and X_2 are important for this example according to Proposition 1 in Section 2.6.2. Table 2.1 shows that the new nonparametric method SKDA performs competitively in terms of variable selection, even though the data generation strongly favors LDA and SLDA.

We plot the two important predictors, X_1 versus X_2 , for one training sample in the left panels of Figure 2.2 with classes represented by blue “o” and red “+.” Box plots of test errors

Table 2.1 Predictor selection frequencies for SKDA and SLDA for the study in Section 2.4.3.1.

ρ		Frequency									
		X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0	SKDA	100	100	10	14	6	15	11	9	9	6
	SLDA	100	100	1	1	3	1	2	2	3	3
	SCADlogist	100	100	13	11	10	10	7	12	12	11
	SCADsvm	100	100	13	14	11	15	18	12	9	10
0.3	SKDA	100	100	7	15	16	10	10	7	8	10
	SLDA	100	100	2	0	0	1	1	0	0	0
	SCADlogist	100	100	10	12	7	5	6	10	11	14
	SCADsvm	100	100	6	6	5	5	9	5	7	9
0.6	SKDA	100	100	16	12	9	8	6	14	13	10
	SLDA	100	100	0	0	0	0	0	0	0	0
	SCADlogist	100	100	0	0	1	0	2	1	0	3
	SCADsvm	100	100	0	2	1	1	0	2	0	3

over the independent test set across 100 repetitions are in the right panels. The top, middle, and bottom rows correspond to $\rho = 0, 0.3$ and 0.6 , respectively. As the setting is ideal for LDA, and SLDA, and only marginally less so for SCAD-Logistic and SCAD-SVM, these methods perform very well with SLDA having a slight edge. KDA neither exploits the marginal normality of the data, nor does it do variable selection, and thus performs poorly. However, although SKDA does not exploit the marginal normality, its performance in terms of classification error and selection is competitive.

2.4.3.2 Two-Group Mixtures of Equicorrelated Normals

This is another binary example with $Y \sim \text{Bernoulli}(0.5)$. In this example, we consider an AR(1) type correlation $\Sigma = (\sigma_{ij})$ with $\sigma_{ij} = \rho^{|i-j|}$. We partition predictors with $p_1 = 2$ and $p_2 = p - 2$. Predictors are generated as $\mathbf{X} \mid (Y = 0) \sim N(\mathbf{0}, \Sigma)$ and $\mathbf{X} \mid (Y = 1) \sim 0.5\mathcal{N}((\boldsymbol{\theta}_1^T \Sigma_{11}, \boldsymbol{\theta}_1^T \Sigma_{12})^T, \Sigma) + 0.5\mathcal{N}(-(\boldsymbol{\theta}_1^T \Sigma_{11}, \boldsymbol{\theta}_1^T \Sigma_{12})^T, \Sigma)$ with $\boldsymbol{\theta}_1 = (3, -3)^T$. The ‘0’ class is multivariate normal, while the ‘1’ class is a mixture of two multivariate normal distributions. Thus the true classification rule is nonlinear and LDA and SLDA are misspecified in this case. Yet it can be checked that the true classification rule depends only on the first two predictors

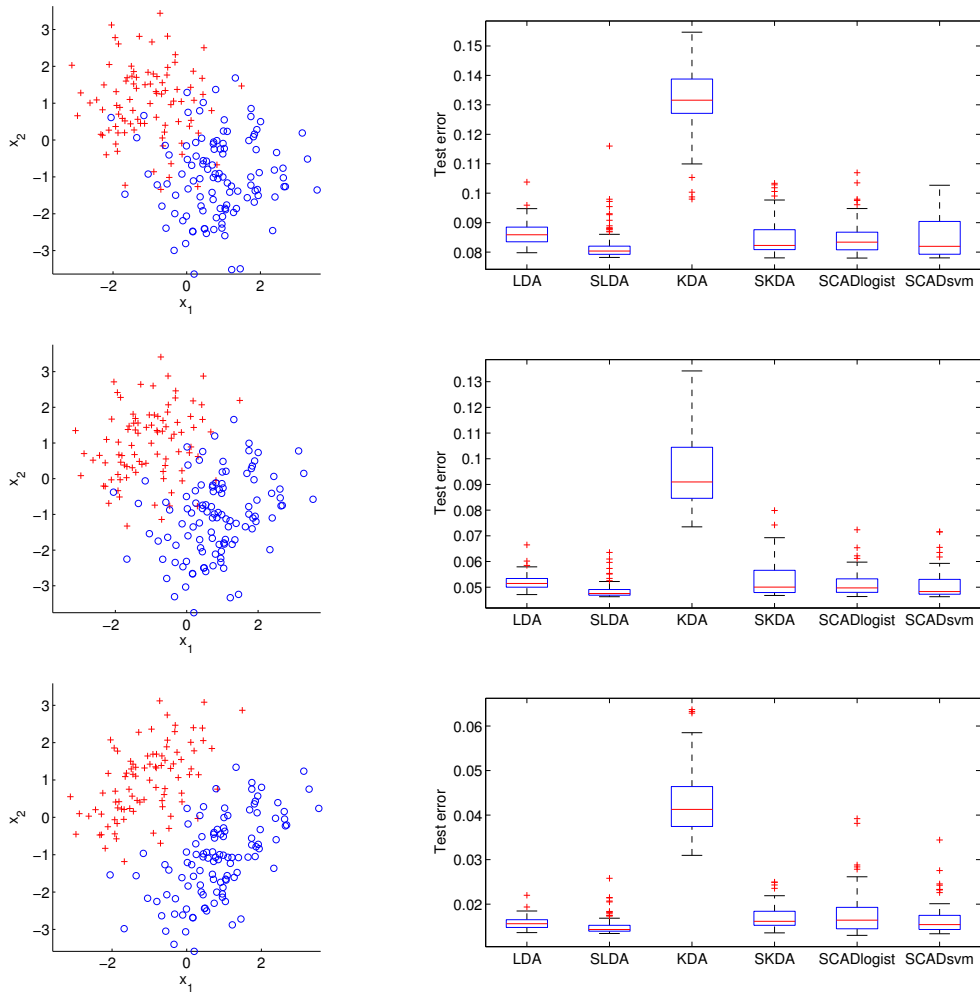


Figure 2.2 Plots of typical training samples and boxplots of test errors for data generated with $\rho = 0$ (top), 0.3 (middle), and 0.6 (bottom), from Section 2.4.3.1.

X_1 and X_2 (according to Proposition 1 in Section 2.6.2). Training sets of size 200 and a test set of size 20,000 are used. We set $p = 10$ and consider three different ρ values 0, 0.3, and 0.6. Results over 100 repetitions are reported in Table 2.2 and Figure 2.3 in a similar way as the previous example.

Table 2.2 indicates that the two important predictors X_1 and X_2 are selected by SKDA for all 100 repetitions while unimportant predictors are selected infrequently. SLDA, SCAD logistic regression, and SCAD SVM fail to select the two important predictors for most repetitions. The

Table 2.2 Predictor selection frequencies for SKDA and SLDA for the study in Section 2.4.3.2.

ρ		Frequency									
		X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0	SKDA	100	100	9	9	8	6	14	10	7	8
	SLDA	0	0	5	4	3	5	6	6	4	8
	SCADlogist	0	0	8	10	9	7	12	11	8	11
	SCADsvm	38	38	50	54	55	56	59	56	53	57
0.3	SKDA	100	100	3	1	3	5	8	6	4	5
	SLDA	0	0	4	5	5	3	6	6	6	9
	SCADlogist	4	0	4	11	10	5	8	11	12	9
	SCADsvm	46	39	52	52	51	54	56	60	60	60
0.6	SKDA	100	100	6	0	4	0	1	2	3	4
	SLDA	2	1	4	5	7	3	4	3	7	6
	SCADlogist	4	6	6	7	9	6	10	4	13	7
	SCADsvm	51	54	46	52	52	53	56	56	60	59

poor performance of these methods is not unexpected in light of the data generation model. Variables X_1 and X_2 from one simulated training data set are plotted in the left panels of Figure 2.3 for three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom). There is no good way to linearly separate the two classes well. Test error results are consistent as shown in the right panels of Figure 2.3. SKDA performs well whereas LDA, SLDA, SCAD logistic regression, and SCAD SVM perform like random guessing.

High dimensional case: We repeated the study changing only $p = 10$ to $p = 50$. Boxplots of the test errors over 100 repetitions are shown in right panels of Figure 2.4 for different methods and three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom). Out of 100 repetitions, the frequency of X_1 and X_2 and average frequency of X_3, \dots, X_{50} being selected by SKDA and SLDA are reported in Table 2.3.

The true classifier in this example is nonlinear, and thus LDA, SLDA, SCAD logistic regression, and SCAD SVM are misspecified. SKDA is nonparametric and thus more flexible. hence it performs well at both variable selection and classification accuracy.

To address a review comment, we increase the dimension further to $p = 100$. For the case

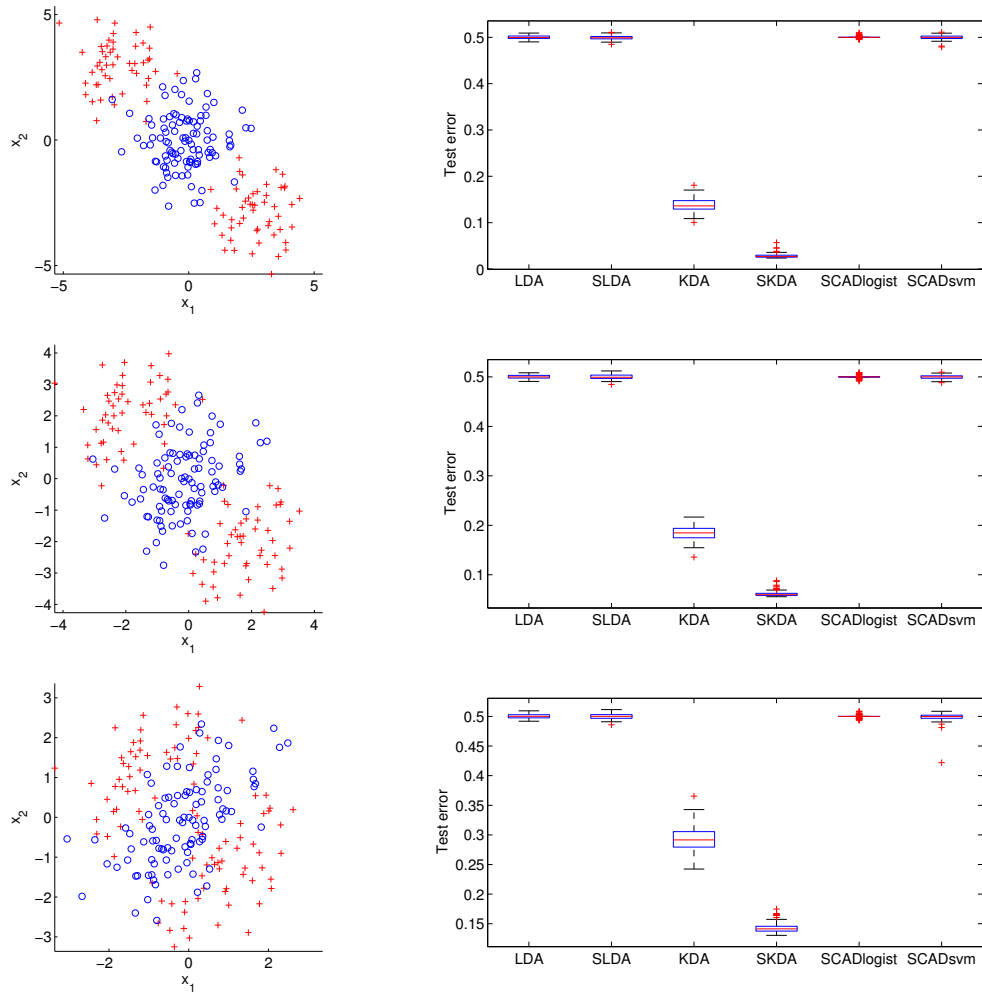


Figure 2.3 Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.2 with $p = 10$ and three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom).

with $\rho = 0$, the SKDA selects the important predictors X_1 and X_2 for all 100 repetitions and the average selection frequency for the other 98 unimportant predictors is 5.2959. The average testing error for the SKDA is 0.0426 with a standard deviation of 0.0095. We expect a similar performance for the cases with $\rho = 0.3$ and $\rho = 0.6$.

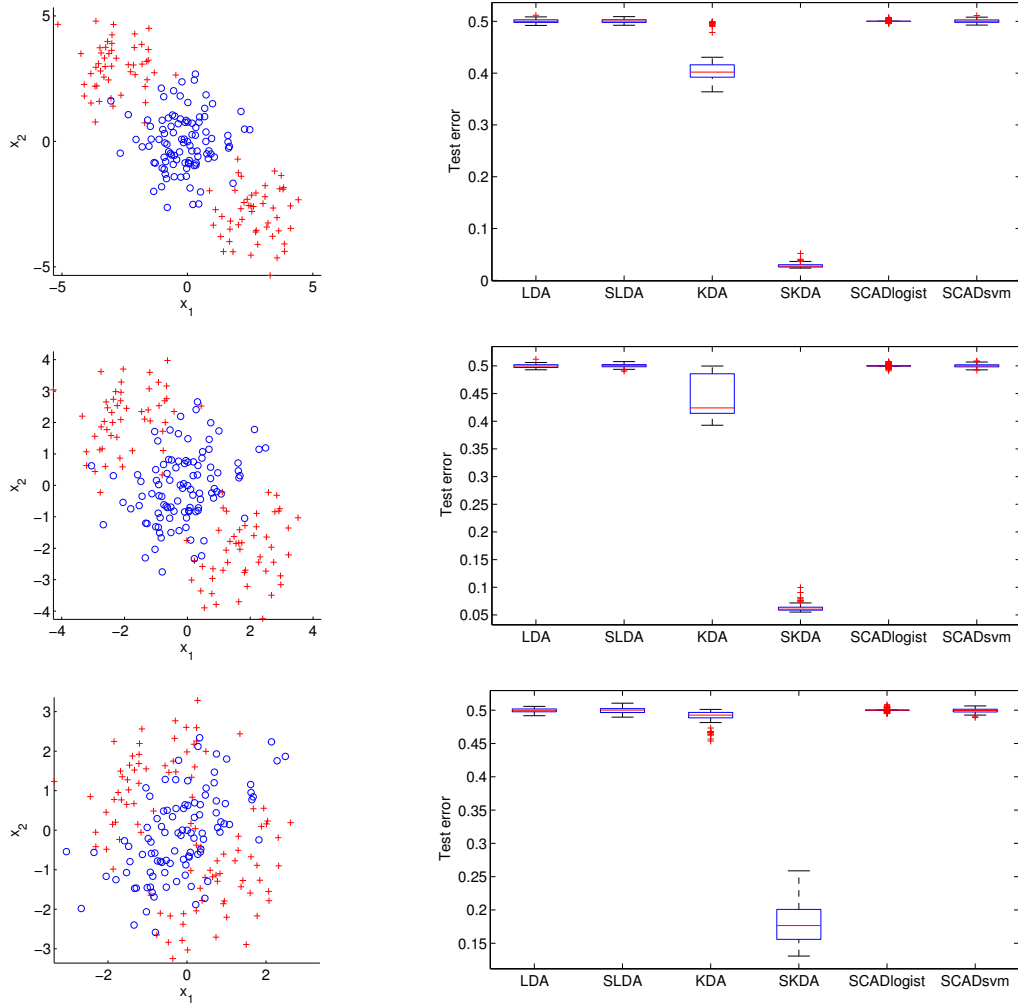


Figure 2.4 Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.2 with $p = 50$ and three values of ρ : 0 (top), 0.3 (middle), and 0.6 (bottom).

2.4.3.3 Three-group, Normal Linear Discriminant Model

This is a 3-class example with $p = 10$ and an AR(1) correlation $\Sigma = (\sigma_{ij})$ with $\sigma_{ij} = \rho^{|i-j|}$. We partition predictors with $p_1 = 2$ and $p_2 = p - 2$. Data are generated in two steps: generate Y uniformly over $\{1, 2, 3\}$. Conditional on Y , predictors are generated as $\mathbf{X} \mid (Y = 1) \sim \mathcal{N}(\mathbf{0}, \Sigma)$; $\mathbf{X} \mid (Y = 2) \sim \mathcal{N}((\boldsymbol{\theta}_{12}^T \Sigma_{11}, \boldsymbol{\theta}_{12}^T \Sigma_{12})^T, \Sigma)$ with $\boldsymbol{\theta}_{12} = (2, 2\sqrt{3})^T$; $\mathbf{X} \mid (Y = 3) \sim \mathcal{N}((\boldsymbol{\theta}_{13}^T \Sigma_{11}, \boldsymbol{\theta}_{13}^T \Sigma_{12})^T, \Sigma)$ with $\boldsymbol{\theta}_{13} = (-12, 2\sqrt{3})^T$. According to Proposition 1 in Section 2.6.2,

Table 2.3 Frequency of X_1 and X_2 and the average frequency of X_3, \dots, X_{50} being selected by SKDA and SLDA.

ρ	Method	(Average) Frequency		
		X_1	X_2	X_3, \dots, X_{50}
0	SKDA	100	100	2.5
	SLDA	1	0	4.2
	SCADlogist	0	0	4.8
	SCADsvm	31	23	44.6
0.3	SKDA	100	100	1.5
	SLDA	1	0	4.4
	SCADlogist	1	1	4.6
	SCADsvm	28	37	41.8
0.6	SKDA	100	100	6.0
	SLDA	1	2	2.1
	SCADlogist	2	0	2.2
	SCADsvm	47	43	45.7

the Bayes classification rule depends only on X_1 and X_2 .

Table 2.4 Predictor selection frequencies for MSKDA for the study in Section 2.4.3.3.

ρ	Frequency									
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0	100	100	1	1	1	0	1	1	0	0
0.3	100	100	0	0	0	1	0	0	0	0
0.6	100	100	1	0	0	0	0	0	0	1

For this 3-class example, training sets of size 300 and a test set of size 30,000 are used. As it is not clear how to extend SLDA to multicategory case, we only compare MSKDA to LDA and MKDA in multicategory examples. Results over 100 repetitions are reported in Table 2.4 and Figure 2.5. In particular, Table 2.4 reports the frequency of each predictor selected by MSKDA. It shows that MSKDA gives very good variable selection performance. Note that, in this example, each class is represented by a multivariate normal distribution. Thus LDA is correctly specified and should do well in terms of classification error. This is confirmed by results

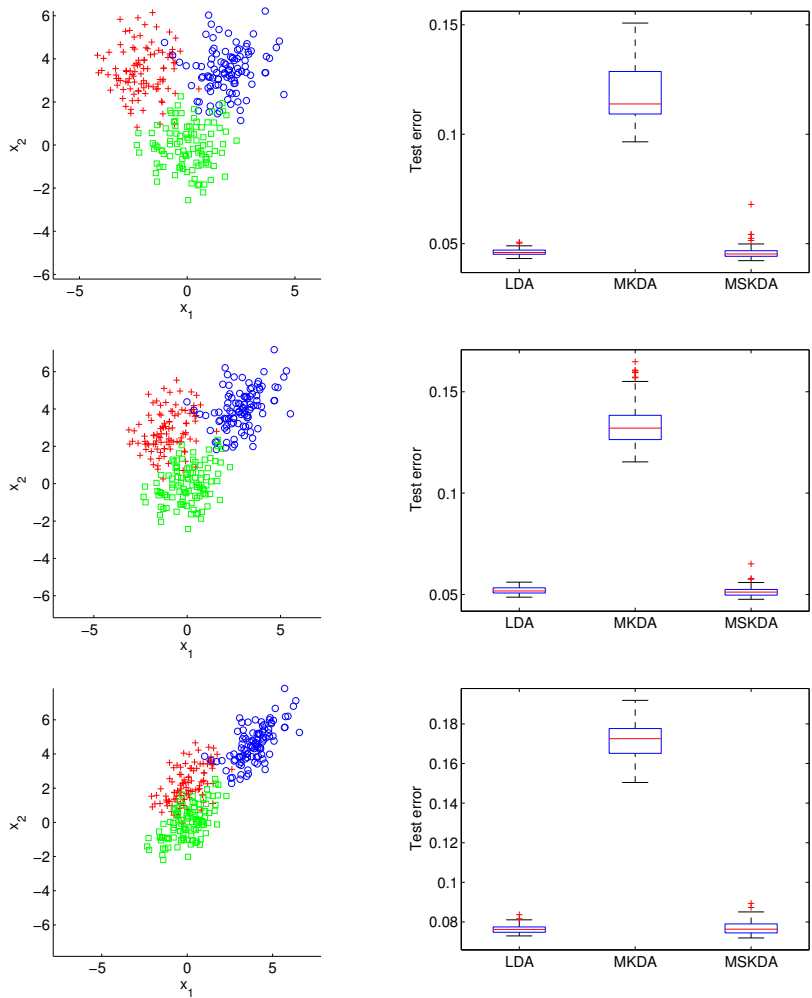


Figure 2.5 Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.3.

in Figure 2.5. MSKDA reduces the test error in comparison to MKDA as a result of its variable selection. Because MSKDA performs variable selection very well, its performance is similar to that of LDA even though it is nonparametric.

To address a reviewer’s query about the performance of the SKDA and MSKDA when some predictors have discrete distributions, we reran the $\rho = 0$ case above replacing the informative continuous predictor X_2 with the ‘new’ informative discrete predictor X_2 generated conditionally on Y according to $X_2 = 3(Y - 1) + U$ where U has a discrete uniform distribution on the set

$\{1, 2, 3, 4, 5\}$, and replacing the noninformative continuous X_7 with the ‘new’ noninformative discrete $X_7 \sim \text{Poisson}(4)$. The selection frequencies for variables X_1, \dots, X_{10} over 100 simulated data sets were 100, 100, 0, 1, 0, 1, 0, 1, 0, 1, respectively (compare to the top row of Table 2.4). A training sample and boxplots of test errors for this discrete case are displayed in Figure 2.6 (compare to the top row of Figure 2.5). These results are promising although far from a thorough study of SKDA and MSKDA with discrete predictors. The latter will likely prove fruitful just as the study of discrete predictors in the framework of kernel regression without variable selection has been; see [58] and [47] and the reference therein.

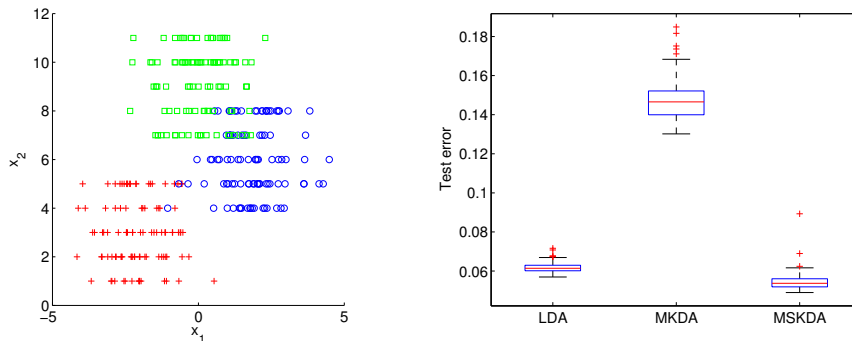


Figure 2.6 Plot of a training sample (left) and boxplots of test errors (right) for the discrete predictor simulation study in Section 2.4.3.3.

2.4.3.4 Three-Group, Mixtures of Equicorrelated Normals

This is another 3-class example with $p = 10$ and an AR(1) correlation $\Sigma = (\sigma_{ij})$ with $\sigma_{ij} = \rho^{|i-j|}$. We also partition predictors with $p_1 = 2$ and $p_2 = p - 2$. Data are generated in two steps: generate Y uniformly over $\{1, 2, 3\}$. Conditional on $Y = k$, predictors are generated as $\mathbf{X} \mid (Y = k) \sim 0.5\mathcal{N}((\boldsymbol{\theta}_{1k}^T \Sigma_{11}, \boldsymbol{\theta}_{1k}^T \Sigma_{12})^T, \Sigma) + 0.5\mathcal{N}(-(\boldsymbol{\theta}_{1k}^T \Sigma_{11}, \boldsymbol{\theta}_{1k}^T \Sigma_{12})^T, \Sigma)$, a mixture of two multivariate Gaussian distributions, for $k = 1, 2, 3$. Here $\boldsymbol{\theta}_{11} = (5, \mathbf{0}_9^T)^T$, $\boldsymbol{\theta}_{12} = (2.5, 5\sqrt{3}/2, \mathbf{0}_8^T)^T$, and $\boldsymbol{\theta}_{13} = (2.5, -5\sqrt{3}/2, \mathbf{0}_8^T)^T$. According to Proposition 1 in Section 2.6.2, the Bayes classification rule depends only on X_1 and X_2 .

Table 2.5 Predictor selection frequencies for MSKDA for the study in Section 2.4.3.4.

ρ	Frequency									
	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
0	100	100	0	0	0	0	0	0	0	0
0.3	100	100	0	0	0	0	0	0	0	0
0.6	100	100	0	0	1	0	0	0	1	3

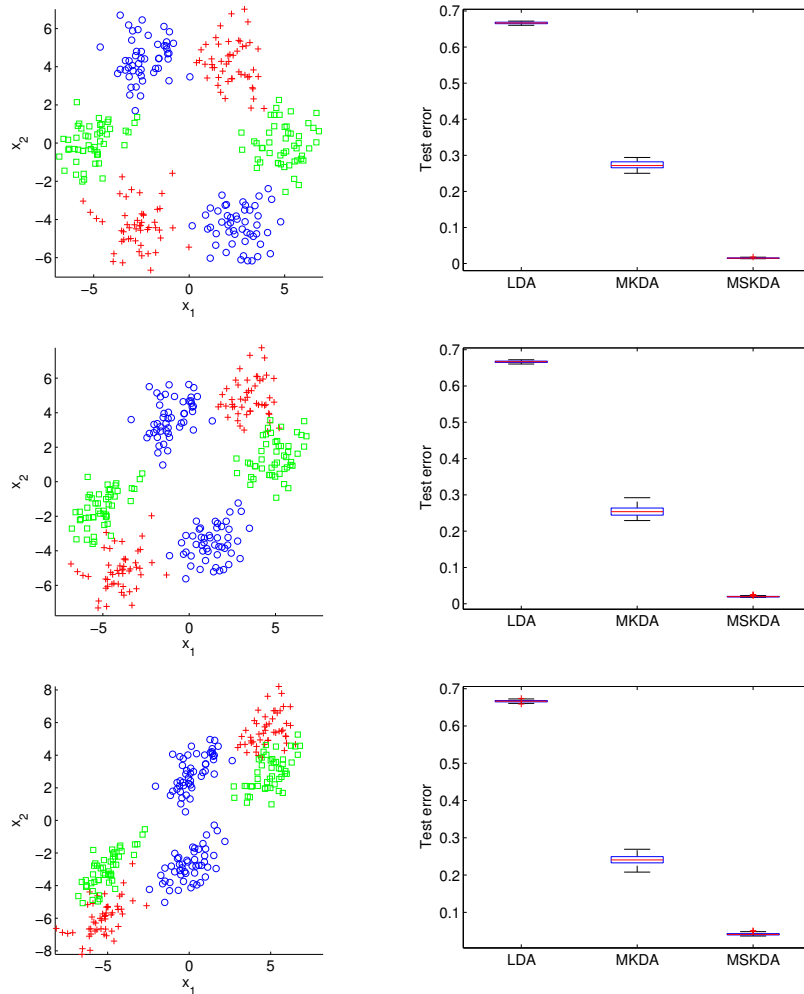


Figure 2.7 Plot of one training sample (left) and boxplots of test errors (right) for the study in Section 2.4.3.4.

Training sets of size 300 and a test set of size 30,000 are used. Results over 100 repetitions are in Table 2.5 and Figure 2.7 in the same format as the previous example. For this data

generation model, LDA is badly misspecified. Consequently LDA performs close to random guessing as seen in by Figure 2.7. MSKDA improves upon MKDA in terms of classification error due to its variable selection ability.

2.4.4 Illustrations with Real Data

2.4.4.1 Wisconsin Breast Cancer Data

We apply SKDA to the Wisconsin Breast Cancer Data (WBCD) containing information on 569 patients, including a binary indicator of tumor malignancy (malignant/benign) and thirty purported predictors.¹ We standardized each predictor to mean zero and variance one. In each repetition, we randomly select 300 observations as the training set and the remaining 269 are used as the test set to report performance by calculating the classification error for each method. Ten-fold cross validation was used for estimating tuning parameters. Test error results over 40 repetitions are reported in Table 2.6 and displayed in Figure 2.8. SKDA had a test error comparable to LDA, KDA, SCAD logistic regression, and SCAD SVM but a smaller test error than SLDA.

Table 2.6 Test error summary for 40 random splits of the WBCD data.

	LDA	SLDA	KDA	SKDA	SCADlogit	SCADsvm
Test Error Average	0.049	0.130	0.043	0.045	0.045	0.038
Test Error Std Dev	0.011	0.020	0.010	0.011	0.014	0.010

The average number of predictors selected by SKDA, SLDA, SCAD logistic regression, and SCAD SVM are 3.8, 11.0, 6.63, and 8.3, respectively. Predictor selection frequencies for SLDA and SKDA for this example are given in Table 2.7. SKDA selects far fewer predictors yet is competitive in terms of classification error.

¹Available at the UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/>

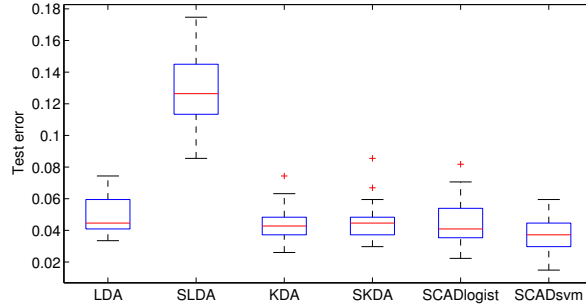


Figure 2.8 Boxplots of test errors for the WBCD example

Table 2.7 Predictor selection frequencies for SLDA and SKDA for the WBCD data example. Asterisks indicate variables selected using the entire data set.

X_j	SLDA	SKDA	SCAD logist	SCAD svm	X_j	SLDA	SKDA	SCAD logist	SCAD svm
X_1	5	0	0	0	X_{16}	19	0	3*	4
X_2	14	3	4	4	X_{17}	12*	0	0	0
X_3	0	0	0	0	X_{18}	4	0	0	0
X_4	1	0	0	0	X_{19}	10	0	0	0
X_5	2	0	0	0	X_{20}	7	0	7	5
X_6	18*	0	0	1	X_{21}	38*	40*	32*	40*
X_7	2	0	2	7	X_{22}	25*	37*	36*	37*
X_8	30*	6	15*	24*	X_{23}	1	0	4	6
X_9	9	0	0	1	X_{24}	9*	0	8*	1
X_{10}	22	0	1	2*	X_{25}	14	14*	6	29*
X_{11}	26*	7	35*	17*	X_{26}	1	0	0	0
X_{12}	10	0	2	2	X_{27}	18*	2	9*	14*
X_{13}	2	1	0	0	X_{28}	32*	28*	25*	24
X_{14}	21*	0	0	0	X_{29}	30*	13	7	27*
X_{15}	32*	0	1	4*	X_{30}	26*	0	2	0

2.4.4.2 Chemical Signatures

The proliferation of natural gas wells due to economically feasible welling technologies has heightened awareness and concern about emissions from these wells and their impacts on human health and the environment. On-ground, off-site, remote emissions measuring technology will play a key role in obtaining quality data for assessing emissions from oil and gas production

sites. We illustrate the application and performance of SKDA using data collected during the development of a remote measuring technology [73]. The data contain measurements of 38 chemical compounds recorded at each of $n = 170$ sites. Secondary to the remote-sensing method development research, the data were clustered using k -means clustering with $k = 2$ (and using all 38 compounds) ostensibly aligning with well type (*dry*, *wet*). We are not concerned with the accuracy or interpretation of this initial clustering. Rather for our illustration we take the classification as given and address the question of whether a smaller subset of the 38 compounds explains it. Because the study is still in progress, we identify the chemical compounds only as X_1, \dots, X_{38} .

The heat map in Figure 2.9 illustrates the data and classification. The figure is too small for the well labels, $1, \dots, 170$, to be legible on the heat map plot. However, it is sufficient to know that the first 50 columns correspond to the “ $Y = 0$ ” group, and that the last 120 columns correspond to the “ $Y = 1$ ” group. Chemical compound labels, identified here only as X_1, \dots, X_{38} , are legible on the right vertical axis.

To assess the relative performance of methods on this data set we used 40 random splits of the data into training and test sets of sizes 136 and 34 respectively. The training data were analyzed using eight-fold cross validation to select tuning parameters where required. Average test errors (with standard errors) are reported in Table 2.8.

Table 2.8 Test error summary for 40 random splits of the chemical signature data.

	LDA	SLDA	KDA	SKDA	SCADlogit	SCADsvm
Test Error Average	0.15	0.38	0.17	0.02	0.31	0.02
Test Error Std Dev	0.06	0.07	0.06	0.03	0.03	0.03

In addition to having the lowest test error, the collection of forty models selected by SKDA clearly suggests that two variables, X_2 and X_5 , provide the bulk of the discriminatory power, whereas the collection of models selected by SLDA is far more diverse and less suggestive with

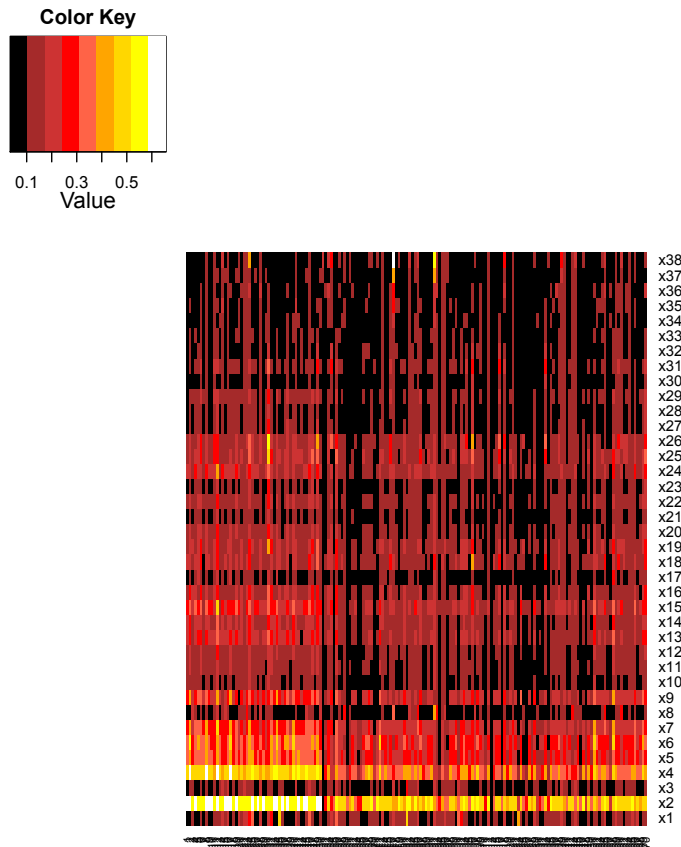


Figure 2.9 Chemical data heat map. First 50 columns (left to right) are the “ $Y = 0$ ” group. Columns within groups randomly ordered. Rows identify compounds X_1, \dots, X_{38} .

regard to possible sparsity in a good classification rule.

2.4.5 Consistency of SKDA

We now establish consistency of SKDA. Below we establish the notation used for the statements of the results below, and in the proofs in the appendix Section 2.6.3.

Partition \mathbf{X} into important, $\mathbf{X}_{\mathcal{I}}$, and unimportant, $\mathbf{X}_{\mathcal{U}}$, components. Without loss of generality, assume that $\mathbf{X} = (\mathbf{X}_{\mathcal{I}}^T, \mathbf{X}_{\mathcal{U}}^T)^T$. Denote the conditional density function of \mathbf{X} given $Y = k$ by $f_k(\mathbf{x}) \equiv f_k(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}})$ and the conditional probability of $Y = 1$ given $\mathbf{X} = \mathbf{x}$ by

Table 2.9 Predictor selection frequencies for SLDA and SKDA for the chemical signature data. Asterisks indicate variables selected using the entire data set.

X_j	SLDA	SKDA	SCAD logist	SCAD svm	X_j	SLDA	SKDA	SCAD logist	SCAD svm
X_1	26*	3	0	0	X_{20}	11	0	0	0
X_2	40*	40*	40*	40*	X_{21}	22*	0	0	0
X_3	28*	0	0	0	X_{22}	3	0	0	0
X_4	29*	8	2	24*	X_{23}	10*	0	0	0
X_5	23*	33*	15*	18	X_{24}	23*	0	0	0
X_6	13*	0	0	0	X_{25}	11*	0	0	0
X_7	19*	0	0	0	X_{26}	9	0	0	0
X_8	21*	0	0	0	X_{27}	10	0	0	0
X_9	25*	0	0	0	X_{28}	19*	0	0	0
X_{10}	20*	0	0	0	X_{29}	23*	0	0	0
X_{11}	21	0	0	1	X_{30}	14*	0	0	0
X_{12}	14*	0	0	0	X_{31}	24*	0	0	0
X_{13}	10	1	0	0	X_{32}	20	0	0	1
X_{14}	4	0	0	0	X_{33}	17*	5	0	4
X_{15}	16*	0	0	0	X_{34}	22*	0	0	2
X_{16}	10	0	0	0	X_{35}	20*	3	1	10
X_{17}	19*	2	0	1	X_{36}	18*	0	0	0
X_{18}	15	0	0	0	X_{37}	14*	0	0	0
X_{19}	10*	0	0	0	X_{38}	16*	0	0	0

$P(\mathbf{x}) \equiv P(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) = \frac{\pi_1 f_1(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}})}{\pi_0 f_0(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) + \pi_1 f_1(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}})}$. Note that $\mathbf{X}_{\mathcal{U}}$ being unimportant means

$$P(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) = P(\tilde{\mathbf{x}}_{\mathcal{I}}, \tilde{\mathbf{x}}_{\mathcal{U}}) \text{ as long as } \tilde{\mathbf{x}}_{\mathcal{I}} = \mathbf{x}_{\mathcal{I}}. \quad (2.13)$$

Note further that this partition is not unique since $\mathcal{I} \cup \{j\}$ and $\mathcal{U} \setminus \{j\}$ is another partition of important and unimportant predictors for any $j \in \mathcal{U}$ as long as \mathcal{I} and \mathcal{U} satisfy (2.13). To ensure uniqueness, assume that \mathcal{I} has minimal cardinality among all such partitions.

Extending the partition notation to $\tilde{P}(\mathbf{x}) \equiv \tilde{P}(\mathbf{x}; \boldsymbol{\lambda}) = \hat{\pi}_1 \tilde{f}_1(\mathbf{x}) / (\hat{\pi}_0 \tilde{f}_0(\mathbf{x}) + \hat{\pi}_1 \tilde{f}_1(\mathbf{x}))$, we define $\tilde{P}(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) \equiv \tilde{P}(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}; \boldsymbol{\lambda}_{\mathcal{I}}, \boldsymbol{\lambda}_{\mathcal{U}}) = \hat{\pi}_1 \tilde{f}_1(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) / (\hat{\pi}_0 \tilde{f}_0(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}) + \hat{\pi}_1 \tilde{f}_1(\mathbf{x}_{\mathcal{I}}, \mathbf{x}_{\mathcal{U}}))$, and $\ell(\boldsymbol{\lambda}_{\mathcal{I}}, \boldsymbol{\lambda}_{\mathcal{U}}) = \sum_{i=1}^n \left\{ y_i \log(\tilde{P}(\mathbf{x}_{i\mathcal{I}}, \mathbf{x}_{i\mathcal{U}}; \boldsymbol{\lambda}_{\mathcal{I}}, \boldsymbol{\lambda}_{\mathcal{U}})) + (1 - y_i) \log(1 - \tilde{P}(\mathbf{x}_{i\mathcal{I}}, \mathbf{x}_{i\mathcal{U}}; \boldsymbol{\lambda}_{\mathcal{I}}, \boldsymbol{\lambda}_{\mathcal{U}})) \right\}$.

For any $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_d)^T$, denote $\mathcal{A}(\boldsymbol{\lambda}) = \{j : \lambda_j > 0\}$. For any vector \mathbf{x} and set \mathcal{A} ,

denote \mathbf{x}_A to be the subvector of \mathbf{x} with indices in A . We use this notation often. For example, \mathbf{s}_B and \mathbf{c}_B are subvectors of \mathbf{s} and \mathbf{c} with indices in B , and so on. By slight abuse of notation, let $f_k(\mathbf{x}_A)$ be the conditional density function of \mathbf{X}_A given $Y = k$ for $k = 0, 1$. For a symmetric kernel $K(\cdot)$, denote $\mu_2 = \int t^2 K(t) dt$ and $\nu_0 = \int K^2(t) dt$.

Lemma 1 Consider an index subset $A \subset \{1, 2, \dots, d\}$. If $\lambda_j \rightarrow \infty$ as $\min_k \{n_k\} \rightarrow \infty$ when $j \in A$ and $\lambda_j = 0$ otherwise, and $\max_k \left\{ \prod_{j \in A} \lambda_j / n_k \right\} \rightarrow 0$, then $\hat{f}_k(\mathbf{x}_A) = \frac{1}{n_k} \sum_{i: y_i = k} \prod_{j \in A} \lambda_j K(\lambda_j(x_j - x_{ij}))$ is a consistent estimator of $f_k(\mathbf{x}_A)$ for $k = 0, 1$. Asymptotically, the bias and variance are given by $E\hat{f}_k(\mathbf{x}_A) - f_k(\mathbf{x}_A) = \frac{\mu_2}{2} \sum_{j \in A} f_{k;jj}^{(2)}(\mathbf{x}_A) / (\lambda_j^2) + o(\sum_{j \in A} \lambda_j^{-2})$ and $\text{Var}(\hat{f}_k(\mathbf{x}_A)) = f_k(\mathbf{x}_A) \prod_{j \in A} (\nu_0 \lambda_j) / n_k + o(\frac{1}{n_k} \prod_{j \in A} \lambda_j)$.

Lemma 1 shows that the estimator for the conditional density is consistent when all nonzero elements of $\boldsymbol{\lambda}$ diverge to infinity when $n \rightarrow \infty$. We establish next that the estimator of the conditional density is not consistent when some elements of $\boldsymbol{\lambda}$ converges to finite positive numbers while $n \rightarrow \infty$.

Lemma 2 Consider subsets A and B of $\{1, 2, \dots, d\}$. If $\lambda_j \rightarrow \infty$ as $n \rightarrow \infty$ when $j \in A$, $\lambda_j \rightarrow c_j$ for some constant $c_j > 0$ when $j \in B$, and $\lambda_j = 0$ otherwise, $\hat{f}_k(\mathbf{x}_A, \mathbf{x}_B) = \frac{1}{n_k} \sum_{i: y_i = k} \prod_{j \in A \cup B} \lambda_j K(\lambda_j(x_j - x_{ij}))$ is not a consistent estimator of $f_k(\mathbf{x}_A, \mathbf{x}_B)$ for $k = 0, 1$. Asymptotically, we have $E\hat{f}_k(\mathbf{x}_A, \mathbf{x}_B) = \int_{\mathbf{s}_B} (\prod_{j \in B} K(s_j)) f_k(\mathbf{x}_A, \mathbf{x}_B + \mathcal{D}_{\{\mathbf{c}_B\}} \mathbf{s}_B) d\mathbf{s}_B + \frac{\mu_2}{2} \sum_{j \in A} \frac{1}{\lambda_j^2} \int_{\mathbf{s}_B} (\prod_{j \in B} K(s_j)) f_{k;jj}^{(2)}(\mathbf{x}_A, \mathbf{x}_B + \mathcal{D}_{\{\mathbf{c}_B\}} \mathbf{s}_B) d\mathbf{s}_B + o(1 + \sum_{j \in A} \lambda_j^{-2})$ and $\text{Var}(f_k(\mathbf{x}_A)) = \prod_{j \in B} (\lambda_j) \left(\int_{\mathbf{s}_B} (\prod_{j \in B} K(s_j)) f_k(\mathbf{x}_A, \mathbf{x}_B + \mathcal{D}_{\{\mathbf{c}_B\}} \mathbf{s}_B) d\mathbf{s}_B \right) \prod_{j \in A} (\nu_0 \lambda_j) / n_k + o(1 + \frac{1}{n_k} \prod_{j \in A} \lambda_j)$.

The extra 1 in the little- o terms is due to the assumption that $\lambda_j \rightarrow c_j > 0$ for $j \in B$.

Theorem 1 Assume that the domain \mathcal{X} is compact and $\tau \rightarrow \infty$ and $\tau^d/n \rightarrow 0$ as $n \rightarrow \infty$ in (2.11). Then the optimizer $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_d)^T$ is such that $\hat{\lambda}_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\hat{\lambda}_j \rightarrow 0$ for $j \in \mathcal{U}$.

Corollary 1 Assume that the domain \mathcal{X} is compact and $\tau \rightarrow \infty$ and $\tau^d/n \rightarrow 0$ as $n \rightarrow \infty$ in (2.12). Then the optimizer $\hat{\boldsymbol{\lambda}} = (\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_d)^T$ satisfies that $\hat{\lambda}_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\hat{\lambda}_j \rightarrow 0$ for $j \in \mathcal{U}$.

Note. Because (2.11) and (2.12) are not convex in $\boldsymbol{\lambda}$ establishing asymptotic properties other than consistency is more involved and we leave that for future work.

2.5 Summary

We introduced a new approach to variable selection that adapts naturally to nonparametric models. The method is derived from a measurement error model likelihood, and exploits the fact that a variable containing a lot of measurement error is not useful for modeling. When the approach is applied to linear regression, the resulting new method is closely related to the LASSO, yielding a novel interpretation for the LASSO and lending credibility to the new MEM selection likelihood approach. When the approach is used for variable selection in nonparametric classification, a new method is obtained, *sparse kernel discrimination analysis* (SKDA). SKDA was shown to be competitive with existing methods in the case that the true classifier is linear (the case for which the existing methods were derived), and generally superior in cases where the true classifier is nonlinear.

In Section 2.3.2 we showed that for linear regression with λ_j parameterized in terms of precisions, MESSO optimization is convex and amenable to efficient computation algorithms. For other cases that of interest MESSO optimization is not convex. For the simulation results and examples in this paper we used the MATLAB constrained minimization function *fmincon*.² We have also had good results with coordinate descent (modified to handle the sum-to- τ constraint) and a standard BFGS iteration after reparametrization to eliminate constraints, e.g., setting $\lambda_j = \tau\eta_j^2/(\eta_1^2 + \dots + \eta_p^2)$. A web search of “*parallel coordinate descent*” returns numerous recent technical reports describing promising modifications of coordinate descent that take advantage of parallel processing. We are investigating the use of these algorithms to enable MESSO to be routinely used with even larger-dimension problems than those studied in Section 2.4.3.2 ($p = 50, 100$).

In 1986 R. Prentice wrote “There appears to be a dirth [sic] of realistic covariate measurement error models that lead to explicit forms ...” for distributions of the observed data in a measurement error model [56]. The same is true today and thus the construction of a MEM Selection Likelihood as defined in Section 2.2.3 is not always straightforward. A popular solution

²MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

to this dilemma in the MEM literature is *regression calibration* [11]. It provides a ready-made (approximate) solution to the construction of the MEM likelihoods (Step 2 of the algorithm in Section 2.2.3). Our initial attempts at using regression calibration to implement our MEM-based variable selection in generalized linear models are promising. Additionally, unlike in 1986, it is now feasible to compute likelihoods even when explicit forms are not available. Thus we anticipate that a second route to realizing the full potential of the MEM-based approach to variable selection will likely entail numerical (Monte Carlo) evaluation of MEM likelihoods.

Acknowledgments. The authors thank the entire review team and especially the Associate Editor and Editor for suggestions and comments that lead to substantial improvements in the paper. K. White was funded by NIH training grant T32HL079896 and NSF grant DMS-1055210; Y. Wu by NSF grant DMS-1055210 and NIH/NCI grant R01-CA149569; and L. Stefanski by NIH grants R01CA085848 and P01CA142538, and NSF grant DMS-0906421.

2.6 Appendix (Supplemental Files)

This section contains details provided in the supplemental files of [68]. Sections 2.6.2 and 2.6.3 are copied verbatim from the supplemental files. Section 2.6.1 contains additional details that are not in the supplemental files, but that facilitate presentation and understanding of the key convexity proof.

2.6.1 Equivalence of LASSO and $\widehat{L}_{\text{SEL}_1}$ -MESSO

We now prove the solution-path equivalence between maximizing $\widehat{L}_{\text{SEL}_1}$ in (2.6) and LASSO. We consider the transformation $\widehat{\sigma}^2(\boldsymbol{\lambda}) = \left(\widehat{L}_{\text{SEL}_1}\right)^{-2/n}$, so that

$$\widehat{\sigma}^2(\boldsymbol{\lambda}) = \widehat{V}_Y - \widehat{\mathbf{V}}_{\mathbf{X}Y}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}\right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y}. \quad (2.14)$$

Maximizing $\widehat{L}_{\text{SEL}_1}$ is equivalent to minimizing $\widehat{\sigma}^2(\boldsymbol{\lambda})$.

We first establish convexity of $\widehat{\sigma}^2(\boldsymbol{\lambda})$ and then we prove that the KKT conditions for the

LASSO and $\widehat{L}_{\text{SEL}_1}$ -MESSO are equivalent. Note that although we started Section 2.3 with an assumption of multivariate normality, the results proved in this subsection do not depend on any distributional assumptions.

CONVEXITY. We first establish definitions, identities, and results that will help in proving convexity of $\widehat{\sigma}^2(\boldsymbol{\lambda})$.

Diagonal matrix definition and identities.

1. For a given vector \mathbf{v} , $\mathcal{D}_{\{\mathbf{v}\}}$ is the diagonal matrix with \mathbf{v} on the main diagonal.
2. $\mathcal{D}_{\{\mathbf{1}\}} = \mathbf{I}$
3. $\mathcal{D}_{\{\mathbf{v}\}}\mathbf{1} = \mathbf{v}$ and $\mathbf{1}^T\mathcal{D}_{\{\mathbf{v}\}} = \mathbf{v}^T$
4. $\mathcal{D}_{\{\mathbf{v}_1\}}\mathbf{v}_2 = \mathcal{D}_{\{\mathbf{v}_2\}}\mathbf{v}_1$ and $\mathbf{v}_2^T\mathcal{D}_{\{\mathbf{v}_1\}} = \mathbf{v}_1^T\mathcal{D}_{\{\mathbf{v}_2\}}$
5. $\mathcal{D}_{\{\mathcal{D}_{\{\mathbf{v}_1\}}\mathbf{v}_2\}} = \mathcal{D}_{\{\mathbf{v}_2\}}\mathcal{D}_{\{\mathbf{v}_1\}} = \mathcal{D}_{\{\mathbf{v}_1\}}\mathcal{D}_{\{\mathbf{v}_2\}}$

Lemma 3 *If \mathbf{V}_X is nonnegative definite and $\boldsymbol{\lambda} \geq \mathbf{0}$, then $\mathbf{I} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}$ and $\mathbf{I} + \mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}$ are full rank.*

Proof. Suppose there exists \mathbf{c} such that

$$\left(\mathbf{I} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\right)\mathbf{c} = \mathbf{0}, \quad \text{or} \quad \left(\mathbf{I} + \mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}\right)\mathbf{c} = \mathbf{0}. \quad (2.15)$$

Clearly, $\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{c} = \mathbf{0}$ implies $\mathbf{c} = \mathbf{0}$ in both cases. If $\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{c} \neq \mathbf{0}$, then $\lambda_j^{1/2}c_j \neq 0$ for some j , from which it follows that $\mathbf{c}^T\mathbf{c} > 0$ and $\mathbf{c}^T\mathcal{D}_{\{\boldsymbol{\lambda}\}}\mathbf{c} > 0$. Note that $\left(\mathbf{I} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\right)\mathbf{c} = \mathbf{0}$ implies $\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{c} = -\mathbf{c}$, and thus that

$$\mathbf{c}^T\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{1/2}\mathbf{c} = -\mathbf{c}^T\mathbf{c} < 0.$$

Likewise $\left(\mathbf{I} + \mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}\right)\mathbf{c} = \mathbf{0}$ implies that $\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}\mathbf{c} = -\mathbf{c}$, and thus that

$$\mathbf{c}^T\mathcal{D}_{\{\boldsymbol{\lambda}\}}\mathbf{V}_X\mathcal{D}_{\{\boldsymbol{\lambda}\}}\mathbf{c} = -\mathbf{c}^T\mathcal{D}_{\{\boldsymbol{\lambda}\}}\mathbf{c} < 0.$$

The latter two inequalities are contradictions because \mathbf{V}_X is nonnegative definite. It follows that $(\mathbf{I} + \mathcal{D}_{\{\lambda\}}^{1/2} \mathbf{V}_X \mathcal{D}_{\{\lambda\}}^{1/2}) \mathbf{c} = \mathbf{0}$ iff $\mathbf{c} = \mathbf{0}$, and the same is true for $\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}}$. Thus both matrices are full rank. ■

Function Definitions and Identities. In the following \mathbf{V}_X is a nonnegative definite matrix and $\lambda \geq \mathbf{0}$. Under these conditions $\mathbf{I} + \mathcal{D}_{\{\lambda\}}^{1/2} \mathbf{V}_X \mathcal{D}_{\{\lambda\}}^{1/2}$ and $\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}}$ are full rank, and so is the transpose of the latter by Lemma 3.

$$\begin{aligned} (\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}})^{-1} &= \mathbf{I} - \mathbf{V}_X (\mathbf{I} + \mathcal{D}_{\{\lambda\}} \mathbf{V}_X)^{-1} \mathcal{D}_{\{\lambda\}} \\ (\mathbf{I} + \mathcal{D}_{\{\lambda\}} \mathbf{V}_X)^{-1} &= \mathbf{I} - \mathcal{D}_{\{\lambda\}} (\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}})^{-1} \mathbf{V}_X \end{aligned}$$

$$\begin{aligned} \widehat{\beta}(\lambda) &= \mathcal{D}_{\{\lambda\}}^{1/2} (\mathbf{I} + \mathcal{D}_{\{\lambda\}}^{1/2} \mathbf{V}_X \mathcal{D}_{\{\lambda\}}^{1/2})^{-1} \mathcal{D}_{\{\lambda\}}^{1/2} \mathbf{V}_{XY} \\ &= \mathcal{D}_{\{\lambda\}} (\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}})^{-1} \mathbf{V}_{XY} \\ &= (\mathbf{I} + \mathcal{D}_{\{\lambda\}} \mathbf{V}_X)^{-1} \mathcal{D}_{\{\lambda\}} \mathbf{V}_{XY} \\ &\stackrel{\lambda > \mathbf{0}}{=} (\mathbf{V}_X + \mathcal{D}_{\{\lambda\}}^{-1})^{-1} \mathbf{V}_{XY} \end{aligned}$$

$$(\mathbf{I} + \mathbf{V}_X \mathcal{D}_{\{\lambda\}})^{-1} \mathbf{V}_{XY} = \mathbf{V}_{XY} - \mathbf{V}_X \widehat{\beta}(\lambda)$$

$$\begin{aligned} \widehat{\beta}(\lambda) &= M \mathcal{D}_{\{M \mathbf{V}_{XY}\}} \quad \text{for } M = (\mathbf{I} + \mathcal{D}_{\{\lambda\}} \mathbf{V}_X)^{-1} \\ &= M \mathcal{D}_{\{\Delta\}} \quad \text{for } \Delta = \mathbf{V}_{XY} - \mathbf{V}_X \widehat{\beta}(\lambda) \\ &\stackrel{\lambda > \mathbf{0}}{=} (\mathbf{V}_X + \mathcal{D}_{\{\lambda\}}^{-1})^{-1} \mathcal{D}_{\{\lambda\}}^{-2} \mathcal{D}_{\{\widehat{\beta}(\lambda)\}} \end{aligned}$$

$$\widehat{\sigma}^2(\lambda) = V_Y - \mathbf{V}_{XY}^T \widehat{\beta}(\lambda)$$

$$\begin{aligned}
\hat{\sigma}^2(\boldsymbol{\lambda}) &= \{\partial \hat{\sigma}^2(\boldsymbol{\lambda}) / \partial \boldsymbol{\lambda}\} \\
&= -\mathbf{V}_{\mathbf{X}Y}^T \hat{\beta}(\boldsymbol{\lambda}) \\
&= \boldsymbol{\Delta}^T \mathcal{D}_{\{\boldsymbol{\Delta}\}} \quad \text{for } \boldsymbol{\Delta} = \mathbf{V}_{\mathbf{X}Y} - \mathbf{V}_{\mathbf{X}} \hat{\beta}(\boldsymbol{\lambda}) \\
&\stackrel{\boldsymbol{\lambda} > \mathbf{0}}{=} -\mathbf{V}_{\mathbf{X}Y}^T \left(\mathbf{V}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \right)^{-1} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\hat{\beta}(\boldsymbol{\lambda})\}}
\end{aligned}$$

$$\ddot{\sigma}^2(\boldsymbol{\lambda}) = 2\mathcal{D}_{\{\hat{\beta}(\boldsymbol{\lambda})\}} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \left\{ \mathcal{D}_{\{\boldsymbol{\lambda}\}} - \left(\mathbf{V}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \right)^{-1} \right\} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\hat{\beta}(\boldsymbol{\lambda})\}}$$

$$\begin{aligned}
\mathcal{D}_{\{\boldsymbol{\lambda} + \mathbf{h}\}}^{-1} &= \left\{ \mathcal{D}_{\{\boldsymbol{\lambda}\}} (I + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \mathcal{D}_{\{\mathbf{h}\}}) \right\}^{-1} \\
&= \left(I + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \mathcal{D}_{\{\mathbf{h}\}} \right)^{-1} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}
\end{aligned} \tag{2.16}$$

$$(I + A)^{-1} \approx I - A + A^2 \quad \text{for } A \text{ "small"} \tag{2.17}$$

$$\mathcal{D}_{\{\mathbf{a}\}} \mathcal{D}_{\{\mathbf{b}\}} = \mathcal{D}_{\{\mathbf{b}\}} \mathcal{D}_{\{\mathbf{a}\}} \tag{2.18}$$

$$\mathcal{D}_{\{\mathbf{a}\}} \mathbf{b} = \mathcal{D}_{\{\mathbf{b}\}} \mathbf{a}. \tag{2.19}$$

Convexity proof. We now give a direct proof that (2.14) is convex in the $\boldsymbol{\lambda} \in \mathbb{R}^p$ parameter. Define $P(\boldsymbol{\lambda}) = \widehat{\mathbf{V}}_{\mathbf{X}Y}^T (\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1})^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y}$. For notational simplicity, let $M = (\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1})^{-1}$ so $P(\boldsymbol{\lambda}) = \widehat{\mathbf{V}}_{\mathbf{X}Y}^T M \widehat{\mathbf{V}}_{\mathbf{X}Y}$. It is equivalent to write $M = (\mathcal{D}_{\{\boldsymbol{\lambda}\}} \widehat{\mathbf{V}}_{\mathbf{X}} + I)^{-1} \mathcal{D}_{\{\boldsymbol{\lambda}\}} = \mathcal{D}_{\{\boldsymbol{\lambda}\}} (\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} + I)^{-1}$ to avoid division by zero when any $\boldsymbol{\lambda}$ component is 0. Let \mathbf{h} be a constant vector of "small"

values so that both $\mathcal{D}_{\{\mathbf{h}\}}$ and $\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}\mathcal{D}_{\{\mathbf{h}\}}$ are “small.” Then by (2.16), (2.17) and (2.18),

$$\begin{aligned}\mathcal{D}_{\{\boldsymbol{\lambda}+\mathbf{h}\}}^{-1} &= \left(I + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}\mathcal{D}_{\{\mathbf{h}\}}\right)^{-1}\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \\ &\approx \left(I - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}\mathcal{D}_{\{\mathbf{h}\}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}^2\right)\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \\ &= \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2.\end{aligned}$$

Using this approximation in $P(\boldsymbol{\lambda} + \mathbf{h})$,

$$\begin{aligned}P(\boldsymbol{\lambda} + \mathbf{h}) &\approx \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2\right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &= \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left\{M^{-1} \left(I + M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right)\right)\right\}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &= \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left\{I + M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right)\right\}^{-1} M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}.\end{aligned}$$

The expression inside of the parenthesis is again considered to be “small” and (2.17) can be used on the expression in the braces,

$$\begin{aligned}P(\boldsymbol{\lambda} + \mathbf{h}) &\approx \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left\{I - M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right)\right. \\ &\quad \left.+ M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right) M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right)\right\} M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &= \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right) M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &\quad + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right) M \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}}\right) M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &= P(\boldsymbol{\lambda}) - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &\quad + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &\quad - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &\quad - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2}\mathcal{D}_{\{\mathbf{h}\}} M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\ &\quad + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 M\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3}\mathcal{D}_{\{\mathbf{h}\}}^2 M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}.\end{aligned}$$

As this is a second-order expansion of $P(\boldsymbol{\lambda} + \mathbf{h})$, any term with more than two $\mathcal{D}_{\{\mathbf{h}\}}$ factors is

ignored (the last three terms). Utilizing (2.18) and (2.19) this expression can be simplified to

$$\begin{aligned}
P(\boldsymbol{\lambda} + \mathbf{h}) &\approx P(\boldsymbol{\lambda}) + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\mathbf{h}\}} M \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&\quad + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T M \left(\mathcal{D}_{\{\mathbf{h}\}} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\mathbf{h}\}} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \mathcal{D}_{\{\mathbf{h}\}}^2 \right) M \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&= P(\boldsymbol{\lambda}) + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\mathbf{h}\}} M \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&\quad + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T M \mathcal{D}_{\{\mathbf{h}\}} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \right) \mathcal{D}_{\{\mathbf{h}\}} M \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&= P(\boldsymbol{\lambda}) + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T (\mathcal{D}_{\{\boldsymbol{\lambda}\}} \widehat{\mathbf{V}}_{\mathbf{X}} + I)^{-1} \mathcal{D}_{\{\boldsymbol{\lambda}\}} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}_{\{\mathbf{h}\}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} (\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} + I)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&\quad + \mathbf{h}^T \left\{ \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \right) \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \right\} \mathbf{h} \\
&= P(\boldsymbol{\lambda}) + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T (\mathcal{D}_{\{\boldsymbol{\lambda}\}} \widehat{\mathbf{V}}_{\mathbf{X}} + I)^{-1} \mathcal{D}_{\{\mathbf{h}\}} (\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} + I)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y} \\
&\quad + \mathbf{h}^T \left\{ \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \right) \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \right\} \mathbf{h} \\
&= P(\boldsymbol{\lambda}) + \widehat{\mathbf{V}}_{\mathbf{X}Y}^T (\mathcal{D}_{\{\boldsymbol{\lambda}\}} \widehat{\mathbf{V}}_{\mathbf{X}} + I)^{-1} \mathcal{D}_{\{(\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} + I)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \mathbf{h} \\
&\quad + \mathbf{h}^T \left\{ \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \right) \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \right\} \mathbf{h}
\end{aligned}$$

In general, a second-order expansion looks like

$$P(\boldsymbol{\lambda} + \mathbf{h}) \approx P(\boldsymbol{\lambda}) + (\nabla P(\boldsymbol{\lambda}))^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T \nabla(\nabla P(\boldsymbol{\lambda}))^T \mathbf{h}$$

so, letting $\boldsymbol{\Delta}(\boldsymbol{\lambda}) = (\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}} + I)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y}$, it must be that

$$\begin{aligned}
\nabla P(\boldsymbol{\lambda}) &= \mathcal{D}_{\{\boldsymbol{\Delta}(\boldsymbol{\lambda})\}} \boldsymbol{\Delta}(\boldsymbol{\lambda}), \\
\nabla \nabla^T P(\boldsymbol{\lambda}) &= 2 \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} \right) \mathcal{D}_{\{M \widehat{\mathbf{V}}_{\mathbf{X}Y}\}}.
\end{aligned}$$

The MESSO objective function is $\widehat{\sigma}^2(\boldsymbol{\lambda}) = \widehat{V}_Y - P(\boldsymbol{\lambda})$ so the Hessian of $\widehat{\sigma}^2(\boldsymbol{\lambda})$ is simply the negative Hessian of $P(\boldsymbol{\lambda})$. Convexity is proved by showing that $-\nabla \nabla^T P(\boldsymbol{\lambda})$ is non-negative

definite. First, note that

$$\begin{aligned}
& -\nabla\nabla^T P(\boldsymbol{\lambda}) \\
&= 2\mathcal{D}\{M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\} \left(\mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-3} - \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} M \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \right) \mathcal{D}\{M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\} \\
&= 2\mathcal{D}\{M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \left\{ \mathcal{D}_{\{\boldsymbol{\lambda}\}} - \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \right)^{-1} \right\} \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-2} \mathcal{D}\{M\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\}.
\end{aligned}$$

Let $A \succeq B$ mean that the matrix $A - B$ is non-negative definite. Then, $\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \succeq \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1}$ since $\widehat{\mathbf{V}}_{\mathbf{X}} \succeq 0$. So $\left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \right)^{-1} \preceq \mathcal{D}_{\{\boldsymbol{\lambda}\}}$ showing that $\mathcal{D}_{\{\boldsymbol{\lambda}\}} - \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}\}}^{-1} \right)^{-1}$ is non-negative definite. Then $-\nabla\nabla^T P(\boldsymbol{\lambda}) \succeq 0$ since, for any $\mathbf{z} \in \mathbb{R}^p$, $\mathbf{z}^T \{-\nabla\nabla^T P(\boldsymbol{\lambda})\} \mathbf{z}$ is a quadratic form where the innermost matrix is non-negative definite. Thus $\widehat{\sigma}^2(\boldsymbol{\lambda})$ is convex. \blacksquare

MESSO-LASSO EQUIVALENCE PROOF. Consider the penalty form of the LASSO estimator (2.3), which we reproduce here using the short-hand subscript L for LASSO.

$$\widehat{\boldsymbol{\beta}}_L(\eta) = \underset{\beta_1, \dots, \beta_p}{\operatorname{argmin}} \left\{ \frac{1}{n-1} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \eta \sum_{j=1}^p |\beta_j| \right\}, \quad (2.20)$$

Consider the weighted-penalty version of (2.20) obtained by replacing the L_1 norm in (2.20) with the weighted L_2 penalty $\eta \boldsymbol{\beta}^T \mathcal{D}_{\{\mathbf{w}\}}^{-1} \boldsymbol{\beta}$, where \mathbf{w} is a vector of nonnegative weights. The resulting estimator is

$$\widehat{\boldsymbol{\beta}}_{\mathbf{w}}(\eta) = \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\mathbf{w}/\eta\}}^{-1} \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} = \left(\mathbf{I} + \mathcal{D}_{\{\mathbf{w}/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}} \right)^{-1} \mathcal{D}_{\{\mathbf{w}/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}.$$

Note that if $\mathbf{w} = |\boldsymbol{\beta}|$, then $\eta \boldsymbol{\beta}^T \mathcal{D}_{\{\mathbf{w}\}}^{-1} \boldsymbol{\beta} = \eta \sum_j |\beta_j|$. It follows that the iteration

$$\boldsymbol{\beta}^{(0)} = \widehat{\boldsymbol{\beta}}_{OLS}, \quad \boldsymbol{\beta}^{(t+1)} \leftarrow \left(\mathbf{I} + \mathcal{D}_{\{|\boldsymbol{\beta}^{(t)}|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}} \right)^{-1} \mathcal{D}_{\{|\boldsymbol{\beta}^{(t)}|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}},$$

converges to the LASSO estimator, $\widehat{\boldsymbol{\beta}}_L(\eta)$, having the representation

$$\widehat{\boldsymbol{\beta}}_L(\eta) = \left(\mathbf{I} + \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_L|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}} \right)^{-1} \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_L|/\eta\}} \widehat{\mathbf{V}}_{\mathbf{X}Y}. \quad (2.21)$$

Comparing (2.21) to (2.7) shows that $\widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) = \widehat{\boldsymbol{\beta}}_L(\eta)$ when $\boldsymbol{\lambda} = |\widehat{\boldsymbol{\beta}}_L|/\eta$.

To prove equivalence of LASSO and $\widehat{L}_{\text{SEL}_1}$ -MESSO consider the optimization problem

$$\widehat{\boldsymbol{\lambda}}_M = \underset{\lambda_1, \dots, \lambda_p}{\operatorname{argmin}} \widehat{\sigma}^2(\boldsymbol{\lambda}), \quad \text{subject to} \quad \sum_{j=1}^p \lambda_j = \tau; \quad \lambda_j \geq 0, \quad j = 1, \dots, p, \quad (2.22)$$

and the regression estimator obtained from it (subscript M is short for MESSO)

$$\widehat{\boldsymbol{\beta}}_M = \widehat{\boldsymbol{\beta}}(\widehat{\boldsymbol{\lambda}}_M) = \left(\mathbf{I} + \mathcal{D}_{\{\boldsymbol{\lambda}_M\}} \widehat{\mathbf{V}}_{\mathbf{X}} \right)^{-1} \mathcal{D}_{\{\boldsymbol{\lambda}_M\}} \widehat{\mathbf{V}}_{\mathbf{X}Y}. \quad (2.23)$$

Reexpress the constraint functions in (2.22) as

$$h(\boldsymbol{\lambda}) = \sum_{j=1}^p \lambda_j - \tau, \quad \text{and} \quad g_j(\boldsymbol{\lambda}) = -\lambda_j, \quad j = 1, \dots, p.$$

The objective function $\widehat{\sigma}^2(\boldsymbol{\lambda})$ and the constraints are convex, continuously differentiable, and $h(\boldsymbol{\lambda})$ is affine. Thus the KKT conditions for optimality in (2.22) are necessary and sufficient [5, p. 207]. We now show that $\widehat{\boldsymbol{\lambda}}_L$ given by $\widehat{\lambda}_{L,j} = |\widehat{\beta}_{L,j}|/\eta$, $j = 1, \dots, p$, satisfies the KKT conditions for (2.22) when $\tau = \sum_j |\widehat{\beta}_{L,j}|/\eta$, thus establishing that $\widehat{\boldsymbol{\lambda}}_L$ is the minimizer to the MESSO problem, so that $\widehat{\boldsymbol{\lambda}}_L = \widehat{\boldsymbol{\lambda}}_M$ and thus $\widehat{\boldsymbol{\beta}}_L = \widehat{\boldsymbol{\beta}}_M$ in light of (2.21) and (2.23).

To this end we next prove the existence of constants ν and μ_j , $j = 1, \dots, p$, such that:

- (i) $\widehat{\sigma}^2(\boldsymbol{\lambda}) + \nu \dot{h}(\boldsymbol{\lambda}) + \sum_j \mu_j \dot{g}_j(\boldsymbol{\lambda}) = \mathbf{0}$;
- (ii) $g_j(\boldsymbol{\lambda}) \leq 0$, $j = 1, \dots, p$, $h(\boldsymbol{\lambda}) = 0$;
- (iii) $\mu_j \geq 0$, $j = 1, \dots, p$;
- (iv) $\mu_j g_j(\boldsymbol{\lambda}) = 0$, $j = 1, \dots, p$.

Substituting expressions for the derivatives in (i) reveals that it is equivalent to

(ia) $-\mathcal{D}_{\{\Delta(\lambda)\}}\Delta(\lambda) + \nu\mathbf{1}_p - \sum_{j=1}^p \mu_j e_j = \mathbf{0}$, with $e_j = j^{\text{th}}$ elementary vector.

Condition (ii) is satisfied by definition of $\widehat{\lambda}_L$. Since $\widehat{\beta}_L$ solves the LASSO problem, it satisfies the LASSO KKT conditions: $\widehat{\mathbf{V}}_{XY} - \widehat{\mathbf{V}}_X \widehat{\beta}_L = \eta\boldsymbol{\gamma}$, where $\boldsymbol{\gamma}$ is some $p \times 1$ vector with j^{th} element

$$\gamma_j \in \begin{cases} \{\text{sign}(\widehat{\beta}_{L,j})\} & \widehat{\beta}_{L,j} \neq 0, \\ [-1, 1] & \widehat{\beta}_{L,j} = 0. \end{cases} \quad (2.24)$$

Recall that $\Delta(\lambda) = (\mathbf{I} + \widehat{\mathbf{V}}_X \mathcal{D}_{\{\lambda\}})^{-1} \widehat{\mathbf{V}}_{XY}$ and so

$$\begin{aligned} \Delta(\lambda_L) &= (\mathbf{I} + \widehat{\mathbf{V}}_X \mathcal{D}_{\{\lambda_L\}})^{-1} \widehat{\mathbf{V}}_{XY} \\ &= \left(\mathbf{I} - \widehat{\mathbf{V}}_X (\mathbf{I} + \mathcal{D}_{\{\lambda_L\}} \widehat{\mathbf{V}}_X)^{-1} \mathcal{D}_{\{\lambda_L\}} \right) \widehat{\mathbf{V}}_{XY} = \widehat{\mathbf{V}}_{XY} - \widehat{\mathbf{V}}_X \widehat{\beta}_L. \end{aligned}$$

Now let $\nu = \eta^2$. Then $\widehat{\sigma}^2(\widehat{\lambda}_L) = -\mathcal{D}_{\{\Delta(\widehat{\lambda}_L)\}}\Delta(\widehat{\lambda}_L) \geq -\eta^2\mathbf{1}$ elementwise, ensuring that $-\mathcal{D}_{\{\Delta(\widehat{\lambda}_L)\}}\Delta(\widehat{\lambda}_L) + \nu\mathbf{1}_p$ is elementwise non-negative and so a vector $\boldsymbol{\mu}$ of non-negative constants always exists to solve (ia). This gives (i) and (iii). When $\widehat{\beta}_{L,j} \neq 0$ it must be that $\widehat{\lambda}_{L,j} > 0$ which implies that $g_j < 0$. In these components, $\gamma_j = \pm 1$ by (2.24) and so the j^{th} component of $\left(-\mathcal{D}_{\{\Delta(\widehat{\lambda}_L)\}}\Delta(\widehat{\lambda}_L) + \nu \right) = -(\pm\eta)^2 + \eta^2 = 0$ exactly. Thus μ_j can be set to 0 to maintain (i) and satisfy (iv).

The MESSO KKT conditions are satisfied by $\widehat{\lambda}_L$, so it must be that $\widehat{\lambda}_L = \widehat{\lambda}_M$ for a generic η and thus any η . Finally, $\widehat{\beta}_L = \beta_M(\widehat{\lambda}_L) = \beta_M(\widehat{\lambda}_M) = \widehat{\beta}_M$ and the two regression coefficient vectors are equal. (The first equality comes from the iterated ridge solution to LASSO which shares the MESSO form.) ■

After publication, a reviewer brought to our attention that a connection between LASSO and a perturbed design matrix in the context of robust regression had been established in [83]. One may view the false measurement error injected by MESSO as perturbations coming from an uncertainty set with a fixed norm.

2.6.2 Generating Sparse Classification Data

Proposition 1 *Suppose that the variance matrix $\Sigma_{p \times p}$ and mean vector $\mu_{p \times 1}$ are conformably partitioned as $(\Sigma_{11}, \Sigma_{12}, \Sigma_{21}, \Sigma_{22})$ and (μ_1, μ_2) . Let $\Psi(\cdot, \cdot, \nu_1, \nu_2, \Omega)$ denote the partitioned multivariate normal density with mean vector $(\nu_1^T, \nu_2^T)^T$ and variance matrix Ω . Then for any conformable vector θ_1 the ratio $\Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1, \mu_2, \Sigma) / \Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1 + \Sigma_{11}\theta_1, \mu_2 + \Sigma_{21}\theta_1, \Sigma)$ does not depend on \mathbf{x}_2 (it is a function of \mathbf{x}_1 only).*

Proof of Proposition 1. Using moment generating function identities it is straightforward to show that $\Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1, \mu_2, \Sigma) \exp(\mathbf{x}_1^T \theta_1 - \mu_1^T \theta_1 - \theta_1^T \Sigma_{11} \theta_1 / 2)$ integrates to one and is thus a density function, and that its moment generating function is the same as that of $\Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1 + \Sigma_{11}\theta_1, \mu_2 + \Sigma_{21}\theta_1, \Sigma)$. Thus

$$\frac{\Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1 + \Sigma_{11}\theta_1, \mu_2 + \Sigma_{21}\theta_1, \Sigma)}{\Psi(\mathbf{x}_1, \mathbf{x}_2; \mu_1, \mu_2, \Sigma)} = \exp(\mathbf{x}_1^T \theta_1 - \mu_1^T \theta_1 - \theta_1^T \Sigma_{11} \theta_1 / 2).$$

■

2.6.3 Proofs of Asymptotic Results

Proof of Lemma 1. Note that

$$\begin{aligned} E \widehat{f}_k(\mathbf{x}_{\mathcal{A}}) &= E \left(\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - X_{ij})) \mid Y_i = k \right) \\ &= \int_{\mathbf{x}_{i\mathcal{A}}} \prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - x_{ij})) f_k(\mathbf{x}_{i\mathcal{A}}) d\mathbf{x}_{i\mathcal{A}} = \int_{\mathbf{s}_{\mathcal{A}}} \prod_{j \in \mathcal{A}} K(s_j) f_k(\mathbf{x}_{\mathcal{A}} + \mathcal{D}_{\{\lambda_{\mathcal{A}}\}}^{-1} \mathbf{s}_{\mathcal{A}}) d\mathbf{s}_{\mathcal{A}} \\ &= \int_{\mathbf{s}_{\mathcal{A}}} \left[f_k(\mathbf{x}_{\mathcal{A}}) + \sum_{j \in \mathcal{A}} f_{k;j}^{(1)}(\mathbf{x}_{\mathcal{A}}) \frac{s_j}{\lambda_j} + \frac{1}{2} \sum_{j,m \in \mathcal{A}} f_{k;jm}^{(2)}(\mathbf{x}_{\mathcal{A}}) \frac{s_j s_m}{\lambda_j \lambda_m} \right] \times \left\{ \prod_{j \in \mathcal{A}} K(s_j) \right\} d\mathbf{s}_{\mathcal{A}} \\ &\quad + \text{higher order terms} \\ &= f_k(\mathbf{x}_{\mathcal{A}}) + \frac{\mu_2}{2} \sum_{j \in \mathcal{A}} f_{k;jj}^{(2)}(\mathbf{x}_{\mathcal{A}}) / (\lambda_j^2) + o\left(\prod_{j \in \mathcal{A}} \lambda_j^{-2}\right), \end{aligned} \tag{2.25}$$

where $\mathcal{D}_{\{\lambda_{\mathcal{A}}\}}$ is the diagonal matrix with elements $\lambda_j, j \in \mathcal{A}$, and $\mathcal{D}_{\{\lambda_{\mathcal{A}}\}}^{-1} \mathbf{s}_{\mathcal{A}}$ is a vector with elements $s_j/\lambda_j, j \in \mathcal{A}$. Define $f_{k;j}^{(1)}(\mathbf{x}_{\mathcal{A}}) = \frac{\partial}{\partial x_j} f_k(\mathbf{x}_{\mathcal{A}})$, $f_{k;jm}^{(2)}(\mathbf{x}_{\mathcal{A}}) = \frac{\partial^2}{\partial x_j \partial x_m} f_k(\mathbf{x}_{\mathcal{A}})$. Then

$$\begin{aligned} \text{Var}(\widehat{f}_k(\mathbf{x}_{\mathcal{A}})) &= \frac{1}{n_k} \text{Var} \left(\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - X_{ij})) | Y_i = k \right) \\ &= \frac{1}{n_k} \left(E \left(\left[\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - X_{ij})) \right]^2 | Y_i = k \right) - (E(\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - X_{ij})) | Y_i = k))^2 \right). \end{aligned}$$

Note that

$$\begin{aligned} E \left(\left[\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - X_{ij})) \right]^2 | Y_i = k \right) &= \int_{\mathbf{x}_{i\mathcal{A}}} \left[\prod_{j \in \mathcal{A}} \lambda_j K(\lambda_j(x_j - x_{ij})) \right]^2 f_k(\mathbf{x}_{i\mathcal{A}}) d\mathbf{x}_{i\mathcal{A}} \\ &= \int_{\mathbf{s}_{\mathcal{A}}} \prod_{j \in \mathcal{A}} [\lambda_j K^2(s_j)] f_k(\mathbf{x}_{\mathcal{A}} + \mathcal{D}_{\{\lambda_{\mathcal{A}}\}}^{-1} \mathbf{s}_{\mathcal{A}}) d\mathbf{s}_{\mathcal{A}} \\ &= \left(\prod_{j \in \mathcal{A}} \lambda_j \right) \int_{\mathbf{s}_{\mathcal{A}}} \left[f_k(\mathbf{x}_{\mathcal{A}}) + \sum_{j \in \mathcal{A}} f_{k;j}^{(1)}(\mathbf{x}_{\mathcal{A}}) \frac{s_j}{\lambda_j} + \frac{1}{2} \sum_{j,m \in \mathcal{A}} f_{k;jm}^{(2)}(\mathbf{x}_{\mathcal{A}}) \frac{s_j s_m}{\lambda_j \lambda_m} \right] \times \left\{ \prod_{j \in \mathcal{A}} K^2(s_j) \right\} d\mathbf{s}_{\mathcal{A}} \\ &\quad + \text{higher order terms} \\ &= f_k(\mathbf{x}_{\mathcal{A}}) \prod_{j \in \mathcal{A}} (\nu \lambda_j) + \text{higher order terms.} \end{aligned}$$

Combining (2.25), we have $\text{Var}(\widehat{f}_k(\mathbf{x}_{\mathcal{A}})) = f_k(\mathbf{x}_{\mathcal{A}}) \prod_{j \in \mathcal{A}} (\nu \lambda_j) / n_k + o\left(\frac{1}{n_k} \prod_{j \in \mathcal{A}} \lambda_j\right)$. This completes the proof. \blacksquare

Proof of Lemma 2. The proof follows from that of Lemma 1. \blacksquare

Proof of Theorem 1. We first consider the regular case that $\widehat{\lambda}_j \rightarrow c_j$ for some $c_j \in [0, \infty]$ for $j = 1, 2, \dots, p$. Denote $\mathcal{I}_0 = \{j : j \in \mathcal{I}; c_j = 0\}$, $\mathcal{I}_1 = \{j : j \in \mathcal{I}; 0 < c_j < \infty\}$, $\mathcal{I}_\infty = \{j : j \in \mathcal{I}; c_j = \infty\}$, $\mathcal{U}_0 = \{j : j \in \mathcal{U}; c_j = 0\}$, $\mathcal{U}_1 = \{j : j \in \mathcal{U}; 0 < c_j < \infty\}$, and $\mathcal{U}_\infty = \{j : j \in \mathcal{U}; c_j = \infty\}$. Note first that all the little- o terms in Lemmas 1 and 2 can be made uniform with respect to $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ due to the assumption that \mathcal{X} is compact [34]. By the strong

law of large numbers, $\widehat{\pi}_k \rightarrow \pi_k$ almost surely for $k = 0, 1$. First we prove by contradiction that $\widehat{\lambda}_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\mathcal{U}_1 = \emptyset$.

Recall that $\widehat{\boldsymbol{\lambda}} = (\widehat{\lambda}_1, \dots, \widehat{\lambda}_p)^T$ solves the optimization problem

$$\begin{aligned} & \max_{\lambda_1, \lambda_2, \dots, \lambda_p} \quad \frac{1}{n} \sum_{i=1}^n \left\{ y_i \log(\tilde{P}(\mathbf{x}_i; \boldsymbol{\lambda})) + (1 - y_i) \log(1 - \tilde{P}(\mathbf{x}_i; \boldsymbol{\lambda})) \right\} \\ & \text{subject to} \quad \sum_{j=1}^p \lambda_j = \tau; \quad \lambda_j \geq 0, j = 1, 2, \dots, p, \end{aligned} \quad (2.26)$$

$\tilde{f}_k(\mathbf{x}; \boldsymbol{\lambda}) = \sum_{i: y_i = k} \prod_{j=1}^p K(\lambda_j(x_j - x_{ij}))/n_k$ and $\tilde{P}(\mathbf{x}; \boldsymbol{\lambda}) = \widehat{\pi}_1 \tilde{f}_1(\mathbf{x}; \boldsymbol{\lambda}) / (\widehat{\pi}_0 \tilde{f}_0(\mathbf{x}; \boldsymbol{\lambda}) + \widehat{\pi}_1 \tilde{f}_1(\mathbf{x}; \boldsymbol{\lambda}))$.

Notations analogous to $\widehat{f}_k(\mathbf{x}; \boldsymbol{\lambda})$, $\tilde{f}_k(\mathbf{x}; \boldsymbol{\lambda})$, and $\tilde{P}(\mathbf{x}; \boldsymbol{\lambda})$ are used below. Note that

$$\begin{aligned} & \tilde{P}(\mathbf{x}_{\mathcal{I}_0}, \mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_0}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_0}, \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_0}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty}) \\ &= \left(1 + \frac{\widehat{\pi}_0 \tilde{f}_0(\mathbf{x}_{\mathcal{I}_0}, \mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_0}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_0}, \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_0}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})}{\widehat{\pi}_1 \tilde{f}_1(\mathbf{x}_{\mathcal{I}_0}, \mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_0}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_0}, \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_0}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})} \right)^{-1} \\ &= \left(1 + \frac{\widehat{\pi}_0 \tilde{f}_0(\mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})}{\widehat{\pi}_1 \tilde{f}_1(\mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})} + o_p(1) \right)^{-1} \\ &= \left(1 + \frac{\widehat{\pi}_0 \widehat{f}_0(\mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})}{\widehat{\pi}_1 \widehat{f}_1(\mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})} + o_p(1) \right)^{-1}. \end{aligned}$$

The first equality follows from definition of \tilde{P} ; the second from the fact that $K(\lambda_j(x_j - x_{ij})) = K(0) + o(1)$ when $\lambda_j \rightarrow 0$ and the domain \mathcal{X} is compact; the third via cancellation of $\prod_{j \in \mathcal{I}_1 \cup \mathcal{I}_\infty \cup \mathcal{U}_1 \cup \mathcal{U}_\infty} \lambda_j$ in the denominator and numerator. Thus by Lemma 2,

$\tilde{P}(\mathbf{x}_{\mathcal{I}_0}, \mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_0}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \boldsymbol{\lambda}_{\mathcal{I}_0}, \boldsymbol{\lambda}_{\mathcal{I}_1}, \boldsymbol{\lambda}_{\mathcal{I}_\infty}, \boldsymbol{\lambda}_{\mathcal{U}_0}, \boldsymbol{\lambda}_{\mathcal{U}_1}, \boldsymbol{\lambda}_{\mathcal{U}_\infty})$ converges in probability to

$$A(\mathbf{x}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}; \mathbf{c}_{\mathcal{I}_1}, \mathbf{c}_{\mathcal{U}_1}) = \left(1 + \frac{\pi_0 Q_0}{\pi_1 Q_1} \right)^{-1}$$

where

$$Q_k = \int_{\mathbf{s}_{\mathcal{I}_1 \cup \mathcal{U}_1}} \left(\prod_{j \in \mathcal{I}_1 \cup \mathcal{U}_1} K(s_j) \right) f_k(\mathbf{x}_{\mathcal{I}_1} + \mathcal{D}_{\{\mathbf{c}_{\mathcal{I}_1}\}} \mathbf{s}_{\mathcal{I}_1}, \mathbf{x}_{\mathcal{I}_\infty}, \mathbf{x}_{\mathcal{U}_1} + \mathcal{D}_{\{\mathbf{c}_{\mathcal{U}_1}\}} \mathbf{s}_{\mathcal{U}_1}, \mathbf{x}_{\mathcal{U}_\infty}) d\mathbf{s}_{\mathcal{I}_1 \cup \mathcal{U}_1}.$$

All convergence and little- o terms can be made uniform with respect to \mathbf{x} due the compactness

assumption on \mathcal{X} . Correspondingly the objective function of (2.26) converges to

$$E(Y \log [A(\mathbf{X}_{\mathcal{I}_1}, \mathbf{X}_{\mathcal{I}_\infty}, \mathbf{X}_{\mathcal{U}_1}, \mathbf{X}_{\mathcal{U}_\infty}; \mathbf{c}_{\mathcal{I}_1}, \mathbf{c}_{\mathcal{U}_1})] + (1 - Y) \log [1 - A(\mathbf{X}_{\mathcal{I}_1}, \mathbf{X}_{\mathcal{I}_\infty}, \mathbf{X}_{\mathcal{U}_1}, \mathbf{X}_{\mathcal{U}_\infty}; \mathbf{c}_{\mathcal{I}_1}, \mathbf{c}_{\mathcal{U}_1})]) \quad (2.27)$$

in probability. Consider another case $\lambda_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\lambda_m \rightarrow 0$ or ∞ for $m \in \mathcal{U}$. By noting that predictors $\mathbf{X}_{\mathcal{U}}$ are unimportant, the corresponding limit of the objective function (2.26) is

$$E \left(Y \log \left[\left\{ 1 + \frac{\pi_0 f_0(\mathbf{X}_{\mathcal{I}})}{\pi_1 f_1(\mathbf{X}_{\mathcal{I}})} \right\}^{-1} \right] + (1 - Y) \log \left[1 - \left\{ 1 + \frac{\pi_0 f_0(\mathbf{X}_{\mathcal{I}})}{\pi_1 f_1(\mathbf{X}_{\mathcal{I}})} \right\}^{-1} \right] \right)$$

which is larger than the limit in (2.27) due to our assumption on the distribution of \mathbf{X} and Y .

Thus $\widehat{\lambda}_j$ must satisfy $\widehat{\lambda}_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\widehat{\lambda}_j \rightarrow 0$ or ∞ in probability for $j \in \mathcal{U}$.

Next we prove $\widehat{\lambda}_j \rightarrow 0$ in probability for $j \in \mathcal{U}$ by noting the convergence speed. Denote $d_{\mathcal{I}}$ to be cardinality of \mathcal{I} and $d_{\mathcal{U}_\infty}$ denotes the cardinality of \mathcal{U}_∞ . According to Lemma 1, the asymptotic bias and variance are of order $\sum_{j \in \mathcal{I} \cup \mathcal{U}_\infty} \lambda_j^{-2}$ and $\frac{1}{n_k} \prod_{j \in \mathcal{I} \cup \mathcal{U}_\infty} \lambda_j$, respectively, which are optimized with λ_j diverges at the same speed as τ , say $\lambda_j = \tau / (d_{\mathcal{I}} + d_{\mathcal{U}_\infty})$ for each $j \in \mathcal{I} \cup \mathcal{U}_\infty$ and have optimal speeds of $(d_{\mathcal{I}} + d_{\mathcal{U}_\infty})^2 / \tau^2$ and $\tau^{d_{\mathcal{I}} + d_{\mathcal{U}_\infty}} / n_k$, respectively. Thus the asymptotic bias is of the same order τ^{-2} with a larger constant $(d_{\mathcal{I}} + d_{\mathcal{U}_\infty})^2$ for a larger $d_{\mathcal{U}_\infty}$ but the asymptotic variance is of order $\tau^{d_{\mathcal{I}} + d_{\mathcal{U}_\infty}} / n_k$ which is increasing in $d_{\mathcal{U}_\infty}$. Although the limit of the objective function of (2.26) is same as long as $\lambda_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\lambda_j \rightarrow 0$ or ∞ for $j \in \mathcal{U}$, the convergence speed to the same limit is different for different number $d_{\mathcal{U}_\infty}$ of $\lambda_j \rightarrow \infty$ for $j \in \mathcal{I}$. A fast convergence is preferred, which corresponds to $d_{\mathcal{U}_\infty} = 0$. This proves that $\widehat{\lambda}_j \rightarrow 0$ in probability for $j \in \mathcal{U}$.

We assumed initially that $\lim \widehat{\lambda}_j$ exists for all j (either 0, a finite number or infinity). We now argue that the limits exist. Consider every convergent subsequence of $\widehat{\boldsymbol{\lambda}}$ and apply the above proof to the subsequence. It shows that every convergent subsequence of $\widehat{\boldsymbol{\lambda}}$ has the same limit, which implies that the limit of $\widehat{\boldsymbol{\lambda}}$ exists. This completes the proof. \blacksquare

Proof of Corollary 1. The proof follows from that of Theorem 1. ■

CHAPTER

3

MEKRO

3.1 Introduction

Stefanski, Wu, and White [henceforth SWW; 68] describe a very general variable selection method that results from modeling predictors as if they were contaminated with measurement error. A model is first embedded in a measurement error model (MEM) framework, then the resulting MEM selection likelihood is maximized subject to a lower bound on the total measurement error. The feasible region set by the constraints has sharp corners that admit feature sparsity. The total measurement error serves as a tuning parameter to balance model sparsity and fit.

When applied to linear models, the SWW procedure generates solution paths identical to those of LASSO [68, 75]. Thus, one can regard SWW's procedure as an extension of LASSO to any model—in this paper, to nonparametric regression. We show that applying the SWW procedure to nonparametric regression results in the Nadaraya-Watson (NW) estimator, but with a novel method of bandwidth estimation that simultaneously performs smoothing and finite-sample variable selection as demonstrated by our simulation studies. Though bandwidth

selection is much studied for the NW estimator, variable selection is less studied and generally only asymptotically. The measurement error kernel regression operator (MEKRO) integrates both.

Intentionally contaminating observations with noise has been previously studied under the terms “noise injection” or “training with noise,” among others. Predictor contamination is well-studied in general for artificial neural networks where small amounts of noise reduce overfitting and generalization error [30, 32, 39, 65]. Simulation-extrapolation (SIMEX) estimation is a method to correct for measurement error in predictors by adding increasing amounts of known measurement error and extrapolating back to a hypothetical version of the data without error [44, 67]. Importantly, our method is distinguished from these noise-addition methods; we develop a likelihood under the false assumption that noise is present instead of contaminating observations.

This paper is organized as follows. Derivation of MEKRO and computational aspects of fitting and tuning are presented in Section 3.2. We extend the method to accommodate categorical covariates in Section 3.3. Section 3.4 describes related methods in the literature and provides numerical support for MEKRO with both simulated and real data examples. In Section 3.5 we study selection consistency. Section 3.6 closes with a discussion.

We observe data $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$, where Y_i is the response, $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,p})^T$ is the $p \times 1$ vector of covariates for the i th observation, and p is fixed. The (continuous) covariates are standardized so that $\sum_{i=1}^n \mathbf{X}_i = \mathbf{0}_{p \times 1}$ and $\sum_{i=1}^n X_{i,j}^2 / (n-1) = 1$, $j = 1, \dots, p$. Denote a generic observation as (\mathbf{X}, Y) where \mathbf{X} has j th component X_j . We assume the model

$$Y = g(\mathbf{X}) + \epsilon,$$

where $g(\mathbf{x}) = g_{Y|\mathbf{X}}(\mathbf{x}) \stackrel{\text{def}}{=} E(Y | \mathbf{X} = \mathbf{x})$ is the unknown regression function, and ϵ is a random error independent of \mathbf{X} with $E(\epsilon) = 0$ and $E(\epsilon^2) = \sigma_\epsilon^2 < \infty$. For presentation simplicity, assume that Y and \mathbf{X} are both continuous unless otherwise stated.

3.2 Measurement Error Kernel Regression Operator

In a supervised learning problem, if a covariate can be contaminated with a substantial amount of error without adversely affecting prediction performance, then it is not useful for predicting Y . Measurement error model (MEM) selection likelihoods introduced by SWW implement this concept by forcing ‘false’ Gaussian measurement error into the covariates \mathbf{X} . We first build a selection likelihood that describes the prediction degradation for a certain allocation of measurement error to each covariate. Then we perform constrained optimization of the likelihood where the constraints force ‘false’ measurement error into the likelihood while the optimizer determines the distribution of errors that results in the least degradation. The likelihood optimization ensures that the least relevant covariates will be assigned the most (possibly infinite) error.

Denote the measurement error variance associated with X_j as $\sigma_{u,j}^2$. MEM selection likelihoods describe model degradation through $\lambda_j = 1/\sigma_{u,j}$ and apply the optimization constraint $\mathbf{1}^T \boldsymbol{\lambda} = \tau$ where $\tau > 0$ is a tuning parameter. This constraint is equivalent to an equality constraint on the harmonic mean of $\boldsymbol{\sigma}_u$ and allows one or more $\sigma_{u,j} = \infty$ when $\tau > 0$, implying that each corresponding X_j can be measured with an infinite amount of ‘false’ measurement error and thus is irrelevant in the model. A constraint on the un-transformed $\boldsymbol{\sigma}_u$ could not achieve this as elegantly.

Applying the MEM selection likelihood framework from SWW to nonparametric regression results in a kernel regression bandwidth and variable selection method, the measurement error kernel regression operator (MEKRO). The MEM selection likelihood is

$$\hat{L}_{\text{SEL}}(\boldsymbol{\lambda}) = -\frac{1}{n} \sum_{i=1}^n \{Y_i - \hat{g}(\mathbf{X}_i, \boldsymbol{\lambda})\}^2, \quad (3.1)$$

where

$$\hat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) = \frac{\sum_{k=1}^n Y_k \prod_{j=1}^p \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2 / 2\}}{\sum_{k=1}^n \prod_{j=1}^p \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2 / 2\}}. \quad (3.2)$$

See Appendix A for the full derivation. Notice that (3.1) is simply the (negative) mean squared error, and, more interestingly, (3.2) is the familiar Nadaraya-Watson estimator [53, 78] of $g(\cdot)$ from the data $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$, computed using a Gaussian product kernel and diagonal bandwidth matrix. One key difference is that the traditional smoothing bandwidths, h_j , are parameterized as inverse bandwidths, $\lambda_j = 1/h_j$. In this setting, $\lambda_j = 0 \implies h_j = \infty$, or covariate X_j is infinitely smoothed and thus selected out. This parameterization is also found in [29].

Estimation of the (inverse-)bandwidths is done as prescribed in SWW, via maximizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ subject to an L_1 -type constraint in the non-negative orthant,

$$\lambda_j \geq 0, \quad j = 1, \dots, p; \quad \sum_{j=1}^p \lambda_j = \tau, \quad \text{for fixed } \tau > 0, \quad (3.3)$$

where τ is a tuning parameter controlling the roughness of $\widehat{g}(\cdot)$. A small τ keeps $\|\boldsymbol{\lambda}\|$ small, implying large bandwidths and substantial smoothing; a large τ permits smaller individual bandwidths and more roughness in $\widehat{g}(\cdot)$. Adding generated noise to predictors with this constraint has been successful in artificial neural networks [31].

Although MEKRO is the focus of this paper, the derivation of an estimator using MEM selection likelihoods itself is interesting and worth highlighting. SWW proved that applying MEM selection likelihoods to the linear model is equivalent to LASSO. The proof hinged on equivalent ways to express linear model coefficients subjected to shrinkage, either through ridge regression, LASSO, or MEM selection likelihoods. Although the same relationships or concepts do not exist in more complicated models, MEM selection likelihoods have been shown via simulation to produce LASSO-like, finite-sample variable selection in a density-based classification procedure [68] and in kernel regression (MEKRO).

A MEKRO solution $\widehat{\boldsymbol{\lambda}}_\tau$ is the result of optimizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ under the constraints in (3.3). To avoid constrained optimization, we introduce $\boldsymbol{\gamma} \in \mathbb{R}^p$ and let $\lambda_j(\boldsymbol{\gamma}) = \tau \gamma_j^2 / (\sum_{k=1}^p \gamma_k^2)$, $j = 1, \dots, p$, for a fixed τ . We then maximize $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda}(\boldsymbol{\gamma}))$ with respect to $\boldsymbol{\gamma}$. This guarantees that the constraints (3.3) on $\boldsymbol{\lambda}$ are satisfied for any $\boldsymbol{\gamma}$, at the cost of one additional parameter. Optimization is done in C using the gradient-based algorithm L-BFGS [55]. With $\pi_{ik} =$

$\prod_{j=1}^p \exp\{-\lambda_j^2(X_{i,j} - X_{k,j})^2/2\}$ and $\Gamma = \sum_{j=1}^p \gamma_j^2$, then $\widehat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) = \sum_{k=1}^n Y_k \pi_{ik} / \sum_{k=1}^n \pi_{ik}$ and the required gradients are,

$$\partial \widehat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) / \partial \lambda_t = \left[\sum_{k=1}^n Y_k \pi_{ik} \{-\lambda_t (X_{i,t} - X_{k,t})^2\} \times \sum_{k=1}^n \pi_{ik} - \sum_{k=1}^n \pi_{ik} \{-\lambda_t (X_{i,t} - X_{k,t})^2\} \times \sum_{k=1}^n Y_k \pi_{ik} \right] \left(\sum_{k=1}^n \pi_{ik} \right)^{-2}.$$

Also, $\partial \lambda_t / \partial \gamma_j = -2\tau \gamma_t^2 \gamma_j \Gamma^{-2}$ when $t \neq j$ and $2\tau \gamma_j (\Gamma - \gamma_j^2) \Gamma^{-2}$ when $t = j$. Finally,

$$\partial \widehat{L}_{\text{SEL}} / \partial \gamma_j = -\frac{4\tau \gamma_j}{n\Gamma^2} \sum_{t=1}^p \left[\sum_{i=1}^n \{Y_i - \widehat{g}(\mathbf{X}_i, \boldsymbol{\lambda})\} \times (\partial \widehat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) / \partial \lambda_t) \right] (\gamma_t^2 - \Gamma \mathbf{1}_{t=j}), \quad (3.4)$$

where $\mathbf{1}_{(\cdot)}$ is the indicator function. $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ is not concave but can be maximized well with neutral starting values. However, starting values near where at least one component of $\boldsymbol{\lambda}$ is zero and the initial gradient points in an unfavorable direction tends to trap the optimizer in non-global maxima. Further, (3.4) shows that $\partial \widehat{L}_{\text{SEL}} / \partial \gamma_j$ is 0 when $\gamma_j = 0$. Thus, using warm starts with components of $\boldsymbol{\lambda}$ set at or near 0 is ill-advised. We always start at $\boldsymbol{\gamma}_{\text{start}} = \mathbf{1}_p$, equivalent to $\boldsymbol{\lambda}_{\text{start}} = (\tau/p)\mathbf{1}_p$.

3.2.1 Example

We generate $n = 100$ iid observations from the model,

$$Y = \sin(2\pi X_1) + \sin(\pi X_2) + 0.5\epsilon, \quad (3.5)$$

where $p = 3$, $X_1, X_2, X_3 \sim U(0, 1)$ and $\epsilon \sim \mathcal{N}(0, 1)$, independent of \mathbf{X} . The X_1 component has the same amplitude but oscillates twice as quickly as the X_2 component, and thus X_1 is more important in describing the variation in Y ; X_3 is an irrelevant predictor. Figure 3.1 illustrates how the inverse-bandwidth parameterization and constraint (3.3) encourages sparse solutions.

When $\tau = 1$, the smallest kernel bandwidth, h , permitted in $g(\cdot)$ is $h = 1/\tau = 1$, which results in considerable smoothing (recall that each X_j is scaled to have mean zero and unit

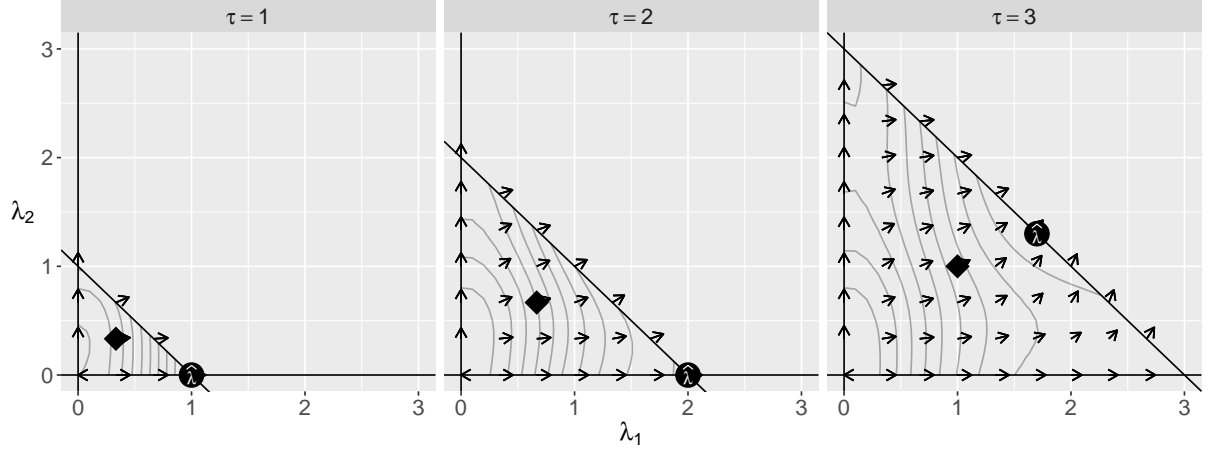


Figure 3.1 $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ contours and gradient vector fields of example model (3.5) for $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3 = \tau - \lambda_1 - \lambda_2)$ and $\tau \in \{1, 2, 3\}$; global maxima are denoted with solid circles and the neutral starting values $\boldsymbol{\lambda}_{\text{start}} = (\tau/p)\mathbf{1}_p$ are denoted with solid diamonds.

variance before fitting, so at $\tau = 1$ the data and kernel weights have equal variances). When such a smooth model is forced, the maximizer of $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ rests in a corner of the feasible region defined by the constraints at $\widehat{\boldsymbol{\lambda}}_\tau = (1, 0, 0)$. At this solution, both λ_2 and λ_3 have infinite kernel bandwidths and X_2 and X_3 are selected out. When $\tau = 2$, maximizing $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ still results in the solution $\widehat{\boldsymbol{\lambda}}_\tau = (\tau, 0, 0)$, however, the contours hint at the importance of X_2 by bending along the diagonal boundary. When more roughness is permitted at $\tau = 3$, the maximizer slides along the boundary and splits τ between λ_1 and λ_2 , leaving $\lambda_3 = 0$ (note that solutions along the line $\lambda_2 = \tau - \lambda_1$ imply that $\lambda_3 = 0$). As τ increases, the maximizer approaches $(\tau/3)\mathbf{1}_3$ and results in overfitting (plot not shown).

3.2.2 Tuning and Solution Paths

An optimal τ is chosen via small-sample nonparametric AIC, AIC_c , suggested in [43] resulting in a sparse inverse-bandwidth solution $\widehat{\boldsymbol{\lambda}}_\tau$. Cross-validation is prohibitively slow. In preliminary simulation studies, AIC_c worked as well or better than other criteria [37]. The degrees of freedom

are approximated by $\text{tr}(\mathbf{S}_\tau)$, where \mathbf{S}_τ is the $n \times n$ smoothing matrix with $[r, s]$ element,

$$\mathbf{S}_\tau[r, s] = \frac{\prod_{j=1}^p \exp\{-\lambda_j^2(X_{s,j} - X_{r,j})^2/2\}}{\sum_{k=1}^n \prod_{j=1}^p \exp\{-\lambda_j^2(X_{r,j} - X_{k,j})^2/2\}} \Big|_{\lambda=\hat{\lambda}_\tau}.$$

Thus, $\hat{\tau}$ minimizes,

$$\text{AIC}_c(\tau) = \ln \left\{ -\hat{L}_{\text{SEL}}(\hat{\lambda}_\tau) \right\} + \frac{n + \text{tr}(\mathbf{S}_\tau)}{n - \text{tr}(\mathbf{S}_\tau) - 2}.$$

In practice, we first compute $\hat{\tau}_0 = \text{argmin}_{\tau \in \tau^*} \text{AIC}_c(\tau)$, where τ^* is a predetermined coarse grid. Then we create a finer τ^* grid around $\hat{\tau}_0$ and repeat the search for the final $\hat{\tau}$.

The plot of $\hat{\lambda}_\tau$ versus τ is an inverse-bandwidth solution path, similar to LASSO solution paths. To illustrate, again consider the example from Section 3.2.1 except with two additional irrelevant predictors (X_4 and X_5). The solution path is shown in Figure 3.2; solid dots represent active predictors and open dots represent irrelevant predictors. Overlaid is the scaled $\text{AIC}_c(\tau)$ curve (dashed line). Predictor indices are shown in the right margin. In this example, $\hat{\tau} = 5$, for which $\hat{\lambda}_1, \hat{\lambda}_2 > 0$ and $\hat{\lambda}_3 = \hat{\lambda}_4 = \hat{\lambda}_5 = 0$ (perfect selection after tuning). Note that at the final solution, the inverse-bandwidth associated with the more rapidly varying predictor (X_1) is larger than the more slowly varying one (X_2), as expected.

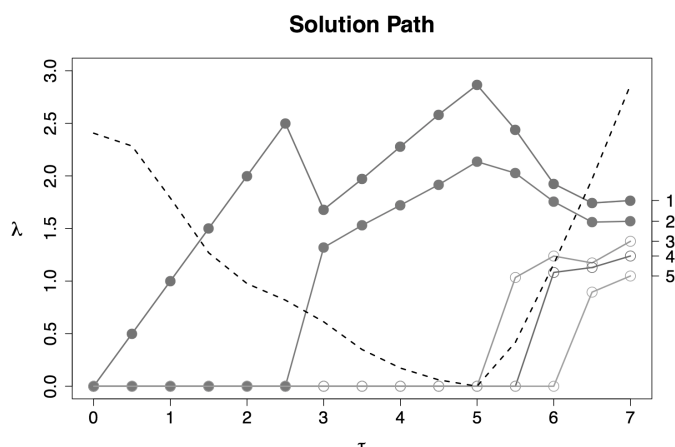


Figure 3.2 Solution paths of $\hat{\lambda}_\tau$ versus τ for Section 3.2.1 example with two active (solid) and three irrelevant (open) predictors. Dashed line: scaled $\text{AIC}_c(\tau)$, $\tau \in \tau^* = \{0, 0.5, \dots, 7\}$.

3.3 Extension to Categorical Predictors

Let $\mathcal{C} = \{j : X_j \text{ is continuous}\}$ and $\mathcal{D} = \{j : X_j \text{ is categorical}\}$. If $j \in \mathcal{D}$ then assume without loss of generality that X_j takes values in the label set $\{0, \dots, D_j - 1\}$ where there is no natural ordering. The ‘frequency approach’ [see 47] estimates a separate regression function for each permutation of observed discrete variables, but reduces the effective sample size of each separate estimator by a factor of approximately $\prod_{j \in \mathcal{D}} D_j^{-1}$. We describe an extension of MEKRO for mixed continuous and categorical variables based on the approach in [58]. The kernel for smoothing categorical X_j is

$$l_j(X_j, x_j) = (1 - \delta_j) \mathbf{1}_{X_j \neq x_j}, \quad (3.6)$$

where $\delta_j \in [0, 1]$. If $\delta_j = 0$, l_j is identically equal to 1 and does not depend on X_j . If $\delta_j = 1$, l_j is zero unless $X_j = x_j$. Any $\delta_j \in (0, 1)$ smooths the effect of covariate j , borrowing weight across the D_j different values of x_j .

Simply letting δ_j play the role of λ_j in the MEKRO algorithm fails because δ_j is bounded above by 1; thus, continuous and categorical predictors would be penalized unequally by the sum constraint in (3.3) because of the scaling differences. To alleviate the scaling problem, we propose the univariate categorical kernel

$$k_j^d(X_j, x_j) = \exp\left(-\frac{1}{2}\lambda_j^2 w_j \mathbf{1}_{X_j \neq x_j}\right), \quad (3.7)$$

where λ_j is the same inverse bandwidth parameter used throughout this paper, and w_j is a weight. This is similar to the continuous kernel, except that the indicator and weight replace $(X_{k,j} - X_{i,j})^2$. To weight the categorical and continuous kernels similarly, note that if $j \in \mathcal{C}$, $\mathbb{E}[(X_{k,j} - X_{i,j})^2] = 2$ for $i \neq k$. If $j \in \mathcal{D}$, and again for $i \neq k$, $\mathbb{E}[\mathbf{1}_{X_{k,j} \neq X_{i,j}}] = 1 - P(X_{k,j} = X_{i,j}) = 1 - \sum_{t=1}^{D_j} [P(X_{k,j} = t)]^2$. Then set $w_j = 2/[1 - \sum_{t=1}^{D_j} \{\widehat{P}(X_{k,j} = t)\}^2]$ where $\widehat{P}(X_{k,j} = t) = n^{-1} \sum_{k=1}^n \mathbf{1}_{X_{k,j}=t}$. The weight requires that realizations be spread across two or more categories. When the data are balanced across the D_j categories, the weight reduces to $w_j = 2D_j/(D_j - 1)$. Observe that, like

(3.6), $\lambda_j = 0$ implies that categorical covariate j is selected out, and λ_j large implies $\hat{g}(\cdot, \boldsymbol{\lambda})$ is different for each category in D_j .

The estimator for $g(\cdot)$ incorporating categorical variables is then

$$\hat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) = \frac{\sum_{k=1}^n Y_k \prod_{j \in \mathcal{C}} \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2 / 2\} \prod_{j \in \mathcal{D}} \exp(-\lambda_j^2 w_j \mathbb{1}_{X_{kj} \neq X_{ij}} / 2)}{\sum_{k=1}^n \prod_{j \in \mathcal{C}} \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2 / 2\} \prod_{j \in \mathcal{D}} \exp(-\lambda_j^2 w_j \mathbb{1}_{X_{kj} \neq X_{ij}} / 2)},$$

where w_j is described above. This is substituted into (3.1) and optimized under (3.3) by methods in Section 3.2.

3.4 Method Comparison and Numerical Results

Much of the nonparametric regression methodology incorporating variable importance can be separated into two classes: methods that downweight features with little or no effect, and methods that perform feature subset selection. Arguments can be made for either, based on either modeling philosophy or the particular application. It is unlikely that any judicious real-world regression application includes *truly* irrelevant variables, and downweighting can be superior to selection for predictions when there are a larger number of small effects present [ridge regression vs. LASSO in 75]. On the other hand, sparsity attained from selection is valuable for parsimonious model descriptions, avoiding the curse of dimensionality [46], and predictions where there are only a few large effects.

MEKRO falls into the selection class, along with several other popular methods. Friedman [23] developed MARS, a method that flexibly estimates models using a basis of linear splines with one knot, but it is prone to overfitting [4]. COSSO extends smoothing spline ANOVA models to perform selection by penalizing a least-squares loss similar to that of LASSO [50, 75]. Adaptive COSSO uses an adjusted weighting scheme analogous to the adaptive LASSO [70]. Both versions of COSSO typically truncate the model complexity at or below two-way interactions. SPAM (sparse additive models) is similar to COSSO in that it truncates complexity, but it allows $p \gg n$ [59]. Kernel iterative feature extraction (KNIFE) by Allen [2] imposes L_1 -regularization on L_2 -penalized splines.

Many of the downweighting methods are similar to MEKRO by attaching individual weights to the separate input dimensions in a flexible model. Automatic relevance determination (ARD) first described by [54] puts prior distributions on weights for each input in a Bayesian neural network, and input weights of only irrelevant predictors remain concentrated around 0. [80] put weights on the distance metric for each input dimension in the covariance function of a Gaussian process and demonstrate results similar to ARD. [31] add noise to each input of an artificial neural network and use the harmonic mean to control the total noise added. They show greatly reduced generalization errors against trees and k -nearest neighbors on classification problems, but do not consider examples with irrelevant inputs. Adaptive metric kernel regression [AMKR; 29] is a kernel regression bandwidth selection procedure that parameterizes the local-constant estimator with inverse-bandwidths. However, it directly optimizes the leave-one-out cross-validation loss instead of MEKRO’s approach of choosing an optimal smoothness from an entire path of solutions with sparsity via cross-validation. RODEO [46] thresholds derivatives of the local-linear estimator to keep bandwidths associated with irrelevant variables large.

3.4.1 Simulation Preliminaries

This section presents numerical studies on the performance of MEKRO (MEK) against other variable selection methods for nonparametric regression, including KNIFE (KNI), two “regular” COSSO variants (additive COSSO, RC1; two-way interaction COSSO, RC2), two adaptive COSSO, or ACOSSO variants (additive ACOSSO, AC1; two-way interaction ACOSSO, AC2), and MARS (additive, M1; two-way interaction, M2; three-way interaction, M3). We use the default GCV criterion for MARS. For KNIFE, we fix $\lambda_1 = 1$ and use a radial kernel with $\gamma = 1/p$ as suggested in [2]. The weight power for the ACOSSO is fixed at $\gamma = 2$, as suggested by [70]. Although these parameters serve as additional tuning parameters, we tune only one parameter per method for fairness. We also include AMKR (AM) because of its close relationship with MEKRO.

Each simulation sets $Y_i = g(\mathbf{X}_i) + \epsilon_i$ where $\epsilon_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2)$ and $g(\cdot)$ is defined for each model.

We set σ_ϵ^2 so that the theoretical model $R^2 = \text{Var}\{g(\mathbf{X})\}/[\text{Var}\{g(\mathbf{X})\} + \sigma_\epsilon^2]$ is 0.75 (a SNR of 3) unless otherwise noted. The predictors are generated as $X_j = (U_j + kU^*)/(1 + k)$; $U_j \sim U(0, 1)$, $j = 1, \dots, p$; $U^* \sim U(0, 1)$, so that $X_j \in [0, 1]$ and \mathbf{X} has compound symmetric correlation $\rho = k^2/(1 + k^2)$. The covariates are independent when $\rho = 0$.

Results are summarized in terms of Type I selection error (irrelevant predictor inclusion rate), Type II selection error (active predictor exclusion rate), and average integrated squared error (AISE) over $M = 100$ Monte Carlo (MC) replications. AISE estimates $\text{MISE} = \text{E}\{\widehat{g}(\mathbf{X}, \widehat{\boldsymbol{\lambda}}_{\widehat{\tau}}) - g(\mathbf{X})\}^2 = \text{E}_{\mathbf{T}}[\text{E}\{\widehat{g}(\mathbf{X}, \widehat{\boldsymbol{\lambda}}_{\widehat{\tau}}) - g(\mathbf{X})\}^2 \mid \mathbf{T}]$ by averaging the mean squared difference between $\widehat{g}(\cdot, \widehat{\boldsymbol{\lambda}}_{\widehat{\tau}})$ and $g(\cdot)$ evaluated on test data over 100 MC replicates, where \mathbf{T} is random training data used in defining the estimator $\widehat{g}(\cdot, \widehat{\boldsymbol{\lambda}}_{\widehat{\tau}})$, and \mathbf{X} and \mathbf{T} follow the same distribution. We use a test set of 10,000 \mathbf{X} data vectors for each model and vary the dimension of \mathbf{T} as a simulation factor. AISE comparison plots (Fig. 3.3, 3.5-3.10) show 95% confidence bars for the MISE of each method. MISEs with non-overlapping confidence bars are statistically different based on the more powerful paired-difference test (not shown).

We give a measure of predictor effect size to provide additional context to regression functions and simulation results. In models with complex interactions, it is difficult to quantify the contribution of each covariate to the regression function variance. We quantify effect size with the scaled root mean squared risk difference between the regression function with and without X_j is replaced by its mean. Define $\nu_j = \text{E}\{g(\mathbf{X}) - g(\mathbf{X}|_{X_j=\text{E}(X_j)})\}^2$. Then the predictor effect size for X_j is defined as $\nu_j^{1/2} / \max\{\nu^{1/2}\}$. We compute effect sizes to two decimal places via numerical integration. When this method is applied to linear regression, the effect sizes are the scaled absolute regression coefficients. Predictors with near-zero effect sizes are effectively irrelevant and inflate Type II selection errors for all methods.

As an example, consider two models used in [23] for assessing variable selection and prediction

performance of MARS,

$$Z(X_1, X_2, X_3, X_4) = \left\{ X_1^2 + \left(X_2 X_3 - \frac{1}{X_2 X_4} \right)^2 \right\}^{1/2}, \text{ and}$$

$$\phi(X_1, X_2, X_3, X_4) = \arctan \left\{ \frac{X_2 X_3 - 1/(X_2 X_4)}{X_1} \right\},$$

where $X_1 \in [0, 100]$, $X_2 \in [40\pi, 560\pi]$, $X_3 \in [0, 1]$, and $X_4 \in [1, 11]$, all uniformly distributed. Both models contain all orders of interactions, although the contribution of the covariates to the model varies widely. Predictor effect sizes for active predictors (X_1, X_2, X_3, X_4) are $(0.03, 0.90, 1.00, 0.00)$ in $Z(\cdot)$ and $(0.55, 0.59, 1.00, 0.00)$ in $\phi(\cdot)$. Because the contributions from X_1 in $Z(\cdot)$ and X_4 in both models are so low, $Z(\cdot)$ is well-approximated by two-way interaction models, and $\phi(\cdot)$ is well-approximated by three-way interaction models. This was recognized in [50] after model fitting and observing the performance of a two-way interaction model against a saturated model.

3.4.2 Simulation Results

Model 1. Nonlinear, three-way interaction; $g(\mathbf{X}) = \sin\{2\pi(X_1 + X_2)/(1 + X_3)\}$; $p = 10$ (7 irrelevant variables included); $\rho = 0$; $n \in \{50, 100, 200, 400\}$. Predictor effect sizes for active predictors (X_1, X_2, X_3) are $(1.00, 1.00, 0.48)$. Selection errors are displayed in Table 3.1 and average integrated squared errors (AISEs) are in Figure 3.3. MEKRO (MEK) dominates in both prediction and selection, achieving perfect selection when $n \geq 100$, and having a comparable AISE to AMKR only when $n = 400$. AMKR (AM) overselects irrelevant covariates. KNIFE (KNI) has approximately the same AISE as the two-way interaction COSSO models (RC2, AC2), but greatly underselects for smaller n . The two-way interaction COSSO models show a clear advantage over the additive COSSO (RC1, AC1) models for prediction in larger samples; the large effect sizes for X_1 and X_2 and smaller effect size of X_3 indicates that $g(\cdot)$ is well-approximated by a two-way interaction model but not an additive model. The three-way interaction MARS model (M3) performs worst when $n \leq 100$, but demonstrates good selection rates for $n = 400$.

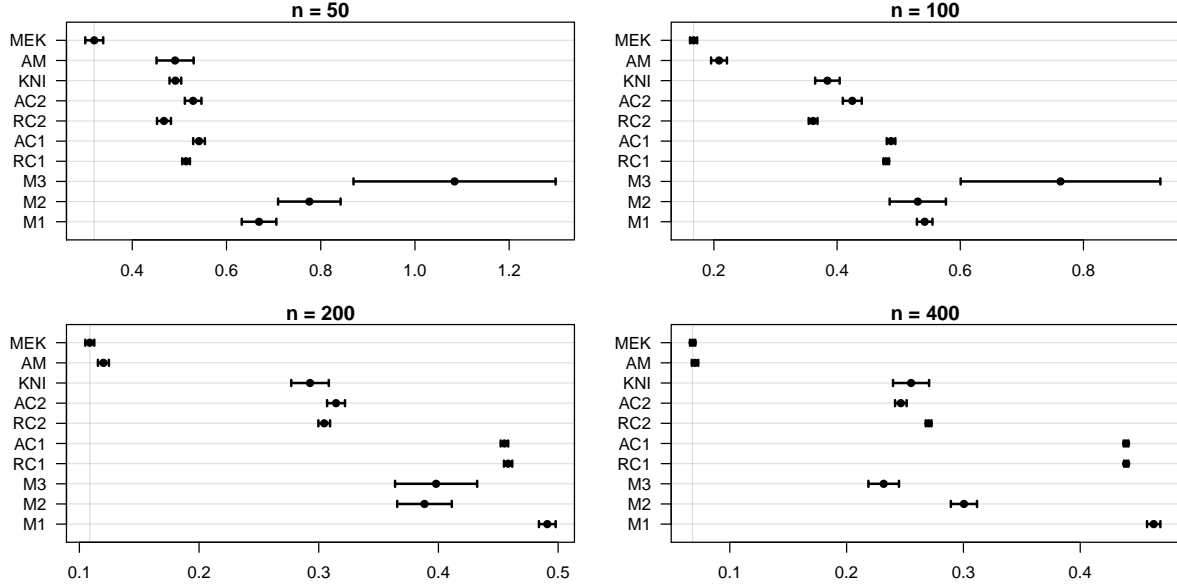


Figure 3.3 AISEs for Model 1. Note the scale differences. Out of the 400 MC samples, 3 AISE outliers are omitted from M3.

We now elaborate on the selection performance for adaptive metric kernel regression (AMKR). Simulation studies in [29] suggest that inverse-bandwidth estimates for irrelevant covariates are shrunk from AMKR, but are frequently positive (non-zero). Thus, for comparing to a selection method, one must select a cutoff to operationalize when an inverse-bandwidth is small enough to be selected out. Our simulation studies show that AMKR-estimated inverse-bandwidths are

Table 3.1 Selection error rates for Model 1. MC standard errors for all cells ≤ 0.03 .

	Error	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1
$n = 50$	Type I	0.05	0.55	0.09	0.14	0.19	0.29	0.19	0.56	0.54	0.70
	Type II	0.13	0.06	0.73	0.58	0.49	0.53	0.62	0.21	0.21	0.19
$n = 100$	Type I	0.00	0.46	0.05	0.24	0.11	0.27	0.12	0.49	0.52	0.64
	Type II	0.00	0.00	0.40	0.36	0.42	0.40	0.46	0.04	0.05	0.12
$n = 200$	Type I	0.00	0.48	0.03	0.26	0.08	0.31	0.07	0.28	0.38	0.65
	Type II	0.00	0.00	0.17	0.31	0.38	0.22	0.36	0.00	0.00	0.12
$n = 400$	Type I	0.00	0.52	0.03	0.22	0.05	0.30	0.04	0.07	0.22	0.65
	Type II	0.00	0.00	0.10	0.31	0.39	0.20	0.31	0.00	0.00	0.09

either near machine zero or large enough to be regarded as relevant. The left panel of Figure 3.4 shows the AMKR estimates for Model 1, $n = 400$, where X_4 through X_{10} are irrelevant and should each have $1/h = 0$. Of the 700 samples (100 MC replicates for 7 predictors), 48% of them had an AMKR estimate of 0. The right panel of Figure 3.4 shows the \log_{10} -value for the other 52% of estimates that were positive; many clump around $10^{-0.5}$, a smooth kernel bandwidth, but still large enough to be considered relevant. There are very few positive estimates below 10^{-4} even in this moderate sample size case, thus we chose a cutoff of 10^{-4} below which an AMKR inverse bandwidth was considered 0.

The high Type-I selection error for AMKR is reflective of a researcher being uncertain whether small inverse-bandwidths represent prunable features or not. However, despite this binary classification, small inverse-bandwidths will not greatly impact prediction error. In our

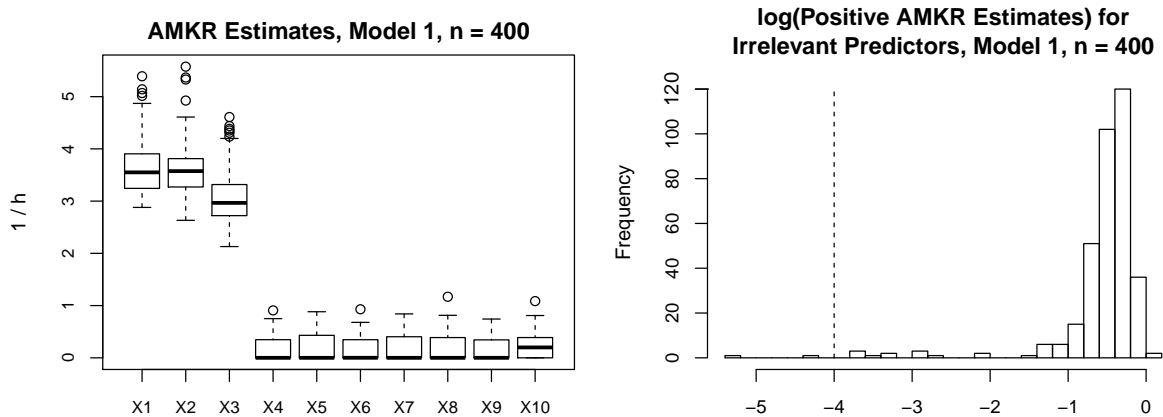


Figure 3.4 Study of AMKR estimates for Model 1, $n = 400$.

simulations, it is generally true that AMKR approaches MEKRO's prediction error as n increases, but the Type-I selection error remains high.

Model 2. Identical to Model 1 with $\rho = 0.5$. Predictor effect sizes for active predictors are (1.00, 1.00, 0.44) and 0 for irrelevant predictors. The selection errors and average integrated squared errors for this model are given in Table 3.2 and Figure 3.5, respectively. Again, MEKRO

(MEK) dominates in prediction and has the best selection rates for $n \geq 100$, including perfect

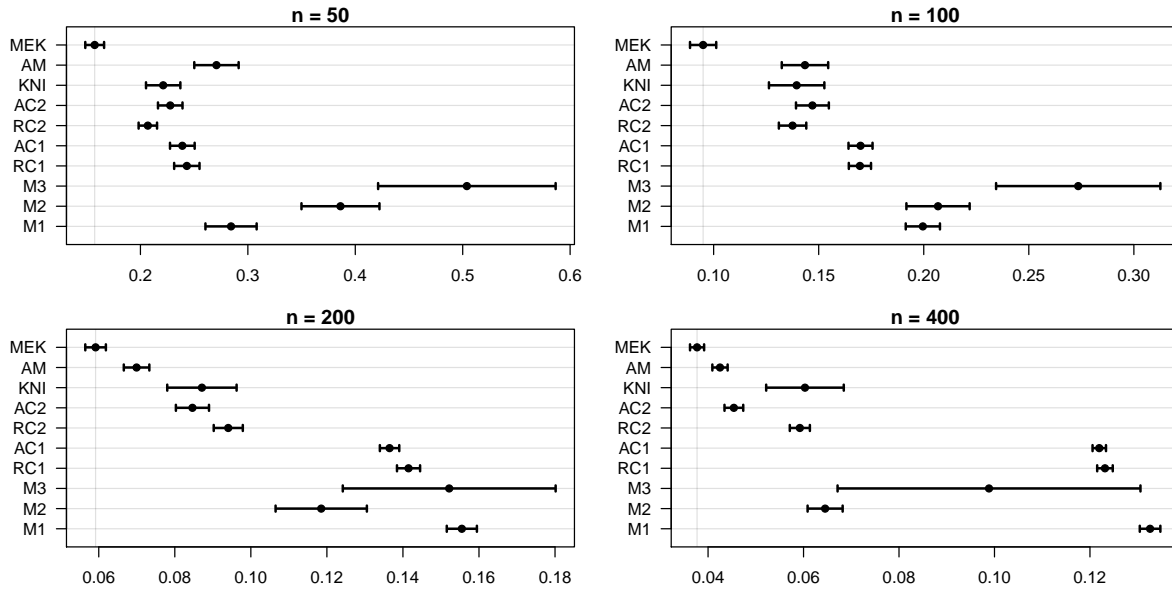


Figure 3.5 AISEs for Model 2. Note the scale differences. Out of the 400 MC samples, 19 and 6 AISE outliers are omitted from M3 and M2, respectively.

Table 3.2 Selection error rates for Model 2. MC standard errors for all cells ≤ 0.03 .

	Error	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1
$n = 50$	Type I	0.03	0.57	0.08	0.15	0.16	0.34	0.16	0.49	0.50	0.67
	Type II	0.27	0.05	0.49	0.51	0.47	0.26	0.34	0.07	0.06	0.05
$n = 100$	Type I	0.02	0.52	0.04	0.25	0.08	0.31	0.04	0.50	0.48	0.63
	Type II	0.06	0.00	0.28	0.34	0.43	0.11	0.21	0.00	0.00	0.02
$n = 200$	Type I	0.00	0.45	0.03	0.24	0.06	0.29	0.03	0.39	0.39	0.59
	Type II	0.00	0.00	0.11	0.30	0.42	0.05	0.18	0.00	0.00	0.01
$n = 400$	Type I	0.00	0.48	0.08	0.24	0.03	0.35	0.01	0.26	0.24	0.55
	Type II	0.00	0.00	0.04	0.26	0.38	0.00	0.04	0.00	0.00	0.00

selection for $n \geq 200$. The other models show improvements in prediction with correlated predictors because the three-way interaction in $g(\cdot)$ can be approximated more accurately by

one- or two-way interactions. However, only additive COSSO (RC1) and KNIFE (KNI) show selection errors comparable to MEKRO at $n = 400$. Generally, but especially in the presence of correlation, three-way interaction MARS (M3) can produce unstable predictions by including near-degenerate basis functions in the training fit.

In response to a reviewer comment, we replicated Model 2 but changed the covariate distribution from uniform to Gaussian. Results of this additional experiment, described fully in Appendix C, show that prediction performance was adversely affected but that selection performance was not.

Model 3. Interaction model with categorical covariates; $g(\mathbf{X}) = \arctan[10\{X_1(2X_3 - 1) + X_2\}/(-\mathbb{1}_{X_4=0} + \mathbb{1}_{X_4=1} + 2\mathbb{1}_{X_4=2})]$; X_1, X_2 continuous, $X_3 \in \{0, 1\}, X_4 \in \{0, 1, 2\}; p = 10; \rho = 0; n \in \{50, 100, 200, 400\}$. Irrelevant predictors $X_5 \in \{0, 1\}, X_6 \in \{0, 1, 2\}, X_7 \in \{0, 1, 2, 3\}$, and X_8, X_9, X_{10} are continuous. All of the discrete covariates follow a discrete uniform distribution and the continuous covariates are generated in the same manner as above. Predictor effect sizes for active predictors (X_1, X_2, X_3, X_4) are (0.38, 0.37, 0.65, 1.00). Both MARS and COSSO are designed to handle categorical covariates without modification. AMKR (AM) does not include a kernel for categorical covariates, however, it will still approximate the ‘frequency approach’ [47] as n and thus the inverse-bandwidths grow. Selection errors are displayed in Table 3.3 and average integrated squared errors are in Figure 3.6. MEKRO’s good prediction and selection performance apparent in Table 3.3 and Figure 3.6 support the definition of the weights in (3.7). The only competitor to MEKRO on prediction is AMKR when $n = 400$, lending insight that kernel regression is well-suited to pick up the complexities in this model.

Model 4. This example uses the functions $Z(\cdot)$ and $\phi(\cdot)$ taken from [23]; see Section 3.4.1 for a description. We add a variable selection aspect to the original simulation in Friedman by including six additional irrelevant covariates having iid $U(0, 1)$ distributions, for a total of ten covariates. We also increase the residual error so the model R^2 is 0.75 (lowering the signal-to-noise ratio from 9 to 3) to match Models 1-3. From Section 3.4.1, we know that X_1 in $Z(\cdot)$ and X_4 in both models are essentially irrelevant predictors, and we consider them as

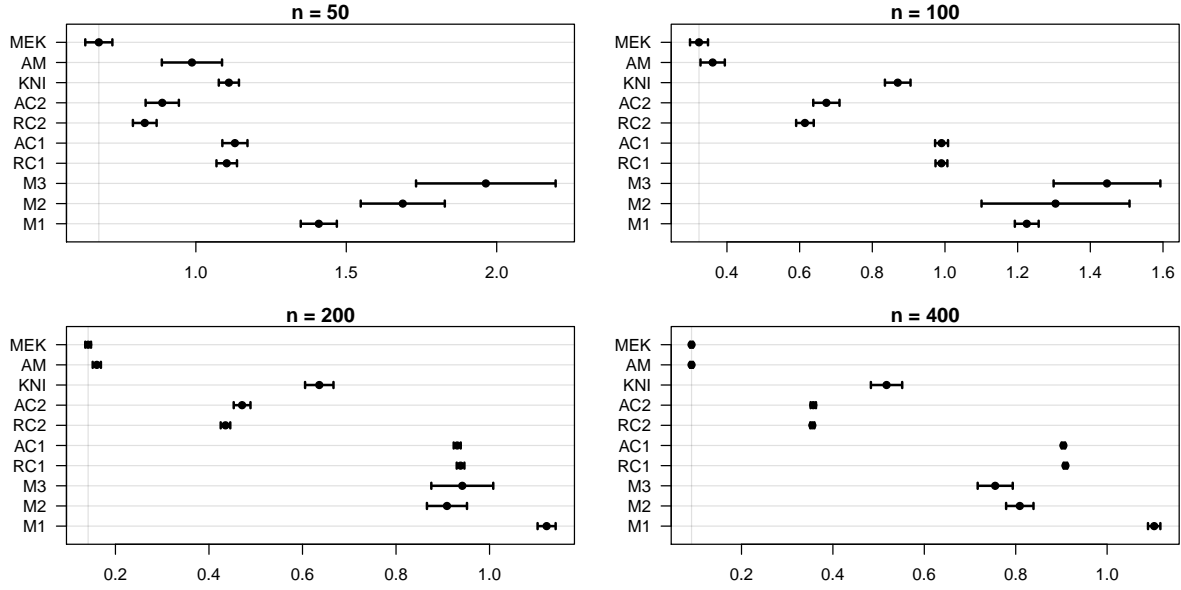


Figure 3.6 AISEs for Model 3. Note the scale differences. Out of the 400 MC samples, 2 and 1 AISE outlier(s) are omitted from M3 and M2, respectively.

Table 3.3 Selection error rates for Model 3. MC standard errors for all cells ≤ 0.03 .

	Error	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1
$n = 50$	Type I	0.07	0.57	0.24	0.14	0.11	0.23	0.27	0.55	0.54	0.65
	Type II	0.36	0.06	0.47	0.68	0.70	0.60	0.59	0.19	0.19	0.22
$n = 100$	Type I	0.03	0.51	0.17	0.21	0.12	0.27	0.24	0.44	0.49	0.58
	Type II	0.10	0.00	0.39	0.54	0.64	0.54	0.56	0.10	0.11	0.21
$n = 200$	Type I	0.00	0.46	0.06	0.23	0.10	0.26	0.21	0.31	0.38	0.54
	Type II	0.00	0.00	0.24	0.43	0.52	0.48	0.54	0.02	0.02	0.18
$n = 400$	Type I	0.00	0.51	0.03	0.18	0.04	0.29	0.29	0.17	0.31	0.52
	Type II	0.00	0.00	0.15	0.37	0.49	0.33	0.38	0.01	0.01	0.11

irrelevant when calculating selection error rates.

Selection error rates for Model 4 are displayed in Table 3.4. Average integrated squared errors (AISE) are shown in Figure 3.7; AISEs too large to display in the plot windows are indicated by dashed horizontal lines. Although MEKRO (MEK) exhibits very good selection rates for both sample sizes and response functions, it falls short in predictions to COSSO (RC and AC variants) depending on the setup. MEKRO suffers from the same boundary effect problems

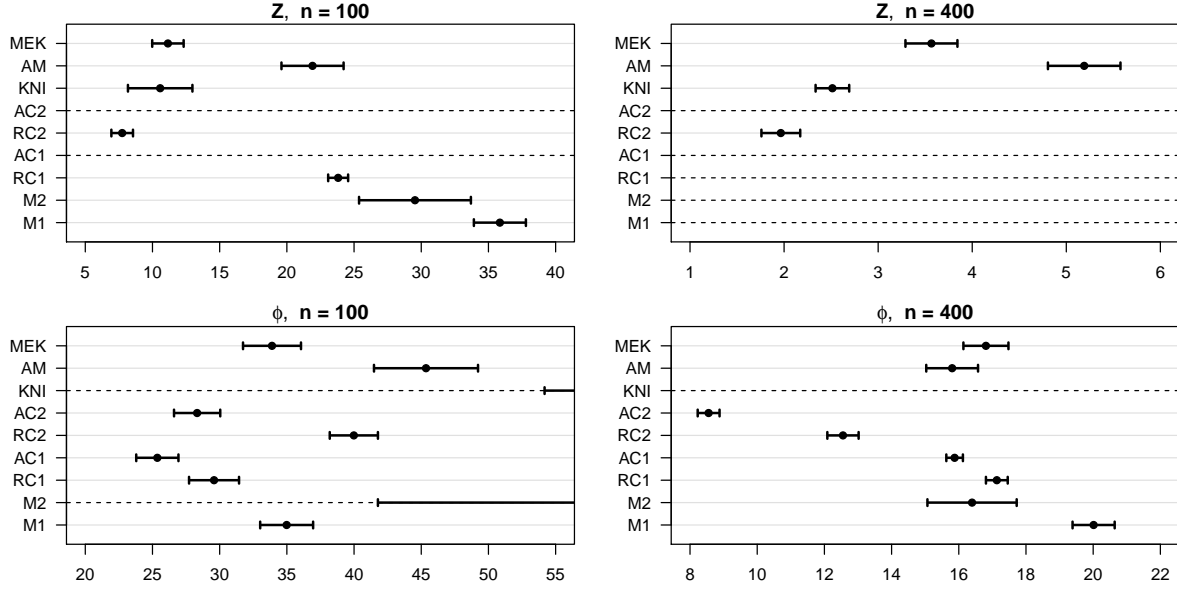


Figure 3.7 AISEs for Model 4; units for $Z(\cdot)$ plots are 10^3 and units for $\phi(\cdot)$ plots are 10^{-3} . Note the scale differences. Dashed lines indicate methods with AISEs too large to display.

Table 3.4 Selection error rates for Model 4, with X_1 in $Z(\cdot)$ and X_4 classified irrelevant. MC standard errors for all cells ≤ 0.04 .

	Error	MEK	AM	KNI	AC2	RC2	AC1	RC1	M2	M1
$Z(\cdot)$	Type I	0.03	0.49	0.03	0.49	0.05	0.08	0.15	0.60	0.67
$n = 100$	Type II	0.00	0.00	0.48	0.02	0.00	0.09	0.00	0.00	0.00
$Z(\cdot)$	Type I	0.01	0.50	0.01	0.04	0.03	0.01	0.13	0.58	0.63
$n = 400$	Type II	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
$\phi(\cdot)$	Type I	0.05	0.54	0.00	0.27	0.09	0.28	0.10	0.54	0.64
$n = 100$	Type II	0.02	0.00	0.56	0.16	0.32	0.00	0.01	0.00	0.01
$\phi(\cdot)$	Type I	0.00	0.51	0.12	0.21	0.01	0.18	0.04	0.37	0.59
$n = 400$	Type II	0.00	0.00	0.26	0.01	0.06	0.00	0.00	0.00	0.00

as the Nadaraya-Watson estimator [61]. Both $Z(\cdot)$ and $\phi(\cdot)$ vary rapidly near their boundary points (see [23] for surface plots), inflating MEKRO's prediction error rate. Examining a plot of $Z(\cdot, X_2, X_3, \cdot)$ (not shown) reveals that much of the surface variation is attributed to the X_2X_3 interaction. The additive COSSOs (RC1, AC1) cannot pick this effect out and predict poorly. When $n = 100$, the weights in two-way interaction ACOSSO (AC2) reduce the component

penalties too far and irrelevant covariates are overly selected. Even in the larger sample size, when two-way interaction ACOSSO selects well, the weights impart too much component variation leading to poor predictions. The two-way interaction COSSO (RC2) performs well.

Model 5. For this model, data are generated from the deterministic function describing the kinematics of the Puma 560 robotic arm (the data are available from the DELVE¹ data repository; see www.cs.toronto.edu/~delve/data/pumadyn/desc.html² for details). The arm has six independently-operating joints. The goal is to estimate the linear acceleration in Joint 3, given the position, velocity, and torque of all of the joints. This example sets several parameters to zero to reduce the number of active covariates to eight. We append four irrelevant covariates to judge selection performance. DELVE adds noise to both the input parameters and the response in two levels, medium and high ('pumadyn-8nm' and 'pumadyn-8nh' respectively in DELVE). We cannot estimate predictor effect sizes because we do not have access to the data generating function.

There are $N = 8192$ observations available. Because we do not have the luxury of generating a test data set, we randomly select n training observations without replacement and use the remaining $N - n$ samples to estimate the conditional squared prediction error, $\widehat{\text{SPE}} = (N - n)^{-1} \sum_{i=1}^{N-n} \{Y_i - \widehat{g}(\mathbf{x}_i)\}^2$. The sampling process is repeated 100 times, and the average of the $\widehat{\text{SPE}}$ values, the ASPE, estimates the squared prediction error. We report results for training sizes of $n = 100, 200$.

ASPEs are given in Figure 3.8 and main effect selection rates, the proportion of main effects selected out of the 100 MC samples, are given in Table 3.5. Note that these are not selection errors as shown on the previous tables. Interaction effect selection rates are excluded because we do not know which interactions are weak and effectively irrelevant. Table 3.5 shows selection rates averaged over the four simulations (main effect selection rates are similar across the four simulations). The 'IRR' row is the average inclusion rate for the four irrelevant covariates that are extraneous to the original data set.

¹Copyright (c) 1995-1996 by The University of Toronto, Toronto, Ontario, Canada. All Rights Reserved.

²Updated: 08 Oct. 1996. Accessed: 02 Mar. 2014.

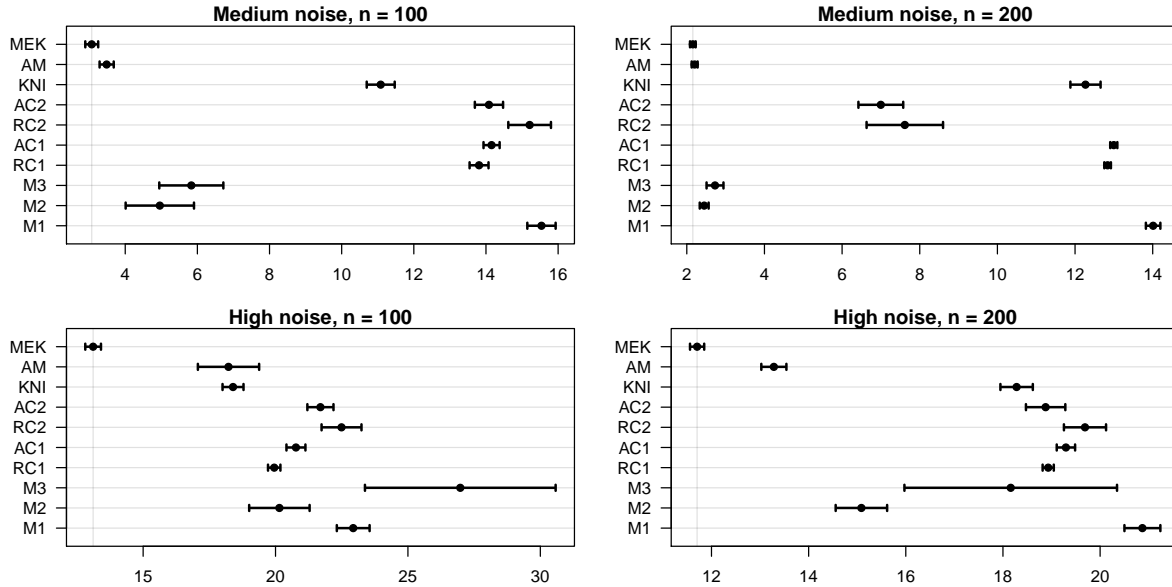


Figure 3.8 ASPEs (average squared prediction errors) for Model 5. Note the scale differences. Out of the 400 MC samples, 3 ASPE outliers are omitted from both M3 and M2.

Table 3.5 Main effect selection rates (not errors) for Model 5, averaged over the four simulation settings. The IRR row is the average selection rate for the four irrelevant predictors that were independently generated. MC standard errors for all cells ≤ 0.03 .

	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1
X_1	0.01	0.49	0.62	0.41	0.44	0.22	0.09	0.47	0.29	0.01
X_2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_3	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.99	0.97
X_4	0.01	0.52	0.66	0.50	0.52	0.25	0.09	0.48	0.30	0.01
X_5	0.01	0.55	0.66	0.66	0.65	0.24	0.09	0.44	0.29	0.01
X_6	0.01	0.50	0.60	0.44	0.47	0.26	0.10	0.48	0.32	0.01
X_7	0.01	0.48	0.62	0.39	0.44	0.23	0.09	0.46	0.31	0.01
X_8	0.01	0.48	0.69	0.45	0.45	0.24	0.09	0.47	0.30	0.01
IRR	0.01	0.51	0.65	0.41	0.44	0.25	0.10	0.16	0.12	0.01

MEKRO includes X_2 and X_3 , the positions of the second and third joints, on every replicate, and excludes every other variable at a very high rate. Additive MARS (M1) shows a very similar selection performance. KNIFE, AMKR, and the four COSSO variants (KNI, AM, RC and AC) show generally higher selection rates for both active and irrelevant predictors, suggesting that the selection procedures are discriminating poorly. The two-way and three-way interaction

MARS (M2 and M3) models show elevated active covariate selection rates, while keeping the irrelevant covariate selection rate low. Despite only selecting two of the eight active covariates, MEKRO has a better prediction rate than any other method, including the two MARS methods that show better covariate discrimination.

[29] benchmark AMKR (AM) against an artificial neural network (without ARD) and Gaussian process on the same Puma DELVE data sets, giving us indirect comparisons on prediction error. AMKR outperformed the artificial network, suggesting that MEKRO would do the same if ARD is not implemented. The Gaussian process generally predicted better than AMKR by 2-5% (quadratic loss comparison as a percentage) for n near 100 or 200, indicating that MEKRO would enjoy the best prediction rates in the high noise scenario and similar prediction rates in the medium noise scenario.

Prostate Data Example. The data are from a study of 97 men with prostate cancer [66] and were used in the original LASSO paper [75]. The data contain the log level of a prostate-specific biomarker (response) along with eight other clinical measures (predictors): log cancer volume, log prostate weight, age, log benign prostatic hyperplasia amount, seminal vesicle invasion (binary), log capsular penetration, Gleason score, and percentage of Gleason scores equal to 4 or 5.

Table 3.6 Main effect selection rates (not errors) for the prostate data. ‘Avg Model’ is the method’s average model size; ‘Corr’ is the selection rate correlation of each method with LAS. MC standard errors for all selection rates ≤ 0.07 and all average model sizes ≤ 0.55 .

	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1	LAS
X_1	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X_2	0.78	0.99	0.76	0.71	0.77	0.95	0.96	0.99	0.98	1.00	1.00
X_3	0.01	0.46	0.00	0.44	0.52	0.53	0.54	0.86	0.83	0.96	0.48
X_4	0.29	0.75	0.19	0.44	0.46	0.76	0.70	0.84	0.82	0.91	0.75
X_5	0.64	0.91	0.82	0.94	0.90	0.99	0.97	0.82	0.84	0.97	1.00
X_6	0.01	0.58	0.01	0.33	0.12	0.39	0.40	0.81	0.80	0.89	0.21
X_7	0.14	0.71	0.25	0.21	0.17	0.30	0.51	0.26	0.32	0.27	0.39
X_8	0.10	0.52	0.11	0.41	0.42	0.48	0.53	0.71	0.73	0.90	0.65
Avg Model	2.97	5.92	3.12	4.48	4.36	5.40	5.61	6.29	6.32	6.90	5.48
Corr	0.89	0.80	0.86	0.87	0.93	0.94	0.96	0.55	0.60	0.51	1.00

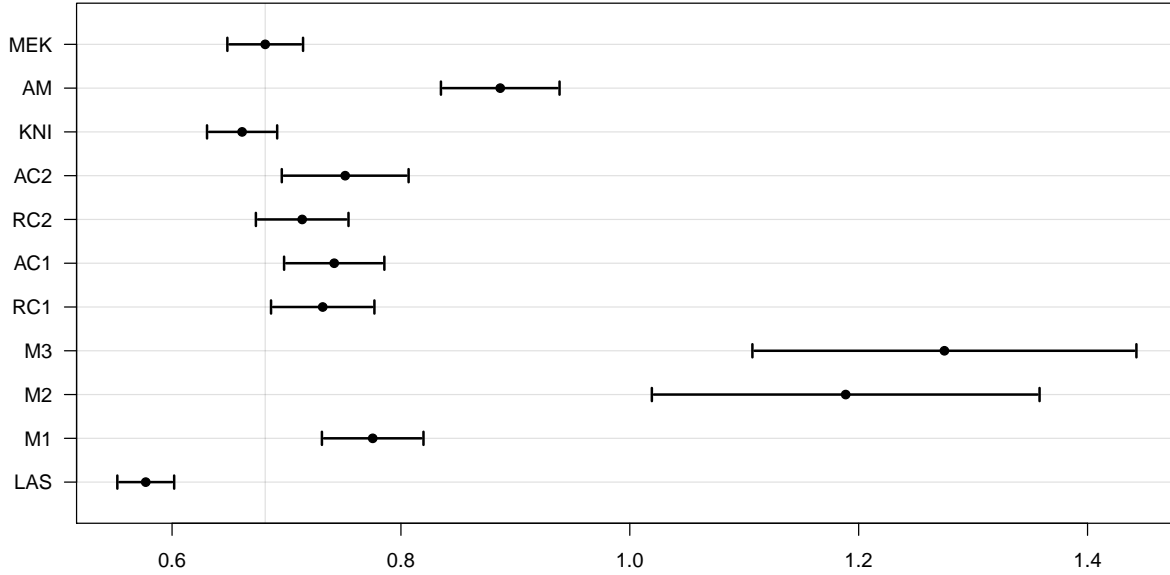


Figure 3.9 ASPEs (average squared prediction errors) for the prostate data. Out of the 100 MC samples, 2 and 4 ASPE outliers are omitted from M3 and M2, respectively.

We evaluate the nonparametric methods by training on two-thirds of the data and evaluating the predictions on the remaining third. This process is repeated 100 times and the squared prediction errors are averaged. We also include LASSO, tuned with 10-fold cross-validation, and evaluate it in the same way.

The average squared prediction errors (ASPE) are given in Figure 3.9 and the selection rates (not errors) are given in Table 3.6. Predictions in the prostate data favor simpler methods as evidenced by LASSO and additive MARS (M1, versus M2 and M3, the higher-order MARS methods). Among the nonparametric methods, MEKRO (MEK) and KNIFE (KNI) have the smallest average model size while maintaining a low prediction error and high correlation with LASSO (LAS) selection. The MARS methods overfit and have high prediction errors. All COSSO (RC and AC variants) methods perform similarly in terms of prediction, selection, and correlation with LASSO, and have both higher average model sizes and higher prediction errors than MEKRO.

3.5 Asymptotic Results

Consider the model $Y = g(\mathbf{X}) + \epsilon$, where $\text{Var}(\epsilon|\mathbf{X}) = \sigma_\epsilon^2$ and $g(\mathbf{x}) = E(Y|\mathbf{X} = \mathbf{x})$. Define the important predictor set $\mathcal{I} = \{j : X_j \text{ is important in } g(\cdot)\}$ and so its complement \mathcal{I}^c is the set of unimportant predictors. We argue in Appendix 3.7.2 that we can generally expect, if $\tau \rightarrow \infty$ and satisfies $\tau^{|\mathcal{I}|+4}/n \rightarrow 0$ and $\tau^p \log(n)/n \rightarrow 0$ as $n \rightarrow \infty$, then the maximizer $\hat{\lambda}$ of (3.1) subject to the constraints (3.3) satisfies $\hat{\lambda}_j \rightarrow \infty$ and $\hat{\lambda}_{j'} \rightarrow 0$ in probability for $j \in \mathcal{I}$ and $j' \in \mathcal{I}^c$, i.e., the MEKRO asymptotically discriminates important from unimportant predictors and achieves variable selection consistency.

3.6 Discussion

We developed a new method for performing simultaneous variable and bandwidth selection in nonparametric regression using the SWW [68] paradigm. The resulting method is kernel regression with a novel bandwidth estimator (MEKRO). The bandwidth selection strategy is such that certain bandwidths are set to infinity (inverse bandwidth of 0), thereby allowing for complete removal of variables from the model. It is also attractive in that it does not require a complexity truncation and can fit models with many interactions. Simulation studies show that MEKRO is a viable option for selection and prediction generally, and especially useful when the underlying model is nonlinear with complex interactions.

Measurement error model selection likelihoods in linear models share a connection with LASSO, and also perform well when used for nonparametric classification. Although current implementations of the SWW approach focus on estimators with closed-form selection likelihoods, the favorable performance of such estimators suggests further study of selection likelihoods in more complex cases.

Despite the advantages of the new selection strategy, MEKRO is a local-constant kernel regression estimator and does not avoid the known drawbacks of the method. Future work will address boundary corrections, and an adaptive-bandwidth MEKRO that we suspect will

boost prediction performance. Also, the scope of MEKRO can be expanded by adapting an ordinal kernel analogous to that in [58] or allowing different response types. It is likely that major computational gains can be realized by implementing an approximate MEKRO that takes advantage of binning.

Acknowledgements. We thank the referees, Associate Editor, and Editor for alerting us to additional references and for their thoughtful comments and suggestions that greatly improved the paper.

3.7 Appendix

This section contains details copied verbatim from the supplemental files of [79].

3.7.1 MEKRO Selection Likelihood Derivation

[68] proposed a four-step approach for building a measurement error model (MEM) selection likelihood from any ‘traditional’ likelihood of covariates and a response. The measurement error kernel regression operator (MEKRO) is derived from these steps, and so they are included below for completeness. See [68] for comprehensive details on the motivation for MEM selection likelihoods, their relationship to LASSO, and an application that yields a nonparametric classifier that performs variable selection. Let $\mathcal{D}_{\{\mathbf{a}\}}$ be a diagonal matrix with the vector \mathbf{a} on the diagonal. The MEM selection likelihood construction proceeds in four basic steps:

- S1. Start with an assumed ‘true’ likelihood for $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$, denoted $L_{\text{TRUE}}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ could be finite (parametric) or infinite dimensional (nonparametric).
- S2. Construct the associated measurement error model likelihood under the ‘false’ assumption that the components of \mathbf{X} are measured with independent error. That is, assume that \mathbf{W} is observed in place of \mathbf{X} where $\mathbf{W} \mid \mathbf{X} \sim \mathcal{N}(\mathbf{X}, \mathcal{D}_{\{\boldsymbol{\sigma}_u^2\}})$ with $\boldsymbol{\sigma}_u^2 = (\sigma_{u,1}^2, \dots, \sigma_{u,p}^2)$. The resulting likelihood depends on $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}_u^2$ and is denoted $L_{\text{MEM}}(\boldsymbol{\theta}, \boldsymbol{\sigma}_u^2)$. Note that even though $L_{\text{MEM}}(\boldsymbol{\theta}, \boldsymbol{\sigma}_u^2)$ is derived under a measurement error model assumption, it is calculated from the error-free data $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$.
- S3. Replace $\boldsymbol{\theta}$ in $L_{\text{MEM}}(\boldsymbol{\theta}, \boldsymbol{\sigma}_u^2)$ with an estimate, $\hat{\boldsymbol{\theta}}$, resulting in the pseudo-profile likelihood $\hat{L}_{\text{pMEM}}(\boldsymbol{\sigma}_u^2) = L_{\text{MEM}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\sigma}_u^2)$. Note that $\hat{\boldsymbol{\theta}}$ is an estimator for $\boldsymbol{\theta}$ calculated from the observed data without regard to the ‘false’ measurement error assumption, e.g., $\hat{\boldsymbol{\theta}}$ could be the maximum likelihood estimator from $L_{\text{TRUE}}(\boldsymbol{\theta})$.
- S4. Reexpress the pseudo-profile likelihood $\hat{L}_{\text{pMEM}}(\boldsymbol{\sigma}_u^2)$ in terms of precision (or square-root precision) $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)$ where $\lambda_j = 1/\sigma_{u,j}^2$ (or $\lambda_j = 1/\sigma_{u,j}$), resulting in the MEM

selection likelihood $\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$.

$\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ is maximized subject to: $\lambda_j \geq 0$, $j = 1, \dots, p$; and $\sum_{j=1}^p \lambda_j = \tau$. Setting the tuning parameter $\tau < \infty$ in the latter constraint ensures that the harmonic mean of the measurement error standard deviations is $p/\tau > 0$. This is how the approach forces ‘false’ measurement error into the likelihood.

We show that application of the measurement error model selection likelihood approach to nonparametric regression results in MEKRO. Consider the quadratic loss pseudo-likelihood (negative loss) functional,

$$L_{\text{TRUE}}(F_{Y,\mathbf{X}}) = -\frac{1}{n} \sum_{i=1}^n \{Y_i - g_{Y|\mathbf{X}}(\mathbf{X}_i)\}^2,$$

where

$$g_{Y|\mathbf{X}}(\mathbf{x}) = \frac{\int y f_{Y,\mathbf{X}}(y, \mathbf{x}) dy}{\int f_{Y,\mathbf{X}}(y, \mathbf{x}) dy},$$

and $f_{Y,\mathbf{X}}(y, \mathbf{x}) = \partial^{p+1}/(\partial y \partial x_1 \cdots \partial x_p) F_{Y,\mathbf{X}}(y, \mathbf{x})$. Note that $F_{Y,\mathbf{X}}(\cdot, \cdot)$ plays the role of $\boldsymbol{\theta}$ in the four-step algorithm. Assume that \mathbf{W}_i is observed instead of \mathbf{X}_i , where $\mathbf{W}_i = \mathbf{X}_i + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{U}_i$ and $\mathbf{U}_i \stackrel{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p)$ and is independent of all other data, to give

$$L_{\text{MEM}}(F_{Y,\mathbf{W}}, \boldsymbol{\sigma}_u^2) = -\frac{1}{n} \sum_{i=1}^n \{Y_i - g_{Y|\mathbf{W}}(\mathbf{X}_i)\}^2,$$

where $g_{Y|\mathbf{W}}(\cdot)$ depends on $\boldsymbol{\sigma}_u$ implicitly. We derive an expression for $g_{Y|\mathbf{W}}(\cdot)$; observe,

$$\begin{aligned} f_{Y,\mathbf{W}}(y, \mathbf{w}) &= \frac{\partial^{p+1}}{\partial y \partial w_1 \cdots \partial w_p} P(Y \leq y, \mathbf{W} \leq \mathbf{w}) \\ &= \int f_{Y,\mathbf{X}}(y, \mathbf{w} - \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{u}) \prod_{j=1}^p \phi(u_j) du, \end{aligned}$$

where the interchange of differentiation and integration is justified for the Gaussian product kernel and many others. Consequently,

$$\begin{aligned}
g_{Y|\mathbf{W}}(\mathbf{x}) &= \mathbb{E}(Y \mid \mathbf{W} = \mathbf{x}) \\
&= \int y f_{Y|\mathbf{W}}(y \mid \mathbf{x}) dy \\
&= \frac{\int y \int f_{Y,\mathbf{X}}(y, \mathbf{x} - \mathcal{D}_{\{\sigma_u\}} \mathbf{u}) \prod_{j=1}^p \phi(u_j) du dy}{\iint f_{Y,\mathbf{X}}(y, \mathbf{x} - \mathcal{D}_{\{\sigma_u\}} \mathbf{u}) \prod_{j=1}^p \phi(u_j) du dy} \\
&= \frac{\int y \int f_{Y,\mathbf{X}}(y, \mathbf{t}) \prod_{j=1}^p \phi\{(x_j - t_j)/\sigma_j\} (\sigma_{u,j})^{-1} dt dy}{\iint f_{Y,\mathbf{X}}(y, \mathbf{t}) \prod_{j=1}^p \phi\{(x_j - t_j)/\sigma_{u,j}\} (\sigma_{u,j})^{-1} dt dy} \\
&= \frac{\iint y \prod_{j=1}^p \phi\{(x_j - t_j)/\sigma_{u,j}\} F_{Y,\mathbf{X}}(dy, dt)}{\iint \prod_{j=1}^p \phi\{(x_j - t_j)/\sigma_{u,j}\} F_{Y,\mathbf{X}}(dy, dt)},
\end{aligned}$$

after noting the change of variables $u_j = (x_j - t_j)/\sigma_{u,j}$ and that $\phi(\cdot)$ is the standard normal pdf. Step S3 in the four-step algorithm calls for estimation of $\boldsymbol{\theta}$, which in this setting means estimation of $F_{Y,\mathbf{X}}(\cdot, \cdot)$. The empirical cdf is substituted to give \hat{L}_{pMEM} (not shown). Finally, the measurement error standard deviations are parameterized as inverse standard deviations (S4) to produce the MEM selection likelihood,

$$\hat{L}_{\text{SEL}}(\boldsymbol{\lambda}) = -\frac{1}{n} \sum_{i=1}^n \{Y_i - \hat{g}(\mathbf{X}_i, \boldsymbol{\lambda})\}^2,$$

where,

$$\hat{g}(\mathbf{X}_i, \boldsymbol{\lambda}) = \frac{\sum_{k=1}^n Y_k \prod_{j=1}^p \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2/2\}}{\sum_{k=1}^n \prod_{j=1}^p \exp\{-\lambda_j^2 (X_{i,j} - X_{k,j})^2/2\}}.$$

There is now an explicit dependence of $\hat{g}(\cdot)$ on $\boldsymbol{\lambda}$ that have entered the selection likelihood as inverse smoothing bandwidth parameters.

3.7.2 Asymptotic Selection Consistency

Using a mix of known results, detailed derivations, and heuristics, we explain the apparent large-sample selection consistency manifest in our simulation studies. For the multivariate Nadaraya-Watson estimator, we denote the smoothing bandwidth for predictor j by h_j and

$\mathbf{h} = (h_1, \dots, h_p)^T$. In Section 2.1 of [47], the pointwise asymptotic bias and variance are rigorously established for the multivariate Nadaraya-Watson estimator, $\widehat{g}(\mathbf{x}, 1/\mathbf{h}) - g(\mathbf{x}) = O_p\left(\sum_{j=1}^p h_j^2 + (n \prod_{j=1}^p h_j)^{-1/2}\right)$, where $1/\mathbf{h} = (1/h_1, \dots, 1/h_p)^T$.

In MEKRO, one maximizes the MEM selection likelihood (3.1) subject to constraint (3.3), which is equivalent to minimizing the fitted mean squared error $-\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ subject to this constraint. In order to study the asymptotic properties for the optimizer, we need to characterize the asymptotic behavior of $-\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ in a similar format as Lemma A1 in [82]. Their Lemma A1 follows from technical proofs of [18], whose techniques can be used to extend the pointwise asymptotic results of [47] and argue that under regularity conditions of the type in [18], if bandwidths satisfy $h_j \rightarrow 0$ for $j = 1, \dots, p$ and $n \prod_{j=1}^p h_j / \log(n) \rightarrow \infty$ as $n \rightarrow \infty$, it holds that,

$$\frac{1}{n} \sum_{i=1}^n \{Y_i - \widehat{g}(\mathbf{X}_i, 1/\mathbf{h})\}^2 = \sigma_\epsilon^2 + O_p\left(\sum_{j=1}^p h_j^4 + \left(n \prod_{j=1}^p h_j\right)^{-1}\right). \quad (3.8)$$

In (3.8), the smoothing bandwidth for each predictor shrinks to zero as the sample size diverges to infinity. Yet, it is well known that the use of a small bandwidth in local polynomial smoothing reduces approximation bias in Taylor-series expansions and thus also estimation bias [17]. This is echoed in the asymptotic bias formula that appears below Equation (2.8) on page 62 of [47], where it is explicitly shown that the bias corresponding to predictor j has a factor of $2 \frac{\partial f(\mathbf{x})}{\partial x_j} \frac{\partial g(\mathbf{x})}{\partial x_j} + f(\mathbf{x}) \frac{\partial^2 g(\mathbf{x})}{\partial x_j^2}$, where $f(\cdot)$ is the density of \mathbf{X} . If $j \notin \mathcal{I}$, the index set of important predictors in $g(\cdot)$, this factor is equal to zero because $\frac{\partial g(\mathbf{x})}{\partial x_j} = 0$ and $\frac{\partial^2 g(\mathbf{x})}{\partial x_j^2} = 0$ when predictor j is not important. Thus in a Taylor-series expansion of the multivariate Nadaraya-Watson estimator, predictor j does not contribute to the approximate bias if $j \notin \mathcal{I}$ and the corresponding smoothing bandwidth is not required to shrink to zero as the sample size diverges. This suggests that (3.8) can be further refined to show that if bandwidths satisfy $h_j \rightarrow 0$ for $j \in \mathcal{M}$, $h_{j'} \geq c_0 > 0$ for $j' \in \mathcal{M}^c$ and some $c_0 > 0$, and $n \prod_{j \in \mathcal{M}} h_j / \log(n) \rightarrow \infty$ as $n \rightarrow \infty$ for a set \mathcal{M} satisfying

$\mathcal{I} \subseteq \mathcal{M} \subseteq \{1, \dots, p\}$, we have,

$$\frac{1}{n} \sum_{i=1}^n \{Y_i - \widehat{g}(\mathbf{X}_i, 1/\mathbf{h})\}^2 = \sigma_\epsilon^2 + O_p \left(\sum_{j \in \mathcal{M}} h_j^4 + \left(n \prod_{j \in \mathcal{M}} h_j \right)^{-1} \right). \quad (3.9)$$

To gain insight, we now consider a deterministic version of the O_p in (3.9) with transformation $t_j = 1/h_j$. For a set \mathcal{M} satisfying $\mathcal{I} \subseteq \mathcal{M} \subseteq \{1, \dots, p\}$ and n large enough, denote $\widehat{\mathbf{t}}_{\mathcal{M}} = (\widehat{t}_j, j \in \mathcal{M})^T$ as the minimizer of

$$\sum_{j \in \mathcal{M}} t_j^{-4} + n^{-1} \prod_{j \in \mathcal{M}} t_j, \quad \text{subject to } t_j \geq \log(n), \quad j \in \mathcal{M}, \quad \text{and } \sum_{j \in \mathcal{M}} t_j = T_n > 0,$$

where T_n satisfies $T_n \rightarrow \infty$ and $T_n^p \log(n)/n \rightarrow 0$ as $n \rightarrow \infty$. Here the constraint $t_j \geq \log(n)$ guarantees that $h_j = 1/t_j$ converges to zero as required by (3.9), where $\log(n)$ can be replaced by any sequence that slowly diverges to infinity. Note that the optimization problem is symmetric in t_j and thus it follows that the minimizer is given by $\widehat{t}_j = T/|\mathcal{M}|$, $j \in \mathcal{M}$, and the corresponding objective function takes value $|\mathcal{M}|^5/T^4 + (T/|\mathcal{M}|)^{|\mathcal{M}|}/n$, where $|\mathcal{M}|$ denotes the cardinality of set \mathcal{M} and $T \equiv T_n$. By treating $|\mathcal{M}|$ as a continuous variable and examining the first derivative with respect to $|\mathcal{M}|$, we conclude $|\mathcal{M}|^5/T^4 + (T/|\mathcal{M}|)^{|\mathcal{M}|}/n$ is monotonically increasing in $|\mathcal{M}|$ if $0 < |\mathcal{M}| < T/e$. We next appeal to these results to assert that $\widehat{\lambda}_j \rightarrow \infty$ for $j \in \mathcal{I}$ and $\widehat{\lambda}_j \rightarrow 0$ for $j \in \mathcal{I}^c$

Selection consistency of MEKRO: We first argue that $\widehat{\lambda}_j \rightarrow \infty$ in probability for $j \in \mathcal{I}$ as $n \rightarrow \infty$. According to (3.9), $-\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ converges to σ_ϵ^2 as long as the smoothing parameters of all important predictors shrink to zero as the sample size diverges to infinity. On the other hand, according to the proof of the asymptotic bias and variance in [47], the multivariate Nadaraya-Watson estimator is not consistent if smoothing bandwidths of any important predictor do not shrink to zero as the sample size diverges to infinity. Correspondingly, $-\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ will converge to σ_ϵ^2 plus a squared bias term that does not shrink to zero asymptotically. Recall that λ_j is the reciprocal smoothing bandwidth and $\widehat{\boldsymbol{\lambda}}$ is the solution that minimizes $-\widehat{L}_{\text{SEL}}(\boldsymbol{\lambda})$ subject

to subject to constraint (3.3), thus, minimization will not lead to $\widehat{\lambda}_j \not\rightarrow \infty$ for $j \in \mathcal{I}$ as the corresponding limit of the objective function is larger than σ_ϵ^2 , which is attainable. Consequently, we have $\widehat{\lambda}_j \rightarrow \infty$ in probability for any $j \in \mathcal{I}$ as $n \rightarrow \infty$.

Consistency of MEKRO is achieved provided $\widehat{\lambda}_j \rightarrow 0$ in probability for $j \in \mathcal{I}^c$, which we now argue. First we show $\widehat{\lambda}_j \not\rightarrow \infty$ in probability for $j \in \mathcal{I}^c$. The MEKRO solution converges to σ_ϵ^2 with an asymptotic rate of $O_p\left(\sum_{j \in \mathcal{M}} h_j^4 + (n \prod_{j \in \mathcal{M}} h_j)^{-1}\right)$ according to (3.9). The deterministic version of this rate is a monotonically increasing function of the cardinality $|\mathcal{M}|$ of the set of predictors whose corresponding $\lambda_j \rightarrow \infty$ by noting $\lambda_j = 1/h_j$. Thus, the MEKRO solution must satisfy $\widehat{\lambda}_j \not\rightarrow \infty$ in probability for $j \in \mathcal{I}^c$ because minimization favors a faster convergence rate, and so $\mathcal{M} = \mathcal{I}$. Further, $\widehat{\lambda}_j$ has the same order as τ for $j \in \mathcal{I}$.

It remains to argue that $\widehat{\lambda}_j$ converging to a positive constant in probability for $j \in \mathcal{I}^c$ is not favored. Denote $\widehat{\mathcal{A}}_\infty = \{j : \widehat{\lambda}_j \rightarrow \infty \text{ in probability as } n \rightarrow \infty\}$, $\widehat{\mathcal{A}}_0 = \{j : \widehat{\lambda}_j \rightarrow 0 \text{ in probability as } n \rightarrow \infty\}$, and $\widehat{\mathcal{A}}_1 = \{1, \dots, p\} \setminus (\widehat{\mathcal{A}}_0 \cup \widehat{\mathcal{A}}_\infty)$. From the above argument we have $\widehat{\mathcal{A}}_\infty = \mathcal{I}$. Then, for $j \in \widehat{\mathcal{A}}_1$, the sequence $\widehat{\lambda}_j$ is asymptotically bounded away from both 0 and ∞ . We assume without loss of generality that $\widehat{\lambda}_j \rightarrow c_j$ in probability for $j \in \widehat{\mathcal{A}}_1$ and some $0 < c_j < \infty$; otherwise, we consider any convergent subsequence of $\widehat{\lambda}_j$. Thus $\tau - \sum_{j \in \widehat{\mathcal{A}}_\infty} \widehat{\lambda}_j \rightarrow \sum_{j' \in \widehat{\mathcal{A}}_1} c_{j'}$ in probability. Now consider an alternative solution sequence $\widetilde{\lambda}_j = \widehat{\lambda}_j \tau / (\tau - \sum_{j' \in \mathcal{I}^c} \widehat{\lambda}_{j'})$ for $j \in \mathcal{I}$ and $\widetilde{\lambda}_{j'} = 0$ for $j' \in \mathcal{I}^c$. Equation (3.9) gives $\sigma_\epsilon^2 + O_p(\sum_{j \in \mathcal{I}} \widehat{\lambda}_j^{-4} + \prod_{j \in \mathcal{I}} \widehat{\lambda}_j/n)$ for the solution $\widehat{\lambda}_j$ and $\sigma_\epsilon^2 + O_p(b^{-4} \sum_{j \in \mathcal{I}} \widehat{\lambda}_j^{-4} + b^{|\mathcal{I}|} \prod_{j \in \mathcal{I}} \widehat{\lambda}_j/n)$ for the alternative solution $\widetilde{\lambda}_j$, where $b = \tau / (\tau - \sum_{j' \in \mathcal{I}^c} \widehat{\lambda}_{j'}) \geq 1$. Note that $b = 1$ iff $\mathcal{I}^c = \emptyset$ in which case all $\widehat{\lambda}_j \rightarrow \infty$ as desired; we henceforth assume at least one predictor is unimportant and thus $b > 1$. We argued above that $\widehat{\lambda}_j$ has the same order as τ for $j \in \mathcal{I}$ and because $\tau \rightarrow \infty$ satisfies $\tau^{|\mathcal{I}|+4}/n \rightarrow 0$ as $n \rightarrow \infty$ by assumption, the asymptotic bias $O_p(\sum_{j \in \mathcal{I}} \widehat{\lambda}_j^{-4})$ dominates the asymptotic variance $O_p(\prod_{j \in \mathcal{I}} \widehat{\lambda}_j/n)$. The alternative solution $\widetilde{\lambda}_j$ will be favored in the process of minimizing the fitted mean squared error because $b > 1$ and thus the leading term of $O_p(b^{-4} \sum_{j \in \mathcal{I}} \widehat{\lambda}_j^{-4} + b^{|\mathcal{I}|} \prod_{j \in \mathcal{I}} \widehat{\lambda}_j/n)$ has a smaller constant than that of $O_p(\sum_{j \in \mathcal{I}} \widehat{\lambda}_j^{-4} + \prod_{j \in \mathcal{I}} \widehat{\lambda}_j/n)$ even though they share the same asymptotic rate. This implies $\widehat{\mathcal{A}}_1 = \emptyset$ and $\widehat{\mathcal{A}}_0 = \mathcal{I}^c$, completing the argument for every

convergent subsequence of $\widehat{\lambda}_j$ and thus $\widehat{\lambda}_j$ in general.

3.7.3 Numerical Study with Gaussian \mathbf{X}

We explore MEKRO's performance when \mathbf{X} follows a Gaussian distribution to address a concern from reviewers that it may greatly underperform without uniform data. We copied Model 2 from Section 3.4.2, but with \mathbf{X} drawn from $\mathcal{N}(0, 1)$ such that $\text{Corr}(\mathbf{X})$ has an AR(1) structure with $\rho = 0.5$. To generate the predictor matrix \mathbf{X} of stacked predictor vectors \mathbf{X}^T , we generate a $n \times p$ matrix \mathbf{Z} with each element iid $\mathcal{N}(0, 1)$ and define $\mathbf{X} = \mathbf{Z}\boldsymbol{\Sigma}$ where $\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}$ is the population AR(1) correlation matrix with $\rho = 0.5$. The predictors are scaled to be in $[0, 1]$ to generate Y_i , $i = 1, \dots, n$, then scaled to have mean 0 and unit variance. Recall that Y is generated according to $g(\mathbf{X}) = \sin\{2\pi(X_1 + X_2)/(1 + X_3)\}$ so that there are three active and seven irrelevant predictors. The predictor effect sizes are $(1.00, 1.00, 0.32)$ for (X_1, X_2, X_3) and 0 for X_4 through X_{10} .

The average integrated squared errors (AISE) and selection errors for Model 2 with Gaussian \mathbf{X} are shown in Figure 3.10 and Table 3.7, respectively. MEKRO (MEK) does not do as well with prediction in this scenario. Gaussian data are spread too thinly near the boundaries to give MEKRO good surface estimates; see Section 3.4.2, Model 4 for more details. However, MEKRO maintains superior selection performance when compared to all other methods at $n \geq 100$ and achieves perfect selection at $n = 400$. The additive COSSO (RC1) that is similar to MEKRO for selection in Model 2 at $n = 400$ falls short with Gaussian \mathbf{X} by frequently failing to include the weak predictor, X_3 . Adding a boundary correction to boost MEKRO's prediction performance is part of future work.

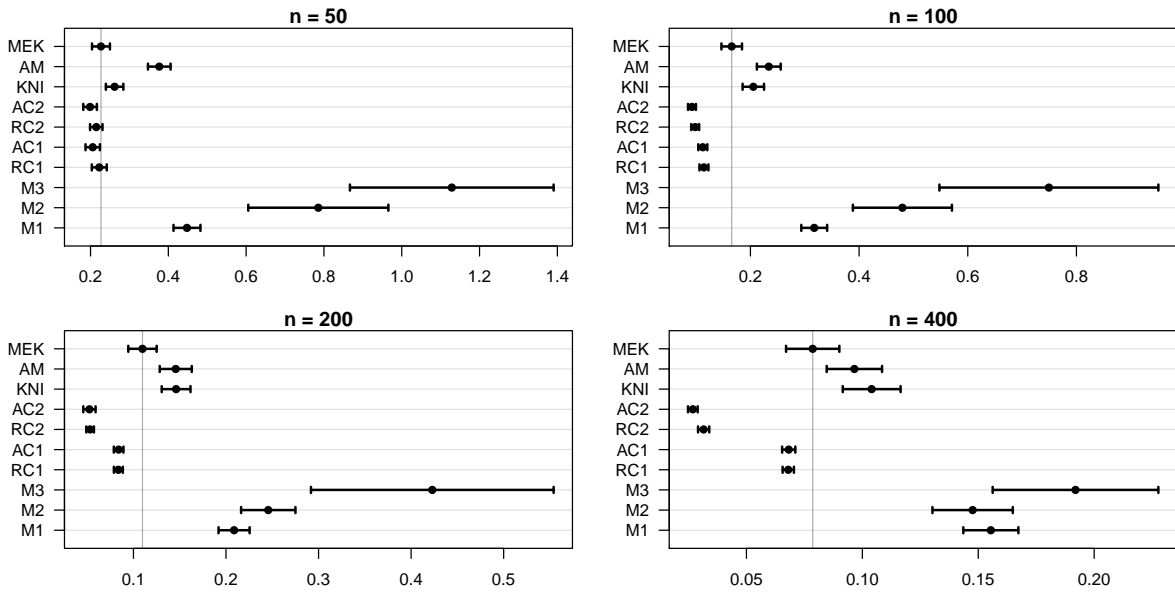


Figure 3.10 AISEs (average integrated squared errors) for Appendix C Model. Note the scale differences. Out of the 400 MC samples, 2 AISE outliers are omitted from both M2 and M3.

Table 3.7 Selection error rates for the model in Appendix C. MC standard errors for all cells ≤ 0.03 .

	Error	MEK	AM	KNI	AC2	RC2	AC1	RC1	M3	M2	M1
$n = 50$	Type I	0.06	0.53	0.15	0.13	0.14	0.33	0.14	0.52	0.53	0.69
	Type II	0.28	0.10	0.42	0.61	0.57	0.42	0.51	0.10	0.09	0.11
$n = 100$	Type I	0.03	0.47	0.07	0.26	0.06	0.27	0.04	0.51	0.50	0.60
	Type II	0.09	0.00	0.30	0.33	0.46	0.22	0.32	0.00	0.01	0.05
$n = 200$	Type I	0.01	0.44	0.06	0.23	0.05	0.33	0.02	0.41	0.39	0.59
	Type II	0.02	0.00	0.28	0.31	0.42	0.11	0.24	0.00	0.00	0.03
$n = 400$	Type I	0.00	0.45	0.11	0.23	0.03	0.29	0.01	0.26	0.27	0.56
	Type II	0.00	0.00	0.13	0.24	0.35	0.08	0.20	0.00	0.00	0.03

CHAPTER

4

GENERAL MEASUREMENT ERROR MODEL SELECTION LIKELIHOODS

4.1 Introduction

Previous work by Stefanski, Wu, and White [68] and White, Stefanski, and Wu [79] show that specific variants of Measurement Error Model Selection Likelihoods (MEMSEL) are equivalent to the LASSO in linear models and produce favorable results when applied to kernel density-based classification and regression. However, both of these works use a similar covariate contamination strategy that results in closed-form solutions for kernel methods. Although closed-form MEMSEL is preferred when a closed-form version exists, it is possible to build MEMSELS for a broader class of models using Monte Carlo simulation or approximations and consider a wide range of contamination strategies that will wrap variable selection around any predictive modeling procedure. This chapter describes such a generalization of MEMSEL.

The contributions of this chapter are twofold. First, a new tuning criterion is developed,

Selection Information Criterion (SIC), as a fast way to tune MEMSEL models. SIC is shown to outperform many popular tuning methods in the linear model including AIC and BIC when the true underlying model is sparse. Second, a standardized framework for predictor contamination is developed and studied in the context of linear models for the purpose of identifying useful error contamination strategies in a familiar setting. MEMSEL is applied to random forests, a method that is not fit using likelihoods, to show how variable selection can be implemented in the absence of a likelihood.

Common notation for the chapter is defined here. The data set $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$ is observed where $\mathbf{Y} = (Y_1, \dots, Y_n)^T$ is a response vector, \mathbf{X}_i is a $p \times 1$ vector of predictors, and thus $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)^T$ is the design matrix. Assume that predictors are continuous and standardized such that $\mathbf{1}^T \mathbf{X} = \mathbf{0}_{1 \times p}$ and $\widehat{\mathbf{V}}_{\mathbf{X}} \stackrel{\text{def}}{=} n^{-1} \mathbf{X}^T \mathbf{X} = \widehat{\text{Corr}}(\mathbf{X})$. When Y is continuous, assume further that $\widehat{\mathbf{V}}_Y \stackrel{\text{def}}{=} \mathbf{1}^T \mathbf{Y} = 0$, $n^{-1} \mathbf{Y}^T \mathbf{Y} = 1$, and $\widehat{\mathbf{V}}_{\mathbf{X}Y} \stackrel{\text{def}}{=} n^{-1} \mathbf{X}^T \mathbf{Y} = \widehat{\text{Corr}}(\mathbf{X}, Y)$.

4.2 Selection Information Criterion

Variable shrinkage and selection methods depend on at least one tuning parameter that controls model fit and sparsity. The tuning parameter(s) must be selected in practice, often out of a grid of candidate values, to produce a model that generalizes well. A straightforward solution is folded cross-validation (CV), where observed data are subdivided into folds for training and testing. Models are fit to each subset of data with one fold omitted to compute test loss [69]. The tuning parameter set producing the lowest aggregated test loss across the folds is chosen. CV is relatively computationally expensive for a slow fitting procedure because it must refit several models across different subsets. Other tuning methods avoid refitting by estimating the best tuning parameter(s) directly from the training fit and model degrees of freedom. Akaike's information criterion (AIC) seeks to minimize the approximate Kullback-Leibler divergence between candidate models and the true model estimated from the observed data [1, 9]. AIC is consistent for minimizing prediction error among the class of models under consideration [84]. Use of a small-sample version of AIC is suggested when the ratio of n to selected model

parameters is under 40 [9, 42]. Bayesian information criterion (BIC) uses a stronger model complexity penalty than AIC and consistently chooses the “correct” model [60]. Generalized cross validation (GCV) [28] was introduced to select ridge regression parameters, but is also used more generally for smoothing methods where $\widehat{\mathbf{Y}} = \mathbf{S}\mathbf{Y}$ for a smoothing matrix \mathbf{S} and performs well when choosing LASSO tuning parameters [75].

Assume that data are generated according to

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon\sigma_\epsilon, \quad \epsilon \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1). \quad (4.1)$$

Consider a modeling procedure that fits an entire path of $\boldsymbol{\beta}$ solutions depending on a tuning parameter η . For any η , The predicted response is $\widehat{\mathbf{Y}}(\eta)$ and number of selected predictors is $k(\eta) \leq p = \dim(\boldsymbol{\beta}) < n$. Then define the Selection Information Criterion (SIC) as

$$\text{SIC}(\eta) = \frac{n^{-1}\|\mathbf{Y} - \widehat{\mathbf{Y}}(\eta)\|^2}{p - k(\eta)}.$$

If $k(\eta) = p$, then $\text{SIC}(\eta) = \infty$. Like AIC and BIC, the minimizer of SIC is the optimal tuning parameter.

4.2.1 Comparing Tuning Criteria Penalties

SIC employs a heavier overfitting penalty than AIC, BIC, or GCV. Under the linear model (4.1), the term that describes fit for AIC and BIC can be expressed as $-2\log \text{likelihood} = n \log(\widehat{\sigma}_\epsilon^2)$ where $\widehat{\sigma}_\epsilon^2$ is the residual mean squared error. Both GCV and SIC can be redefined with monotonic transformations to preserve their minimizers but give valid comparisons between penalty terms.

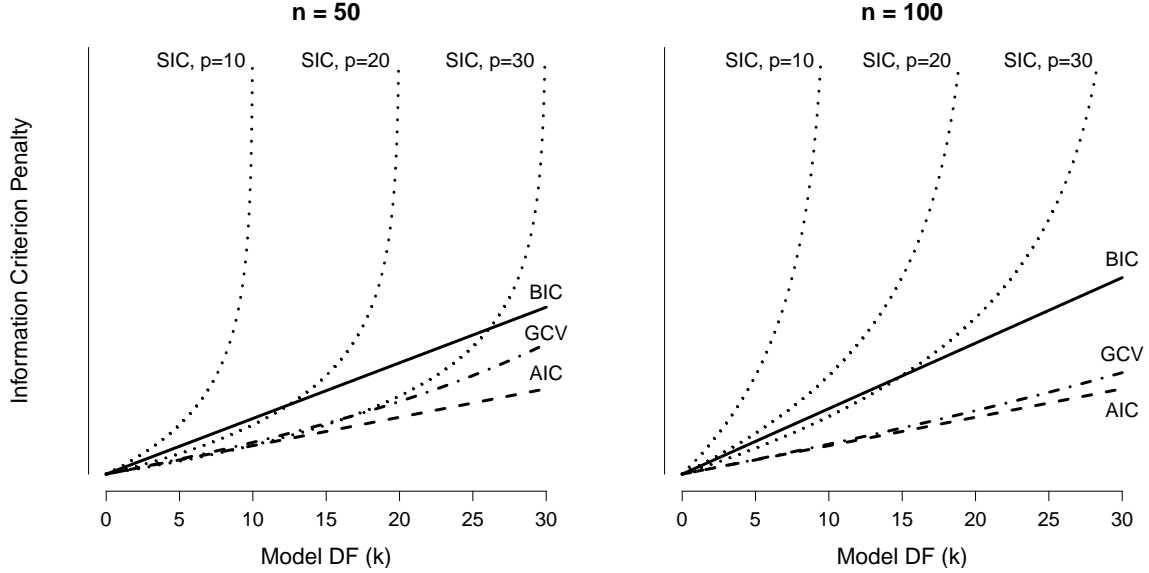


Figure 4.1 Comparing overfitting penalties in AIC (dashed line), BIC (solid line), GCV (dash-dot-dash line), and SIC (dotted lines) as model degrees of freedom (k) increase. Left pane: $n = 50$. Right pane: $n = 100$. Larger penalties for the same model degrees of freedom result in sparser models.

For GCV, define $k = \text{tr}(\mathbf{S})$. Then,

$$\text{AIC} = n \log(\hat{\sigma}_\epsilon^2) + 2k,$$

$$\text{BIC} = n \log(\hat{\sigma}_\epsilon^2) + k \log(n),$$

$$n \log(\text{GCV}) = n \log(\hat{\sigma}_\epsilon^2) - 2n \log(1 - k/n),$$

$$n \log(\text{SIC}) - n \log(p) = n \log(\hat{\sigma}_\epsilon^2) - n \log(p - k).$$

The $-n \log(p)$ term in the SIC equation above normalizes SIC with the other criteria when $k = 0$. Now the penalty terms can be compared directly. Figure 4.1 shows the penalties as k increases for choices of n and p . Clearly, SIC will favor sparser models relative to the others, but graphically the criteria have similar penalties under certain conditions. For SIC, the rate of change in the penalty is given by $\partial\{-n \log(p - k)\}/\partial k = n/(p - k)$. The rates of change between BIC and SIC are then equivalent when $k = p - n/\log(n) \geq 0$. If one considers that the SIC penalty increasing twice as fast as another penalty as being “significantly faster,” then

a rule-of-thumb is that SIC will be different from BIC when $k \geq p - n/\{2 \log(n)\}$. Obviously this disregards the likelihood component from the criterion and exceptions will apply. Note that GCV converges to AIC as n increases [64].

4.2.2 Tuning Method Simulation Study

The selection performance of SIC is evaluated against other tuning methods on data simulated from linear model (4.1) and fit with least squares. Assume rows of \mathbf{X} are drawn from $\mathcal{N}(0, 1)$ such that $\text{Corr}(\mathbf{X})$ has an AR(1) correlation structure with correlation $\rho = 0.5$ unless otherwise noted. The parameters $\boldsymbol{\beta}$, n , and σ_ϵ are varied as simulation parameters. Define model $R^2 = \text{Var}(\mathbf{X}^T \boldsymbol{\beta}) / \text{Var}(Y)$. Many of these models are borrowed from Examples 1-4 in [75]. We compute the full LASSO solution path for each model,

$$\hat{\boldsymbol{\beta}}_L(\eta) \stackrel{\text{def}}{=} \underset{\boldsymbol{\beta}}{\text{argmin}} \ n^{-1} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \eta \|\boldsymbol{\beta}\|_1,$$

and thus $\hat{\mathbf{Y}}(\eta) = \mathbf{X}\hat{\boldsymbol{\beta}}_L(\eta)$. Optimal η values are chosen using the tuning methods SIC, AICc (small-sample-corrected AIC), BIC, GCV (see [28, 75]), and two versions of 5-fold cross-validation (CV). The “min” version of CV selects the tuning parameter by taking the minimum value for the 5-fold CV curve. The “1se” version of CV selects the tuning parameter by choosing the most parsimonious model that is not more than one standard error away from the minimum of the CV curve to favor sparsity [21]. Standard errors are computed using the variance of five separate folds. Although AIC (uncorrected for small samples) and 10-fold CV variants are tested, they are not different from AICc and the 5-fold CV variants in general and are omitted for brevity. A holdout set of 10,000 points generated from the underlying model is used to compute test error.

Simulations are done in R [57] using the `glmnet` package [22] to compute LASSO solution paths. Tuning with CV-min and CV-1se is done with `cv.glmnet()`. Other tuning method results are constructed using the training error and number of variables selected in paths from `glmnet()`. We set `nlambda=1000` in both functions for a finer grid of solutions along the path.

The simulation setups considered are:

Model 1-3, few moderate coefficients. Let $\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)$ so $p = 8$ with three important predictors. For Model 1, let $n = 20$ and σ be set so that $R^2 = 0.70$ (low sample size). Models 2 and 3 set $n = 100$ but take $R^2 = 0.40$ and 0.70 respectively to test larger sample sizes with low and high model signal.

Model 4-6, many small coefficients. Identical to Models 1-3 except $\beta = 0.85\mathbf{1}_8$.

Model 7-9, one large coefficient. Identical to Models 1-3 except $\beta = (5, 0, \dots, 0)$, with $p = 8$ and R^2 is increased to $0.85, 0.60,$ and 0.85 for each model, respectively.

Model 10-11, decreasing coefficients. Let $n = 100$ and $p = 20$ including ten important predictors with $\beta = (9, 10, 0, 5, 4, 0, 0, 7, 0, 0, 6, 1, 0, 0, 0, 0, 3, 0, 8, 2)$. The model R^2 is set to 0.75 and 0.90 respectively.

Model 12, moderate p . Let $n = 100$ and $p = 40$ including 20 important predictors with $\beta = 2(\mathbf{0}_{10}^T, \mathbf{1}_{10}^T, \mathbf{0}_{10}^T, \mathbf{1}_{10}^T)$ and $R^2 = 0.90$.

Model 13, large p . Identical to Model 12 except $\beta = 2(\mathbf{0}_{10}^T, \mathbf{1}_{10}^T, \mathbf{0}_{10}^T, \mathbf{1}_{10}^T, \mathbf{0}_{40}^T)$ so that $p = 80$ with 20 important predictors.

Model 14-16, high $\text{Corr}(\mathbf{X})$. Identical to Models 1-3 except $\rho = 0.85$ and R^2 is increased to $0.85, 0.60,$ and 0.85 for each model, respectively.

Model 17-19, t_3 errors. Identical to Models 1-3 except ϵ_i is generated from a t distribution with three degrees of freedom and $\sigma = \sqrt{3}$ to maintain model R^2 values.

Model 20, pure error. Let $n = 100$ and $p = 8$ but with no important predictors; $\sigma = 1$.

Results are assessed over 100 Monte Carlo (MC) replicates in terms of conditional perfect path selection error. A “perfect selection slice” is defined as a point along a solution path where all irrelevant coefficients are zero and important coefficients are non-zero. Each tuning method selects a tuning parameter that corresponds to a solution path slice. Depending on the underlying data, there may or may not be perfect selection slices in a path. Thus, we define the “conditional perfect path selection rate” as the fraction of times that a tuning method achieved perfect selection out of the total paths that had at least one perfect selection slice, over 100 MC

replicates.

Figures 4.2 and 4.3 shows conditional perfect selection rates. Figure 4.4 shows mean squared test errors when each tuned method is used to predict values on the holdout set of 10,000 points. Each of the 20 models are shown in a separate vertical across the x -axis, with shaded areas to group similar setups. Each tuning method is shown as a different symbol within those verticals. Differences in results between SIC and other methods that are statistically significant based on a paired t -test at $\alpha = 0.05$ are colored in teal and otherwise in red.

Figure 4.2 shows that, in general, SIC performs significantly better than every other tuning method in terms of perfect selection. SIC is followed by CV-1se for selection performance, but does not require repeated evaluations over separate folds and thus is faster than CV methods. Notable exceptions in SIC performance are when there are many small effects (Models 4-6) where SIC is too selective. SIC specifically offers a strong advantage over competitors when there are a moderate to low number of predictors with moderate to high effect sizes. Selection using the min of the CV curve (CV-min) in general performed the worst. Selection in models where p is moderate or high is challenging for all methods. Every model in this simulation has over 50 perfect selection slices except for Model 1 (40), Models 10-13 (4, 16, 29, and 9, respectively), and Models 15-16 (36 and 35, respectively).

This simulation study is conducted again with all factors held constant except for n reset to $10n$ for larger-sample considerations. The results are shown in Figure 4.3. SIC remains a better method for selection than all others except in Models 4-6 where many weak signals are present. SIC obtains over 90% perfect selection in Models 1, 3, 7-9, 17, 19, and 20. Competing non-CV methods, specifically BIC, fall behind in perfection selection by 10-50 percentage points in these cases (except for Model 20, the pure error model). CV-1se remains the closest to SIC and CV-min remains the worst method for selection. Every model in this simulation has over 70 perfect selection slices except for Models 10 and 15 (35 and 64, respectively).

Figure 4.4 shows the average mean squared test errors from predictions using the tuned methods. Unsurprisingly, the most selective methods (SIC, CV-1se) show the worst prediction

performance. All of the other tuning methods perform better than selective methods and similarly to each other, with BIC being slightly worst in models with many small coefficients (Models 5 and 10).

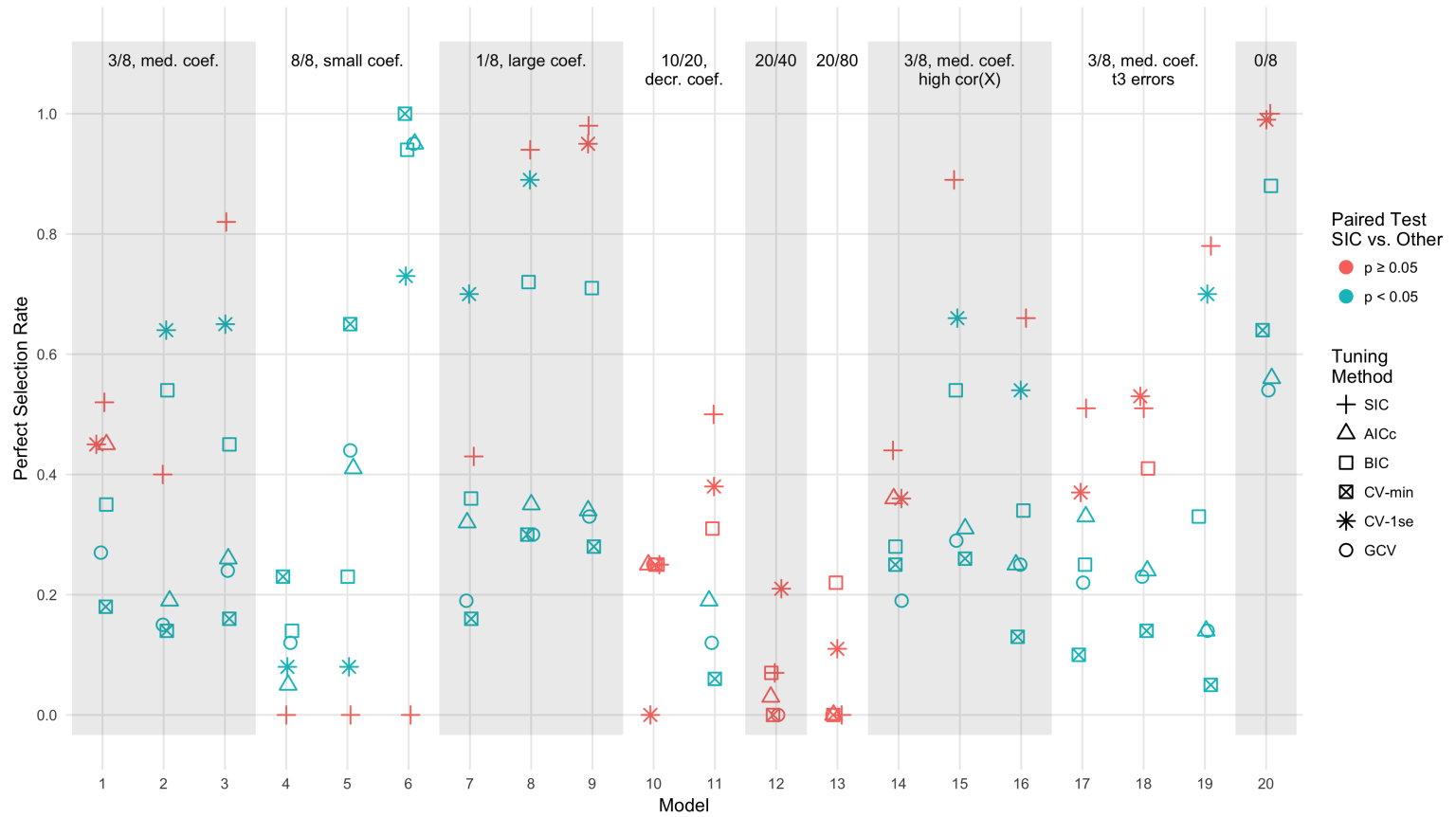


Figure 4.2 Comparison of perfection selection rates (higher is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$.

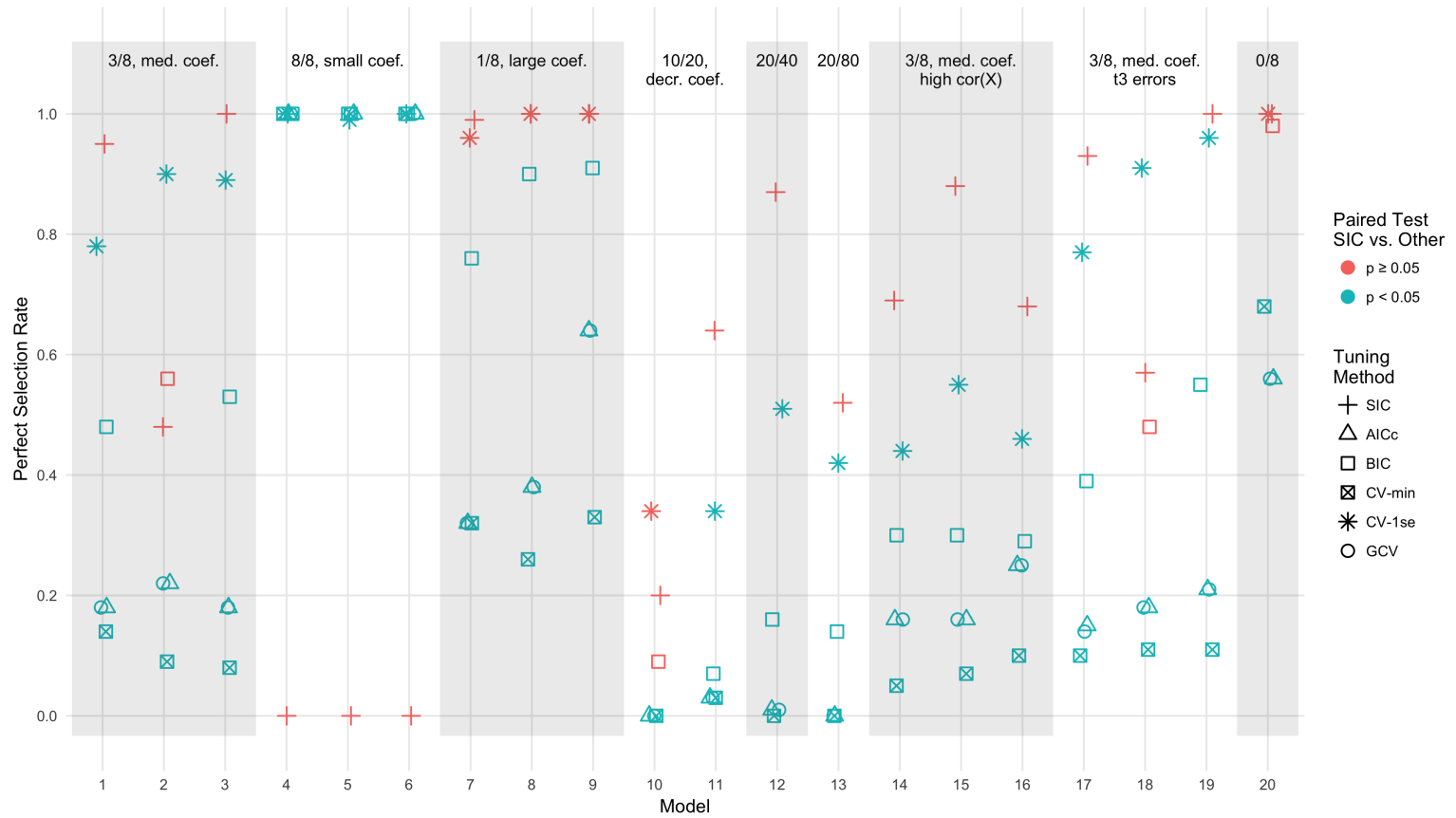


Figure 4.3 Comparison of perfection rates with increased n (higher is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$.

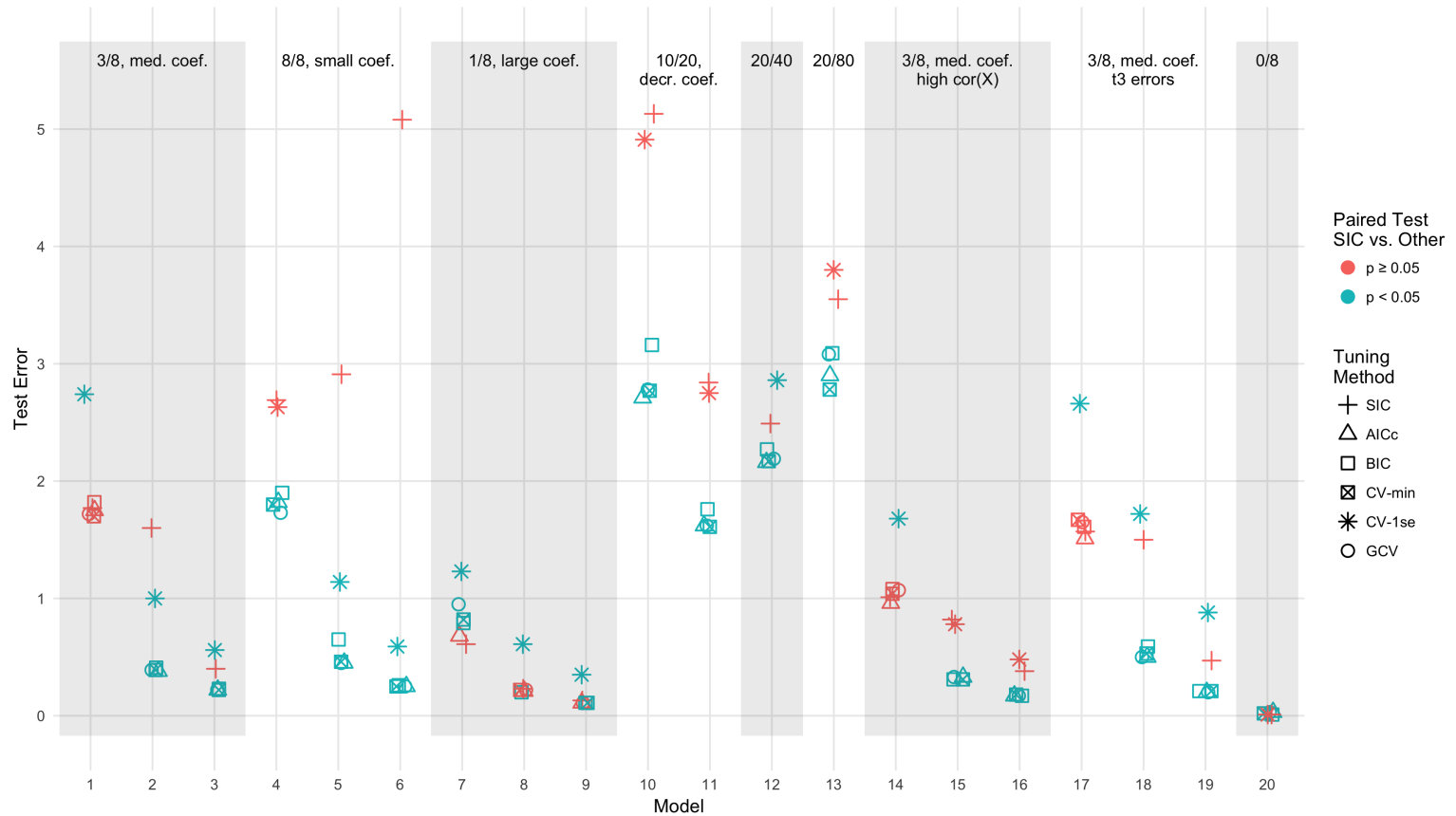


Figure 4.4 Comparison of test errors in SIC simulation study (lower is better). 100 MC replicates are used for each model. Paired t -tests of SIC versus other methods are displayed in teal if statistically different from SIC and red otherwise at $\alpha = 0.05$.

4.2.3 SIC Sensitivity Analysis to p

SIC depends on p unlike other criteria. Intuitively, increasing p by adding purely noise covariates should *not* severely hamper tuning performance. We conduct another simulation study to determine the impact that the relationship between n , p , and $p_1 =$ the number of important variables has on SIC's performance. Assume the same setup as described in 4.2.2 with $R^2 = 0.75$ and $\rho = 0.5$. Three basic coefficient vectors are used to build β vectors for the models in this simulation. Let $\beta_1 = (1, \mathbf{0}_7^T)^T$, $\beta_2 = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$, and $\beta_3 = (3, 1.5, 1, 0, 2, 1.5, 2, 0)^T$. Define $\beta_j^{[r]}$ as r stacked copies of β_j . In cases where $p > \dim(\beta_j^{[r]})$, the unspecified components are filled with 0s such that $\dim(\beta) = p$. The 84 simulation models are:

Table 4.1 Simulation setup to study effects of n , p , p_1 on SIC. Both p and n are explicitly chosen as simulation factors. The number of non-zero coefficients p_1 is implied by the choice of β and r . There are 84 total simulations created from full crosses of r , p , and n in each row.

β	r	p	n	Implied p_1
$\beta_1^{[r]}$	1	{8,25,50,75}	{100,200,400,1000}	1
$\beta_2^{[r]}$	1	{8,25,50,75}	{100,200,400,1000}	3
$\beta_2^{[r]}$	{2,3}	25	{100,200,400,1000}	{6,9}
$\beta_2^{[r]}$	{2,3,4,5,6}	50	{100,200,400,1000}	{6,9,12,15,18}
$\beta_3^{[r]}$	1	8	{100,200,400,1000}	6
$\beta_3^{[r]}$	{2,3}	25	{100,200,400,1000}	{12,18}
$\beta_3^{[r]}$	{4,5,6}	50	{100,200,400,1000}	{24,30,36}

The same six tuning methods described above (SIC, AICc, BIC, GCV, CV-1min, CV-1se) are again compared over 100 MC replicates. The output of each simulation is the method that had the best conditional perfect selection rate as defined in Section 4.2.2 and grouped into one of four categories:

- SIC performs statistically significantly better than every other method,
- SIC performs statistically no differently from the best method,

- SIC performs statistically worse than the best method, or
- no solution paths had any “perfect selection slice” regions.

Comparisons are all paired t -tests at $\alpha = 0.05$. Figure 4.5 plots the resultant category from each of the 84 simulations against p/n and p_1/p to determine where SIC dominates. SIC does significantly better at perfect selection when $p/n < 0.2$ and $p_1/p < 0.75$. This is not overly restrictive of p_1 which is unknown to the analyst. Tuning with CV-1se beats SIC in the lower-left corner of the plot around $p/n = 0.2$ and BIC dominates in the lower-middle portion of the plot when p/n is moderate. No method does well on perfect selection when there are many covariates and many of them are important, but in those instances selection is not a priority. On the other hand, if an analyst can pare down the list of input variables as much as possible based on pre-screening methods or domain knowledge, SIC is a competitive tuning algorithm compared to BIC and the more computationally-intensive CV-1se.

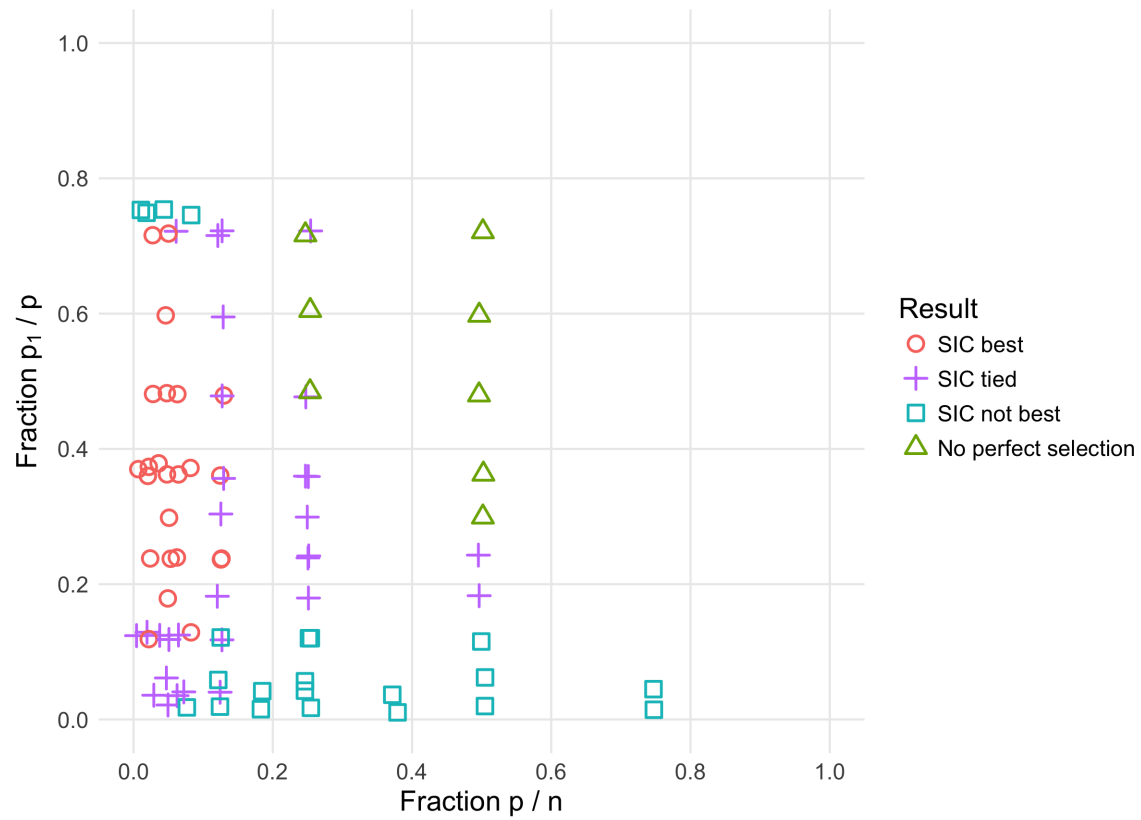


Figure 4.5 Results of sensitivity analysis of SIC. Each of the 84 points coincides with one simulation described in Table 4.1. Results are the tuning method that had the best perfect selection rate over 100 MC replicates. The results “best,” “tied,” and “not best” are judged with a paired t -test at $\alpha = 0.05$.

4.3 MEMSEL

Measurement Error Modeling Selection Likelihoods (MEMSEL) provide a wrapper method that performs variable selection on top of any predictive model as described in [45]. The predictive model serves as a “black-box” algorithm that produces a prediction function from input data. The variable selection layer evaluates the relevance of each feature in the model based on how much black-box predictions degrade when false error is added to the fitting or evaluation data. The fitting procedure is defined succinctly as $input\ data \rightarrow \blacksquare \rightarrow prediction\ function$. The black-box method can be arbitrarily complex provided it produces a consistent estimator of the true prediction function. Define the true prediction function as $\mu_{\mathbf{T}}(\mathbf{t}) \stackrel{\text{def}}{=} E(Y \mid \mathbf{T} = \mathbf{t})$ where the subscript shows what data are used in black-box fitting. Thus write $(\mathbf{Y}, \mathbf{T}) \rightarrow \blacksquare \rightarrow \hat{\mu}_{\mathbf{T}}(\mathbf{t}) = \hat{E}(Y \mid \mathbf{T} = \mathbf{t})$ to indicate that the black box purportedly produces an estimator of the conditional mean of Y given $\mathbf{T} = \mathbf{t}$. As an example, if the black box is ordinary least squares, then $(\mathbf{Y}, \mathbf{X}) \rightarrow \blacksquare \rightarrow \hat{\mu}_{\mathbf{X}}(\mathbf{x}) = \mathbf{x}^T \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}$.

MEMSEL contaminates the black-box inputs, prediction function arguments, or both with a controlled amount of measurement error and evaluates the degradation in predicting \mathbf{Y} versus when error-free data are used. An optimizer determines the distribution of measurement error that minimizes the impact of predictor contamination. The amount of added measurement error is controlled by adjusting the harmonic mean of the measurement error spread parameter to allow some variances to be infinite. Predictors measured with an infinite amount of error are considered to be selected out. The harmonic mean is a regularization tuning parameter much like the penalty parameter in LASSO.

When first formulated in [68], MEMSEL is described in terms of four broad steps that started with a true likelihood for model parameters. The general MEMSEL approach tweaks the prior four-step process to require only a predictive black box and allow for more general contamination. The general MEMSEL approach is:

- S1. Obtain measured data $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$ and define a fitting procedure $\rightarrow \blacksquare \rightarrow$ that is sensible for the data (i.e., logistic regression or random forest classifier for binary response data). Define the contaminated pseudo-measurements $\mathbf{W} = \mathbf{X} + \mathbf{Z}$ where $\mathbf{Z} \sim \mathcal{N}\{\mathbf{0}, \text{diag}(\boldsymbol{\sigma}_u^2)\}$, $\boldsymbol{\lambda}^m = \boldsymbol{\sigma}_u^{-2}$, and $m > 0$.
- S2. Derive $\mu_{\mathbf{T}}(\mathbf{t}) \stackrel{\text{def}}{=} \text{E}(Y \mid \mathbf{T} = \mathbf{t})$ where the analyst takes \mathbf{T} to be either \mathbf{X} or \mathbf{W} . It is desirable to obtain a closed form for $\mu_{\mathbf{T}}(\mathbf{t})$ but not always possible; approximations or Monte Carlo estimation may be necessary. If $\mathbf{T} = \mathbf{X}$ then this step must be done only once. Otherwise, $\mu_{\mathbf{T}}(\mathbf{t})$ depends on $\boldsymbol{\lambda}$ and must be recomputed when $\boldsymbol{\lambda}$ changes.
- S3. Let $\hat{\mu}_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \hat{\mu}_{\mathbf{T}}(\mathbf{S})$ be a prediction functional, evaluating $\hat{\mu}_{\mathbf{T}}(\cdot)$ at \mathbf{S} that is taken to be either \mathbf{X} or \mathbf{W} . The argument $\boldsymbol{\lambda}$ is introduced through either \mathbf{T} or \mathbf{S} (or both) playing the role of \mathbf{W} . Define a loss function \mathcal{L} , typically the loss function used in $\rightarrow \blacksquare \rightarrow$, and let $Q_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \mathcal{L}\{\mathbf{Y}, \hat{\mu}_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda})\}(\boldsymbol{\lambda})$ be the MEMSEL loss function.
- S4. Minimize $Q_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda})$ subject to $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\mathbf{1}^T \boldsymbol{\lambda} = \tau$ where τ is a fixed tuning parameter. Tuning validation over a grid of τ values yields the optimal MEMSEL fit, $\hat{\boldsymbol{\lambda}}_{\hat{\tau}}$. Components of $\hat{\boldsymbol{\lambda}}_{\hat{\tau}}$ that are zero indicate components of \mathbf{X} that are unimportant in the model, while larger components $\hat{\boldsymbol{\lambda}}_{\hat{\tau}}$ indicate more important components of \mathbf{X} .

These steps are described in detail in Sections 4.3.1 and 4.3.2.

4.3.1 Contamination Via Pseudo-Measured Predictors

The four-step method above describes broadly how predictors are contaminated but skips over important details for implementation, especially in Steps S2 and S3. This section explores a plurality of ways to introduce contamination for the sake of completeness. Later sections will demonstrate both through theory and simulation that not every contamination scheme presented in this section is useful, however, starting with the broad MEMSEL framework and narrowing down to viable schemes gives a more holistic view of the approach.

Step S2 above seeks $\mu_{\mathbf{T}}(\mathbf{t})$. Recall that

$$\begin{aligned}\mu_{\mathbf{T}}(\mathbf{t}) &= \text{E}(Y \mid \mathbf{T} = \mathbf{t}) = \text{E}\{\text{E}(Y \mid \mathbf{X}) \mid \mathbf{T} = \mathbf{t}\} \\ &= \text{E}\{\mu_{\mathbf{X}}(\mathbf{X}) \mid \mathbf{T} = \mathbf{t}\}\end{aligned}\tag{4.2}$$

$$\approx \mu_{\mathbf{X}}\{\text{E}(\mathbf{X} \mid \mathbf{T} = \mathbf{t})\},\tag{4.3}$$

where the final approximation is equality if $\mu_{\mathbf{X}}(\cdot)$ is linear. Estimation of (4.2) is the most desirable and known as “exact” MEMSEL. If $\mathbf{T} = \mathbf{X}$ then no contamination is used and estimation of $\mu_{\mathbf{X}}(\cdot)$ proceeds as in typical regression. If $\mathbf{T} = \mathbf{W}$, estimation of $\mu_{\mathbf{W}}(\cdot)$ requires the pseudo-measured predictors defined generically as in Step S1. Specifically, let

$$\mathbf{W}_{i,b} = \mathbf{X}_i + \mathcal{D}_{\{\boldsymbol{\lambda}^m\}}^{-1/2} \mathbf{Z}_{i,b} = \mathbf{X}_i + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{Z}_{i,b}, \quad i = 1, \dots, n\tag{4.4}$$

where $\mathbf{Z}_{i,b}$ are iid $\mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, $i = 1, \dots, n$ and $b = 1, 2, \dots, B$, and $\mathcal{D}_{\{\boldsymbol{\lambda}^m\}}$ is a diagonal matrix with diagonal elements $\lambda_j^m = \sigma_{u,j}^{-2}$, $j = 1, \dots, p$, and $m > 0$. When $m = 1$ then $\boldsymbol{\lambda}^m$ is a measurement error precision vector. Note that (4.4) can be written equivalently in matrix form as $\mathbf{W}_b = \mathbf{X} + \mathbf{Z}_b \mathcal{D}_{\{\boldsymbol{\lambda}^m\}}^{-1/2} = \mathbf{X} + \mathbf{Z}_b \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}$ where $\mathbf{Z}_b = (\mathbf{Z}_{1,b}, \dots, \mathbf{Z}_{n,b})^T$. Rather than generate the $\mathbf{Z}_{i,b}$ as iid $\mathcal{N}(\mathbf{0}, \mathbf{I}_p)$, assume that the linear effects of $\mathbf{1}$, \mathbf{Y} , and \mathbf{X} have been “swept out” and the resulting residuals normalized so that

$$\mathbf{1}^T \mathbf{Z}_b = \mathbf{0}_{1 \times p}, \quad \mathbf{Y}^T \mathbf{Z}_b = \mathbf{0}_{1 \times p}, \quad \mathbf{X}^T \mathbf{Z}_b = \mathbf{0}_{p \times p}, \quad \text{and} \quad \mathbf{Z}_b^T \mathbf{Z}_b = n \mathbf{I}_p\tag{4.5}$$

for each b . See Appendix 4.7.1 for details. This assumption is made throughout the rest of the paper leading to closed-form representations in the linear model.

Many times a closed-form representation of $\mu_{\mathbf{W}}(\cdot)$ is available only in certain special cases leaving Monte Carlo (MC) estimation as a brute-force approach to MEMSEL. Define a single contaminated estimator $\hat{\mu}_{\mathbf{W}_b}(\mathbf{t})$ as $(\mathbf{Y}, \mathbf{W}_b) \rightarrow \blacksquare \rightarrow \hat{\mu}_{\mathbf{W}_b}(\mathbf{t})$, $b = 1, \dots, B$ so that $\hat{\mu}_{\mathbf{W}}(\mathbf{t}) = B^{-1} \sum_{b=1}^B \hat{\mu}_{\mathbf{W}_b}(\mathbf{t})$. Note that $\hat{\mu}_{\mathbf{W}}(\mathbf{t})$ is a successively less variable estimator of $\text{E}(Y \mid \mathbf{W} = \mathbf{t})$ as B increases. However, choosing B large may result in $\hat{\mu}_{\mathbf{W}}(\mathbf{t})$ being prohibitively slow to compute

if the black-box prediction method $\rightarrow \blacksquare \rightarrow$ is a complicated procedure.

If closed-form representations are unavailable and MC estimation proves to be too slow, one may use regression calibration from measurement error literature to approximate (4.2) using (4.3). If $\mathbf{T} = \mathbf{W}$, then $E(\mathbf{X} \mid \mathbf{W} = \mathbf{t}) = \widehat{\mathbf{V}}_{\mathbf{X}} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \mathbf{t}$ if \mathbf{X} and \mathbf{W} are assumed to be jointly normal. This assumption is rarely valid but gives a sensible way to introduce $\boldsymbol{\lambda}$ into the fitting procedure via \mathbf{W} . Unlike exact MEMSEL, \mathbf{W} is not actually observed and \mathbf{t} is taken to be \mathbf{X} . Thus, the argument in (4.3) may be replaced by

$$\mathbf{M} \stackrel{\text{def}}{=} \widehat{\mathbf{V}}_{\mathbf{X}} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \mathbf{X} \quad (4.6)$$

to contaminate predictors in what we call “regression calibration” MEMSEL. Because speed is of the essence for MEMSEL methods, “diagonal regression calibration” is also considered where the inverse in \mathbf{M} is avoided by assuming $\mathbf{V}_{\mathbf{X}} = \mathbf{I}_p$ so that $E(\mathbf{X} \mid \mathbf{W} = \mathbf{t})$ may be replaced by

$$\mathbf{D} \stackrel{\text{def}}{=} \mathcal{D}_{\{1 + \sigma_u^2\}}^{-1} \mathbf{X}. \quad (4.7)$$

Up to this point, variable contamination has been considered as only an input to $\rightarrow \blacksquare \rightarrow$. It may also be introduced when evaluating the output from the black box, $\widehat{\mu}_{\mathbf{T}}(\cdot)$. Let \mathbf{S} represent the data used in evaluation. We define the prediction functional $\widehat{\mu}_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \widehat{\mu}_{\mathbf{T}}(\mathbf{S})$ where $\widehat{\mu}_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda})$ implicitly takes the observed data and black-box fitter as inputs and produces predictions for \mathbf{Y} under a contamination scheme and level. The contamination scheme here is choosing \mathbf{T} and \mathbf{S} each to be one of $\{\mathbf{X}, \mathbf{W}, \mathbf{M}, \mathbf{D}\}$, where “ \mathbf{X} ” denotes no contamination, “ \mathbf{W} ” denotes exact MEMSEL, “ \mathbf{M} ” denotes regression calibration MEMSEL, and “ \mathbf{D} ” denotes diagonal regression calibration MEMSEL. This choice can be succinctly defined through (\mathbf{T}, \mathbf{S}) pairs, e.g., (\mathbf{X}, \mathbf{M}) MEMSEL represents $(\mathbf{Y}, \mathbf{X}) \rightarrow \blacksquare \rightarrow \widehat{\mu}_{\mathbf{X}}(\mathbf{t})$, then $\widehat{\mathbf{Y}} = \widehat{\mu}_{\mathbf{X}}(\mathbf{M})$. If m is specified then we use the abbreviation “TSm” so (\mathbf{X}, \mathbf{M}) MEMSEL with $m = 2$ is written as XM2.

Either \mathbf{T} or \mathbf{S} , or both, may be a set of pseudo-measured predictors. If \mathbf{S} and \mathbf{T} are both taken to be \mathbf{X} , then the prediction functional is identical to “typical” regression absent of

measurement error contamination. If $\mathbf{S} = \mathbf{W}$ and the MC approach is used, then define

$$\hat{\mu}_{\mathbf{T},\mathbf{W}}(\boldsymbol{\lambda}) = R^{-1} \sum_{r=1}^R \hat{\mu}_{\mathbf{T}}(\mathbf{W}_r),$$

where R is the number of replicate contaminated data sets to use for evaluation.

4.3.2 MEMSEL Objective, Optimization, and Tuning

The MEMSEL loss function measures the distance between the observed response and evaluated prediction functional as a function of $\boldsymbol{\lambda}$ and is chosen as the same loss (negative likelihood) function in the black-box prediction method $\rightarrow \blacksquare \rightarrow$. Define the loss function as \mathcal{L} and for notational simplicity let the MEMSEL objective be $Q_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda}) = \mathcal{L}\{\mathbf{Y}, \hat{\mu}_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda})\}(\boldsymbol{\lambda})$, or $Q(\boldsymbol{\lambda})$ generically. The loss function is minimized with respect to $\boldsymbol{\lambda}$ to determine the optimal distribution of inverse measurement errors on each predictor. In general, $Q_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda}) > Q_{\mathbf{X},\mathbf{X}}(\boldsymbol{\lambda})$ if *any* contamination is introduced and thus optimizing $Q_{\mathbf{T},\mathbf{S}}(\boldsymbol{\lambda})$ diverges all components of $\boldsymbol{\lambda}$. As such, contamination must be forced into the model. $Q(\boldsymbol{\lambda})$ is optimized under the constraints

$$\boldsymbol{\lambda} \geq \mathbf{0} \quad \text{and} \quad \boldsymbol{\lambda}^T \mathbf{1} = \tau > 0. \tag{4.8}$$

The former constraint allows infinite measurement errors, recalling that $\boldsymbol{\lambda}^m = \boldsymbol{\sigma}_u^{-2}$. The latter constraint forces in contamination by constraining the harmonic mean of the measurement error parameters to be $\tau/p > 0$. Smaller values of τ force in more contamination resulting in more predictor shrinkage ($0 < \lambda < \infty$). If \mathbf{T} and \mathbf{S} are properly selected when building the MEMSEL, the feasible region defined by (4.8) will encourage sparse solutions ($\lambda = 0$), similarly to how the choice of penalty in penalized linear regression dictates sparsity. See Section 3.2.1 for a graphical example of the MEMSEL feasible region.

$Q(\boldsymbol{\lambda})$ is minimized under the constraints in (4.8) to determine the optimal distribution of $\boldsymbol{\lambda}$ such that contamination impacts predictions the least. A modified coordinate descent algorithm that fixes the L_1 -norm of candidate $\boldsymbol{\lambda}$ solution at τ and cycles through univariate grid-search updates works well to find an approximate global minimum (see Section 3.2 in [82]). A univariate

grid size of 20 is sufficient for most problems. The rough solution following modified coordinate descent is then fine-tuned by parameterizing the $\boldsymbol{\lambda}$ vector as $\lambda_j = \tau \gamma_j^2 / (\sum_{k=1}^p \gamma_k^2)$, $j = 1, \dots, p$, and $Q(\boldsymbol{\lambda}(\boldsymbol{\gamma}))$ is minimized with respect to $\boldsymbol{\gamma} \in \mathbb{R}^p$ for a fixed τ using a general gradient-based optimizer such as BFGS. This re-parameterization enforces the constraints automatically at the cost of an extra parameter. An analytical gradient greatly decreases computation time but is not always available for truly black-box models. A final MEMSEL solution is denoted $\widehat{\boldsymbol{\lambda}}_\tau$.

Repeating the MEMSEL procedure over a grid of τ values produces a MEMSEL solution path. A good default grid choice is $\boldsymbol{\tau} = (0.01, 0.1, 0.2, 0.3, e^{-1+0.13}, e^{-1+2 \times 0.13}, \dots, e^{-1+60 \times 0.13})$. This approximates the linear spacing of LASSO $\boldsymbol{\beta}$ estimates between $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ and $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{\text{OLS}}$ in a linear model to represent the solution spaces equally for tuning. An optimal value $\widehat{\tau}$ must be estimated in practice. Poor choices for $\widehat{\tau}$ will underfit or overfit the data. Folded cross validation (CV) is a popular choice for tuning parameter selection but proves to be prohibitively slow in computationally-intensive MEMSEL modeling. Unlike some ensemble black-box methods, MEMSEL can estimate model degrees of freedom to permit the use of faster tuning methods like SIC (defined in Section 4.2), GCV, AICc and BIC.

After a MEMSEL fit has been tuned, predictions may be done in two different ways. One option is a two-step method where the first step calculates $\widehat{\boldsymbol{\lambda}}_{\widehat{\tau}}$ and the second step refits the black-box model to $\{(\mathbf{X}_i, Y_i)_{i=1}^n\}$ after omitting the columns of \mathbf{X} where $\widehat{\boldsymbol{\lambda}}_{\widehat{\tau}} = \mathbf{0}$. Predictions are done using the reduced data. This practice is sensible with “regression calibration” MEMSEL because, despite the loss function remaining bounded for all $\boldsymbol{\lambda}$, the evaluation data for both training and predictions may be contaminated with an unbounded error. Two-step methods avoid this complication. Another option is using the resulting estimator of $\mu_{\mathcal{T}}(\mathbf{t})$ from “exact” MEMSEL directly for predictions. This allows attenuation or shrinkage of effects when components of $\boldsymbol{\lambda}$ are small unlike the two-step method, and is how predictions are done in [68] and [79].

Recall that $\boldsymbol{\lambda}^m = \boldsymbol{\sigma}_u^{-2}$. The parameter m controls how τ is distributed to $\boldsymbol{\lambda}$. Generally, larger values of m promote sparser solution paths when τ is small. The constraint $\boldsymbol{\lambda}^T \mathbf{1} = \tau$ is equivalent to $(\boldsymbol{\xi}^{1/m})^T \mathbf{1} = \tau$ by regarding $\boldsymbol{\xi} = \boldsymbol{\sigma}_u^{-2}$ as an approximate “effect size vector.” The

optimizer must decide on which effects to spend the fixed τ . If m is large then there is a big cost in moving a component of $\boldsymbol{\xi}$ away from zero. For components of $\boldsymbol{\xi}$ that are already positive there is a smaller cost in increasing any of them the same amount. Thus, the optimizer will not enter a new variable into the model unless $Q(\boldsymbol{\lambda})$ will greatly benefit, but is then more free to shuffle the distribution of τ around between active variables. Thus choosing $m = 2$ results in both sparser and more stable solution path estimates when τ is roughly $< p$. On the other hand, choosing m smaller means relatively little cost for entering a new variable and thus more noise in a solution path from variables entering and exiting. We take $m \in \{1, 2\}$ to coincide with measurement error precisions or root precisions, but any $m > 0$ is feasible.

As a final note, the flexibility of model-agnostic variable selection has a high computational cost. The cost is directly proportional to the cost of fitting and predicting on one model via $\rightarrow \blacksquare \rightarrow$, and the efficiency of the chosen optimizer. For example, if $p = 10$ and the default modified coordinate descent grid size is 20, a single coordinate descent iteration costs 200 black-box fits and predictions. Considering multiple coordinate descent iterations, fine-tuning optimization, and computing $\hat{\boldsymbol{\lambda}}$ over a grid of τ values, there can easily be over 10,000 black-box fits or predictions to determine a MEMSEL solution path. Using $\mathbf{T} = \mathbf{X}$, so that $\hat{\mu}(\cdot)$ may be computed only once in fitting and reused in predictions, and using the regression calibration approximation \mathbf{M} and its diagonal counterpart \mathbf{D} from (4.6) and (4.7) instead of \mathbf{W} will greatly increase computational speed.

4.4 MEMSEL in Linear Models

Studying MEMSEL in the linear model draws concrete parallels to the abstract steps presented in Section 4.3 and suggests expected behavior to MEMSEL in more complex models. Assume the linear model $\mathbf{Y} = \mathbf{T}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon}$ are iid, $E(\boldsymbol{\epsilon} \mid \mathbf{X}) = 0$ and $\text{Var}(\boldsymbol{\epsilon}) = \sigma_{\boldsymbol{\epsilon}}^2$ so that $\mu_{\mathbf{T}}(\mathbf{t}) = E(Y \mid \mathbf{T} = \mathbf{t}) = \mathbf{t}^T \boldsymbol{\beta}$. Selecting $\rightarrow \blacksquare \rightarrow$ as minimizing $\|\mathbf{Y} - \mathbf{T}\boldsymbol{\beta}\|^2/n$ gives the estimator $\hat{\boldsymbol{\beta}} = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T \mathbf{Y}$. This leads to the following definitions of $\hat{\mu}(\cdot)$ in the linear model (with

derivations relegated to Appendix 4.7.2):

$$\hat{\mu}_{\mathbf{X}}(\mathbf{t}) = \mathbf{t}^T \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}$$

$$\hat{\mu}_{\mathbf{W}}(\mathbf{t}) = \mathbf{t}^T \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \quad (\text{for any choice of } B \geq 1)$$

$$\hat{\mu}_{\mathbf{M}}(\mathbf{t}) = \mathbf{t}^T \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\lambda^m\}}^{-1} \right) \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}$$

$$\hat{\mu}_{\mathbf{D}}(\mathbf{t}) = \mathbf{t}^T \mathcal{D}_{\{1 + \lambda^{-m}\}} \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}$$

Now we can derive the MEMSEL objective function Q for (\mathbf{T}, \mathbf{S}) pairs of interest. The MEMSEL loss \mathcal{L} is taken to be squared error loss to match with least squares in linear regression and the black box. As above, full derivations are available in Appendix 4.7.2. The totality of possible MEMSEL objective functions are:

$$Q_{x,x}(\boldsymbol{\lambda}) = \widehat{V}_Y - \widehat{V}_{XY}^T \widehat{V}_X^{-1} \widehat{V}_{XY} \quad (\text{constant in } \boldsymbol{\lambda})$$

$$Q_{x,w}(\boldsymbol{\lambda}) = \widehat{V}_Y - 2\widehat{V}_{XY}^T \widehat{V}_X^{-1} \widehat{V}_{XY} + \widehat{V}_{XY}^T \widehat{V}_X^{-1} \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right) \widehat{V}_X^{-1} \widehat{V}_{XY}$$

$$Q_{w,x}(\boldsymbol{\lambda}) = \widehat{V}_Y - 2\widehat{V}_{XY}^T \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \widehat{V}_{XY} \\ + \widehat{V}_{XY}^T \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \widehat{V}_X \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \widehat{V}_{XY}$$

$$Q_{w,w}(\boldsymbol{\lambda}) = \widehat{V}_Y - \widehat{V}_{XY}^T \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right)^{-1} \widehat{V}_{XY}$$

$$Q_{M,x}(\boldsymbol{\lambda}) = \widehat{V}_Y - 2\widehat{V}_{XY}^T \widehat{V}_X^{-1} \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right) \widehat{V}_X^{-1} \widehat{V}_{XY} \\ + \widehat{V}_{XY}^T \widehat{V}_X^{-1} \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right) \widehat{V}_X^{-1} \left(\widehat{V}_X + \mathcal{D}_{\{\lambda^m\}}^{-1} \right) \widehat{V}_X^{-1} \widehat{V}_{XY}$$

$$Q_{x,M}(\boldsymbol{\lambda}) = Q_{w,x}(\boldsymbol{\lambda})$$

$$Q_{D,x}(\boldsymbol{\lambda}) = \widehat{V}_Y - 2\widehat{V}_{XY}^T \mathcal{D}_{\{1+\lambda^{-m}\}} \widehat{V}_X^{-1} \widehat{V}_{XY} \\ + \widehat{V}_{XY}^T \widehat{V}_X^{-1} \mathcal{D}_{\{1+\lambda^{-m}\}} \widehat{V}_X^{-1} \mathcal{D}_{\{1+\lambda^{-m}\}} \widehat{V}_X^{-1} \widehat{V}_{XY}$$

$$Q_{x,D}(\boldsymbol{\lambda}) = \widehat{V}_Y - 2\widehat{V}_{XY}^T \mathcal{D}_{\{1+\lambda^{-m}\}}^{-1} \widehat{V}_X^{-1} \widehat{V}_{XY} \\ + \widehat{V}_{XY}^T \widehat{V}_X^{-1} \mathcal{D}_{\{1+\lambda^{-m}\}}^{-1} \widehat{V}_X \mathcal{D}_{\{1+\lambda^{-m}\}}^{-1} \widehat{V}_X^{-1} \widehat{V}_{XY}$$

$$Q_{x,x}(\boldsymbol{\lambda}) = Q_{M,M}(\boldsymbol{\lambda}) = Q_{D,D}(\boldsymbol{\lambda}) \quad (\text{constant in } \boldsymbol{\lambda})$$

The contamination variants that have constant $Q(\cdot)$ functions in $\boldsymbol{\lambda}$ are trivially inadmissible as MEMSEL selection methods in the linear model. Additionally we rule out MEMSEL variants (\mathbf{X}, \mathbf{W}) , (\mathbf{M}, \mathbf{X}) , and (\mathbf{D}, \mathbf{X}) from performing selection by observing that the common factor $(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}^m\}}^{-1})$ in each does not permit any component of $\boldsymbol{\lambda}$ to be 0. These methods are comparable to ridge regression methods. Conversely, $(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\lambda}^m\}}^{-1})^{-1}$ can be written as $\left\{ \left(\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}^m\}} + \mathbf{I} \right) \mathcal{D}_{\{\boldsymbol{\lambda}^m\}}^{-1} \right\}^{-1} = \mathcal{D}_{\{\boldsymbol{\lambda}^m\}} \left(\widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}^m\}} + \mathbf{I} \right)^{-1}$ where the inverse always exists and any component of $\boldsymbol{\lambda}$ is permitted to be zero. Thus, (\mathbf{W}, \mathbf{X}) , (\mathbf{W}, \mathbf{W}) , (\mathbf{X}, \mathbf{M}) , and (\mathbf{X}, \mathbf{D}) are admissible forms of MEMSEL.

Chapter 1 [68] contains the full proof that (\mathbf{W}, \mathbf{W}) with $m = 1$, or WW1, MEMSEL in the linear model produces an equivalent solution path to LASSO. Specifically, for any $\tau \geq 0$ there exists an $\eta \geq 0$ such that the minimizer of $n^{-1} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \eta \|\boldsymbol{\beta}\|_1$ and constrained minimizer of $Q_{\mathbf{W}, \mathbf{W}}(\boldsymbol{\lambda})$ satisfy $\widehat{\boldsymbol{\lambda}} = |\widehat{\boldsymbol{\beta}}|/\eta$, noting the change in parameterization from [68] to accommodate for the change in standardization of \mathbf{X} and Y between chapters. The remarkable relationship between the coefficient vector and tuning parameter in LASSO and the coefficient vector in MEMSEL show that (\mathbf{W}, \mathbf{W}) MEMSEL is a generalization of LASSO.

The equivalence proof requires both a linear model and standardized contamination data as described in (4.5). More complex (nonlinear) models that exhibit sparsity from L_1 -penalization do not generally have MEMSEL equivalents. However, (\mathbf{W}, \mathbf{W}) MEMSEL or other MEMSEL variants still produce LASSO-like sparse solution paths outside of the linear model. Other variants are explored below.

While it is true that the (absolute) LASSO and WW1 MEMSEL solution paths are equivalent, tuning parameter methodology that relies on the training fit will be different. The LASSO objective is subject to the L_1 -penalty term, but the penalty term is not included as part of the loss when evaluating training fit. Conversely, the MEMSEL objective and loss are identical and equivalent to the penalized LASSO objective. Note from [68] that for any given η , the LASSO solution can be written as $\widehat{\boldsymbol{\beta}}_L = \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{|\widehat{\boldsymbol{\beta}}_L|/\eta\}}^{-1} \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y} = \mathcal{D}_{\{\boldsymbol{\lambda}\}} (\mathbf{I} + \widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}})^{-1} \widehat{\mathbf{V}}_{\mathbf{X}Y}$. Further, $\tau = \mathbf{1}^T \boldsymbol{\lambda} = \|\boldsymbol{\beta}\|_1/\eta$. Thus the LASSO penalty $\eta \|\boldsymbol{\beta}\|_1$ is

$P(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \left(\sum_j \left| [\mathcal{D}_{\{\boldsymbol{\lambda}\}}(\mathbf{I} + \widehat{\mathbf{V}}_{\mathbf{X}} \mathcal{D}_{\{\boldsymbol{\lambda}\}})^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}]_j \right| \right)^2 / \sum_j \lambda_j$. To get equivalence between LASSO and WW1 MEMSEL for both solution paths and tuning, $Q_{\mathbf{w},\mathbf{w}}(\boldsymbol{\lambda})$ is optimized but $Q_{\mathbf{w},\mathbf{w}}(\boldsymbol{\lambda}) - P(\boldsymbol{\lambda})$ is used as the training fit for tuning. The term $P(\boldsymbol{\lambda})$ is used as a correction between the objective and training fit in other MEMSEL methods as well.

4.4.1 Simulation Study Setup

The performance of MEMSEL variants in the linear model is compared in a simulation study to gain insight about their relative performance in a familiar and well-studied setting. The MEMSEL black box is as described in Section 4.4. Experimental setups are inspired from those in the original LASSO paper [75]. Each assumes linear model (4.1) with theoretical $R^2 = 0.75$ and AR(1) correlation in \mathbf{X} , $\rho = 0.50$, unless otherwise noted. The sample size is fixed at $n = 100$ and prediction errors are evaluated on a hold-out set of 10,000 test points. Each simulation comprises 100 Monte Carlo replicates.

Linear-model MEMSEL variants are compared to elastic net (E.N.) and LASSO (LAS.) over three tasks, active selection error (the number of important predictors excluded), irrelevant selection error (the number of unimportant predictors included), and test loss as the average squared error between actual and predicted test set \mathbf{Y} vectors. MEMSEL variants are labeled in the form “TSm”, recalling that \mathbf{T} is the data set used for fitting, \mathbf{S} is the data set used for evaluation, and m relates $\boldsymbol{\lambda}$ to the measurement error precision by $\boldsymbol{\lambda}^m = \boldsymbol{\sigma}_u^{-2}$. MEMSEL predictions are done with the two-step fitting method. Elastic net is restricted to have the L_1 - and L_2 -penalty terms are equally weighted [87]. The base ordinary least squares (OLS) model with all predictors included (Naive) and only the important predictors included (Oracle) are also shown to compare predictions without shrinkage or selection and with perfect selection and no shrinkage, respectively. All simulations are performed in R [57] and non-MEMSEL fitting and tuning is done with the package `glmnet` [22].

The tuning criteria considered in this study are similar to those in Section 4.2: small-sample-corrected AIC (AICc), BIC, Selection Information Criterion (SIC), and two forms of five-fold

cross-validation (CV), CV-min and CV-1se. Cross-validated tuning methods are excluded for any MEMSEL model that fits or evaluates on \mathbf{W} in the interest of speed. The simulation models under consideration are:

Model 1. Let $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)$ so $p = 8$ with three important predictors.

Model 2. Identical to Model 1 except $\boldsymbol{\beta} = (5, 0, 0, 0, 0, 0, 0, 0)$ so the true model is sparse.

Model 3. Let $\boldsymbol{\beta} = (8, \dots, 1, 0, \dots, 0)$ with $p = 16$ and eight important predictors.

Recall WW1 MEMSEL is equivalent to the LASSO. However, test losses between LASSO and WW1 MEMSEL are not expected to be equal because MEMSEL is using the two-step predictions and LASSO is not. Further, minor selection error discrepancies arise between the two methods in practice because MEMSEL optimization may not attain the global minimum of $Q(\boldsymbol{\lambda})$. As τ increases, the point at which a new covariate enters the model is in a “corner” of the $\boldsymbol{\lambda}$ space. See Figure 3.1 for a visual reference. The grid used in modified coordinate descent may be too coarse to find minima until they are sufficiently far from a corner. Figure 4.6 shows the effect of the grid size on the accuracy of WW1 MEMSEL on one data set. The left panel uses a grid size of 10 and the MEMSEL solutions (red dotted line) tend to “overshoot” the true LASSO solution (gray solid line) because of wide grid spacing. Tuned MEMSEL solutions then tend to underselect slightly. The effect is reduced when the grid size is increased to 100 (right panel). A recommended grid size that balances computational speed and accuracy is generally between 20 and 50.

4.4.2 Simulation Study Results

Results for Models 1-4 are shown in Figures 4.7-4.9. Of primary interest is the performance of the faster regression calibration variants of MEMSEL against the slower variants. In Model 1, regression calibration MEMSEL variants (XM1, XM2, XD1, XD2) generally perform better than LASSO in terms of both irrelevant selection error and test loss. Tuning with cross validation (CV) is the exception, but CV tuning is slow and undesired for MEMSEL methods anyway.

All methods performed active selection error well with the exception of XM2 missing a small fraction of active variables using SIC. Elastic net performs slightly worse than LASSO. MEMSEL variants with $m = 2$ perform slightly better than those with $m = 1$. This is due to the sparse model and solution paths favoring sparsity when $m = 2$ as mentioned in Section 4.3.2. Because LASSO is theoretically equivalent to WW1, WW2 may prove as an improved generalized LASSO when applied to other black-box models. Neither WX1 nor WX2 offered an improvement over XM1 and XM2, and are both slower to compute. In terms of tuning, SIC generally performed the best on MEMSEL methods but Model 1 is not a good test for active error rates. CV-min is significantly worse than other methods.

Model 2 tests the methods on a sparser linear model than Model 1. Results are shown in Figure 4.8. Every method selected the only important variable in every run. All MEMSEL variants show improvements over elastic net and LASSO. MEMSEL variants with $m = 2$

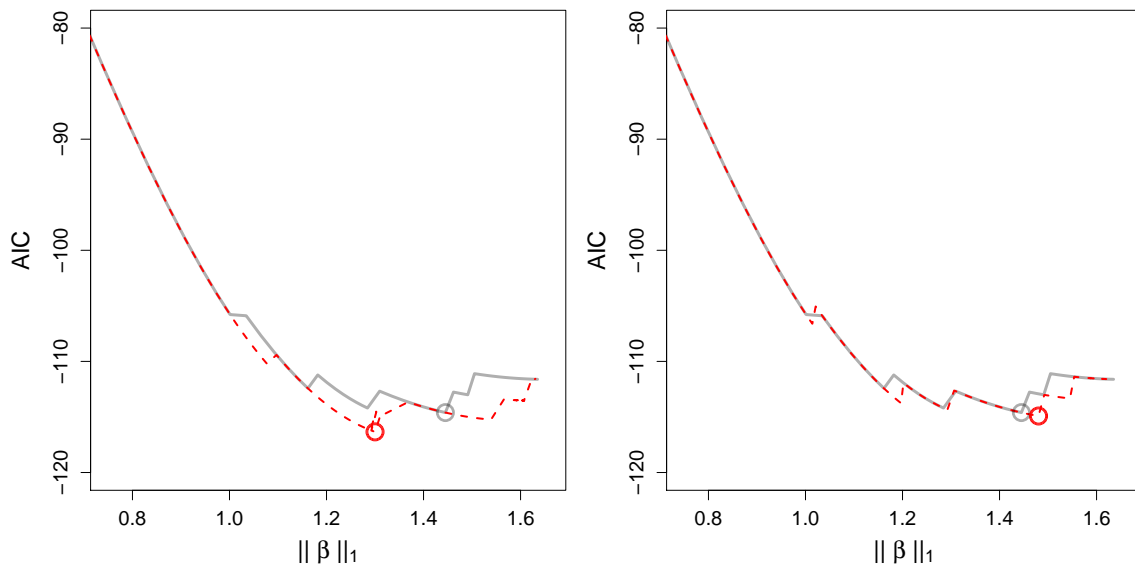


Figure 4.6 Comparing AIC curves against coefficient size between LASSO (gray solid line) and the equivalent MEMSEL variant, (\mathbf{W}, \mathbf{W}) with $m = 1$ (red dashed line). Left: a grid size of 10 is used in MEMSEL modified coordinate descent. Right: a grid size of 100 is used in MEMSEL modified coordinate descent. Identical data are used for both plots. Discrepancies occur when optimized $\hat{\lambda}_\tau$ values are not global minimizers. The gray (red) circle denotes the minimum AIC value for LASSO (MEMSEL).

tuned with BIC, CV-1se, or SIC have near perfect irrelevant selection and therefore test losses equivalent to oracle test loss. Variants with $m = 1$ are slightly less selective.

Model 3 tests the methods on a more difficult model for selection where there is a gradient of effects. Figure 4.9 shows that the more selective MEMSEL variants ($m = 2$) now miss important variables and have worse test loss. This is exacerbated by selective tuning methods (BIC, CV-1se, SIC). XM1 has an advantage over elastic net and LASSO on all tasks. Like in Model 1, XM1 and XM2 are equivalent to WX1 and WX2 in terms of performance but faster to compute.

In each of these models, using \mathbf{D} in place of \mathbf{M} for faster computation as suggested in Section 4.3.1 shows roughly a 10% improvement in speed. This gain will be washed out in slower black-box prediction models where fitting and/or evaluation absorbs the most computation time. Thus, we recommend using \mathbf{M} .

We draw three important conclusions from these simulations. First, the regression calibration variants of MEMSEL (XM1, XM2) perform at least as well as the much slower MEMSEL methods that use \mathbf{W} for contamination (WW1, WW2, WX1, WX2). Second, the use of AICc, BIC, or SIC developed in Section 4.2 is justified to perform faster tuning versus cross validation. Faster MEMSEL computations make possible the use of far more complex and interesting black-box models. Finally, despite the LASSO equivalence when $m = 1$, taking $m = 2$ in MEMSEL models appears to be favorable for selection in sparse models. Considering both speed and performance, XM2 appears to be the most appealing MEMSEL variant, followed by XM1.

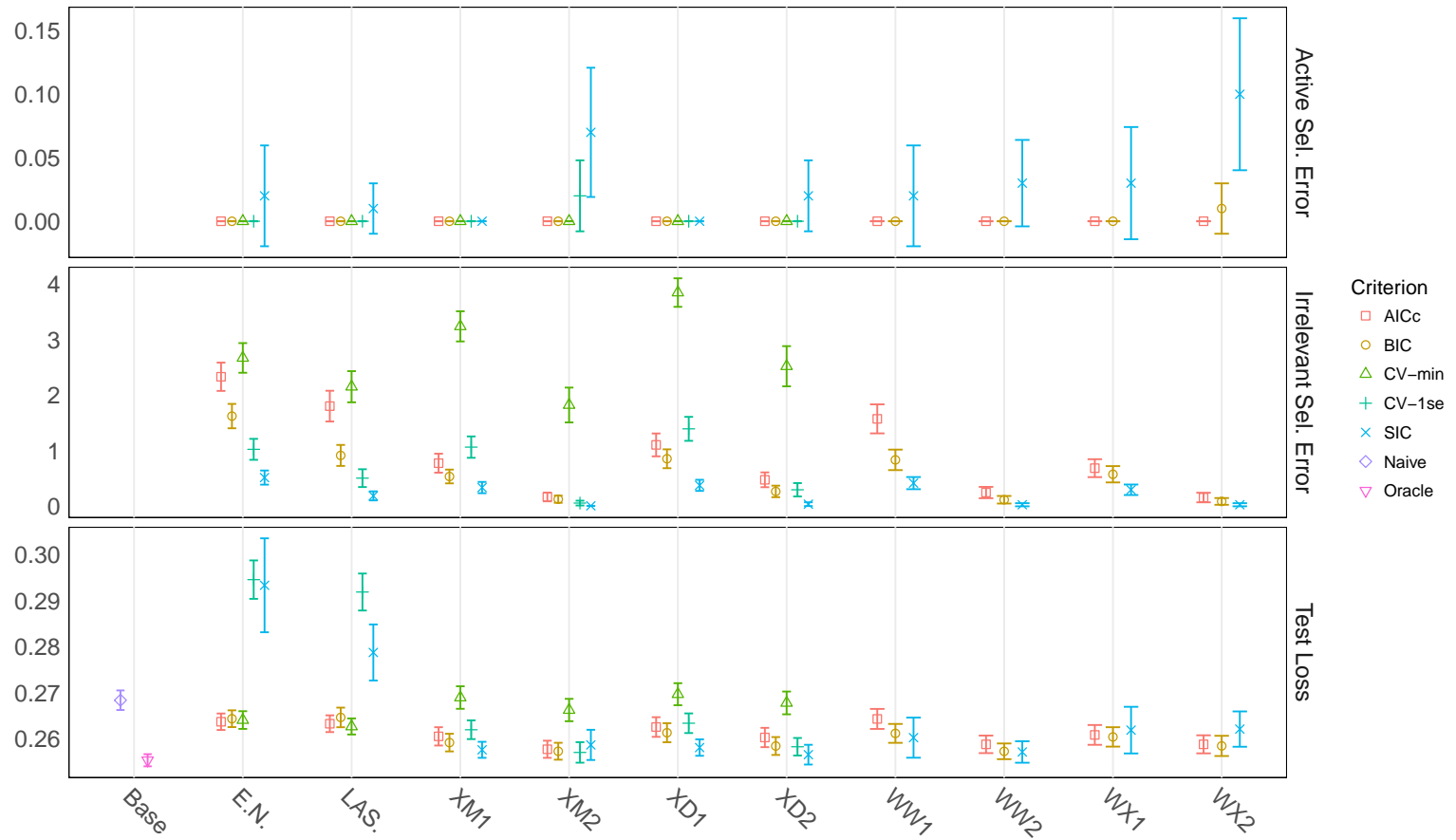


Figure 4.7 Results from linear Model 1. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the test loss axis is truncated.

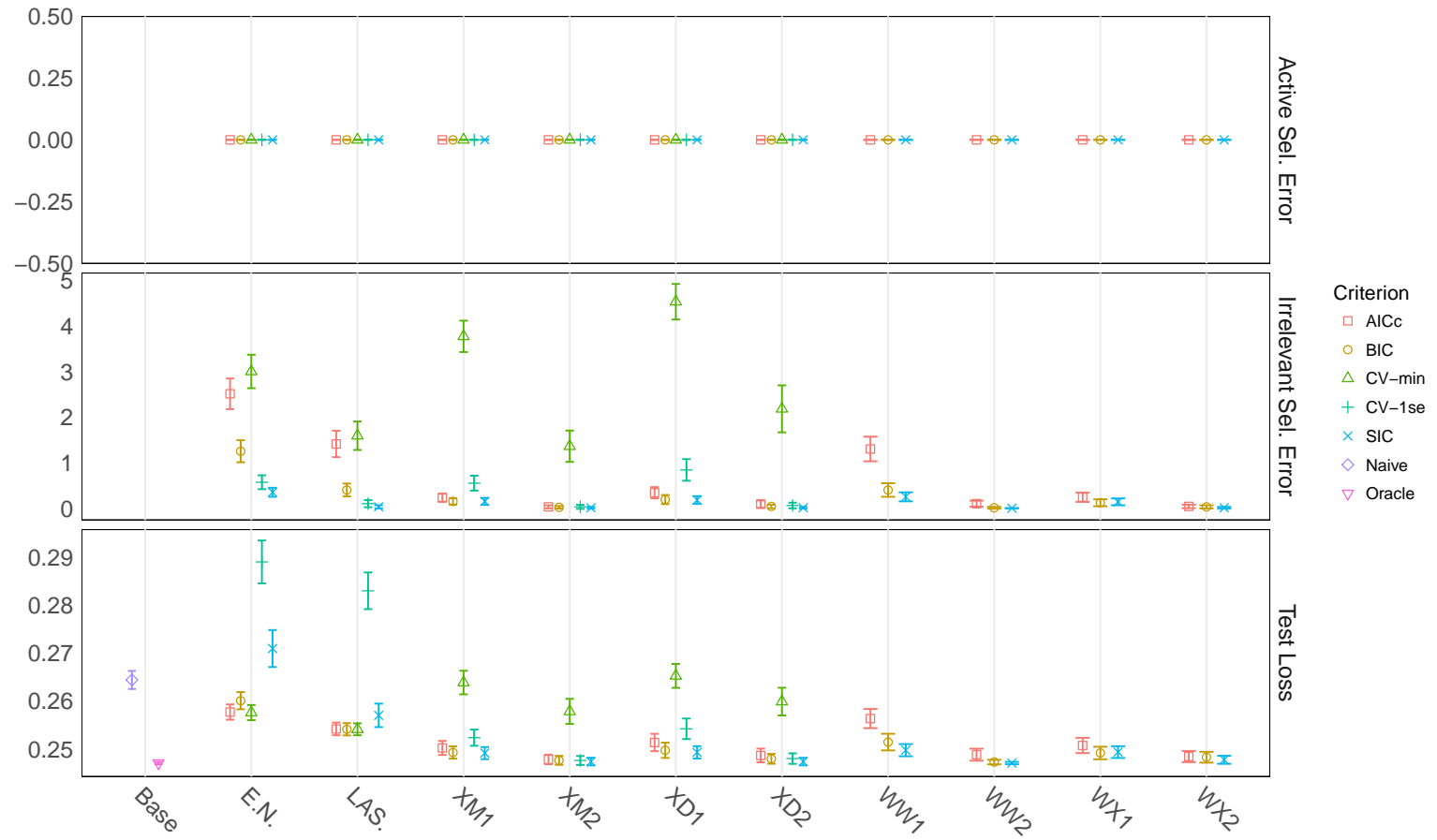


Figure 4.8 Results from linear Model 2. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the test loss axis is truncated.

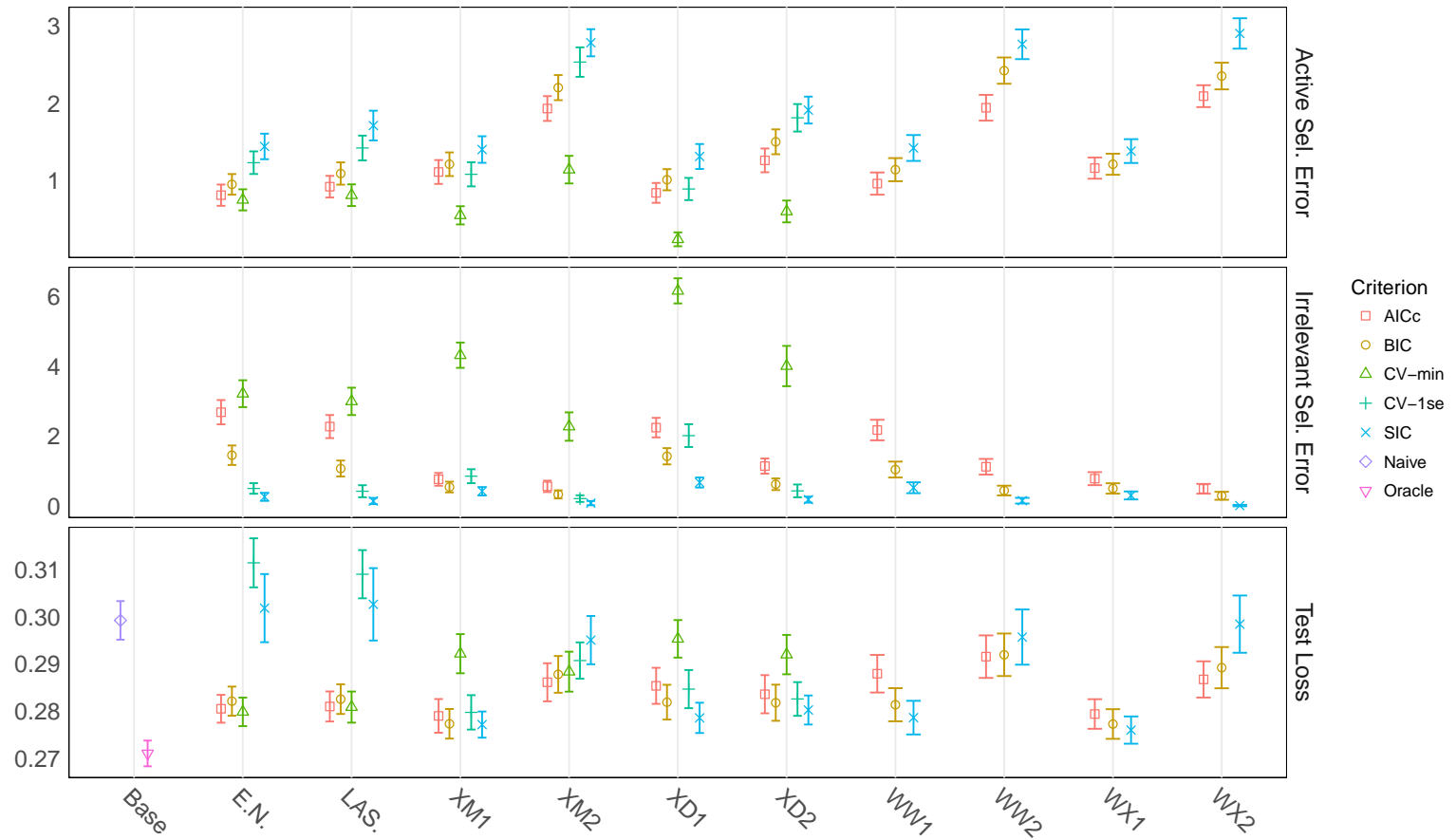


Figure 4.9 Results from linear Model 3. Estimation methods are along the horizontal axis and selection errors and test loss are along the vertical axis separated by panes. Tuning methods are grouped within fitting method and show a 95% confidence interval. Note the active selection error and test loss axes are truncated.

4.5 MEMSEL in Random Forests

4.5.1 Motivation

Random forests (RF) [7] are a simple yet powerful tool to perform predictions when predictive power is valued over model interpretability. A forest’s ensemble of decision trees smooths out the bias produced from a single tree but obscures the effect of each individual variable. RF *can* estimate relative variable importance (VI) with a noising algorithm similar to that of MEMSEL. Individual predictors have their values permuted and the resulting out-of-bag error rates are averaged across all trees and compared to the un-permuted error rates. A large error increase indicates that the variable carries more significant predictive power in the data. However, because VI is a relative measure it alone is not a rule for pruning.

Given the lack of an established routine for variable selection, random forests are a good candidate for MEMSEL; removing irrelevant variables from a forest’s pool of candidate predictors can only improve the out-of-sample loss. We assume the generic model $Y = \mu_{\mathbf{T}}(\mathbf{T}) + \epsilon$ where again ϵ are iid, $E(\epsilon | \mathbf{X}) = 0$ and $\text{Var}(\epsilon) = \sigma_{\epsilon}^2$. Thus $\mu_{\mathbf{T}}(\mathbf{t}) = E(Y | \mathbf{T} = \mathbf{t})$. We choose $\rightarrow \blacksquare \rightarrow$ as forming predictions $\mu_{\mathbf{T}}(\cdot)$ from a random forest [7]. The MEMSEL loss \mathcal{L} is taken as squared error loss. We focus on models with $\mathbf{T} = \mathbf{X}$ and $\mathbf{S} = \mathbf{M}$ for speed considerations because random forests are relatively slow to fit. Taking $\mathbf{T} = \mathbf{X}$ and avoiding \mathbf{W} allows us to reuse one model fit throughout estimation and evaluate $Q(\boldsymbol{\lambda})$ with only one RF prediction on contaminated data.

4.5.2 Selection Using Variable Importance

Recent articles have leveraged RF variable importance to perform variable selection. The primary challenge is choosing a exclusion threshold for either the variable importances directly, or the out-of-bag error (OOB) rates or p -values. It is common to use the bootstrapped resampling in RF to estimate thresholds from the empirical distribution of variable importances. Multi-step methods compute variable importances and OOB error rates on sequentially nested subsets of variables with the highest importance until a stopping criterion is met. A different approach

is to estimate the p -value for each variable via permutation test methodology and choose a threshold on those p -values. See [35] for a detailed overview.

A competing permutation test-based approach to RF variable selection is offered in [35]. We abbreviate this method as “HAP” after the last name of the first author. HAP claims to distinguish important from irrelevant features more often than its competitors and thus it is used as one benchmarking method for MEMSEL. HAP first computes a variable importance for \mathbf{X}_j . It then permutes the values of \mathbf{X}_j 100 times and recomputes importance measures to determine empirical distribution of importances under the assumption that \mathbf{X}_j is independent of \mathbf{Y} and $\mathbf{X}_{-j} = (\mathbf{X}_1, \dots, \mathbf{X}_{j-1}, \mathbf{X}_{j+1}, \dots, \mathbf{X}_p)$. Only if the permutation test p -value falls under a Bonferroni-controlled threshold is \mathbf{X}_j kept. This procedure is repeated for $j = 1, \dots, p$ to determine the inclusion set. Predictions are formed using a new forest grown from only the selected variables.

Variable Selection Using Random Forests (VSURF) [25] is another variable selection approach that we use for benchmarking because it has steps in place to specifically optimize selection and prediction performance separately. VSURF performs three nested selection steps. The first step removes likely irrelevant variables by estimating a VI threshold below which a variable is eliminated. The second step computes sequential out-of-bag (OOB) error rates by using the top rank-ordered variables by importance and chooses the smallest subset not more than one standard deviation from the subset with the lowest error rate. This subset is the “interpretation” subset meant to identify all variables that are relevant. The final step reduces variable redundancy by again sequentially checking the rank-ordered variables by importance and including a new predictor only if the OOB error decreases by more than a threshold estimated from the data. This “prediction” set is a subset of the “interpretation” variables meant to eliminate redundancy in the interpretation subset. The interpretation or prediction step may deem that the mean model is best by excluding every variable. In this case we set predicted values identically to $\bar{\mathbf{Y}}$.

HAP departs from VSURF by using conditional inference random forests [71, 72] to perform predictions and determine variable importance (VI). The splits in trees that make up conditional

inference random forests are chosen from permutation test results as opposed to a single impurity or mean squared error measure [41].

HAP, VSURF and MEMSEL all select variables by adding noise to inputs. However, both HAP and VSURF do so in a binary and independent fashion. Variable permutations aggressively break the relationship with \mathbf{Y} and are done one at a time holding the other $p - 1$ variables uncontaminated. MEMSEL, in contrast, adds continuous amounts of contamination and accounts for dependencies between each input through the selection likelihood optimization. Thus MEMSEL should have more power to distinguish important from irrelevant variables. The major tradeoff is the higher computational cost of MEMSEL.

4.5.3 Simulation Study Setup

We compare the proposed random forest MEMSEL variant (\mathbf{X}, \mathbf{M}) using $m = 1$ (XM1) and $m = 2$ (XM2) against HAP and VSURF. Performance is evaluated on irrelevant variable selection error (average number of irrelevant variables included), important variable selection error (average number of important variables excluded), and prediction error defined as the average squared error between actual and predicted test set \mathbf{Y} vectors of dimension 10,000. Data are generated as described in Section 4.2.2 but using five different base models for $\mu_{\mathbf{T}}(\mathbf{t})$. We consider both smooth and nonsmooth models of $\mu_{\mathbf{T}}(\mathbf{t})$ with respect to \mathbf{t} . Each data model assumes $n = 100$, a theoretical $R^2 = 0.75$, and $\rho = 0.25$ unless otherwise noted. Recall $p_1 =$ count of important variables. Each simulation comprises 100 Monte Carlo replicates. The base simulation models under consideration are listed below.

Smooth Model 1 (few large effects): $\mu_{\mathbf{T}}(\mathbf{t}) = \mathbf{t}^T \boldsymbol{\beta}$; $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)$; $p = 8$; $p_1 = 3$.

Smooth Model 2 (effect gradient): $\mu_{\mathbf{T}}(\mathbf{t}) = \mathbf{t}^T \boldsymbol{\beta}$; $\boldsymbol{\beta} = (10, \dots, 1, 0, \dots, 0)$; $p = 20$; $p_1 = 10$.

Nonsmooth Model 1 (slowly-varying effects): $\mu_{\mathbf{T}}(\mathbf{t}) = (t_1)_+ + (t_3)_+ + S(t_7, t_9)$; $p = 10$; $p_1 = 4$. Define $(t)_+ = \max(t, 0)$ and $S(x, y) =$ the indicator function that $x < 0$ and $y < 0$.

Nonsmooth Model 2 (quickly-varying effects): $\mu_{\mathbf{T}}(\mathbf{t}) = \max(t_1, t_7) + (t_3 + t_9)_+$; $p = 10$; $p_1 = 4$.

Nonsmooth Model 3 (many effects): $\mu_{\mathbf{T}}(\mathbf{t}) = (t_1)_+ + (t_3)_+ + S(t_7, t_9) + S(t_{10}, t_{12}) + S(t_{14}, t_{15}) + S(t_{16}, t_{20})$; $p = 20$; $p_1 = 10$.

Additionally, we may vary one simulation factor at a time to understand how each of the four methods respond to changes in sample size, number of predictors, correlation in \mathbf{X} , or residual error variance. All possible model \times tweak combinations are not considered in the interest of space.

4.5.4 Software Considerations

All simulations are performed in R. The source code for HAP is available in the supplementary files of [35]. HAP relies on `cforest()` in the package `party` for fitting and prediction [40, 71, 72]. VSURF is available in the `VSURF` package [26]. VSURF “interpretation step” variables are used for judging selection and “prediction step” variables are considered for re-building a forest with which to perform predictions. MEMSEL methods use traditional random forests available in the R package `randomForest` [48]. Although fast R packages for building forests exist, MEMSEL requires fast predictions (if uncontaminated data are used to fit); `randomForest()` evaluated predictions faster than both `ranger()` and `Rborist()`.

For HAP, VSURF, and MEMSEL, the number of candidate variables to use for each split is set at `mtry = max{floor(p/3), 1}`. The number of trees used in HAP, VSURF, and MEMSEL are their defaults of 50, 2000, and 500, respectively; increasing this value for either HAP or MEMSEL does not make an improvement. Trees are fully grown without pruning and terminal nodes must have at least five observations. The Bonferroni-adjusted p -value in HAP is $0.05/\dim(\mathbf{X})$. MEMSEL is tuned using SIC. AICc, BIC, and GCV proved to not penalize aggressively enough and favored the largest possible models. Cross validation is prohibitively slow.

4.5.5 Simulation Study Results

The results of the simulations are shown in Table 4.2. Bold entries in the table denote errors that are not statistically different from those with the lowest error in the row using a paired t -test at $\alpha = 0.20$. A lower significance level is chosen to limit the number of examples that

are statistically indistinguishable, and further because researchers likely prefer a method that outperforms competitors even at $\alpha = 0.20$.

Our proposed MEMSEL method XM2 performs exceedingly well in general on the critical tasks of important variable selection and prediction. It tends to significantly overselect irrelevant variables compared to HAP only in cases where there are many irrelevant variables. However, those errors do not impact MEMSEL prediction greatly because they simply will not be used for splitting as often in the refitted random forest. By contrast, XM1 that sets $m = 1$ performs far worse than XM2. Higher important *and* irrelevant selection errors for XM1 suggest that it is overall worse in variable discrimination as opposed to only more or less selective. This reinforces our claim in Section 4.3.2 that MEMSEL with $m = 2$ should be a better selector. VSURF appears to gain an advantage in important selection error when the signal-to-noise ratio is lowered. When the sample size is increased from 100 to 400, MEMSEL shows drastically reduced overall selection errors but both HAP and VSURF have *higher* irrelevant selection errors. This suggests favorable large-sample properties for MEMSEL. There does not appear to be a difference in performance for any method between smooth and nonsmooth models.

Table 4.2 Selection and prediction errors for random forest MEMSEL (XM1, XM2), HAP, and VSURF. Simulation factor tweaks are shown in the “Setup” column, e.g., $R^2 : 0.75 \searrow 0.50$ indicates decreasing R^2 from 0.75 to 0.50 in the model. Values in bold are statistically no different from those with the lowest error rate across the row using a paired t -test at $\alpha = 0.20$.

Setup	Error	XM1	XM2	HAP	VSURF
Smooth Model 1 (SM1)	Sel. (Irrelevant)	0.34	0.09	0.20	0.13
	Sel. (Important)	0.16	0.04	0.12	0.07
	Prediction	0.40	0.37	0.44	0.42
$p : 10 \nearrow 30$	Sel. (Irrelevant)	1.52	0.96	0.33	0.49
	Sel. (Important)	0.02	0.00	0.15	0.05
	Prediction	0.41	0.39	0.45	0.39
$R^2 : 0.75 \searrow 0.50$	Sel. (Irrelevant)	0.26	0.06	0.07	0.53
	Sel. (Important)	0.62	0.57	0.45	0.12
	Prediction	0.68	0.67	0.65	0.70
Smooth Model 2 (SM2)	Sel. (Irrelevant)	0.03	0.03	0.04	0.04
	Sel. (Important)	4.21	3.59	4.54	3.69
	Prediction	0.48	0.46	0.56	0.49
$p : 11 \nearrow 20$	Sel. (Irrelevant)	0.75	0.63	0.14	0.21
	Sel. (Important)	3.01	2.81	4.85	3.80
	Prediction	0.46	0.46	0.57	0.47
Nonsmooth Model 1 (NS1)	Sel. (Irrelevant)	0.50	0.28	0.22	0.67
	Sel. (Important)	0.46	0.50	1.22	0.39
	Prediction	0.49	0.48	0.54	0.57
$p : 10 \nearrow 30$	Sel. (Irrelevant)	3.48	3.22	0.49	1.41
	Sel. (Important)	0.32	0.19	1.39	0.73
	Prediction	0.51	0.51	0.54	0.50
$R^2 : 0.75 \searrow 0.50$	Sel. (Irrelevant)	0.31	0.26	0.10	1.02
	Sel. (Important)	1.44	1.33	1.73	0.85
	Prediction	0.77	0.73	0.74	0.74
$\rho : 0.25 \nearrow 0.75$	Sel. (Irrelevant)	1.65	0.64	2.18	2.14
	Sel. (Important)	0.56	0.72	0.96	0.63
	Prediction	0.43	0.42	0.48	0.43
$n : 100 \nearrow 400$	Sel. (Irrelevant)	0.10	0.02	0.68	0.88
	Sel. (Important)	0.00	0.00	0.00	0.00
	Prediction	0.35	0.35	0.36	0.37
Nonsmooth Model 2 (NS2)	Sel. (Irrelevant)	0.60	0.39	0.32	0.29
	Sel. (Important)	0.15	0.04	0.43	0.08
	Prediction	0.51	0.49	0.60	0.52
Nonsmooth Model 3 (NS3)	Sel. (Irrelevant)	1.11	1.02	0.22	0.70
	Sel. (Important)	4.60	3.98	6.75	4.25
	Prediction	0.71	0.66	0.77	0.69

4.5.6 Real Data Example – Concrete Workability

Before concrete is poured, a sample is drawn from the batch and subject to a slump test to determine if the batch is suitable for use. A cone filled with concrete is inverted to observe how the free-standing wet concrete deforms. If the cone slumps too drastically or not at all, the batch is improperly composed or mixed and must be discarded. Concrete with the correct slump has a high workability and will be easier to pour, compact, and finish while maintaining consistency [85]. Further, superplasticizers are known to increase workability but are costly compared to possible alternatives like fly ash or blast furnace slag. It is then of interest to predict how raw concrete ingredients impact slump to optimize both cost and workability.

We consider a data set of $n = 103$ different concrete recipes and the resulting slump test heights (in cm.), obtained from [49]. The input variables are the $p = 7$ components that define each recipe: cement, slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate, each measured in kg/m^3 . The objective is a model with the lowest root mean squared error (RMSE). The small sample size suggests that RMSE be determined through repeatedly splitting the data into training and test sets.

Previous work in modeling slump has used both linear models with quadratic terms and artificial neural networks (ANN) [85]. The ANN model performs significantly better than the linear model by reducing RMSE from 15.57 to 8.51 cm, although the article implies use of bootstrapped test sets for tuning and this result may be overstated. (Further, the linear model actually performs worse than the simple mean model with $\text{RMSE}=8.71$.) RMSEs were determined by splitting the data into four disjoint subsets, roughly 3/4 for training and 1/4 for testing, and averaging the four test-set RMSEs.

We pursue a lower RMSE with the four random forest models above: our proposed MEMSEL methods XM1 and XM2, and competitors HAP and VSURF. We include a traditional random forest (RF, from `randomForest()`) to assess performance both with and without selection. Indeed, each of the seven variables are truly relevant in determining slump, however, predictions may improve by pruning variables that carry little signal. RMSE results are compared to the

ANN model presented in [85]. We split the $n = 103$ observations into 77 training points and 26 test points. Each method is fit and tuned on the training set. Predictions are done on the test set to determine RMSE. This procedure is repeated 100 times with new randomly split training (75%) and test (25%) sets. We also record the fraction of times each variable is selected and overall fraction of variables selected for each method over the 100 replicates. Software defaults are unchanged from those described in Section 4.5.4.

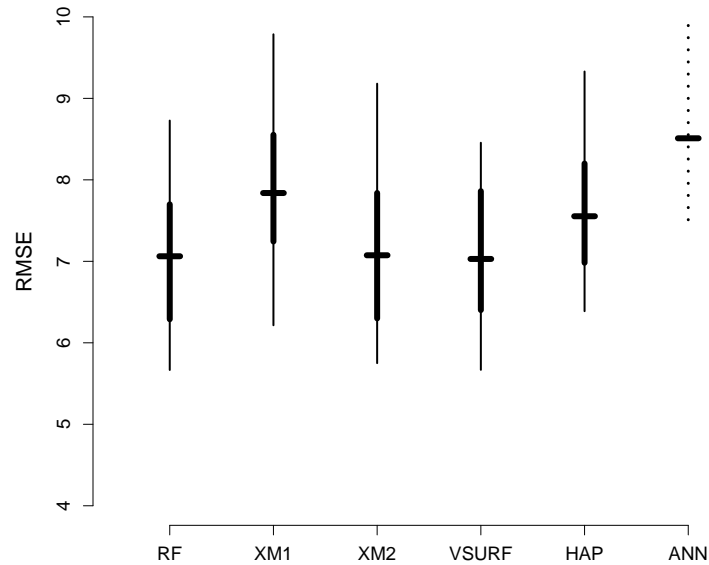


Figure 4.10 Distribution of RMSEs from concrete data. The thin solid vertical line represents the middle 95% of RMSEs, the thick solid vertical line represents the middle 50% of RMSEs, and the horizontal line is the median RMSE over 100 samples. The distribution of RMSEs from ANN is approximated from the range of test-set results in [85].

RMSE results are shown in Figure 4.10 and selection results in Table 4.3. The traditional random forest (RF), MEMSEL XM2, and VSURF exhibit the lowest aggregate RMSEs. (It is important to note that [85] had only 78 observations at the time of publication and the current data set contains an additional 25.) XM2 and HAP are the most selective methods, but XM2 is able to retain good predictive performance relative to HAP while being selective. XM2 estimates that *water* and *slag* densities are the most important in determining slump. Although we know

Table 4.3 Selection results from concrete data. Values are the fraction of times a variable is chosen out of 100 redrawn samples. AVG = average selection rate over all variables.

Variable	XM1	XM2	HAP	VSURF
cement	0.17	0.01	0.00	0.03
slag	0.38	0.90	0.65	1.00
fly ash	0.20	0.07	0.03	0.04
water	1.00	1.00	1.00	1.00
super	0.23	0.01	0.09	0.05
coarse aggr	0.11	0.01	0.13	0.20
fine aggr	0.15	0.01	0.01	0.57
AVG	0.32	0.29	0.27	0.41

that each concrete component is important in determining slump (especially at extreme ratios), this result indicates that water and blast furnace slag be the most carefully measured.

4.6 Summary

This chapter generalizes and extends prior research on Measurement Error Model Selection Likelihoods (MEMSEL) [68, 79]. In addition, a new tuning method, Selection Information Criterion (SIC), is developed and shown to discriminate better than popular competing methods when the true model is sparse. Although there are strong arguments *against* using selection-consistent criteria in models that approximate complex systems (see Section 6.3.2 in [9]), SIC is primarily developed for use in random forest MEMSEL where AIC, BIC, and cross validation do not perform well.

MEMSEL is extended to allow different forms of predictor contamination as inputs to either fitting or predicting (or both) with the black-box fitting method. This requires a slight reframing of the four-step process to build a Measurement Error Model Selection Likelihood described in earlier chapters, although the concept is unchanged. SKDA and MEKRO from Chapters 2 and 3 are special cases of the broadened four-step method. Also, the computationally-less-intensive “regression calibration” approximation introduced in Section 2.5 is described and studied.

The new variants of MEMSEL that arise from its generalization are studied both theoretically

and numerically in the linear model. Many of them are shown to be infeasible by either not introducing λ to the loss function, not achieving variable selection, or performing worse than faster MEMSEL variants. MEMSEL variants that fit on the uncontaminated data \mathbf{X} and predict at the regression-calibration-contaminated data \mathbf{M} exhibit favorable selection and out-of-sample prediction errors when compared to LASSO.

MEMSEL is applied to the random forest black-box predictor, a method with no likelihood to penalize for selection, as a way to introduce variable selection. We demonstrate numerically that MEMSEL outperforms other methods of variable selection for random forests studied in the literature.

Future work on MEMSEL will address its lack of ability to handle inputs with discrete distributions. Although kernel regression MEMSEL [79] allows discrete predictors, it relies on a specific kernel function and cannot be easily generalized outside of kernel methods. We also intend to test MEMSEL on other, more convoluted black-box prediction methods. Finally, a generic proof or proof sketch that constrained minimization of $Q(\lambda)$ provides a selection-consistent $\hat{\lambda}$ is still under investigation.

4.7 Appendix

4.7.1 Centering and Scaling \mathbf{Z}

Let \mathbf{Z}_* be an $n \times p$ matrix of iid $\mathcal{N}(0, 1)$ observations. Define $\mathbf{B} = (\mathbf{1}_n, \mathbf{X}, \mathbf{Y})$. Then set $\mathbf{Z} = \mathbf{Z}_* - \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Z}_*$ so that $\mathbf{1}_n^T\mathbf{Z} = \mathbf{0}_p$, $\mathbf{Y}^T\mathbf{Z} = \mathbf{0}_p$, and $\mathbf{X}^T\mathbf{Z} = \mathbf{0}_{p \times p}$. Note \mathbf{Z} is quickly computed as the residuals after regressing \mathbf{Z}_* on \mathbf{B} . Now \mathbf{Z} must be scaled; we use eigendecomposition. Let $\mathbf{P} = n^{-1}\mathbf{Z}^T\mathbf{Z}$ and because \mathbf{P} is symmetric it can be decomposed as $\mathbf{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ where \mathbf{V} is the orthogonal matrix of stacked eigenvectors of \mathbf{P} and $\mathbf{\Lambda}$ is a matrix with corresponding eigenvalues of \mathbf{P} on the diagonal. Further, \mathbf{P} is positive definite implying $\mathbf{\Lambda}$ is as well. Thus $\mathbf{P}^{1/2} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{V}^T$ and $\mathbf{P}^{-1/2} = \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^T$ by orthogonality of \mathbf{V} . Finally update $\mathbf{Z} \leftarrow \mathbf{Z}\mathbf{P}^{-1/2}$ so that $\mathbf{Z}^T\mathbf{Z} = n\mathbf{I}_p$ as desired.

4.7.2 Derivation of MEMSEL in Linear Models

We derive MEMSEL prediction function variants and MEMSEL loss functions when the black-box model is ordinary least squares as described in Section 4.4. Recall that \mathbf{X} and \mathbf{Y} are standardized, and crossproducts of \mathbf{Z}_b and \mathbf{X} or \mathbf{Y} are the zero matrix of the appropriate dimension because of the choices from (4.5).

Each expression is derived using σ_u^2 as the contamination vector to follow how false measurement error is introduced conceptually. For use in MEMSEL, each expression must be reparameterized as a function of λ using $\lambda^m = \sigma_u^{-2}$.

The $\hat{\mu}(\cdot)$ variants reveal that the exact MEMSEL prediction function $\hat{\mu}_{\mathbf{W}}(\mathbf{t})$ attenuates predictions towards the response mean (centered at 0) when the false measurement errors σ_u^2 are all large. This is expected; if all model variables are poor predictors of \mathbf{Y} , then $\bar{\mathbf{Y}}$ is the best one can do to predict a new response value. Conversely, both $|\hat{\mu}_{\mathbf{M}}(\mathbf{t})|$ and $|\hat{\mu}_{\mathbf{D}}(\mathbf{t})|$ undesirably increase without bound as components of σ_u^2 increase. Thus, given a λ value, one can form sensible predictions from the exact MEMSEL predictor $\hat{\mu}_{\mathbf{W}}(\mathbf{t})$ but not from regression calibration MEMSEL predictors $\hat{\mu}_{\mathbf{M}}(\mathbf{t})$ and $\hat{\mu}_{\mathbf{D}}(\mathbf{t})$.

Examination of the $Q(\cdot)$ equations reveals that only (\mathbf{W}, \mathbf{W}) , (\mathbf{W}, \mathbf{X}) , (\mathbf{X}, \mathbf{M}) , and (\mathbf{X}, \mathbf{D}) are viable MEMSEL variants (in the linear model). The other variants either do not depend on λ or cannot perform selection because they are undefined when any component of λ is 0.

Below are derivations of MEMSEL objective prediction functions $\hat{\mu}(\cdot)$.

$$\begin{aligned}\hat{\mu}_{\mathbf{X}}(\mathbf{t}) &= \mathbf{t}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \\ &= \mathbf{t}^T \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\end{aligned}$$

$$\begin{aligned}\hat{\mu}_{\mathbf{W}}(\mathbf{t}) &= \mathbf{t}^T B^{-1} \sum_{b=1}^B (\mathbf{W}_b^T \mathbf{W}_b)^{-1} \mathbf{W}_b^T \mathbf{Y} \\ &= \mathbf{t}^T B^{-1} \sum_{b=1}^B \left\{ (\mathbf{X} + \mathbf{Z}_b \mathcal{D}_{\{\sigma_u\}})^T (\mathbf{X} + \mathbf{Z}_b \mathcal{D}_{\{\sigma_u\}}) \right\}^{-1} (\mathbf{X} + \mathbf{Z}_b \mathcal{D}_{\{\sigma_u\}})^T \mathbf{Y} \\ &= \mathbf{t}^T B^{-1} \sum_{b=1}^B \left(\mathbf{X}^T \mathbf{X} + 2\mathbf{X}^T \mathbf{Z}_b \mathcal{D}_{\{\sigma_u\}} + \mathcal{D}_{\{\sigma_u\}} \mathbf{Z}_b^T \mathbf{Z}_b \mathcal{D}_{\{\sigma_u\}} \right)^{-1} (\mathbf{X}^T \mathbf{Y} + \mathcal{D}_{\{\sigma_u\}} \mathbf{Z}_b^T \mathbf{Y}) \\ &= \mathbf{t}^T \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \quad (\text{for any choice of } B \geq 1)\end{aligned}$$

$$\begin{aligned}\hat{\mu}_{\mathbf{M}}(\mathbf{t}) &= \mathbf{t}^T (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{Y} \\ &= \mathbf{t}^T \left\{ \hat{\mathbf{V}}_{\mathbf{X}} \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \mathbf{X}^T \mathbf{X} \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \hat{\mathbf{V}}_{\mathbf{X}} \right\}^{-1} \\ &\quad \times \hat{\mathbf{V}}_{\mathbf{X}} \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \mathbf{X}^T \mathbf{Y} \\ &= \mathbf{t}^T \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \left(\hat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\sigma_u\}}^2 \right) \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\end{aligned}$$

$$\begin{aligned}\hat{\mu}_{\mathbf{D}}(\mathbf{t}) &= \mathbf{t}^T (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{Y} \\ &= \mathbf{t}^T \left(\mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \mathbf{X}^T \mathbf{X} \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \right)^{-1} \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \mathbf{X}^T \mathbf{Y} \\ &= \mathbf{t}^T \mathcal{D}_{\{1+\sigma_u^2\}} \hat{\mathbf{V}}_{\mathbf{X}}^{-1} \hat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}\end{aligned}$$

Below are derivations of MEMSEL objective functions for (\mathbf{T}, \mathbf{S}) pairs of interest. The MEMSEL loss \mathcal{L} is squared error loss to match with least squares in linear regression.

$$\begin{aligned}
Q_{\mathbf{X},\mathbf{X}}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right\}^2 \\
&= \widehat{\mathbf{V}}_{\mathbf{Y}} - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \quad (\text{constant in } \boldsymbol{\lambda})
\end{aligned}$$

$$\begin{aligned}
Q_{\mathbf{X},\mathbf{W}}(\boldsymbol{\lambda}) &= \frac{1}{nR} \sum_{i=1}^n \sum_{r=1}^R \left\{ Y_i - (\mathbf{X}_i + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{Z}_{r,i}^*)^T \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right\}^2 \\
&= \widehat{\mathbf{V}}_{\mathbf{Y}} - 2\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right) \widehat{\mathbf{V}}_{\mathbf{X}}^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}
\end{aligned}$$

$$\begin{aligned}
Q_{\mathbf{W},\mathbf{X}}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right\}^2 \\
&= \widehat{\mathbf{V}}_{\mathbf{Y}} - 2\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\
&\quad + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}
\end{aligned}$$

$$\begin{aligned}
Q_{\mathbf{W},\mathbf{W}}(\boldsymbol{\lambda}) &= \frac{1}{nR} \sum_{i=1}^n \sum_{r=1}^R \left\{ Y_i - (\mathbf{X}_i + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{Z}_{r,i})^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right\}^2 \\
&= \frac{1}{R} \sum_{r=1}^R \left\{ \widehat{\mathbf{V}}_{\mathbf{Y}} - 2\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right. \\
&\quad \left. + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + 2n^{-1} \mathbf{X}^T \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{Z}_r + n^{-1} \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \mathbf{Z}_r^T \mathbf{Z}_r \mathcal{D}_{\{\boldsymbol{\sigma}_u\}} \right) \right. \\
&\quad \left. \times \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \right\} \\
&= \widehat{\mathbf{V}}_{\mathbf{Y}} - 2\widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\
&\quad + \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right) \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}} \\
&= \widehat{\mathbf{V}}_{\mathbf{Y}} - \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}^T \left(\widehat{\mathbf{V}}_{\mathbf{X}} + \mathcal{D}_{\{\boldsymbol{\sigma}_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{\mathbf{X}\mathbf{Y}}
\end{aligned}$$

$$\begin{aligned}
Q_{M,X}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \widehat{\mathbf{V}}_X^{-1} \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right) \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= \widehat{\mathbf{V}}_Y - 2 \widehat{\mathbf{V}}_{XY}^T \widehat{\mathbf{V}}_X^{-1} \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right) \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \\
&\quad + \widehat{\mathbf{V}}_{XY}^T \widehat{\mathbf{V}}_X^{-1} \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right) \widehat{\mathbf{V}}_X^{-1} \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right) \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY}
\end{aligned}$$

$$\begin{aligned}
Q_{X,M}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_X \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= \widehat{\mathbf{V}}_Y - 2 \widehat{\mathbf{V}}_{XY}^T \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{XY} \\
&\quad + \widehat{\mathbf{V}}_{XY}^T \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_X \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_{XY} \\
&= Q_{W,X}(\boldsymbol{\lambda})
\end{aligned}$$

$$\begin{aligned}
Q_{M,M}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right)^{-1} \widehat{\mathbf{V}}_X \widehat{\mathbf{V}}_X^{-1} \left(\widehat{\mathbf{V}}_X + \mathcal{D}_{\{\sigma_u\}}^2 \right) \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= Q_{X,X}(\boldsymbol{\lambda}) \quad (\text{constant in } \boldsymbol{\lambda})
\end{aligned}$$

$$\begin{aligned}
Q_{D,X}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \mathcal{D}_{\{1+\sigma_u^2\}} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= \widehat{\mathbf{V}}_Y - 2 \widehat{\mathbf{V}}_{XY}^T \mathcal{D}_{\{1+\sigma_u^2\}} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \\
&\quad + \widehat{\mathbf{V}}_{XY}^T \widehat{\mathbf{V}}_X^{-1} \mathcal{D}_{\{1+\sigma_u^2\}} \widehat{\mathbf{V}}_X^{-1} \mathcal{D}_{\{1+\sigma_u^2\}} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY}
\end{aligned}$$

$$\begin{aligned}
Q_{X,D}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= \widehat{\mathbf{V}}_Y - 2 \widehat{\mathbf{V}}_{XY}^T \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \\
&\quad + \widehat{\mathbf{V}}_{XY}^T \widehat{\mathbf{V}}_X^{-1} \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \widehat{\mathbf{V}}_X \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY}
\end{aligned}$$

$$\begin{aligned}
Q_{D,D}(\boldsymbol{\lambda}) &= \frac{1}{n} \sum_{i=1}^n \left\{ Y_i - \mathbf{X}_i^T \mathcal{D}_{\{1+\sigma_u^2\}}^{-1} \mathcal{D}_{\{1+\sigma_u^2\}} \widehat{\mathbf{V}}_X^{-1} \widehat{\mathbf{V}}_{XY} \right\}^2 \\
&= Q_{X,X}(\boldsymbol{\lambda}) \quad (\text{constant in } \boldsymbol{\lambda})
\end{aligned}$$

BIBLIOGRAPHY

- [1] Akaike, H. “Information theory and an extension of the maximum likelihood principle.” *Selected Papers of Hirotugu Akaike*. Springer, 1998, pp. 199–213.
- [2] Allen, G. I. “Automatic Feature Selection via Weighted Kernels and Regularization.” *Journal of Computational and Graphical Statistics* **22.2** (2013), pp. 284–299.
- [3] Anderson, T. W. *An Introduction to Multivariate Statistical Analysis*. 3rd. New Jersey: Wiley-Interscience, 2003.
- [4] Barron, A. R. & Xiao, X. “Discussion: Multivariate Adaptive Regression Splines.” *The Annals of Statistics* **19.1** (1991), pp. 67–82.
- [5] Bazaraa, M. S. et al. *Nonlinear Programming: Theory and Algorithms*. New Jersey: Wiley-Interscience, 2006.
- [6] Bickel, P. J. & Levina, E. “Some theory for Fisher’s linear discriminant function, ‘naive Bayes’, and some alternatives when there are many more variables than observations.” *Bernoulli* **10** (2004), pp. 989–1010.
- [7] Breiman, L. “Random forests.” *Machine learning* **45.1** (2001), pp. 5–32.
- [8] Buonaccorsi, J. P. *Measurement error: models, methods, and applications*. CRC Press, 2010.
- [9] Burnham, K. P. & Anderson, D. R. *Model selection and multimodel inference: a practical information-theoretic approach*. 2003.
- [10] Cai, T. & Liu, W. “A Direct Estimation Approach to Sparse Linear Discriminant Analysis.” *Journal of the American Statistical Association* **106** (2011), pp. 1566–1577.
- [11] Carroll, R. J. et al. *Measurement error in nonlinear models: a modern perspective*. CRC press, 2006.
- [12] Cooley, C. A. & MacEachern, S. N. “Classification via Kernel Product Estimators.” *Biometrika* **85.4** (1998), pp. 823–833.
- [13] Cristianini, N. & Shawe-Taylor, J. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.
- [14] Duda, R. et al. *Pattern Classification*. New York: Wiley, 2000.
- [15] Fan, J. & Fan, Y. “High dimensional classification using features annealed independence rules.” *The Annals of Statistics* **36** (2008), pp. 2605–2637.

- [16] Fan, J. & Li, R. “Variable selection via nonconcave penalized likelihood and its oracle properties.” *Journal of American Statistical Association* **96** (2001), pp. 1348–1360.
- [17] Fan, J. & Gijbels, I. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Vol. 66. CRC Press, 1996.
- [18] Fan, J. & Jiang, J. “Nonparametric inferences for additive models.” *Journal of the American Statistical Association* **100**.471 (2005), pp. 890–907.
- [19] Fisher, R. A. “The use of multiple measurements in taxonomic problems.” *Annals Eugen.* **7** (1936), pp. 179–188.
- [20] Freund, Y. & Schapire, R. “A decision theoretic generalization of on-line learning and an application to boosting.” *Journal of Computer and System Sciences* **55** (1997), pp. 119–139.
- [21] Friedman, J. et al. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin, 2001.
- [22] Friedman, J. et al. “Regularization paths for generalized linear models via coordinate descent.” *Journal of statistical software* **33**.1 (2010), p. 1.
- [23] Friedman, J. H. “Multivariate Adaptive Regression Splines.” *The Annals of Statistics* **19**.1 (1991), pp. 1–67.
- [24] Fuller, W. “Measurement error models. 1987.” *John Willey* ().
- [25] Genuer, R. et al. “VSURF: An R Package for Variable Selection Using Random Forests.” *R Journal* **7**.2 (2015).
- [26] Genuer, R. et al. *VSURF: Variable Selection Using Random Forests*. R package version 1.0.3. 2016.
- [27] Ghosh, A. K. et al. “Classification Using Kernel Density Estimates.” *Technometrics* **48**.1 (2006), pp. 120–132.
- [28] Golub, G. H. et al. “Generalized cross-validation as a method for choosing a good ridge parameter.” *Technometrics* **21**.2 (1979), pp. 215–223.
- [29] Goutte, C. & Larsen, J. “Adaptive Metric Kernel Regression.” *Journal of VLSI Signal Processing* **26** (2000), pp. 155–167.

- [30] Grandvalet, Y. & Canu, S. “Comments on ”Noise injection into inputs in back propagation learning”.” *Systems, Man and Cybernetics, IEEE Transactions on* **25.4** (1995), pp. 678–681.
- [31] Grandvalet, Y. & Canu, S. “Adaptive Noise Injection for Input Variables Relevance Determination.” *Artificial Neural Networks - ICANN '97, 7th International Conference, Lausanne, Switzerland, October 8-10, 1997, Proceedings*. 1997, pp. 463–468.
- [32] Grandvalet, Y. et al. “Noise Injection: Theoretical Prospects.” *Neural Computation* **9.5** (1997), pp. 1093–1108.
- [33] Hall, P. et al. “Median-Based Classifiers for High-Dimensional Data.” *Journal of the American Statistical Association* **104** (2009), pp. 1597–1608.
- [34] Hansen, B. E. “Uniform Convergence Rates For Kernel Estimation With Dependent Data.” *Econometric Theory* **24** (2008), pp. 726–748.
- [35] Hapfelmeier, A. & Ulm, K. “A new variable selection approach using random forests.” *Computational Statistics & Data Analysis* **60** (2013), pp. 50–69.
- [36] Hastie, T. et al. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag: New York, 2001, p. 552.
- [37] Hastie, T. et al. *The Elements of Statistical Learning*. 2nd. Springer, 2009.
- [38] Hoerl, A. E. & Kennard, R. W. “Ridge Regression: Biased Estimation for Nonorthogonal Problems.” *Technometrics* **12** (1970), pp. 55–67.
- [39] Holmstrom, L. & Koistinen, P. “Using additive noise in back-propagation training.” *Neural Networks, IEEE Transactions on* **3.1** (1992), pp. 24–38.
- [40] Hothorn, T. et al. “Survival ensembles.” *Biostatistics* **7.3** (2005), pp. 355–373.
- [41] Hothorn, T. et al. *ctree: Conditional Inference Trees*. R Foundation for Statistical Computing. Vienna, Austria, 2016.
- [42] Hurvich, C. M. & Tsai, C.-L. “Bias of the corrected AIC criterion for underfitted regression and time series models.” *Biometrika* **78.3** (1991), pp. 499–509.
- [43] Hurvich, C. M. et al. “Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion.” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **60.2** (1998), pp. 271–293.

- [44] J. R. Cook, L. A. S. “Simulation-Extrapolation Estimation in Parametric Measurement Error Models.” *Journal of the American Statistical Association* **89**.428 (1994), pp. 1314–1328.
- [45] John, G. H. et al. “Irrelevant Features and the Subset Selection Problem.” *MACHINE LEARNING: PROCEEDINGS OF THE ELEVENTH INTERNATIONAL*. Morgan Kaufmann, 1994, pp. 121–129.
- [46] Lafferty, J. & Wasserman, L. “RODEO: Sparse, greedy nonparametric regression.” *The Annals of Statistics* **36**.1 (2008), pp. 28–63.
- [47] Li, Q. & Racine, J. *Nonparametric Econometrics: Theory and Practice*. Princeton University Press, 2007.
- [48] Liaw, A. & Wiener, M. “Classification and Regression by randomForest.” *R News* **2**.3 (2002), pp. 18–22.
- [49] Lichman, M. *UCI Machine Learning Repository*. 2013.
- [50] Lin, Y. & Zhang, H. H. “Component Selection and Smoothing in Multivariate Nonparametric Regression.” *Annals of Statistics* **34**.5 (2006), pp. 2272–2297.
- [51] Mai, Q. et al. “A Direct Approach to Sparse Discriminant Analysis in Ultra-high Dimensions.” *Biometrika* (2012). to appear.
- [52] McCullagh, P. & Nelder, J. *Generalized Linear Models*. Chapman & Hall/CRC., 1989.
- [53] Nadaraya, E. A. “On Estimating Regression.” *Theory of Probability & Its Applications* **9**.1 (1964), pp. 141–142.
- [54] Neal, R. M. *Bayesian learning for neural networks*. Springer-Verlag, 1996.
- [55] Okazaki, N. *libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)*. <http://www.chokkan.org/software/liblbfgs/>, 2010.
- [56] Prentice, R. L. “Binary regression using an extended beta-binomial distribution, with discussion of correlation induced by covariate measurement errors.” *Journal of the American Statistical Association* **81**.394 (1986), pp. 321–327.
- [57] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2013.
- [58] Racine, J. & Li, Q. “Nonparametric estimation of regression functions with both categorical and continuous data.” *Journal of Econometrics* **119**.4 (2004), pp. 99–130.

- [59] Ravikumar, P. et al. “Sparse additive models.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71.5** (2009), pp. 1009–1030.
- [60] Schwarz, G. “Estimating the dimension of a model.” *The Annals of Statistics* **6.2** (1978), pp. 461–464.
- [61] Scott, D. W. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 1992.
- [62] Seo, B. & Lindsay, B. G. “A universally consistent modification of maximum likelihood (preprint).” *Statistica Sinica* (2011).
- [63] Shao, J. et al. “Sparse Linear Discriminant Analysis With High Dimensional Data.” *The Annals of Statistics* **39** (2011), pp. 1241–1265.
- [64] Shao, J. “An asymptotic theory for linear model selection.” *Statistica Sinica* (1997), pp. 221–242.
- [65] Sietsma, J. & Dow, R. J. “Creating artificial neural networks that generalize.” *Neural Networks* **4.1** (1991), pp. 67–79.
- [66] Stamey, T. et al. “Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients.” *Journal of Urology* **141.5** (1989), pp. 1076–1083.
- [67] Stefanski, L. A. & Cook, J. R. “Simulation-Extrapolation: The Measurement Error Jackknife.” *Journal of the American Statistical Association* **90.432** (1995), pp. 1247–1256.
- [68] Stefanski, L. A. et al. “Variable Selection in Nonparametric Classification Via Measurement Error Model Selection Likelihoods.” *Journal of the American Statistical Association* **109.506** (2014), pp. 574–589.
- [69] Stone, M. “Cross-validatory choice and assessment of statistical predictions.” *Journal of the royal statistical society. Series B (Methodological)* (1974), pp. 111–147.
- [70] Storlie, C. B. et al. “Surface Estimation, Variable Selection, and the Nonparametric Oracle Property.” *Statistica Sinica* **21.2** (2011), pp. 679–705.
- [71] Strobl, C. et al. “Bias in random forest variable importance measures: Illustrations, sources and a solution.” *BMC bioinformatics* **8.1** (2007), p. 25.
- [72] Strobl, C. et al. “Conditional variable importance for random forests.” *BMC bioinformatics* **9.1** (2008), p. 307.

- [73] Thoma, E. et al. “Assessment of methane and voc emissions from select upstream oil and gas production operations using remote measurements, interim report on recent survey studies.” *Proceedings of 105th Annual Conference of the Air & Waste Management Association, Control*. 2012-A. 2012, pp. 298–312.
- [74] Tibshirani, R. “Regression shrinkage and selection via the lasso.” *Journal of the Royal Statistical Society, Series B* **58** (1996), pp. 267–288.
- [75] Tibshirani, R. “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society. Series B (Methodological)* **58.1** (1996), pp. 267–288.
- [76] Vapnik, V. *The Nature of Statistical Learning Theory*. New York: Springer, 1995.
- [77] Wand, M. P. & Jones, M. C. *Kernel Smoothing*. London: Chapman & Hall/CRC, 1995.
- [78] Watson, G. S. “Smooth Regression Analysis.” *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)* **26.4** (1964), pp. 359–372.
- [79] White, K. R. et al. “Variable Selection in Kernel Regression Using Measurement Error Selection Likelihoods.” *Journal of the American Statistical Association (to appear in print)* (2016). eprint: <http://dx.doi.org/10.1080/01621459.2016.1222287>.
- [80] Williams, C. K. & Rasmussen, C. E. “Gaussian processes for regression.” *MIT Press* (1996).
- [81] Witten, D. & Tibshirani, R. “Covariance-Regularized Regression and Classification for High Dimensional Problems.” *Journal of the Royal Statistical Society, Ser. B* **71** (2009), pp. 615–636.
- [82] Wu, Y. & Stefanski, L. A. “Automatic structure recovery for additive models.” *Biometrika* **102.2** (2015), pp. 381–395.
- [83] Xu, H. et al. “Robust regression and lasso.” *Advances in Neural Information Processing Systems*. 2009, pp. 1801–1808.
- [84] Yang, Y. “Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation.” *Biometrika* **92.4** (2005), pp. 937–950.
- [85] Yeh, I.-C. “Modeling slump flow of concrete using second-order regressions and artificial neural networks.” *Cement and Concrete Composites* **29.6** (2007), pp. 474–480.
- [86] Zhang, H. H. “Variable selection for support vector machines via smoothing spline ANOVA.” *Statistica Sinica* **16** (2006), pp. 659–674.

- [87] Zou, H. & Hastie, T. “Regularization and variable selection via the elastic net.” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67.2** (2005), pp. 301–320.