

# Nonparametric Learning in High Dimensions

Han Liu

December 2010  
CMU-ML-10-112

Machine Learning Department and Department of Statistics  
School of Computer Science  
Carnegie Mellon University

Thesis Committee  
John Lafferty (Co-chair)  
Larry Wasserman (Co-chair)  
Christopher Genovese (Statistics)  
Zoubin Ghahramani (Cambridge University)  
Bin Yu (University of California, Berkeley)

Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy

December 2010

Copyright © December 2010 Han Liu

This research was sponsored by the National Science Foundation under grant numbers CCF-0625879, IIS-0427206, IIS-0312814 and a fellowship from Google. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

**Keywords:** machine learning, statistical inference, nonparametric methods, curse of dimensionality, regression, classification, multi-task learning, density estimation, undirected graphical models, structure learning, spatial-temporal adaptive learning

To my parents Zenghai Liu, Zhen Han, and my wife Xiaolin Yang.

## ABSTRACT

---

This thesis develops flexible and principled nonparametric learning algorithms to explore, understand, and predict high dimensional and complex datasets. Such data appear frequently in modern scientific domains and lead to numerous important applications. For example, exploring high dimensional functional magnetic resonance imaging data helps us to better understand brain functionalities; inferring large-scale gene regulatory network is crucial for new drug design and development; detecting anomalies in high dimensional transaction databases is vital for corporate and government security.

Our main results include a rigorous theoretical framework and efficient nonparametric learning algorithms that exploit hidden structures to overcome the curse of dimensionality when analyzing massive high dimensional datasets. These algorithms have strong theoretical guarantees and provide high dimensional nonparametric recipes for many important learning tasks, ranging from unsupervised exploratory data analysis to supervised predictive modeling. In this thesis, we address three aspects:

- 1 Understanding the statistical theories of high dimensional nonparametric inference, including risk, estimation, and model selection consistency;
- 2 Designing new methods for different data-analysis tasks, including regression, classification, density estimation, graphical model learning, multi-task learning, spatial-temporal adaptive learning;
- 3 Demonstrating the usefulness of these methods in scientific applications, including functional genomics, cognitive neuroscience, and meteorology.

In the last part of this thesis, we also present the future vision of high dimensional and large-scale nonparametric inference.

## KEYWORDS

---

machine learning, statistical inference, nonparametric methods, curse of dimensionality, regression, classification, multi-task learning, density estimation, undirected graphical models, structure learning, spatial-temporal adaptive learning

*We have seen that computer programming is an art,  
because it applies accumulated knowledge to the world,  
because it requires skill and ingenuity, and especially  
because it produces objects of beauty.*

— Donald E. Knuth [Knuth, 1974]

## ACKNOWLEDGMENTS

---

It is impossible to express my gratitude to John Lafferty and Larry Wasserman, my thesis advisors, in languages. To them, I owe too much. Without their patience and insight, it is impossible to get this thesis done. In the past five years, they guided me through my PhD studies like father figures. Not only did they coach me on how to do research, they also let me know how to be a good person. They are extremely nice and sincere to every student, colleague and collaborator. Such an altitude deeply influenced me and will continue to shape me in the future. From them, I have learnt many things that not only help define me, but will make my future years enormously successful — emotionally, professionally, and socially. They are the best advisors I could ever dreamed to have. Thanks again, John and Larry!

This thesis was also shaped by the insightful suggestions of my other thesis committee members: Christopher Genovese from Statistics, Zoubin Ghahramani from Cambridge University, and Bin Yu from University of California, Berkeley. All of them are extremely busy, but they are always trying to find time to talk to me about my research. I am indeed very thankful to them. I also thank Google for generously providing the PhD fellowship to support the research in this thesis.

I deeply appreciate Carlos Guestrin, Tom Mitchell, and Kathryn Roeder for their enormous support and help on my job search. I am very grateful to many other faculty members at Carnegie Mellon: Jaime Carbonell, Anupam Dasgupta, Bill Eddy, Christos Faloutsos, Steve Fienberg, Peter Freeman, Geoff Gordon, Joel Greenhouse, Steve Hanneke, Jiashun Jin, Brian Junker, Jay Kadane, Rob Kass, Ann Lee, Robert Murphy, Rebecca Nugent, Alessandro Rinaldo, Ronald Rosenfeld, Chad Schafer, Mark Schervish, Teddy Seidenfeld, Cosma Shalizi, Aarti Singh, Andrew Thomas, Valerie Ventura, Isabella Verdinelli, Eric Xing, Yiming Yang. In the past years, I have very joyable discussions with these great researchers. They have given valuable suggestions related to my research and pointed me to relevant literature. I would also like to thank Peter Bailey, Ryen White at Microsoft Research and Yoram Singer at Google Research; interning with them is very enjoyable and exciting. I also owe special thanks to Diane Stidle, who has done a perfect administration job and always help me to solve problems in time.

During my stay at Carnegie Mellon, I have made some great friends and life would not have been the same without them. I have had the pleasure to know Amr Ahmed, Edoardo Airoidi, Andrew Arnold, Sivaraman Balakrishnan, Gaia Bellone, Joseph Bradley, Susan Buchman, Andy Carlson, Lucia Castellanos, Hao Cen, Ning Chen, Xi Chen, Anne-Sophie Charest, Nanjun Chu, Khalid El-Arini, Jun Gao, Georg Goerg, Joseph Gonzalez, Justin Gross, Haijie Gu, Fan Guo, Robert Hall, Jingrui He, Darren Homrighausen, Jon Huang, Tzu-Kuo Huang, Yi Jiang, Seyoung Kim, Mladen Kolar, Andreas Krause, Xiaoye Jiang, Jurij Leskovec, Fan Li, Lei Li, Frank Lin, Di Liu, Jun Liu, Li Liu, Zhanwu Liu, Yan Liu, Yucheng Low, Daniel Manrique, Shuhei Okumura, Mark Palatucci, Ankur Parikh, Daniel Percival, Kriti Punyani, Pradeep Ravikumar, Indrayana Rustandi, Avranil Sarkar, Purnamrita Sarkar, James Sharpnack, Xuehua Shen, Runting Shi, Suyash Shringarpure, Ajit Singh, Kyung-Ah Sohn, Le Song, Nan Song, Shijun Song, Sonia Todorova, Hanghang Tong, Charalampos Tsourakakis, Wanjie Wang, Yi Wu, Liang Xiong, Min Xu, Yang Xu, Rong Yan, Hui Yang, Liu Yang, Xiting Yang, Shinjae Yoo, Lubov Zeifman, Ji Zhang, Jian Zhang, Xin Zhang, Yi Zhang, Linqiao Zhao, Shuheng Zhou, Jerry Zhu, Jun Zhu. They have looked out for me in difficult times; brainstormed about new exciting ideas; read my papers at odd hours before pressing deadlines; and last, but not least, helped me during job search.

Finally, I would like to thank my parents Zenghai Liu and Zhen Han, and my wife Xiaolin Yang for their love and endless support in all these years.

## CONTENTS

---

|            |                                                    |           |
|------------|----------------------------------------------------|-----------|
| <b>I</b>   | <b>THESIS OVERVIEW</b>                             | <b>1</b>  |
| 1          | THESIS OVERVIEW                                    | 3         |
| 1.1        | Motivation and Thesis Statement                    | 3         |
| 1.2        | Main Results                                       | 3         |
| 1.3        | Organization of This Thesis                        | 5         |
| 1.3.1      | Part II: Fundamental Theory                        | 5         |
| 1.3.2      | Part III: Unsupervised Learning Methods            | 7         |
| 1.3.3      | Part IV: Supervised Learning Methods               | 11        |
| 1.3.4      | Part V: Regularization Selection                   | 13        |
| 1.4        | Related Publications                               | 15        |
| <b>II</b>  | <b>STATISTICAL THEORY</b>                          | <b>17</b> |
| 2          | BACKGROUND AND STATISTICAL THEORY                  | 19        |
| 2.1        | Literature Overview                                | 19        |
| 2.1.1      | Curse of Dimensionality                            | 19        |
| 2.1.2      | Sparsity in High-dimensional Parametric Methods    | 20        |
| 2.1.3      | Sparsity in High-dimensional Nonparametric Methods | 21        |
| 2.2        | Theoretical Framework of This Thesis               | 22        |
| <b>III</b> | <b>UNSUPERVISED LEARNING</b>                       | <b>27</b> |
| 3          | DENSITY RODEO: NONPARAMETRIC DENSITY ESTIMATION    | 29        |
| 3.1        | Introduction and Motivation                        | 29        |
| 3.2        | The Local Rodeo                                    | 31        |
| 3.2.1      | The Kernel Density Estimator Version               | 31        |
| 3.2.2      | The Local Likelihood Version                       | 33        |
| 3.2.3      | Other Baseline Densities                           | 34        |
| 3.3        | The Global Rodeo                                   | 35        |
| 3.4        | Different Extensions                               | 35        |
| 3.4.1      | Bootstrap Version                                  | 35        |
| 3.4.2      | Reverse Rodeo                                      | 36        |
| 3.5        | Examples                                           | 37        |
| 3.5.1      | One-dimensional Examples                           | 37        |
| 3.5.2      | Two dimensional Examples                           | 39        |
| 3.5.3      | High Dimensional Examples                          | 43        |
| 3.5.4      | Using Other Baseline Densities                     | 44        |
| 3.6        | Theoretical Properties                             | 46        |
| 3.7        | Conclusions                                        | 51        |
| 3.8        | APPENDIX: TECHNICAL PROOFS                         | 51        |

|       |                                                            |     |
|-------|------------------------------------------------------------|-----|
| 3.8.1 | Derivation of the Semiparametric Rodeo Estimator           | 51  |
| 3.8.2 | Proofs of the Main Results                                 | 52  |
| 4     | NONPARANORMAL: LEARNING NONPARAMETRIC UNDIRECTED GRAPHS    | 61  |
| 4.1   | Introduction and Motivation                                | 61  |
| 4.2   | Estimating Undirected Graphs                               | 62  |
| 4.3   | The Nonparanormal                                          | 64  |
| 4.4   | Estimation Method                                          | 67  |
| 4.5   | Theoretical Properties                                     | 69  |
| 4.6   | Experimental Results                                       | 72  |
| 4.6.1 | Neighborhood Graphs                                        | 72  |
| 4.6.2 | Gene Microarray Data                                       | 80  |
| 4.7   | Concluding Remarks                                         | 81  |
| 4.8   | Appendix: Technical Proofs                                 | 83  |
| 4.8.1 | Proof of Theorem 4.1                                       | 84  |
| 4.8.2 | Proof of Theorem 4.2                                       | 94  |
| 5     | FOREST DENSITY ESTIMATION                                  | 97  |
| 5.1   | Introduction and Motivation                                | 97  |
| 5.2   | Preliminaries and Notation                                 | 99  |
| 5.3   | Kernel Density Estimation For Forests                      | 102 |
| 5.3.1 | Step 1: Estimating the marginals                           | 103 |
| 5.3.2 | Step 2: Optimizing the forest                              | 104 |
| 5.3.3 | Building a forest on held-out data                         | 105 |
| 5.4   | Statistical Properties                                     | 106 |
| 5.4.1 | Assumptions on the density                                 | 106 |
| 5.4.2 | Assumptions on the kernel                                  | 107 |
| 5.4.3 | Risk consistency                                           | 108 |
| 5.4.4 | Structure selection consistency                            | 110 |
| 5.4.5 | Estimation consistency                                     | 112 |
| 5.5   | Experimental Results                                       | 113 |
| 5.5.1 | Synthetic data                                             | 113 |
| 5.5.2 | Microarray data                                            | 118 |
| 5.6   | Conclusion                                                 | 120 |
| 5.7   | Appendix: Technical Proofs                                 | 122 |
| 5.7.1 | Computation of the Mutual Information Matrix               | 130 |
| IV    | SUPERVISED LEARNING                                        | 133 |
| 6     | MT-SPAM: MULTI-TASK SPARSE ADDITIVE MODELS                 | 135 |
| 6.1   | Introduction and Motivation                                | 135 |
| 6.2   | Background Materials on Single-task Sparse Additive Models | 138 |
| 6.3   | Muti-task Sparse Additive Models                           | 150 |
| 6.3.1 | Multi-task/Multi-response Sparse Additive Regression       | 150 |
| 6.3.2 | Sparse Multi-Category Additive Logistic Regression         | 151 |
| 6.3.3 | Simultaneous Sparse Backfitting                            | 152 |



|       |                                                           |     |
|-------|-----------------------------------------------------------|-----|
| 6.3.4 | Penalized Local Scoring Algorithm for SMALR               | 154 |
| 6.4   | Theoretical Properties of the Multi-task SpAM             | 156 |
| 6.5   | New Insights on the Smooth Sparse Backfitting Algorithm   | 157 |
| 6.5.1 | Population Version of the Sparse Backfitting Algorithm    | 158 |
| 6.5.2 | Function Space and Semi-Norms                             | 159 |
| 6.5.3 | Smooth Sparse Backfitting Using Kernel Smoothers          | 161 |
| 6.5.4 | Existence and Uniqueness of the Solution                  | 163 |
| 6.6   | Experimental Results                                      | 165 |
| 6.6.1 | Synthetic Data                                            | 166 |
| 6.6.2 | Gene Microarray Data                                      | 166 |
| 6.7   | A Case Study of MT-SPAM: Neural Semantic Basis Discovery  | 168 |
| 6.7.1 | Datasets                                                  | 169 |
| 6.7.2 | Results                                                   | 171 |
| 6.8   | Conclusions                                               | 174 |
| 6.9   | Appendix: Technical Proofs                                | 175 |
| 7     | GREEDY NONPARAMETRIC REGRESSION                           | 177 |
| 7.1   | Introduction                                              | 177 |
| 7.2   | Sparse Nonparametric Learning in High Dimensions          | 179 |
| 7.3   | Additive Forward Regression                               | 179 |
| 7.4   | Generalized Forward Regression                            | 180 |
| 7.5   | Theoretical Properties                                    | 183 |
| 7.6   | Experimental Results                                      | 185 |
| 7.6.1 | The Synthetic Data                                        | 185 |
| 7.6.2 | The real data                                             | 188 |
| 7.7   | Conclusions and Discussions                               | 189 |
| 8     | MDRT: MULTIVARIATE DYADIC REGRESSION TREES                | 191 |
| 8.1   | Introduction                                              | 191 |
| 8.2   | Multivariate Dyadic Regression Trees                      | 194 |
| 8.3   | Statistical Properties                                    | 195 |
| 8.4   | Computational Algorithm                                   | 198 |
| 8.5   | Experimental Results                                      | 200 |
| 8.5.1 | Synthetic Data                                            | 200 |
| 8.5.2 | Real Data                                                 | 201 |
| 8.6   | Conclusions                                               | 203 |
| 8.7   | Appendix: Pseudo-code for Greedy Tree Learning Algorithms | 203 |
| 9     | GRAPH-VALUED REGRESSION                                   | 205 |
| 9.1   | Introduction                                              | 205 |
| 9.2   | Graph-Valued Regression                                   | 206 |
| 9.3   | Graph-Optimized CART                                      | 207 |
| 9.3.1 | Dyadic Partitioning Tree                                  | 208 |
| 9.3.2 | Go-CART: Risk Minimization Estimator                      | 208 |
| 9.3.3 | Go-CART: Greedy Partitioning                              | 210 |
| 9.4   | Theoretical Properties                                    | 211 |

|        |                                                           |     |
|--------|-----------------------------------------------------------|-----|
| 9.5    | Experimental Results                                      | 215 |
| 9.5.1  | Synthetic Data                                            | 215 |
| 9.5.2  | Chain Structure                                           | 218 |
| 9.5.3  | Two-way Grid Structure                                    | 220 |
| 9.5.4  | Climate Data Analysis                                     | 220 |
| 9.6    | Conclusions                                               | 222 |
| 9.7    | Appendix: Technical Proofs                                | 223 |
| V      | REGULARIZATION PARAMETER SELECTION                        | 231 |
| 10     | STARS: STABILITY APPROACH FOR REGULARIZATION SELECTION    | 233 |
| 10.1   | Introduction                                              | 233 |
| 10.2   | Estimating a High-dimensional Undirected Graph            | 235 |
| 10.3   | Regularization Selection                                  | 236 |
| 10.3.1 | Existing Methods                                          | 236 |
| 10.3.2 | StARS: Stability Approach to Regularization Selection     | 237 |
| 10.4   | Theoretical Properties                                    | 238 |
| 10.5   | Experimental Results                                      | 241 |
| 10.5.1 | Synthetic Data                                            | 241 |
| 10.5.2 | Microarray Data                                           | 243 |
| 10.6   | Conclusions                                               | 245 |
| VI     | CONCLUSION                                                | 247 |
| 11     | CONCLUSION AND FUTURE DIRECTIONS                          | 249 |
| 11.1   | Summary and Discussions                                   | 249 |
| 11.1.1 | Building Computational Models to Predict Brain Activities | 249 |
| 11.1.2 | Inferring Gene Regulatory Networks                        | 249 |
| 11.1.3 | Tumor Classification using Microarray Data                | 250 |
| 11.1.4 | Climate Data Analysis                                     | 250 |
| 11.2   | Future Directions                                         | 250 |
| 11.2.1 | Theory                                                    | 250 |
| 11.2.2 | Methods                                                   | 251 |
| 11.2.3 | Applications                                              | 251 |
| VII    | APPENDIX                                                  | 253 |
| A      | MORE TECHNICAL DETAILS OF THE COSSO                       | 255 |
|        | BIBLIOGRAPHY                                              | 257 |

## LIST OF FIGURES

---

- Figure 1 A paradigm illustrating the thesis structure. The applications of this thesis come from modern scientific areas including Genomics, Cognitive Neuroscience, and Meteorology. Under a unified theoretical framework (including risk consistency, estimation consistency, and model selection consistency), we develop high-dimensional nonparametric learning methods for both unsupervised and supervised learning paradigms; To automatically select the regularization parameters of these methods, we also develop a regularization selection method named StARS. 6
- Figure 2 An illustrative example of the density rodeo: Perspective plots of the estimated density functions by the density rodeo (left) and the R built-in method KDE2d (right) on a 2-dimensional synthetic data. 8
- Figure 3 An illustrate example of the nonparanormal: The densities of three 2-dimensional nonparanormals. The component functions have the form  $f_j(x) = \text{sign}(x)|x|^{\alpha_j}$ . Left:  $\alpha_1 = 0.9$ ,  $\alpha_2 = 0.8$ ; center:  $\alpha_1 = 1.2$ ,  $\alpha_2 = 0.8$ ; right  $\alpha_1 = 2$ ,  $\alpha_2 = 3$ . In each case  $\mu = (0, 0)$  and  $\Sigma = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ . 9
- Figure 4 An illustrative example of the forest density estimator: a 934 gene subgraph of the full estimated 4238 gene network. Left: estimated forest graph. Right: estimated Gaussian graph. Red edges in the forest graph are missing from the Gaussian graph and vice versa; the blue edges are shared by both graphs. Note that the layout of the genes is the same for both graphs. 10
- Figure 5 SMALR results on gene data: heat map (left), marginal fits (center), and CV score (right). 12
- Figure 6 An illustrative example of Go-CART: Analysis of the climate data. (a) Estimated partitions for 125 locations projected to the US map, with the estimated graphs for subregions 2, 3, and 65; (b) estimated graph with data pooled from all 125 locations; (c) the re-scaled partition pattern induced by the dyadic tree structure. 13
- Figure 7 An illustrative example of StARS: Two estimated graphs on the microarray data using the StARS and BIC. The StARS graph is more informative than the BIC graph. 14

|           |                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Figure 8  | An paradigm of statistical learning under a function estimation view. the true function: $f^*$ ; the oracle: $f^o$ ; the empirical estimator: $\hat{f}$ . 23                                                                                                                                                                                                                      |
| Figure 9  | The density rodeo algorithm. 32                                                                                                                                                                                                                                                                                                                                                   |
| Figure 10 | Density Rodeo: the bootstrap method to calculate the $s_f^2$ 36                                                                                                                                                                                                                                                                                                                   |
| Figure 11 | Different versions of the density Rodeo algorithms run on the highly skewed unimodal example. The first three plots are results for the different estimators, the last one is the fitted bandwidths for the local rodeo. 38                                                                                                                                                       |
| Figure 12 | Density Rodeo Experiments on data from Highly skewed unimodal distribution: The boxplots of the empirical Heillinger's losses on test samples of estimated densities by the three methods based on 100 simulations. 39                                                                                                                                                            |
| Figure 13 | Density Rodeo: Perspective plots of the estimated density functions by the global density rodeo (left) and the R built-in method KDE2d (right) on a 2-dimensional synthetic data. 40                                                                                                                                                                                              |
| Figure 14 | Marginal distributions of the relevant and the irrelevant dimensions for example 2 41                                                                                                                                                                                                                                                                                             |
| Figure 15 | Density rodeo experiments on the geyser data. Upper: Perspective plots of the estimated density functions by the global rodeo (left) and the R built-in method KDE2d (right) on the geyser data. Lower: Contour plots of the result from the global rodeo (left) and KDE2d (right) 42                                                                                             |
| Figure 16 | The bandwidth output by the local density rodeo for a 30-dimensional synthetic dataset (Left) and its boxplot for 30 trials. (Right) 44                                                                                                                                                                                                                                           |
| Figure 17 | ensity rodeo on the image data: the upper plots are the evaluation digit and the bandwidths output by the reverse rodeo. The lower subplots illustrate a series of bandwidth plots sampled at different rodeo steps: 10, 20, 40, 60, and 100 45                                                                                                                                   |
| Figure 18 | The bandwidth output by the local semiparametric rodeo for a 15-dimensional synthetic dataset (Left) and a 20-dimensional synthetic dataset (Right). Using Gaussian distribution as the irrelevant dimensions 46                                                                                                                                                                  |
| Figure 19 | The density rodeo results for fitting the uniform distribution with the semiparametric rodeo. The first plot shows the true density , the second plot is the estimated density, the lower left plot illustrates the estimated bandwidths at different evaluation points, the last one is the estimated density function by the KDE with bandwidth selected by cross validation 47 |

- Figure 20 Comparison of regression and graphical models. The nonparanormal extends additive models to the graphical model setting. Regularizing the inverse covariance leads to an extension to high dimensions, which parallels sparse additive models for regression. 63
- Figure 21 Densities of three 2-dimensional nonparanormals. The component functions have the form  $f_j(x) = \text{sign}(x)|x|^{\alpha_j}$ . Left:  $\alpha_1 = 0.9, \alpha_2 = 0.8$ ; center:  $\alpha_1 = 1.2, \alpha_2 = 0.8$ ; right  $\alpha_1 = 2, \alpha_2 = 3$ . In each case  $\mu = (0, 0)$  and  $\Sigma = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ . 66
- Figure 22 The power and cdf transformations. The densities are estimated using a kernel density estimator with bandwidths selected by cross-validation. 74
- Figure 23 Regularization paths for the glasso and nonparanormal with  $n = 500$  (top) and  $n = 200$  (bottom). The paths for the relevant variables (nonzero inverse covariance entries) are plotted as solid (black) lines; the paths for the irrelevant variables are plotted as dashed (red) lines. For non-Gaussian distributions, the nonparanormal better separates the relevant and irrelevant dimensions. 75
- Figure 24 Estimated transformations for the first three variables. Winsorization plays a significant role for the power transformation due to its high skewness. 76
- Figure 25 Boxplots of the oracle scores for  $n = 1000, 500, 200$  (top, center, bottom). 77
- Figure 26 ROC curves for sample sizes  $n = 1000, 500, 200$  (top, middle, bottom). 78
- Figure 27 Typical runs for the two methods for  $n = 1000$  using the cdf and power transformations. The dashed (black) lines in the symmetric difference plots indicate edges found by the glasso but not the nonparanormal, and vice-versa for the solid (red) lines. 79
- Figure 28 For Gaussian models, comparison of boxplots of the oracle scores and ROC curves for small  $n$  and relatively large  $d$ . The ROC curves suggest some efficiency loss of the nonparanormal; however, the corresponding boxplots indicate this loss is insignificant. 82
- Figure 29 For the cdf transformation with  $n = 200, d = 500, s = 1/40$ , comparison of the boxplots and a typical run of the regularization paths. The nonparanormal paths separate the relevant from the irrelevant dimensions well. For the glasso, the relevant variables are “buried” in irrelevant variables. 83

- Figure 30 The regularization paths of both methods on the microarray data set. Although the paths for the two methods look similar, there are some subtle differences. 83
- Figure 31 The nonparanormal estimated graph for three values of  $\lambda = 0.2448, 0.2661, 0.30857$  (left column), the closest glasso estimated graph from the full path (middle) and the symmetric difference graph (right). 84
- Figure 32 Estimated transformations for the microarray data set, indicating non-Gaussian marginals. The corresponding genes are among the nodes appearing in the symmetric difference graphs above. 84
- Figure 33 (Gaussian example) Boxplots of  $\hat{I}_{\text{fast}}$ ,  $\hat{I}_{\text{medium}}$ , and  $\hat{I}_{\text{slow}}$  on three different pairs of variables. The red-dashed horizontal lines represent the population values. 116
- Figure 34 Perspective and contour plots of the bivariate Gaussian fits vs. the kernel density estimates for two edges of a Gaussian graphical model. 116
- Figure 35 Synthetic data. Top-left Gaussian, and top-right non-Gaussian: Held-out log-likelihood plots of the forest density estimator (black step function), glasso (red stars), and refit glasso (blue circles), the vertical dashed red line indicates the size of the true graph. Bottom plots show the true and estimated graphs for the Gaussian (second row) and non-Gaussian data (third row). 117
- Figure 36 Results on microarray data. Top: held-out log-likelihood (left) and its zoom-in (right) of the tree-based kernel density estimator (black step function), glasso (red stars), and refit glasso (blue circles). Bottom: estimated graphs using the tree-based estimator (left) and glasso (right). 119
- Figure 37 A 934 gene subgraph of the full estimated 4238 gene network. Upper: estimated forest graph. Lower: estimated Gaussian graph. Red edges in the forest graph are missing from the Gaussian graph and vice versa; the blue edges are shared by both graphs. Note that the layout of the genes is the same for both graphs. 121
- Figure 38 The sparse backfitting algorithm. The first two steps in the iterative algorithm are the usual backfitting procedure; the remaining steps carry out functional soft thresholding. 143
- Figure 39 The SpAM backfitting algorithm is a functional version of the coordinate descent algorithm for the lasso, which computes  $\hat{\beta} = \arg \min \frac{1}{2} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$ . 144

|           |                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Figure 40 | Data version of the soft-thresholding operator. 155                                                                                                                                                                                                                                                                                                                                                                      |
| Figure 41 | The simultaneous sparse backfitting algorithm for MT-SpAM or MR-SpAM. For the multi-response case, the same smoothing matrices are used for each $k$ . 156                                                                                                                                                                                                                                                               |
| Figure 42 | The penalized local scoring algorithm for SMALR. 157                                                                                                                                                                                                                                                                                                                                                                     |
| Figure 43 | (Top) Estimated vs. true functions from MT-SpAM; (Middle) Regularization paths using MT-SpAM. (Bottom) Quantitative comparison between MR-SpAM and MARS 167                                                                                                                                                                                                                                                              |
| Figure 44 | SMALR results on gene data: heat map (left), marginal fits (center), and CV score (right). 168                                                                                                                                                                                                                                                                                                                           |
| Figure 45 | Model for predicting fMRI activation for a stimuli 169                                                                                                                                                                                                                                                                                                                                                                   |
| Figure 46 | The leave-two-out-cross-validation protocols 172                                                                                                                                                                                                                                                                                                                                                                         |
| Figure 47 | Bar and Box plots for accuracies for 9 fMRI participants 173                                                                                                                                                                                                                                                                                                                                                             |
| Figure 48 | The 25 estimated component functions using the MT-SpAM 174                                                                                                                                                                                                                                                                                                                                                               |
| Figure 49 | The Additive Forward Regression Algorithm 181                                                                                                                                                                                                                                                                                                                                                                            |
| Figure 50 | The Generalized Forward Regression Algorithm 182                                                                                                                                                                                                                                                                                                                                                                         |
| Figure 51 | Performance of the different algorithms on synthetic data: MSE versus sparsity level 186                                                                                                                                                                                                                                                                                                                                 |
| Figure 52 | Performance of the different algorithms on real datasets: CV error versus sparsity level 188                                                                                                                                                                                                                                                                                                                             |
| Figure 53 | (a) The 22 subregions defined on $[0, 1]^2$ . The horizontal axis corresponds to the first dimension denoted as $X_1$ while the vertical axis corresponds to the second dimension denoted as $X_2$ . The bottom left point corresponds to $[0, 0]$ and the upper right point corresponds to $[1, 1]$ . (b) The true graph for subregion 4. (c) The true graph for subregion 17. (d) The true graph for subregion 22. 216 |
| Figure 54 | (a) The estimated dyadic tree structure; (b) the induced partition on $[0, 1]^2$ and the number labeled on each subregion corresponds to each leaf node ID of the tree in (a); (c) the held-out negative log-likelihood risk for each split. The order of the splits corresponds the ID of the tree node (from small to large) 217                                                                                       |
| Figure 55 | (a) Estimated tree structure; (b) corresponding partitions 218                                                                                                                                                                                                                                                                                                                                                           |
| Figure 56 | Comparison of our algorithm with glasso (a) Precision; (b) Recall; (c) $F_1$ -score; (d) Estimated graph by applying glasso on the entire dataset 219                                                                                                                                                                                                                                                                    |
| Figure 57 | (a) Estimated tree structure; (b) estimated partitions where the labels correspond to the index of the leaf node in (a) 220                                                                                                                                                                                                                                                                                              |

|           |                                                                                                                                                                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Figure 58 | (a) Color map of $F_1$ -score for glasso run on the entire dataset; (b) color map of $F_1$ -score for Go-CART. Red indicates large values (approaching 1) and blue indicates small values (approaching 0), as shown in the color bar. <a href="#">221</a>                                                           |
| Figure 59 | Analysis of the climate data. (a) Estimated partitions for 125 locations projected to the US map, with the estimated graphs for subregions 2, 3, and 65; (b) estimated graph with data pooled from all 125 locations; (c) the re-scaled partition pattern induced by the dyadic tree structure. <a href="#">222</a> |
| Figure 60 | The estimated dyadic tree structure on the climate data. <a href="#">223</a>                                                                                                                                                                                                                                        |
| Figure 61 | Comparison of different methods on the data from the neighborhood graphs ( $n = 400, d = 100$ ). <a href="#">243</a>                                                                                                                                                                                                |
| Figure 62 | Comparison of different methods on the data from the hub graphs ( $n = 400, d = 100$ ). <a href="#">244</a>                                                                                                                                                                                                         |
| Figure 63 | Microarray data example. The StARS graph is more informative graph than the BIC graph. <a href="#">244</a>                                                                                                                                                                                                          |

## LIST OF TABLES

---

|         |                                                                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Table 1 | Quantitative comparison on the data set using the cdf transformation. For both FPE and FNE, the nonparanormal performs much better in general. <a href="#">80</a>   |
| Table 2 | Quantitative comparison on the data set using the power transformation. For both FPE and FNE, the nonparanormal performs much better in general. <a href="#">81</a> |
| Table 3 | Quantitative comparison on the data set without any transformation. The two methods behave similarly, the glasso is slightly better. <a href="#">82</a>             |
| Table 4 | The semantic basis used in Mitchell et al. (2008) <a href="#">169</a>                                                                                               |
| Table 5 | The 60 stimulus words presented during the fMRI studies. Each row represents a category <a href="#">170</a>                                                         |
| Table 6 | Accuracies for 9 fMRI participants <a href="#">173</a>                                                                                                              |
| Table 7 | An example of 25 learned semantic basis words. <a href="#">175</a>                                                                                                  |
| Table 8 | Comparison of variable selection <a href="#">187</a>                                                                                                                |
| Table 9 | Comparison of Variable Selection and Function Estimation on Synthetic Datasets <a href="#">202</a>                                                                  |



|          |                                                                                                    |     |
|----------|----------------------------------------------------------------------------------------------------|-----|
| Table 10 | Testing MSE on Real Datasets                                                                       | 203 |
| Table 11 | The graph estimation performance over different subregions                                         | 217 |
| Table 12 | Quantitative comparison of different methods on the datasets from the neighborhood and hub graphs. | 242 |

## LISTINGS

---

## ACRONYMS

---



## Part I

### THESIS OVERVIEW



## THESIS OVERVIEW

---

### 1.1 MOTIVATION AND THESIS STATEMENT

Modern data acquisition routinely produces massive amounts of high dimensional and highly complex datasets, including interactive logs from search engines, traffic records from network routing, chip data from high throughput genomic experiments, and image data from functional Magnetic Resonance Imaging (fMRI). Driven by the complexity of these new types of data, highly adaptive and reliable data analysis procedures are crucially needed.

Older high dimensional theories and learning algorithms rely heavily on parametric models, which assume the data come from an underlying distribution that can be characterized by a finite number of parameters. If these assumptions are correct, accurate and precise estimates can be expected. However, given the increasing complexity of modern data, conclusions inferred under these restrictive assumptions can be misleading. To handle this challenge, this thesis focuses on nonparametric methods, which directly conduct inference in infinite-dimensional spaces and thus are powerful enough to capture the subtleties in most modern applications.

The main goal of this thesis is to develop flexible and principled non-parametric learning algorithms to **explore**, **understand**, and **predict** high dimensional and complex datasets.

### 1.2 MAIN RESULTS

Results of this thesis include rigorous statistical theories and efficient non-parametric learning algorithms that exploit hidden structures to overcome the curse of dimensionality when analyzing massive high dimensional datasets. These algorithms have strong theoretical guarantees and provide high dimensional nonparametric recipes for many important learning tasks, ranging from unsupervised exploratory data analysis (e.g. density estimation, graphical model learning, clustering) to supervised predictive modeling (e.g. regression, classification, multi-task learning). The following list provides a summary of the thesis structure and highlights the main results:

#### A UNIFIED THEORETICAL FRAMEWORK

- **PERSISTENCY**: the prediction performance of a procedure;
- **CONSISTENCY**: the estimation performance of a procedure;
- **SPARSISTENCY**: the model selection performance of a procedure;
- **RATES OF CONVERGENCE**: the sample complexity.

#### UNSUPERVISED LEARNING METHODS

- **DENSITY RODEO**: Sparse Nonparametric Density Estimation using the Rodeo [Liu et al., 2007];
- **NONPARANORMAL**: Semiparametric Estimation of High Dimensional Undirected Graphs [Liu et al., 2009a];
- **FOREST DENSITY ESTIMATION**: Nonparametric Density Estimation and Undirected Graphical Model Learning [Liu et al., 2010c].

#### SUPERVISED LEARNING METHODS

- **MULTI-TASK SPAM**: Sparse Additive Models for Multi-task Regression and Multi-Class Classification [Liu et al., 2009b, 2008]
- **GREEDY FORWARD REGRESSION**: Nonparametric Regression with General Mean Function [Liu and Chen, 2009];
- **MULTIVARIATE DYADIC REGRESSION TREES**: Multivariate Regression with General Mean Functions [Liu and Chen, 2010];
- **GRAPH-OPTIMIZED CART**: Graph-Valued Regression [Liu et al., 2010a]

#### REGULARIZATION SELECTION METHOD

- **STARS**: Stability Approach to Regularization Selection for High Dimensional Undirected Graphical Models [Liu et al., 2010b]

#### SCIENTIFIC APPLICATIONS

- **GENOMICS**: tumor classification; biomarker discovery, gene regulatory network construction [Liu et al., 2009a, 2010c, 2008, 2010b];
- **COGNITIVE NEUROSCIENCE**: neural semantic basis discovery [Liu et al., 2009b];
- **METEOROLOGY**: learning spatial-temporal varying interaction graphs of climate factors [Liu et al., 2009b, Chen et al., 2010].

Several items in this list have very interesting extensions and will continue to shape our research agenda in the near future. Although these topics appear to be diverse, their underlying principles are the same, that is, to understand the fundamental mathematical structure of these problems in order to develop better theory and methods. In the following, we provide a high-level summary of the main results of this thesis.

### 1.3 ORGANIZATION OF THIS THESIS

The structure and logic of this thesis are illustrated in Figure 1. This thesis is motivated by novel applications arising from modern scientific domains including Genomics, Cognitive Neuroscience, and Meteorology. Under a unified theoretical framework, we develop flexible and efficient nonparametric learning algorithms for both supervised and unsupervised learning paradigms. Since all these methods involve a tuning parameter, we also propose a general regularization selection approach to automatically choose the tuning parameter. We organize the rest of the thesis into several parts: (Part II) fundamental theory; (Part III) unsupervised learning methods; (Part IV) supervised learning methods; (Part V) regularization selection.

#### 1.3.1 Part II: Fundamental Theory

In Chapter 2, we review some background of high dimensional data analysis and nonparametric statistical inference. Next, we provide an overview of the basic theoretical framework used in this thesis.

Classical nonparametric theory is developed by allowing the data sample size  $n$  to grow while the data dimension  $d$  remains low, typically fixed at one. In contrast, for high dimensional data, it is more realistic to allow both  $n$  and  $d$  to grow, with  $d$  possibly growing much faster than  $n$ . Under this setting, minimax theory shows that, without further structural assumptions, it is hopeless to obtain a consistent learning procedure. This fact characterizes the *statistical curse of dimensionality*. On the other hand, the time complexity of many learning algorithms also increases exponentially with  $d$ , which characterizes the *computational curse of dimensionality*. Together, these facts illustrate a fundamental limit of nonparametric learning methods: Structural assumptions must be traded off with statistical and computational efficiencies.

One concept that plays an important role in our research is *functional sparsity*. For example, in high-dimensional nonparametric regression, even if the observed dimensionality  $d$  is large, the true regression function might only depend on a small number of *relevant* dimensions  $r$  with  $r \ll d$ . In other words, the problem is sparse. This thesis aims to design learning algorithms

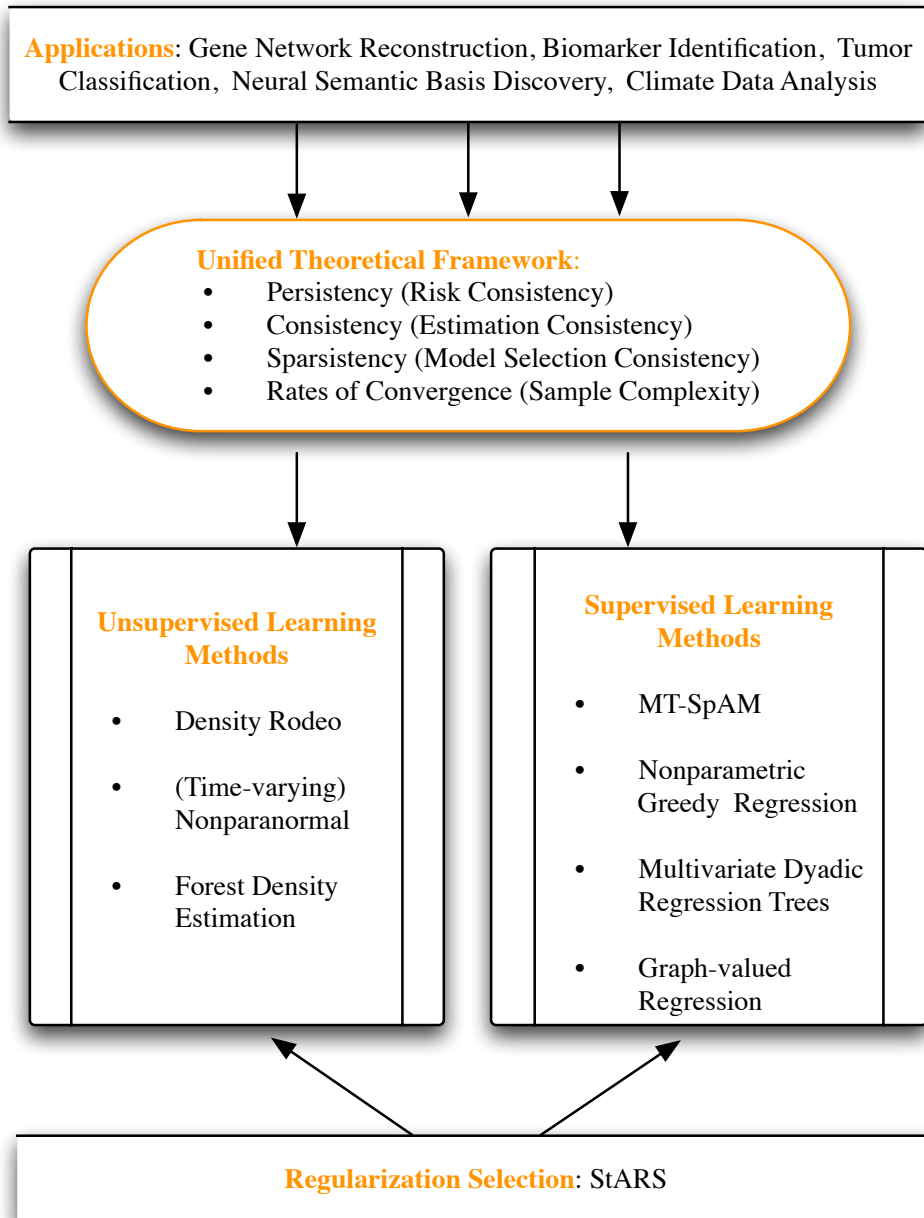


Figure 1.: A paradigm illustrating the thesis structure. The applications of this thesis come from modern scientific areas including Genomics, Cognitive Neuroscience, and Meteorology. Under a unified theoretical framework (including risk consistency, estimation consistency, and model selection consistency), we develop high-dimensional nonparametric learning methods for both unsupervised and supervised learning paradigms; To automatically select the regularization parameters of these methods, we also develop a regularization selection method named StARS.



that can effectively utilize this unknown sparsity pattern to beat the curse of dimensionality. Two key questions are addressed:

- *How can we design effective high-dimensional nonparametric algorithms?*
- *What kinds of theoretical guarantees can we provide?*

In a series of papers with John Lafferty and Larry Wasserman [Liu et al., 2008, 2010a,b,c], we propose an integrated framework consisting of a family of evaluation criteria from modern statistical theory and numerical analysis. These criteria can evaluate an algorithm’s performance from both the statistical and computational perspectives. For example, a criterion called “sparsistency” characterizes whether an algorithm can identify the unknown sparsity pattern with large probability; “persistency” characterizes whether an algorithm can predict as well as the best model within a family; The “rates of convergence” evaluates the asymptotic sample complexity that an algorithm requires to achieve a certain estimation accuracy; “nondegeneracy” and “numerical convergence” characterize the uniqueness of the solution and whether an iterative algorithm is guaranteed to converge to one of its solutions. This framework is quite general and provides the theoretical underpinnings of high dimensional nonparametric methods. We attempt to evaluate all the proposed methods in the thesis using this integrated theoretical framework.

Under this principled theoretical framework, we focus on developing new methods for a number of important data-analysis tasks. These methods roughly fall in two learning paradigms:

(P1) Unsupervised Learning: density estimation, undirected graphical model learning, clustering;

(P2) Supervised Learning: regression, classification, learning conditional undirected graphical models;

These two learning paradigms are broadly applicable, theoretically interesting, and represent learning tasks including exploratory data analysis and predictive modeling.

### 1.3.2 Part III: Unsupervised Learning Methods

#### 1.3.2.1 Density-rodeo: High-dimensional Nonparametric Density Estimation

Accurately estimating the joint density is a fundamental problem in nonparametric exploratory data analysis but is also notoriously difficult in high dimensions due to the curse of dimensionality. Some structure or sparsity

assumptions are needed to avoid the curse. In Chapter 3, we address the following problem—

- *What is an appropriate notion of sparsity in the density estimation setting?*

We assume the joint density can be factored into the product of a high-dimensional parametric baseline and a nonparametric correction term that depends only on an unknown small subset of the variables. This turns out to be a more realistic assumption for the real-world distributions. Under this assumption, we develop a method called density-rodeo [Liu et al., 2007] which can simultaneously achieve a nearly optimal rate of convergence in the sample complexity and a very efficient asymptotic polynomial running time. *This is the first method that can systematically conduct nonparametric density estimation in hundreds of dimensions with strong theoretical guarantees.* Figure 2 illustrates the method on a simulated dataset, showing how it captures the true shape of the density. In this simulation, the true bivariate density factors into the product of two univariate densities: a mixture of Beta distribution and a uniform distribution. The density rodeo perfectly recovers this true density while the R build-in method KDE2d fails.

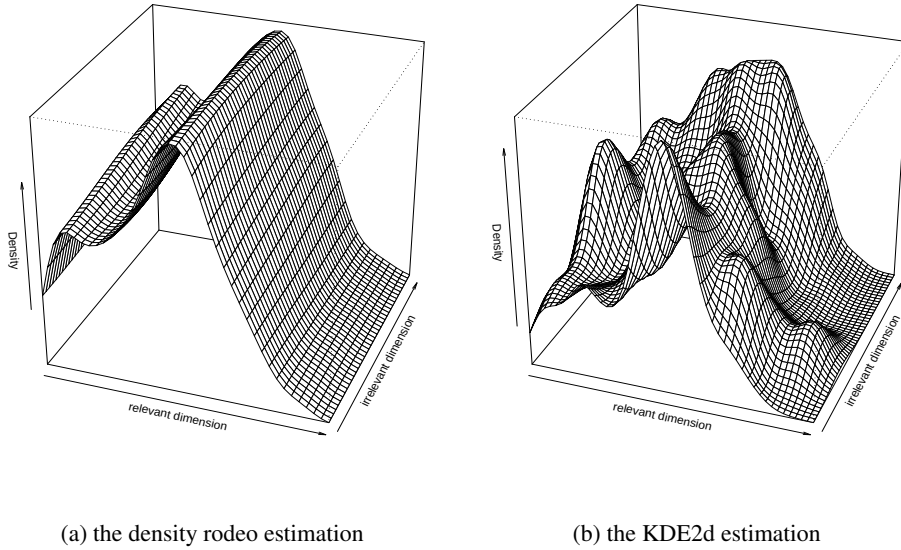


Figure 2.: An illustrative example of the density rodeo: Perspective plots of the estimated density functions by the density rodeo (left) and the R built-in method KDE2d (right) on a 2-dimensional synthetic data.

### 1.3.2.2 The Nonparanormal: Estimation of High-Dimensional Undirected Graphs

Another important exploratory data analysis task is to estimate undirected graphical models, which graphically represent the conditional independence structure among a large number of variables. Current methods for estimating sparse undirected graphs in high-dimensional problems rely heavily on the normality assumption, i.e., the data is assumed to have a multivariate Gaussian distribution, which significantly limits the applications of these methods. Motivated by this, we ask the following question—

- *Can we estimate high-dimensional undirected graphs for non-Gaussian data?*

In Chapter 4, we show that this is possible for a much larger family of distributions, which we call the “nonparanormal.” Just as additive models extend linear models by replacing linear functions with a set of one-dimensional smooth functions, the nonparanormal is a nonparametric extension of the normal that transforms the variables by one-dimensional smooth functions. Figure 3 provides three examples of 2-dimensional nonparanormal density and contour plots, which illustrates the richness of the nonparanormal family. We derive a method for estimating the nonparanormal and provide theoretic

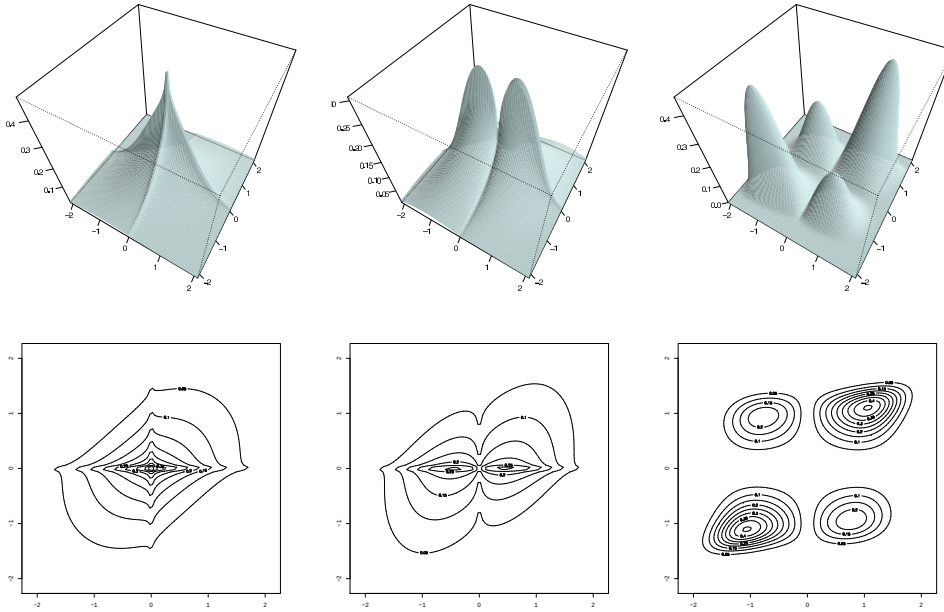


Figure 3.: An illustrate example of the nonparanormal: The densities of three 2-dimensional nonparanormals. The component functions have the form  $f_j(x) = \text{sign}(x)|x|^{\alpha_j}$ . Left:  $\alpha_1 = 0.9, \alpha_2 = 0.8$ ; center:  $\alpha_1 = 1.2, \alpha_2 = 0.8$ ; right  $\alpha_1 = 2, \alpha_2 = 3$ . In each case  $\mu = (0, 0)$  and  $\Sigma = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ .

cal guarantees including persistency, consistency, and sparsistency. We have

applied the nonparanormal to estimate gene-gene interaction graphs for the isoprenoid biosynthetic pathway of *Arabidopsis thaliana* using microarray data. Our method supports different biological conclusions from those obtained using the Gaussian graphical model and graphical lasso. A variant of time-varying nonparanormal has been applied to analyze climate data [Chen et al., 2010]. *Some preliminary analysis shows that our result has a better match with existing Meteorology theory than the competing state-of-the-art approaches.*

### 1.3.2.3 Forest Density Estimation

In Chapter 5, we propose a *forest density estimator* which can simultaneously estimate high dimensional densities and undirected graphical models. The nonparanormal assumes the data can be Gaussianized using a set of univariate monotone functions, but allows for arbitrary undirected graphical models. In contrast, the forest density estimator assumes arbitrary smooth distributions, but only allows for forest graphical models. Together, they reflect a tradeoff between distribution flexibility and graphical model complexity. Both methods have been proven to be persistent, consistent, and sparsistent; they are the first high dimensional nonparametric density estimation methods that have all of these desirable properties.

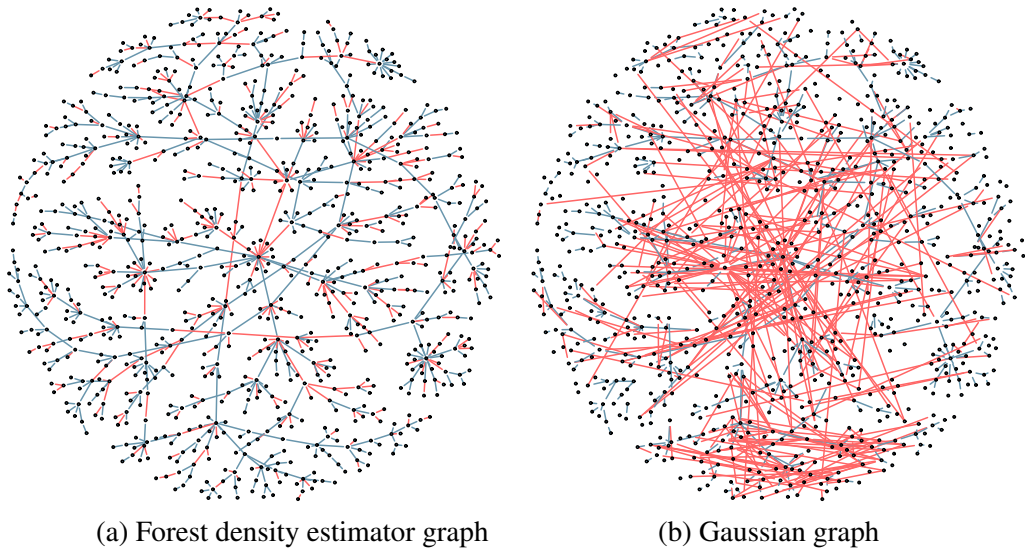


Figure 4.: An illustrative example of the forest density estimator: a 934 gene subgraph of the full estimated 4238 gene network. Left: estimated forest graph. Right: estimated Gaussian graph. Red edges in the forest graph are missing from the Gaussian graph and vice versa; the blue edges are shared by both graphs. Note that the layout of the genes is the same for both graphs.

Figure 4 provides an illustration of the forest density estimator on a human microarray dataset. The data contains Affymetrics chip measured expression

levels of 4238 genes for 295 normal subjects in the *Centre d'Etude du Polymorphisme Humain* (CEPH) and the International HapMap collections. The 295 subjects come from four different groups: 148 unrelated grandparents in the CEPH-Utah pedigrees, 43 Han Chinese in Beijing, 44 Japanese in Tokyo, and 60 Yoruba in Ibadan, Nigeria. We estimate the full 4238 node graph using both the forest density estimator (described in Sections 5.3.1 and 5.3.2) and the state-of-the-art Gaussian graphical models proposed in Meinshausen and Bühlmann [2006] with the regularization parameter chosen so that the number of estimated edges is the same as the forest graph.

The forest density estimator graph reveals one strongly connected component of more than 3000 edges and various isolated genes; this is consistent with the analysis in Nayak et al. [2009] and is realistic for the regulatory system of humans. The Gaussian graph contains similar component structure, but the set of estimated edges differs significantly. For visualization purposes, in Figure 4, we only show a 934 gene subgraph of the strongly connected component among the full 4238 node graphs we estimated.

### 1.3.3 Part IV: Supervised Learning Methods

#### 1.3.3.1 Multi-task Sparse Additive Models

Many application problems can be naturally formulated in terms of multi-task regression or multi-class classification problems, in which several regression or discriminant functions need to be estimated based on different datasets. Sometimes, while the details of the predictors vary from instance to instance, they may share a common sparsity pattern across different regression tasks or class categories. The linear model is a mainstay of statistical inference for these problems and has been extended in several important ways. An extension to high dimensions was achieved by adding a sparsity constraint, leading to different variants of the sparse linear models. An extension to nonparametric models was achieved by replacing linear functions with smooth functions, leading to additive models. These developments motivate a natural question—

- *Can we combine the power of sparse linear modeling and additive models?*

In Chapter 6, we describe a new family of models and algorithms for high-dimensional nonparametric multi-task learning with joint sparsity constraints Liu et al. [2008]. Our approach formulates the problem as a sum of sup-norm penalized least squares regression, which enforces common sparsity patterns across different function components in a nonparametric additive model. The obtained algorithm is called simultaneous sparse backfitting, which is highly efficient since each individual iteration can be solved by a closed-form functional soft-thresholding operator. This framework is very flexible and yields several new models, including multi-task sparse additive models,

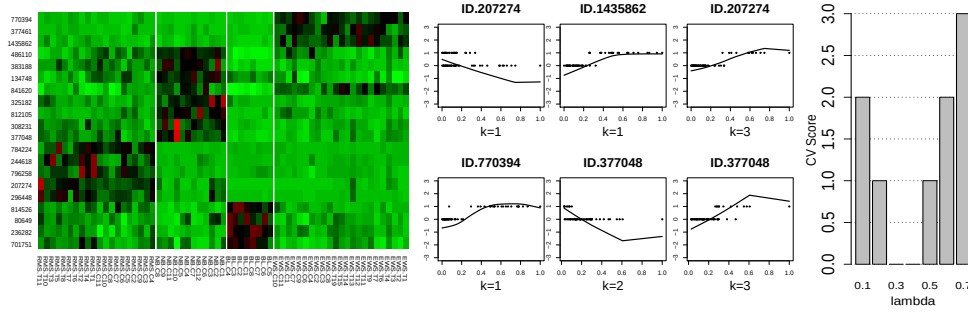


Figure 5.: SMALR results on gene data: heat map (left), marginal fits (center), and CV score (right).

multi-response sparse additive models, and sparse additive multi-category logistic regression. Using the same idea as in [Ravikumar et al. \[2007, 2009a\]](#), the multi-task sparse additive models can be proved to be both persistent and sparsistent.

Figure 5 illustrates our method on a microarray dataset for small round blue cell tumor (SRBCT) classification. This data contains 2,308 genes and 4 categories. Compared with previous analyses on this same data, our method achieves the best predictive accuracy on the test set (100% accuracy) using the most compact set of predictors (20 genes). From the gene heatmap, the selected 20 genes are seen to have an informative block structure, and the fitted components are highly nonlinear. This indicates that high-dimensional nonparametric inference is suitable for this dataset.

### 1.3.3.2 Greedy Regression and Multivariate Dyadic Regression Trees

The multi-task sparse additive models assume the regression or classification functions have an additive form. Such an assumption may still be restrictive in applications. Motivated by this, we address the question:

- Can we conduct high-dimensional nonparametric inference with general multivariate regression/classification functions?

To estimate general multivariate functions, we propose *greedy forward regression* and *multivariate dyadic regression trees* in Chapters 7 and 8. Both methods simultaneously conduct estimation and variable selection in high dimensional nonparametric regression problems. They can be viewed as nonparametric counterparts for two major sparsity-inducing approaches: *greedy pursuit* and *convex regularization*. The greedy pursuit approach regularizes the model by iteratively selecting the current optimal approximation according to some criterion; While the convex regularization approach regularizes the model by adding a sparsity constraint. We provide theoretical justifications for both methods and validate the theoretical arguments on real datasets.



### 1.3.3.3 Graph-valued Regression

Let  $Y$  be a high dimensional random vector with independence relations encoded in a graph  $G$ . In many applications, it is of interest to model  $Y$  given another random vector  $X$  as input. We refer to the problem of estimating the graph  $G(x)$  of  $Y$  conditioned on  $X = x$  as “graph-valued regression.” The key question we address here is

- Can we estimate conditional undirected graphical models with respect to possibly high dimensional covariates?

In Chapter 9, we propose a semiparametric method for estimating  $G(x)$  that builds a tree on the  $X$  space just as in CART (classification and regression trees), but at each leaf of the tree estimates a graph. We call the method “Graph-optimized CART,” or Go-CART. We study the theoretical properties of Go-CART using dyadic partitioning trees, establishing oracle inequalities on risk minimization and tree partition consistency. We also demonstrate the application of Go-CART to a meteorological dataset, showing how graph-valued regression can provide a useful tool for analyzing complex data.

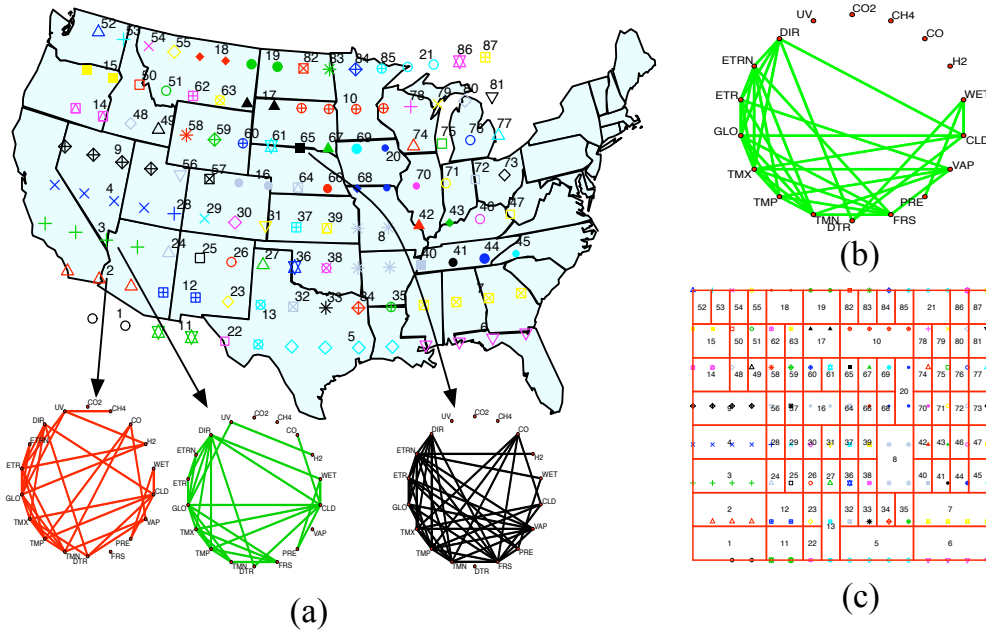


Figure 6.: An illustrative example of Go-CART: Analysis of the climate data. (a) Estimated partitions for 125 locations projected to the US map, with the estimated graphs for subregions 2, 3, and 65; (b) estimated graph with data pooled from all 125 locations; (c) the re-scaled partition pattern induced by the dyadic tree structure.

Figure 6 provides an illustrative example of applying graph-valued regression to analyze a meteorology dataset [Lozano et al., 2009] that contains monthly data of 18 different meteorological factors from 1990 to 2002. The

observations span 125 locations in the US on an equally spaced grid between latitude 30.475 and 47.975 and longitude -119.75 to -82.25. The 18 meteorological factors measured for each month include levels of  $\text{CO}_2$ ,  $\text{CH}_4$ ,  $\text{H}_2$ ,  $\text{CO}$ , average temperature (TMP) and diurnal temperature range (DTR), minimum temperature (TMN), maximum temperature (TMX), precipitation (PRE), vapor (VAP), cloud cover (CLD), wet days (WET), frost days (FRS), global solar radiation (GLO), direct solar radiation (DIR), extraterrestrial radiation (ETR), extraterrestrial normal radiation (ETRN) and UV aerosol index (UV). For further detail, see [Lozano et al. \[2009\]](#).

More detailed analysis of Figure 6 can be found in Chapter 9. Here, the key point is that the fitted results by estimating the conditional independence graphs (conditional on the geographic locations) are more interpretable than those obtained by estimating an unconditional universal graph.

#### 1.3.4 Part V: Regularization Selection

##### 1.3.4.1 Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models

All the new methods proposed in this thesis contain at least one tuning parameter. A challenging problem is to choose the regularization parameter in a data-dependent way. The standard techniques include  $K$ -fold cross-validation ( $K$ -CV), Akaike information criterion (AIC), and Bayesian information criterion (BIC). Though these methods work well for low-dimensional problems, they are not suitable in high dimensional settings. The key challenge is:

- *Can we reliably choose the regularization parameters for high dimensional non-parametric methods?*

In chapter 10, we present StARS: a new stability-based method for choosing the regularization parameter in high dimensional inference. The method is quite general and can be applied to different kinds of parametric and nonparametric models. In this chapter, we only consider the problem of estimating high dimensional undirected graphs as a pilot study. The method has a clear interpretation: we use the least amount of regularization that simultaneously makes a graph sparse and replicable under random sampling of data points. This interpretation requires essentially no conditions. Under mild conditions, we show that StARS is partially sparsistent in terms of graph estimation: i.e. with high probability, all the true edges will be included in the selected model even when the graph size diverges with the sample size.

Empirically, the StARS is expected to provide more informative (sparser) graphs than the other state-of-the-art methods. We illustrate this point in Figure 7 using the fitted graphs of the StARS and Bayesian information criterion (BIC) on a human microarray dataset containing 324 genes.



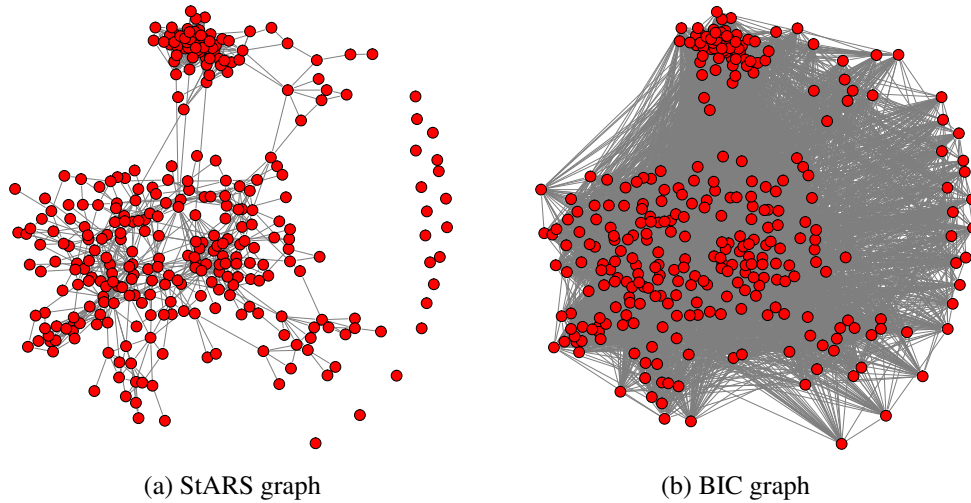


Figure 7.: An illustrative example of StARS: Two estimated graphs on the microarray data using the StARS and BIC. The StARS graph is more informative than the BIC graph.

We see that the StARS graph is remarkably simple and informative, exhibiting some cliques and hub genes. In contrast, the BIC graph is very dense and possible useful association information is buried in the large number of estimated edges.

#### 1.3.4.2 Conclusions

Some concluding remarks and future visions are provided in the last chapter.

### 1.4 RELATED PUBLICATIONS

Parts of the thesis work have been published in major refereed machine learning conferences and journals. Below is an incomplete list:

Related publications of Chapter 3 include

- Han Liu, John Lafferty, and Larry Wasserman (2007). *Sparse Nonparametric Density Estimation using the Rodeo*. Proceedings of the Eleventh Conference on Artificial Intelligence and Statistics (AISTATS), pages 283-290, April 2007.

Related publications of Chapter 4 include

- Han Liu, John Lafferty, and Larry Wasserman (2009). *The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs*. Journal of Machine Learning Research (JMLR), Volume 10, pages 2295-2328, 2009.

- Xi Chen, Yan Liu, Han Liu, and Jamie Carbonell (2010). *Learning Spatial-Temporal Varying Graphs with Applications to Climate Data Analysis*. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI), pages 425-430, June 2010

Related publications of Chapter 5 include

- Han Liu, Min Xu, Haijie Gu, Anupam Dasgupta, John Lafferty, and Larry Wasserman (2010). *Forest Density Estimation*. On the arXiv:1001.1557. A short conference version appeared in the Proceedings of the Twenty-Third Annual Conference on Learning Theory (COLT), pages 394-406, June 2010.

Related publications of Chapter 6 include

- Han Liu, John Lafferty, and Larry Wasserman (2008). *Nonparametric Regression and Classification with Joint Sparsity Constraints*. Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems (NIPS), pages 969-976, December 2008.
- Han Liu, Mark Palatucci, and Jian Zhang (2009). *Blockwise Coordinate Descent Procedures for the Multi-task Lasso, with Applications to Neural Semantic Basis Discovery*. Proceedings of the Twenty-Sixth International Conference on Machine Learning (ICML), pages 82-90, June 2009.
- Han Liu and Jian Zhang (2009). *On the Estimation Consistency of the Group Lasso and its Applications*. Proceedings of the Twelfth Conference on Artificial Intelligence and Statistics (AISTATS), pages 376-383, August 2009
- Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman (2009). *Sparse Additive Models*. Journal of the Royal Statistical Society: Series B (JRSSB), 71(5), pages 1009-1030, 2009.

Related publications of Chapter 7 include

- Han Liu and Xi Chen (2009). *Nonparametric Greedy Algorithm for the Sparse Learning Problems*. Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS), pages 1141-1149, December 2009.

Related publications of Chapter 8 include

- Han Liu and Xi Chen (2010). *Multivariate Dyadic Regression Trees for Sparse Learning Problems*. Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS), December 2010.

Related publications of Chapter 9 include

- Han Liu, Xi Chen, John Lafferty, and Larry Wasserman (2010). *Graph-valued Regression*. Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS), December 2010.

Related publications of Chapter 10 include

- Han Liu, Kathryn Roeder, and Larry Wasserman (2010). *Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models*. Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS), December 2010.



**Part II**

**STATISTICAL THEORY**



In this chapter, we briefly review the literature on high-dimensional nonparametric inference. We are especially interested in the situation where the number of data dimensions can asymptotically increase with the sample size, which is a key challenge for modern statistical theory. We then introduce the key concept of *sparsity* and its usefulness in high dimensional linear models. We also briefly overview existing results on high dimensional nonparametric methods. Finally, we present the unified theoretical framework utilized throughout this thesis.

## 2.1 LITERATURE OVERVIEW

Nonparametric learning aims to develop flexible and computationally efficient algorithms that can effectively explore and predict complex datasets. It can be viewed as a sub-area of both statistics and computer science. Researchers from both communities are attacking essentially the same problem with different emphasis. To conduct nonparametric inferences in high dimensions, we need to estimate infinite-dimensional smooth functions from large-scale data. This task is both theoretically and computationally challenging due to the *curse of dimensionality*, which means that inference becomes exponentially harder when the number of dimensions increases.

### 2.1.1 Curse of Dimensionality

One way to characterize the curse is by *minimax theory*. Given  $n$  observed data points

$$\mathcal{D}_n = \{(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})\}, \quad (2.1)$$

where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T \in \mathbb{R}^d$  and  $Y^{(i)} \in \mathbb{R}$ . We consider a  $d$ -dimensional nonparametric regression problem

$$Y^{(i)} = m(X^{(i)}) + \epsilon^{(i)}, \quad i = 1, \dots, n \quad (2.2)$$

where  $\epsilon^{(i)}$  is independently distributed mean zero noise. Let  $L_2(0, 1)$  be the class of  $L_2$ -functions with support on  $[0, 1]^d$ . If  $m$  is in  $\mathcal{M} = W(2, c)$ , the  $d$ -dimensional Sobolev ball of order two and radius  $c$ , which is defined as

$$W(2, c) = \left\{ f : f \in L_2(0, 1), D^2 f \in L_2(0, 1), \|D^2 f\|_{L_2}^2 \leq c^2 \right\}, \quad (2.3)$$

the risk of an estimator  $\hat{m}_n$ , defined as

$$\mathcal{R}(\hat{m}_n, m) = \mathbb{E}_m \int (\hat{m}_n(x) - m(x))^2 dx$$

satisfies

$$\liminf_{n \rightarrow \infty} n^{4/(4+d)} \inf_{\hat{m}_n} \sup_{m \in W(2, c)} \mathcal{R}(\hat{m}_n, m) > 0. \quad (2.4)$$

Thus, the rate of convergence is  $O(n^{-4/(4+d)})$ . This means, to achieve a certain error rate, the required sample size is exponential in  $d$ , which is practically intractable.

Another way to characterize the curse is by complexity theory. Consider a nonparametric kernel density estimation problem. Let  $X^{(1)}, X^{(2)}, \dots, X^{(n)}$  be a sample from a  $d$ -dimensional density  $p(x)$ . We are interested in estimating the density  $p(x)$  when the dimension  $d$  is large. For an evaluation point  $x$ , the kernel density estimator [Parzen, 1962] is defined as

$$\hat{f}_H(x) = \frac{1}{n \det(H)} \sum_{i=1}^n \mathcal{K}(H^{-1}(x - X^{(i)})) \quad (2.5)$$

$$= \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \frac{1}{h_j} K\left(\frac{x_j - X_j^{(i)}}{h_j}\right) \quad (2.6)$$

where  $\mathcal{K}$  is a symmetric product kernel with  $\int \mathcal{K}(u) du = 1$  and  $\int u \mathcal{K}(u) du = \mathbf{0}_d$ .  $H = \text{diag}(h_1, \dots, h_d)$  is a diagonal bandwidth matrix. If we want to conduct bandwidth selection through cross validation with each dimension has  $q$  candidate bandwidths, the total number of calculations will be  $O(q^d)$ . The computational cost of trying all possible combinations is exponential in the data dimension  $d$ , which is also practically intractable.

#### 2.1.1.1 Theoretical Contributions of this Thesis

A major goal of this thesis is to develop rigorous theoretical frameworks which could overcome the curse of dimensionality when analyzing high dimensional datasets. A central concept in our framework is *sparsity* (or *functional sparsity* in the nonparametric settings), which means that the data has some hidden structures. Even though the observed data dimension  $d$  is very large, the *relevant* (or *intrinsic*) dimension  $r$  can be very low, i.e.  $r \ll d$ . Under such a sparsity assumption, it's possible to develop some flexible methods which can automatically exploit this underlying structure as if it's already known.



### 2.1.2 Sparsity in High-dimensional Parametric Methods

The notion of sparsity has played important role in fitting high-dimensional linear models. One particularly successful case is the Lasso estimator [Tibshirani, 1996], which is also named as basis pursuit estimator in the signal processing community [Chen et al., 1998]. Consider a linear model  $m(x) = x^T \beta$ . The Lasso estimator is defined as

$$\hat{\beta}^\lambda = \arg \min_{\beta} \left\{ \frac{1}{n} \sum_{i=1}^n \left( Y^{(i)} - X^{(i)T} \beta \right)^2 + \lambda \|\beta\|_1 \right\} \quad (2.7)$$

where  $\lambda$  is the regularization parameter for the  $\ell_1$ -norm of the coefficients  $\beta$ . The solution of Lasso can be obtained by standard convex optimization techniques [Osborne et al., 1999]. Furthermore, its entire solution path can be computed in the same complexity of that of least squares [Efron et al., 2004]. Greenshtein and Ritov [2004] prove that the Lasso estimator is persistent, in the sense that the predictive risk of the Lasso estimator converges to the risk obtained by the oracle estimator in probability. However, recent studies [Meinshausen and Bühlmann, 2006, Zhao and Yu, 2007, Zou, 2006] show that the Lasso estimator is not in general model selection consistent (or sparsistent), which means the correct sparse subset of the relevant variables can not be identified asymptotically. In particular, in [Zhao and Yu, 2007], it is shown that in order for Lasso to be model selection consistent, the so-called irrepresentable condition has to be satisfied. Zou [2006] propose the adaptive Lasso and showed that by using adaptive weights for different variables, the  $\ell_1$  penalty can lead to model selection consistent estimator. Besides adaptive Lasso, the non-negative garrotte estimator [Breiman, 1995] has also been shown to be able to achieve model selection consistency in a two-step procedure given that the initial estimator is estimation consistent [Zou, 2006, Yuan and Lin, 2007]. In terms of estimation, it has been shown in Meinshausen and Yu [2009] that under weaker conditions, the Lasso estimator is  $\ell_2$  estimation consistent for high-dimensional setting where  $d$  can grow almost as fast as  $\exp(n)$ . Under a stronger assumption, Bunea et al. [2007] further prove the sparsity oracle inequalities for the Lasso estimator using fixed design. These oracle inequalities can be used to derive the rate of convergence of the Lasso estimator as  $O(\log d/n)$ , a similar result for the random design can be found in [Bunea et al., 2007]. Some newest theoretical results related to Lasso include Negahban and Wainwright [2008], Wainwright [2009], Obozinski et al. [2009], Ravikumar et al. [2010].

All these results show that the Lasso estimator can effectively utilize the hidden sparsity structure in the linear regression to overcome both the statistical and computational curse of dimensionality even when facing increasing dimensions.

### 2.1.3 Sparsity in High-dimensional Nonparametric Methods

It's a natural idea to extend the Lasso estimator to high-dimensional nonparametric methods. There are two basic approaches to utilize sparsity in high dimensional nonparametric inferences: *convex regularization* and *greedy pursuit*.

Substantial progress has been made recently on applying the convex regularization idea to fit sparse additive models. For splines, [Lin and Zhang \[2006\]](#) propose a method called COSSO, which uses the sum of reproducing kernel Hilbert space norms as a sparsity inducing penalty, and can simultaneously conduct estimation and variable selection; A recent work of [Jeon and Lin \[2006\]](#) extended the idea of COSSO to density estimation setting. More technical details of the COSSO are presented in the Appendix A of this thesis. In parallel to the COSSO, [Ravikumar et al. \[2007, 2009a\]](#) develop a method called SpAM. The population version of SpAM can be viewed as a least squared regression problem penalized by the sum of  $L_2(P)$ -norms; [Meier et al. \[2009\]](#) develop a similar method using a different sparsity-smoothness penalty, which guarantees the solution to be a spline. The newest theoretical result on the sparse additive models is developed by [Raskutti et al. \[2010\]](#). All these methods can be viewed as different nonparametric variants of Lasso.

Another way to conduct high dimensional nonparametric inference is through greedy pursuit. Instead of trying to formulate the whole learning task into a global convex optimization, the greedy pursuit approaches adopt iterative algorithms with a local view. During each iteration, only a small number of variables are actually involved in the model fitting so that the whole inference only involves low dimensional models. Thus they naturally extend to the general multivariate regression and do not induce large estimation bias, which makes them especially suitable for high dimensional nonparametric inference. However, the greedy pursuit approaches do not attract as much attention as the convex regularization approaches in the nonparametric literature. For additive models, existing methods include the sparse boosting [[Bühlmann and Yu, 2006](#)] and multivariate adaptive regression splines (MARS) [[Friedman, 1991](#)]. These methods mainly target on additive models or lower-order functional ANOVA models. For general multivariate regression, the only available method we are aware of is rodeo [[Lafferty and Wasserman, 2008](#)]. The rodeo assumes the true regression function only depends on  $r$  covariates and  $r \ll d$ . Using the local linear regression estimator [[Fan and Gijbels, 1996](#)], the rodeo can simultaneously perform bandwidth selection and (implicitly) variable selection to achieve an improved minimax convergence rate of  $O(n^{-4/(4+r)})$  up to a logarithmic factor, as if the  $r$  relevant variables were explicitly isolated in advance. The rodeo algorithm starts with large bandwidths for all dimensions and incrementally shrink the bandwidths by a sequence of hypothesis tests in a greedy manner, the nearly optimal rate of convergence can be achieved even with increasing dimension  $d = O(\log n)$ .

## 2.2 THEORETICAL FRAMEWORK OF THIS THESIS

It is easy to invent new learnings methods and try it on some datasets. But how do we know if a method is good or bad? Especially how can we gain insights on why a method works for specific situations and does not work for other situations. This requires us to provide theoretical justifications of the methods. In this thesis, we aim to design a theoretical framework that could evaluate a learning algorithm from different perspectives, including

1. prediction performance
2. estimation performance
3. model selection performance
4. sample complexity (The number of data points required to obtain a certain error rate).

Before we present the detailed theoretical criteria, in Figure 8, we start with a review of the typical workflow for designing statistical learning methods. We first characterize a function class  $\mathcal{F}$  which does not have to contain the true function  $f^*$ . Given some risk criterion  $R(\cdot)$ , the oracle estimator  $f^o$  is defined as

$$f^o = \arg \min_{f \in \mathcal{F}} R(f).$$

Statistical inferences aim at finding a function  $\hat{f}_n \in \mathcal{F}$  that mimics the oracle  $f^o$ . Our theoretical criteria mainly characterize the relationships among  $f^*$ ,  $f^o$ , and  $\hat{f}_n$ .

The first criterion we are interested in is called *persistency* or *risk consistency*, which characterizes how fast the predictive risk could converge to the oracle risk.

**Definition 2.1.** (Persistency or Risk Consistency) *For an estimator  $\hat{f}_n \in \mathcal{F}$ , the excess risk is defined as*

$$R(\hat{f}_n) - \min_{f \in \mathcal{F}} R(f). \quad (2.8)$$

*The estimator  $\hat{f}_n$  is risk consistent with the rate of convergence  $\delta_n$  if*

$$\lim_{M \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{P} \left( R(\hat{f}_n) - \min_{f \in \mathcal{F}} R(f) \geq M\delta_n \right) = 0. \quad (2.9)$$

*We also write*

$$R(\hat{f}_n) - \min_{f \in \mathcal{F}} R(f) = O_P(\delta_n) \text{ or } R(\hat{f}_n) - R(f^o) = O_P(\delta_n). \quad (2.10)$$

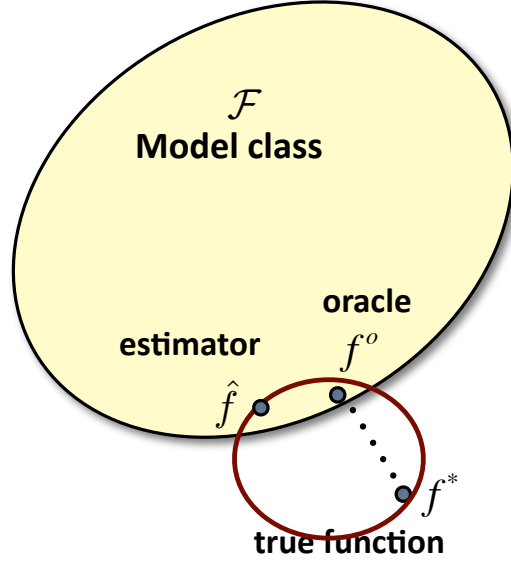


Figure 8.: An paradigm of statistical learning under a function estimation view. the true function:  $f^*$ ; the oracle:  $f^o$ ; the empirical estimator:  $\hat{f}$ .

One thing to note is that the persistency criterion only considers the relationship between the empirical estimator  $\hat{f}_n$  and the oracle  $f^o$ , which does not involve the true function  $f^*$ . Thus a procedure to be persistent does not require the model to be correctly specified.

Let  $D(\cdot, \cdot)$  be some distance (or divergence) defined on  $\mathcal{F}$  and  $D(\hat{f}_n, f^*)$  be the distance (or divergence) between the empirical estimator  $\hat{f}_n$  and  $f^*$ . The second criterion we are interested in is *estimation consistency*, which characterizes how fast  $\hat{f}_n$  converges to  $f^*$  evaluated by  $D(\cdot, \cdot)$ :

**Definition 2.2.** (Estimation Consistency) *Under a distance (or divergence)  $D(\cdot, \cdot)$ , an estimator  $\hat{f}_n \in \mathcal{F}$  is estimation consistent with the rate of convergence  $\delta_n$  if*

$$\lim_{M \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{P} \left( D(\hat{f}_n, f^*) \geq M\delta_n \right) = 0. \quad (2.11)$$

We could also write

$$D(\hat{f}_n, f^*) = O_P(\delta_n). \quad (2.12)$$

Unlike the persistency criterion, the estimation consistency criterion requires the model to be correctly specified.

Many methods in this thesis involve estimating the structures of a function  $f$ , denoted by  $\text{Struct}(f)$ . For example, we might be interested in estimating the conditional independence graph of a high dimensional density function or the sparsity pattern of a regression function. The third criterion we are

interested in is called *sparsistency* (or *model selection consistency*, *structure estimation consistency*), which characterizes the structure estimation performance of a procedure.

**Definition 2.3.** (Sparsistency or Model Selection Consistency, Structure Estimation Consistency, Structural Consistency) *An estimator  $\hat{f}_n \in \mathcal{F}$  is sparsistent if*

$$\limsup_{n \rightarrow \infty} \mathbb{P} \left( \text{Struct}(\hat{f}_n) \neq \text{Struct}(f^*) \right) = 0. \quad (2.13)$$

Similar to the estimation consistency, sparsistency also requires the model to be correctly specified. Of course, we could define criteria like oracle estimation consistency and oracle sparsistency. Instead of evaluating the estimation and structure estimation performances of  $\hat{f}_n$  with respect to  $f^*$ , we compare  $\hat{f}_n$  with the oracle  $f^o$ . These oracle criteria allow the models to be mis-specified.

**Definition 2.4.** (Oracle Estimation Consistency) *Under a distance (or divergence)  $D(\cdot, \cdot)$ , an estimator  $\hat{f}_n \in \mathcal{F}$  is oracle estimation consistent with the rate of convergence  $\delta_n$  if*

$$\lim_{M \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{P} \left( D(\hat{f}_n, f^o) \geq M\delta_n \right) = 0. \quad (2.14)$$

**Definition 2.5.** (Oracle Sparsistency) *An estimator  $\hat{f}_n \in \mathcal{F}$  is sparsistent if*

$$\limsup_{n \rightarrow \infty} \mathbb{P} \left( \text{Struct}(\hat{f}_n) \neq \text{Struct}(f^o) \right) = 0. \quad (2.15)$$

We try to evaluate all the proposed methods in this thesis under this integrated theoretical framework.



### Part III

## UNSUPERVISED LEARNING





*In this Chapter, we consider the problem of estimating the density of a  $d$ -dimensional random vector  $X = (X_1, X_2, \dots, X_d)$  when  $d$  is large. We assume that the density is a product of a parametric baseline component and a nonparametric component. The nonparametric component depends on an unknown subset of the variables. If this subset is small, then nonparametric estimates with fast rates of convergence are possible. Using a modification of a previously developed nonparametric regression framework called rodeo (regularization of derivative expectation operator), we propose a method to exploit this fact. The method selects the bandwidths in an incremental way making it computationally attractive. We empirically show that the density rodeo works well even for very high-dimensional problems. When the unknown density function satisfies some suitably defined sparsity conditions, our approach avoids the curse of dimensionality and achieves an optimal converge rate of the risk. Because it is a greedy algorithm, bandwidth selection is fast. When the parametric baseline is a very smooth distribution, we also provide theoretical guarantees on the behavior of the method.*

### 3.1 INTRODUCTION AND MOTIVATION

Let  $X^{(1)}, X^{(2)}, \dots, X^{(n)}$  be a sample from a distribution  $F$  with density  $p(x)$ . We are interested in estimating the density  $p(x)$  when the dimension  $d$  of  $X^{(i)}$  is moderate or large. Methods for estimating  $p(x)$  include the kernel estimator [Parzen, 1962, Rosenblatt, 1956], local likelihood estimator [Hjort and M.C.Jones, 1996, Hjort and Glad, 1995, Loader, 1996] and others. These methods work very well for low-dimensional problems ( $d \leq 3$ ) but are not effective for high-dimensional problems. The major difficulty is due to the intractable computational cost of cross validation when bandwidths need to be selected for each dimension, and the slow rates of convergence of the estimator. Density estimation in high dimensions is usually done by mixture models [Dempster et al., 1977, Laird, 1978, Escobar and West, 1994, Richardson and Green, 1997]. However, mixture models with a fixed number of components are parametric and only useful to the extent that the assumed model is right. Mixture models without a fixed number of components are nonparametric and achieve, at best, the same rates as kernel estimators. In fact, the theoretical guarantees with mixtures are generally not as good

as for kernel estimators, see [Genovese and Wasserman \[2000\]](#) and [Ghosal et al. \[2000\]](#). Other methods for high dimensional density estimation include projection pursuit [[Friedman et al., 1984](#)], log-spline model [[Stone, 1990](#)] and penalized likelihood [[Silverman, 1982](#)].

In a  $d$ -dimensional space, minimax theory shows that the best convergence rate for the mean squared error under standard smoothness assumptions is  $\mathcal{R}_{opt} = O(n^{-4/(4+d)})$  which represents the “curse of dimensionality” when  $d$  is large. In this paper we present a method that achieves faster rates of convergence when a certain sparsity assumption is satisfied. Moreover, it is a greedy method and so is computationally expedient for large  $d$ .

The idea comes from a newly developed nonparametric regression framework called *rodeo* [[Lafferty and Wasserman, 2008](#)]. For the regression problem,

$$Y^{(i)} = m(X^{(i)}) + \epsilon^{(i)}, \quad i = 1, \dots, n,$$

where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)}) \in \mathbb{R}^d$  is a  $d$ -dimensional vector. Assuming that the true function only depends on  $r$  covariates  $r \ll d$ , the rodeo can simultaneously perform bandwidth selection and (implicitly) variable selection to achieve a better minimax convergence rate of  $O(n^{-4/(4+r)})$  up to a logarithmic factor, as if the  $r$  relevant variables were explicitly isolated in advance. The purpose of this chapter is to extend this idea to the nonparametric density estimation setting. Toward this goal, we need to first define an appropriate sparsity condition in the density estimation setting. Our key assumption is

$$p(x_1, \dots, x_d) = g(x_R) b(x_1, \dots, x_d) \quad (3.1)$$

where  $g$  is an unknown function,  $x_R = (x_j : j \in R)$ ,  $R$  is a subset of  $\{1, \dots, d\}$  and  $b$  is a baseline density (completely known or known up to finitely many parameters). If the number of coordinates in  $R$  is small then we can exploit the fact that the nonparametric component  $g$  only depends on a small number of variables. Two examples of this model are  $b(x) = \text{uniform}$  so that  $p(x) = g(x_R)$  and  $b(x) = \text{Normal}$  as in [Hjort and M.C.Jones \[1996\]](#) and [Hjort and Glad \[1995\]](#). In this chapter, We will consider two versions of the rodeo for density estimation problems: a local version and a global version. The local version estimates  $p(x)$  at a given point  $x$  and results in a local bandwidth selection method. The global version estimates  $p(x)$  at all  $x$  and results in a global bandwidth selection method.

The remaining part of this chapter is organized as follows: In section 2, we derived the local rodeo algorithm for both kernel density estimator and local likelihood estimator. The rodeo algorithm for a semiparametric model when  $b(x) = \text{Normal}$  is also shown. Section 3 and 4 describe the global rodeo algorithm and other variations. Section 5 uses both synthetic and real-world datasets to test our method. Section 6 specifies our main theoretical results about the asymptotic running time, selected bandwidths, and convergence

rate of the risk. The conclusions and more discussion are in section 7. All the proofs are given in the appendix.

### 3.2 THE LOCAL RODEO

Suppose first that the data are on the unit cube  $[0, 1]^d$  and  $b(x)$  is uniform. Let  $x$  be a  $d$ -dimensional target point at which we want to estimate  $p(x)$ . The kernel density estimator is

$$\hat{f}_H(x) = \frac{1}{n \det(H)} \sum_{i=1}^n \mathcal{K}(H^{-1}(x - X^{(i)})) \quad (3.2)$$

where  $\mathcal{K}$  is a symmetric kernel, such that  $\int \mathcal{K}(u) du = 1$ ,  $\int u \mathcal{K}(u) du = 0_d$  while  $\mathcal{K}_H(\cdot) = \frac{1}{\det(H)} \mathcal{K}(H^{-1} \cdot)$  and  $H = \text{diag}(h_1, \dots, h_d)$ . We assume that  $\mathcal{K}$  is a product kernel so

$$\hat{f}_H(x) = \frac{1}{n \det(H)} \sum_{i=1}^n \mathcal{K}(H^{-1}(x - X^{(i)})) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \frac{1}{h_j} K\left(\frac{x_j - X_j^{(i)}}{h_j}\right) \quad (3.3)$$

#### 3.2.1 The Kernel Density Estimator Version

The density rodeo is based on the following idea. We start with a bandwidth matrix  $H = \text{diag}(h_0, \dots, h_0)$  where  $h_0$  is large. We then compute test statistics ( $Z_j : 1 \leq j \leq d$ ) and we reduce bandwidth  $h_j$  if  $Z_j$  is large. The test statistic is

$$Z_j = \frac{\partial \hat{f}_H(x)}{\partial h_j} = \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial h_j} \left( \prod_{k=1}^d \frac{1}{h_k} K\left(\frac{x_k - X_k^{(i)}}{h_k}\right) \right) \equiv \frac{1}{n} \sum_{i=1}^n Z_{ji}. \quad (3.4)$$

Thus,  $|Z_j|$  is large if changing  $h_j$  leads to a substantial difference in the estimator. To carry out the test, we compare  $Z_j$  to its variance

$$\sigma_j^2 = \text{Var}(Z_j) = \text{Var}\left(\frac{1}{n} \sum_{i=1}^n Z_{ji}\right) = \frac{1}{n} \text{Var}(Z_{j1}) \quad (3.5)$$

We estimate  $\sigma_j^2$  with  $s_j^2 = v_j^2/n$  where  $v_j^2$  is the sample variance of the  $Z_{ji}$ 's. The algorithm is given in Figure 9. Some related methods also appeared in [Friedenberg and Genovese \[2009\]](#).

For a general kernel, we have that

$$Z_j = \frac{\partial \hat{f}_H(x)}{\partial h_j} \quad (3.6)$$

$$= -\frac{1}{n} \sum_{i=1}^n \left( \frac{1}{h_j} + \frac{x_j - X_j^{(i)}}{h_j^2} \tilde{K}\left(\frac{x_j - X_j^{(i)}}{h_j}\right) \right) \prod_{k=1}^d \frac{1}{h_k} K\left(\frac{x_k - X_k^{(i)}}{h_k}\right) \quad (3.7)$$

## DENSITY ESTIMATION RODEO

1. *Select* parameter  $0 < \beta < 1$  and initial  $h_0$ , which slowly decreases to zero:

$$h_0 = c_0 / \log \log n$$

for some constant  $c_0$ . Let  $c_n$  be a sequence satisfying  $c_n = O(\frac{\log n}{n})$ .

2. *Initialize* the bandwidths, and activate all dimensions:

$$(a) h_j = h_0, j = 1, 2, \dots, d.$$

$$(b) \mathcal{A} = \{1, 2, \dots, d\}.$$

3. *While*  $\mathcal{A}$  is *nonempty*, do for each  $j \in \mathcal{A}$

$$(a) \text{ Compute the estimated derivative } Z_j \text{ and variance } s_j^2.$$

$$(b) \text{ Compute the threshold } \lambda_j = s_j \sqrt{2 \log(nc_n)}.$$

$$(c) \text{ If } |Z_j| > \lambda_j, \text{ then set } h_j \leftarrow \beta h_j; \text{ otherwise remove } j \text{ from } \mathcal{A}.$$

4. *Output* bandwidths  $H^* = \text{diag}(h_1, \dots, h_d)$  and estimator  $\tilde{f}(x) = \hat{f}_{H^*}(x)$

Figure 9.: The density rodeo algorithm.

where  $\tilde{K}(x) = \frac{d \log K(x)}{dx}$ . In the case where  $K$  is Gaussian this becomes

$$Z_j = \frac{\partial \hat{f}_H(x)}{\partial h_j} \tag{3.8}$$

$$= \frac{1}{nh_j^3} \prod_{k=1}^d \frac{1}{h_k} \sum_{i=1}^n \left( (x_j - X_j^{(i)})^2 - h_j^2 \right) \prod_{k=1}^d K \left( \frac{x_k - X_k^{(i)}}{h_k} \right) \tag{3.9}$$

$$\propto \frac{1}{n} \sum_{i=1}^n \left( (x_j - X_j^{(i)})^2 - h_j^2 \right) \prod_{k=1}^d K \left( \frac{x_k - X_k^{(i)}}{h_k} \right) \tag{3.10}$$

$$= \frac{1}{n} \sum_{i=1}^n \left( (x_j - X_j^{(i)})^2 - h_j^2 \right) \exp \left\{ - \sum_{k=1}^d \frac{(x_k - X_k^{(i)})^2}{2h_k^2} \right\} \tag{3.11}$$

Here, the constant of proportionality  $\frac{1}{h_j^3} \prod_{k=1}^d \frac{1}{h_k}$  can be safely ignored to avoid overflow in the computation as  $h_k \rightarrow 0$  for large  $d$ .

### 3.2.2 The Local Likelihood Version

Hjort and M.C.Jones [1996], Hjort and Glad [1995], Loader [1996] formulate the local likelihood density estimation problem as:

$$\max_{\theta} \mathcal{L}(f, x) = \sum_{i=1}^n \mathcal{K} \left( H^{-1}(X^{(i)} - x) \right) \log f(X^{(i)}; \theta) - n \int_{\mathcal{X}} \mathcal{K} \left( H^{-1}(u - x) \right) f(u; \theta) du \quad (3.12)$$

which is a localized version of the usual loglikelihood function for density estimation problems:

$$\max_{\theta} \mathcal{L}(f, x) = \sum_{i=1}^n \log f(X^{(i)}; \theta) - n \left( \int_{\mathcal{X}} f(u; \theta) du - 1 \right) \quad (3.13)$$

Since the true density function  $p(x)$  is unknown, a polynomial is used to approximate the log density. The large-sample properties of the local likelihood estimator are parallel to those of local polynomial regression. The most appealing property of the resulting estimator is its good performance when facing boundary effects [Loader, 1996]. When assuming a product Gaussian kernel, the closed form of the local likelihood estimator can be written as

$$\tilde{f}_H(x) = \hat{f}_H(x) \exp \left\{ -\frac{1}{2} \sum_{k=1}^d h_k^2 \left( \frac{\sum_{i=1}^n \prod_{j=1}^d K \left( \frac{X_j^{(i)} - x_j}{h_j} \right) \left( \frac{X_k^{(i)} - x_k}{h_k^2} \right)}{\sum_{i=1}^n \prod_{j=1}^d K \left( \frac{X_j^{(i)} - x_j}{h_j} \right)} \right)^2 \right\} \quad (3.14)$$

which can be viewed as a standard kernel density estimator  $\hat{f}_H(x)$  multiplied by an exponential bias correction term. To evaluate  $Z_m = \frac{\partial \hat{f}_H(x)}{\partial h_m}$ ,  $m = 1, \dots, d$ , define

$$\hat{g}_k(x) = \frac{\partial}{\partial x_k} \hat{f}_H(x) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d \frac{1}{h_j} K \left( \frac{X_j^{(i)} - x_j}{h_j} \right) \left( \frac{X_k^{(i)} - x_k}{h_k^2} \right) \quad (3.15)$$

Then

$$\begin{aligned} Z_m &= \frac{\partial}{\partial h_m} \left( \hat{f}_H(x) \exp \left\{ -\frac{1}{2} \sum_{k=1}^d h_k^2 \left( \frac{\hat{g}_k(x)}{\hat{f}_H(x)} \right)^2 \right\} \right) \\ &= \tilde{f}_H(x) \left( \frac{\partial}{\partial h_m} \log \hat{f}_H(x) \right) + \tilde{f}_H(x) \frac{\partial}{\partial h_m} \left( -\frac{1}{2} \sum_{k=1}^d h_k^2 \left( \frac{\hat{g}_k(x)}{\hat{f}_H(x)} \right)^2 \right) \end{aligned} \quad (3.16)$$

where  $\frac{\partial}{\partial h_m} \log \hat{f}_H(x) = \frac{\frac{\partial}{\partial h_m} \hat{f}_H(x)}{\hat{f}_H(x)}$  has been calculated in the previous section. The derivation of the second term, though quite involved, is straightforward. The same algorithm in figure 9 applies.

### 3.2.3 Other Baseline Densities

When using a different baseline (i.e. the Normal distribution), we use the semiparametric density estimator

$$\bar{f}_H(x) = \frac{\hat{b}(x) \sum_{i=1}^n \mathcal{K}_H(X^{(i)} - x)}{n \int \mathcal{K}_H(u - x) \hat{b}(u) du} \quad (3.17)$$

where  $\hat{b}(x)$  is a parametric density estimator at point  $x$ , its parameters are estimated by maximum likelihood. Since the parameters in the parametric form are easy to estimate, we treat them as known. The motivation of this estimator comes from the local likelihood method in equation (3.12): instead of using a polynomial  $P(x)$  to approximate the log density  $\log p(x)$ , we use  $\log b(x) + P(x)$ . Under this setting, we see that starting from a large bandwidth, if the true function is  $b(x)$ , the algorithm will tend to freeze the bandwidth decaying process for the estimator defined in expression (3.17).

Suppose that  $b(x)$  is a multivariate normal density function with a diagonalized variance-covariance matrix  $\Sigma$ . When we use the product Gaussian kernels with bandwidth matrix  $H$ , a closed form estimator can be derived as

$$\begin{aligned} \bar{f}_H(x) = & \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d K \left( \frac{X_j^{(i)} - x_j}{h_j} \right) \\ & \times \sqrt{\frac{|H + \hat{\Sigma}|}{|\hat{\Sigma}|}} \exp \left\{ -\frac{(x - \hat{\mu})^T (\hat{\Sigma}^{-1} - (H + \hat{\Sigma})^{-1}) (x - \hat{\mu})}{2} \right\} \end{aligned} \quad (3.18)$$

where  $\hat{\mu}$  and  $\hat{\Sigma}$  are the M.L.E. for the normal distribution. More details about the derivation of this closed form are given in the appendix. It's easy to see that the local likelihood estimator is a special case of this semiparametric estimator when  $b(x) = \text{uniform}$ . The partial derivative of  $\bar{f}_H(x)$  with respect to the bandwidth  $h_m$  ( $m = 1, \dots, d$ ) is calculated as

$$\begin{aligned} Z_m = \frac{\partial \bar{f}_H(x)}{\partial h_m} = & \sqrt{\prod_{j=1}^d \left( 1 + \frac{h_j^2}{\hat{\sigma}_j^2} \right)} \\ & \times \exp \left\{ \sum_{j=1}^d \left( -\frac{(x_j - \hat{\mu}_j)^2}{2 (\hat{\sigma}_j^2 (\hat{\sigma}_j^2 + h_j^2) / h_j^2)} \right) \left( \frac{\partial \bar{f}_H(x)}{\partial h_m} + M \bar{f}_H(x) \right) \right\}, \end{aligned}$$

where

$$M = \frac{h_m (2(\hat{\sigma}_m^2 + h_m^2) + (x_m - \hat{\mu}_m)^2)}{2(\hat{\sigma}_m^2 + h_m^2)^2} \quad (3.19)$$

and  $\hat{f}_H(x)$  is the standard kernel density estimator as defined in equation (3.3). The variance of  $Z_m$  is estimated using the bootstrap method (see section 3.4.1).

### 3.3 THE GLOBAL RODEO

Instead of using the local rodeo which corresponds to the adaptive density estimation, the idea could be easily extended to carry out global bandwidth selection, in which case each dimension uses a fixed bandwidth. The idea is to average the test statistics for multiple evaluation points  $x_1, \dots, x_k$ , which could be sampled from the empirical distribution of the observed sample points.

As has been pointed out by [Lafferty and Wasserman \[2008\]](#), averaging the  $Z_j$ s directly leads to a statistic whose mean for relevant variables is asymptotically  $\frac{1}{k}h_j \sum_{i=1}^k p_{jj}(x_i)$ . Because of sign changes in  $p_{jj}(x)$ , cancellations can occur resulting in a small value for the statistics. To avoid this problem, the statistic is squared. Let  $x_1, \dots, x_m$  denote the evaluation points and  $Z_j(x_i)$  represents the derivative for the  $i$ -th evaluation point with respect to the bandwidth  $h_j$ . Therefore

$$Z_j(x_i) = \frac{1}{n} \sum_{k=1}^n Z_{jk}(x_i), \quad i = 1, \dots, m, \quad j = 1, \dots, d. \quad (3.20)$$

Let  $\gamma_{jk} = (Z_{j1}(x_k), Z_{j2}(x_k), \dots, Z_{jm}(x_k))^T$  ( $k = 1, \dots, n$ ). Assuming that  $\text{Var}(\gamma_{jk}) = \Sigma_j$ , denote  $Z_j = (Z_{j1}, Z_{j2}, \dots, Z_{jm})^T$ , by the multivariate central limit theorem, we know that  $\text{Var}(Z_j) = \Sigma_j/n \equiv C_j$ . Based on these derivations, we define the test statistic

$$T_j = \frac{1}{m} \sum_{k=1}^m Z_j^2(x_k), \quad j = 1, \dots, d, \quad (3.21)$$

while

$$s_j = \sqrt{\text{Var}(T_j)} = \frac{1}{m} \sqrt{\text{Var}(Z_j^T Z_j)} = \frac{1}{m} \sqrt{2\text{tr}(C_j^2) + 4\hat{\mu}_j^T C_j \hat{\mu}_j} \quad (3.22)$$

where  $\hat{\mu} = \frac{1}{m} \sum_{i=1}^m Z_j(x_i)$ . For the irrelevant dimension  $j \in R^c$ , as will be shown in section 3.6,  $\mathbb{E}Z_j(x_i) = o_P(h_j)$ , so that  $\mathbb{E}T_j \approx \text{Var}(Z_j(x_i))$ . We use  $s_j^2$  as an estimate for  $\text{Var}(Z_j(x_i))$ . Therefore, we take the threshold to be

$$\lambda_j = s_j^2 + 2s_j \sqrt{\log(nc_n)}. \quad (3.23)$$

Several examples of this algorithm and its comparison with the other algorithms are given in the experiment section, the theoretical properties of the global rodeo estimator can be analyzed in a way that is similar to the local version.

### 3.4 DIFFERENT EXTENSIONS

#### 3.4.1 Bootstrap Version

For the previous examples, the explicit closed-form expression for the  $Z_j$  and  $s_j^2$  can be easily derived due to the existence of a closed form for the targeted

density estimator. Sometimes, the density estimator  $\hat{f}_H(x)$  might not have a closed form expression. In these cases, we could still numerically evaluate the derivative  $Z_j$  as

$$Z_j = \frac{\hat{f}_{H+\Delta h_j}(x) - \hat{f}_H(x)}{\Delta h_j} \quad (3.24)$$

where  $H + \Delta h_j$  means adding a small value  $\Delta h_j$  on the  $j$ -th diagonal element of  $H$ . One thing to note is that there exist more sophisticated methods to estimate of  $Z_j$  in a numerically more stable way. In general, a sensitivity analysis may be needed to determine which methods to use. The variance of  $Z_j$  can be calculated by bootstrap, the algorithm is given in figure 10

---

THE BOOTSTRAP METHOD TO CALCULATE  $s_j^2$

---

1. Draw a sample  $X^{(1)*}, \dots, X^{(n)*}$  of size  $n$ , with replacement:

Repeat  $B$  times for the following

Compute the estimate  $Z_{ji}^*$  from data  $X^{(1)*}, \dots, X^{(n)*}, i = 1, \dots, B$ .

2. Compute the bootstrapped variance

$$s_j^2 = \frac{1}{B} \sum_{b=1}^B (Z_{ji}^* - \bar{Z}_{j\cdot})^2. \text{ where } \bar{Z}_{j\cdot} = \frac{1}{B} \sum_{b=1}^B \hat{Z}_j^*$$

3. Output the resulted  $s_j^2$ .
- 

Figure 10.: Density Rodeo: the bootstrap method to calculate the  $s_j^2$

This bootstrap version works for both local and global rodeo algorithms, thus provides a more general framework. We expect that similar analytic results will hold. However, bootstrap needs more computation. In cases that the analytic form of the variance is hard to evaluate, like the local likelihood rodeo and the semiparametric rodeo, this method applies.

### 3.4.2 Reverse Rodeo

The previous rodeo algorithms use a sequence of decreasing bandwidths and estimates the optimal value by a sequence of hypothesis testing. On the contrary, we could begin from a very small bandwidth, and use a sequence of increasing bandwidths to estimate the optimal value. This reversed version does not share the same theoretical property as before, but it's useful in some special cases (i.e. many dimensions need a small bandwidths, while only a few need large bandwidths). More details will be given in an image processing experiment in the next section.



### 3.5 EXAMPLES

In this section, we applied the rodeo algorithm on both synthetic and real data, including one-dimensional, two-dimensional, high-dimensional and very high-dimensional examples to investigate how it performs in various conditions. For the purpose of evaluating the algorithm performance quantitatively, we need some criterion to measure the distance between the estimated density function  $\hat{f}(x)$  with the true density  $p(x)$ . For this, we use the Hellinger distance, defined as

$$D(\hat{f}||p) = \int \left( \sqrt{\hat{f}(x)} - \sqrt{p(x)} \right)^2 dx = 2 - 2 \int p(x) \sqrt{\frac{\hat{f}(x)}{p(x)}} dx \quad (3.25)$$

Assuming we have  $m$  evaluation points, the hellinger distance could be numerically calculated by the Monte Carlo integration

$$D(\hat{f}||p) \approx 2 - \frac{2}{m} \sum_{i=1}^m \sqrt{\frac{\hat{f}_H(X^{(i)})}{p(X^{(i)})}} \quad (3.26)$$

Since this measure utilizes the property that  $p(x)$  is a density function, it's expected to be numerically more stable than the commonly used Kullback-Leibler (KL) divergence as a loss function for evaluating the discrepancy between two density functions. In the following, we first use the simulated data, about which we have known the true distribution function, to investigate the algorithm performance. Then our algorithm is also applied on some real data for analysis and comparison. In the following experiments, if not stated explicitly, the data is always rescaled into a  $d$ -dimensional cube  $[0, 1]^d$ , a product Gaussian kernels are used, the default parameters are  $c_0 = 1$ ,  $c_n = d \log n/n$ , and  $\beta = 0.9$ .

#### 3.5.1 One-dimensional Examples

First, we apply the rodeo algorithm on one dimensional examples. We have conducted a series of comparative study on a list of 15 "test densities" proposed by [Marron and Wand \[1992\]](#), which are all normal mixtures representing many different types of challenges to density estimation methods. Our approach achieves a comparable performance to the built-in kernel density estimator with bandwidth selected by unbiased cross-validation (from the base library of R). Due to the space consideration, only the strongly skewed example is reported here, since it demonstrates the advantage of adaptive bandwidth selection for the local rodeo algorithm.

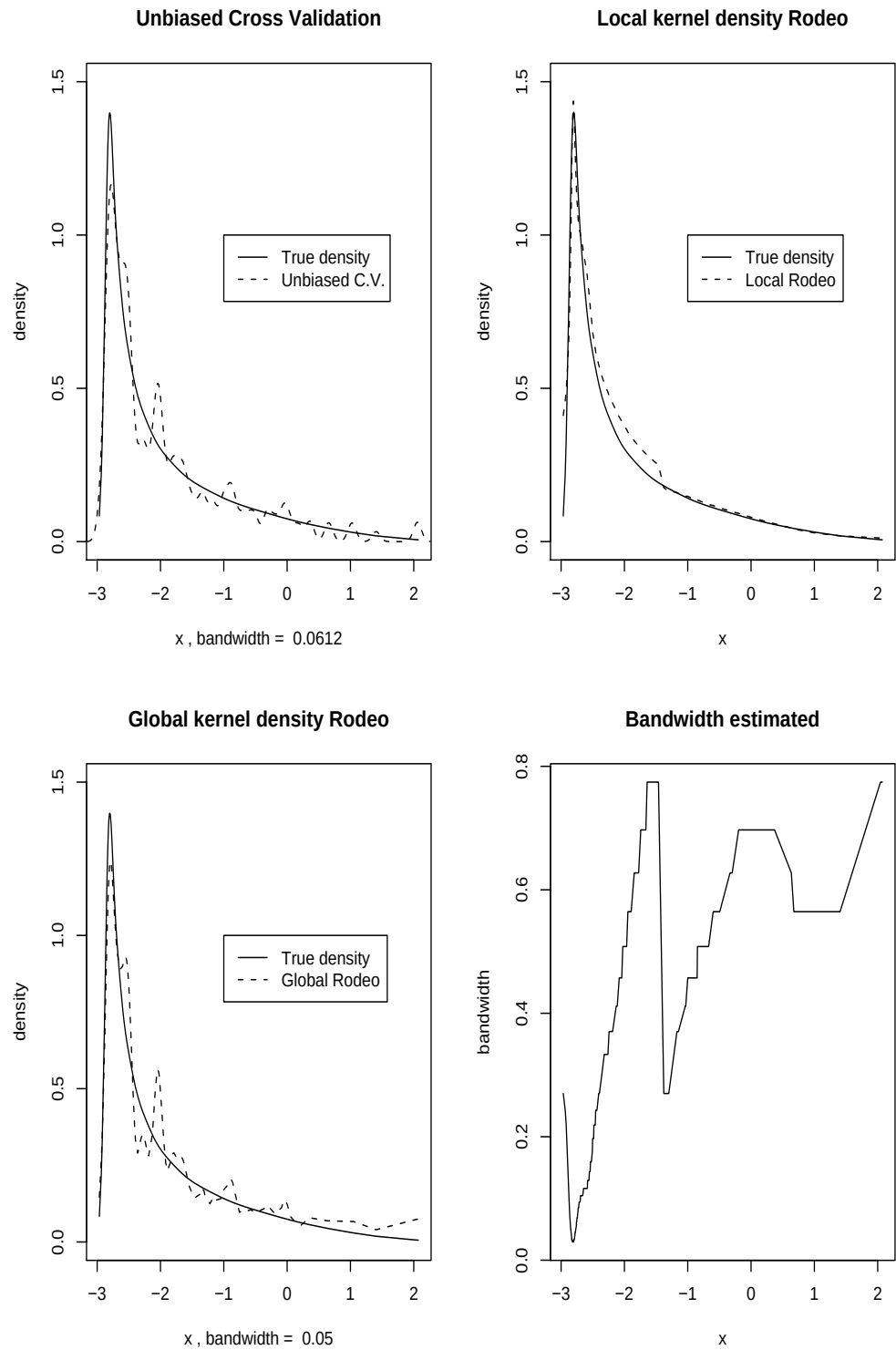


Figure 11.: Different versions of the density Rodeo algorithms run on the highly skewed unimodal example. The first three plots are results for the different estimators, the last one is the fitted bandwidths for the local rodeo.

**Example 1** (Strongly skewed density): This density is chosen to resemble the lognormal distribution, it distributes as

$$X \sim \sum_{i=0}^7 \frac{1}{8} N \left( 3 \left( \left( \frac{2}{3} \right)^i - 1 \right), \left( \frac{2}{3} \right)^{2i} \right). \quad (3.27)$$

200 samples were generated from this distribution, The estimated density functions by the local rodeo, the global rodeo, and the built-in kernel density estimator with bandwidth chosen by unbiased cross validation are shown in figure 11. In which, the solid line is the true density function, the dashed line illustrates the estimated densities by different methods. The local rodeo works the best, this is because the true density function is highly skewed, the fixed bandwidth density estimator fails to fit the very smooth tail. The last subplot from figure 11 illustrates the selected bandwidth for the local rodeo, it illustrates how smaller bandwidths are selected where the function is more rapidly varying.

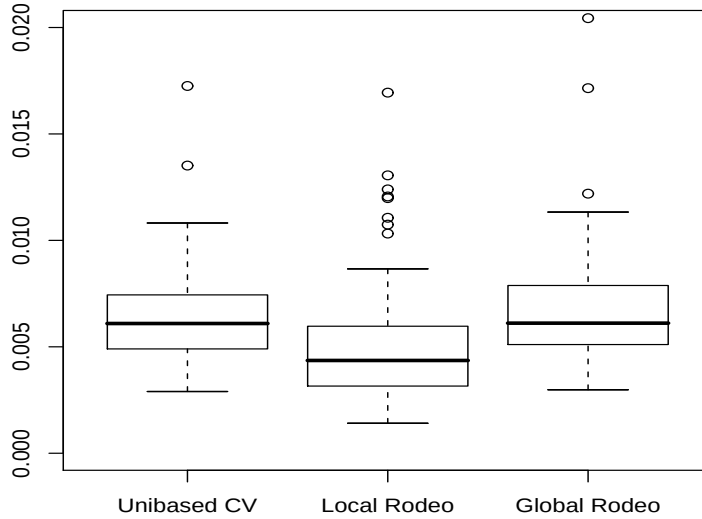


Figure 12.: Density Rodeo Experiments on data from Highly skewed unimodal distribution: The boxplots of the empirical Heillinger's losses on test samples of estimated densities by the three methods based on 100 simulations.

Figure 12 shows the distribution of the empirical Hellinger distances based on 100 simulations. The boxplots show that the local rodeo works the best, while the global rodeo and the unbiased cross-validation methods are comparable in this one dimensional example.

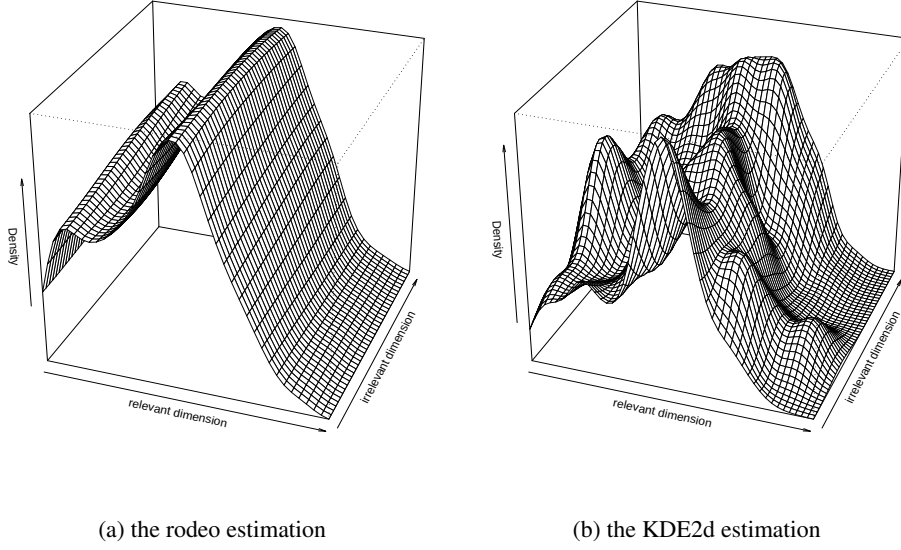


Figure 13.: Density Rodeo: Perspective plots of the estimated density functions by the global density rodeo (left) and the R built-in method KDE2d (right) on a 2-dimensional synthetic data.

### 3.5.2 Two dimensional Examples

We also show some 2-dimensional examples, since they are easy to visualize. One uses a synthetic dataset, the other one uses some real data analyzed by the other authors. The density rodeo's performance is compared with a built-in method named KDE2d ( from MASS package in R ). The empirical results show that the rodeo algorithm works better than the built-in method on the synthetic data, where we know the ground truth. For the real-world dataset, where we do not know the underling density, our method achieves a very similar result as those of the previous authors.

**Example 2:** (Combined Beta distribution with the uniform distribution as irrelevant). We simulate a 2-dimensional dataset with  $n = 500$  points. The two dimensions are independently generated as

$$X_1 \sim \frac{2}{3}\text{Beta}(1,2) + \frac{1}{3}\text{Beta}(10,10) \quad (3.28)$$

$$X_2 \sim \text{Uniform}(0,1) \quad (3.29)$$

Figure 13 illustrates the perspective plots of the estimated density functions by the global rodeo and the built-in method KDE2d. From which, we see that the global rodeo fits the irrelevant uniform dimension perfectly, while KDE2d fails. For a quantitative comparison, we evaluated the empirical Hellinger distance

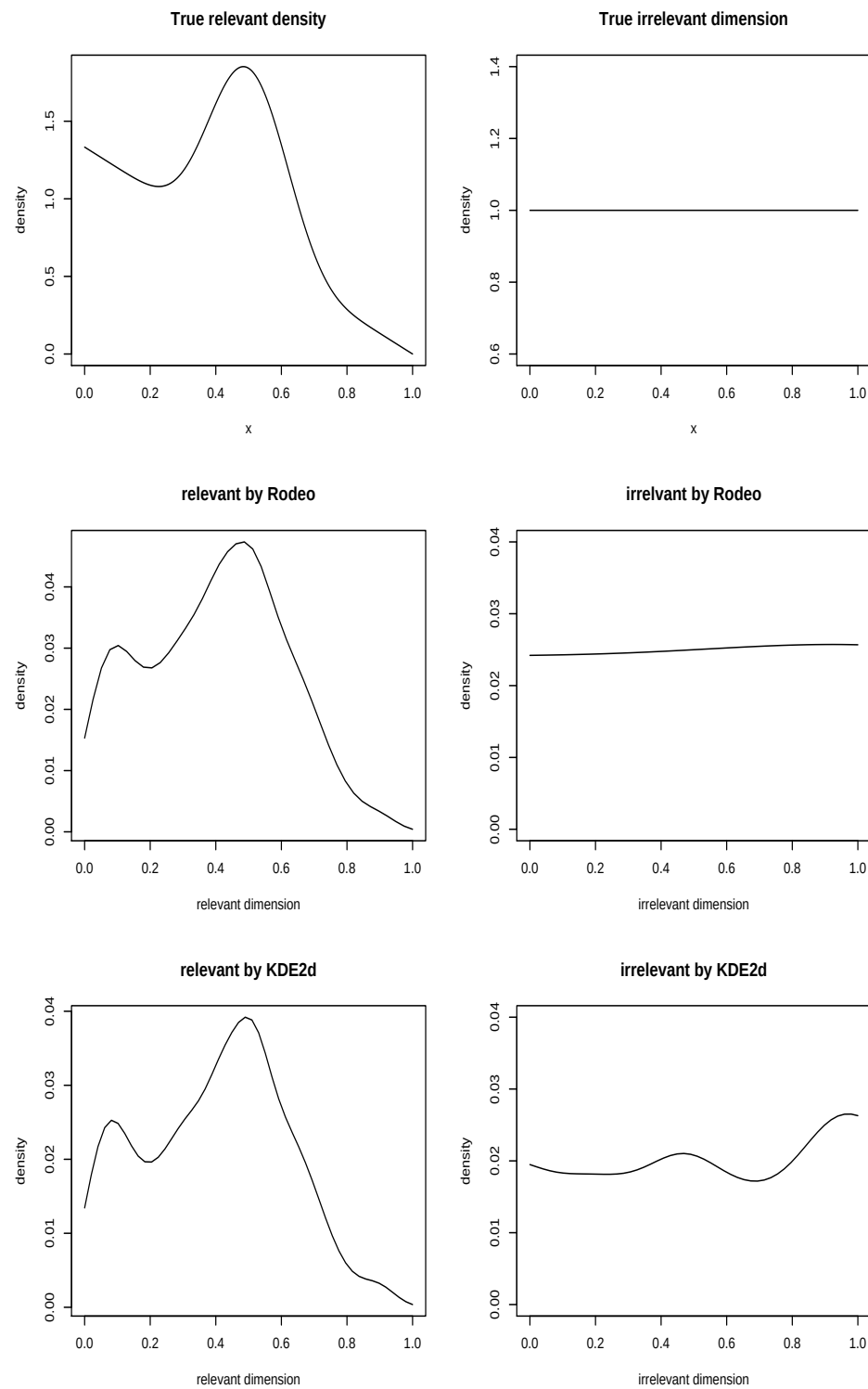


Figure 14.: Marginal distributions of the relevant and the irrelevant dimensions for example 2

between the estimated density and the true density, the global rodeo algorithm outperforms KDE2d uniformly on this example. For a qualitative comparison, figure 14 illustrates the numerically integrated marginal distributions of the two estimators (not normalized). Even with an eye examination, we see that the rodeo's result is better than that of KDE2d, which is consistent with the previous observations.

**Example 3:** (Geyser data). For this example, a real dataset is used. Which is a version of the eruptions data from the "Old Faithful" geyser in Yellowstone National Park, Wyoming. This version comes from Azzalini and Bowman [A.Azzalini and A.W.Bowman \[1990\]](#) and is of continuous measurement from August 1 to August 15, 1985. There are two variables with 299 observations altogether. The first variable, "Duration", represents the numeric eruption time in minutes. The second variable, "waiting", represents the waiting time to next eruption. We apply the global rodeo algorithm on this dataset. The estimated density functions of the rodeo algorithm and the built-in KDE2d method (used by the original authors) are provided in the upper of figure 15. And lower two plots of figure 15 illustrates the corresponding contour plots. Based on a visual examination, our method achieves a very similar estimation as those provided by the previous authors who analyzed this data before.

### 3.5.3 High Dimensional Examples

**Example 4:** (High dimensional case) Figure 16 illustrates the output bandwidths from the local rodeo for a 30-dimensional synthetic dataset with  $r = 5$  relevant dimensions ( $n = 100$ , with 30 trials). The relevant dimensions are generated as

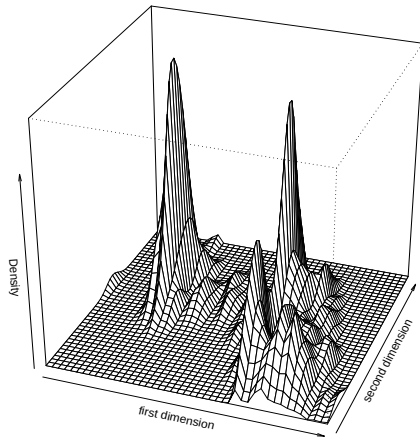
$$X_i \sim N(0.5, (0.02i)^2), \quad \text{for } i = 1, \dots, 5. \quad (3.30)$$

while the irrelevant dimensions are generated as

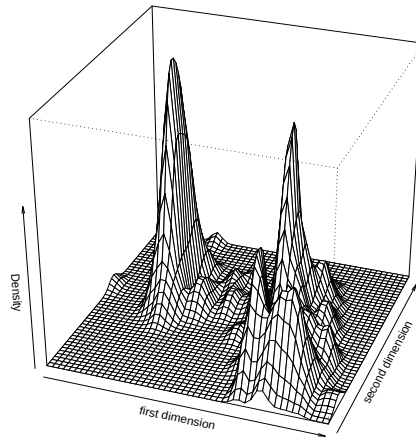
$$X_i \sim \text{Uniform}(0, 1), \quad \text{for } i = 6, \dots, 30. \quad (3.31)$$

The evaluation point is  $x = (\frac{1}{2}, \dots, \frac{1}{2})$ . The boxplot illustrates the selected bandwidths out of 30 trials. The plot shows that the bandwidths of the relevant dimensions shrink towards zero, while the bandwidths of the irrelevant dimensions remain large, indicates that the algorithm's performance is consistent with our analysis. Also, from the bandwidth plot, we see that, for the relevant dimensions, the smaller the variance is, the smaller the estimated bandwidth will be.

**Example 5:** (Image processing). Here we apply the reverse local rodeo on image data. The results are shown in figure 17. The algorithm was run on 1100 grayscale images of digital letter 2s, each with  $256 = 16 \times 16$  pixels



(a) the rodeo estimation



(b) the KDE2d estimation

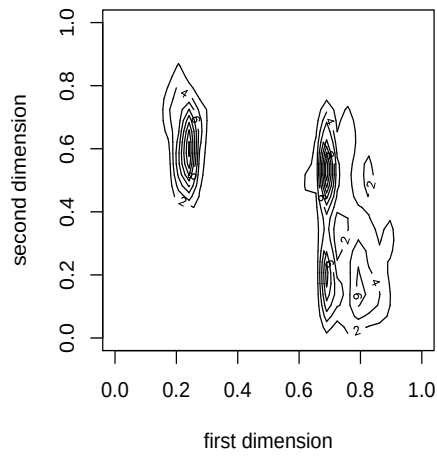
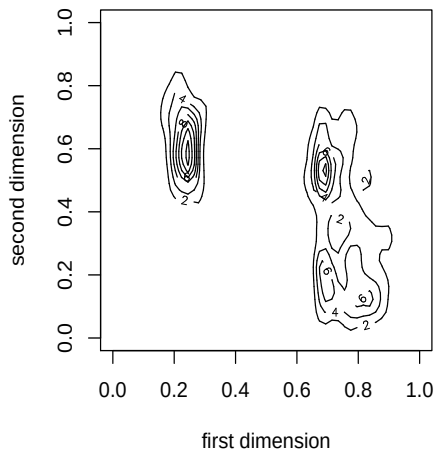


Figure 15.: Density rodeo experiments on the geyser data. Upper: Perspective plots of the estimated density functions by the global rodeo (left) and the R built-in method KDE2d (right) on the geyser data. Lower: Contour plots of the result from the global rodeo (left) and KDE2d (right)

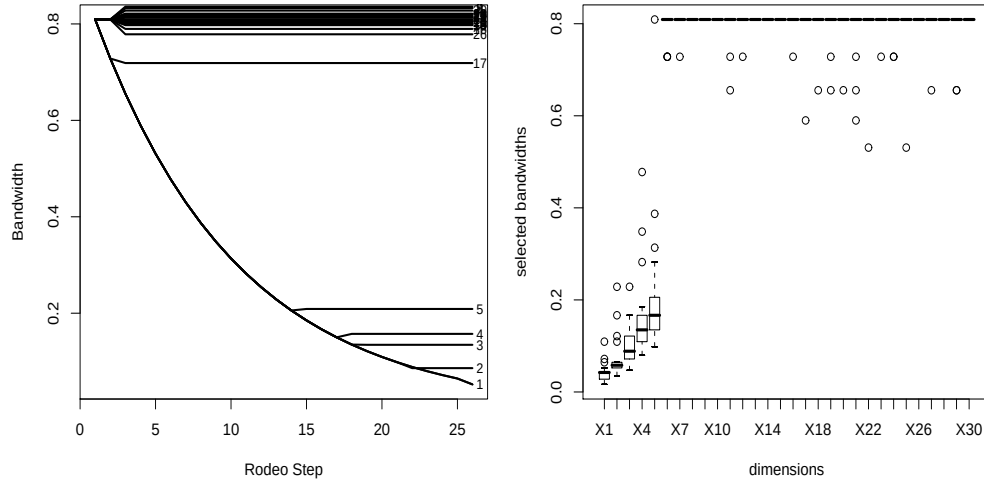


Figure 16.: The bandwidth output by the local density rodeo for a 30-dimensional synthetic dataset (Left) and its boxplot for 30 trials. (Right)

with some unknown background noise; thus this is a 256-dimensional density estimation problem. An evaluation point is shown in the upper left subplot of figure 17, and the bandwidths output by the rodeo algorithm is shown in the upper right subplot. The estimated bandwidth plots in different rodeo steps (step 10, 20, 40, 60, and 100) are shown in the lower series of plots— smaller bandwidths have darker colors, the pixels with larger bandwidth are more informative than those with smaller bandwidths. This is a good example to illustrate the usefulness of the reverse rodeo. For the image data, many background pixels have a density close to point mass, which will pin down the bandwidth to a very small value. The reverse rodeo starts from a small bandwidths, which is more efficient than the original rodeo and is expected to be numerically more stable. Figure 17 visualizes the evolution of the bandwidths and could be viewed as a dynamic process for feature selection — the earlier a dimension’s bandwidth increases, the more informative it is. The reverse rodeo algorithm is quite efficient for this extremely high-dimensional problem. One interesting thing to note is, the early stages of the rodeo reveal that some of the 2s in the data have looped bottoms, while some have straight bottoms; the evaluation point does not have such a loop. This might because in the original dataset, some 2s have this loop while the others not. The density rodeo algorithm could discover these kind of characteristics automatically.



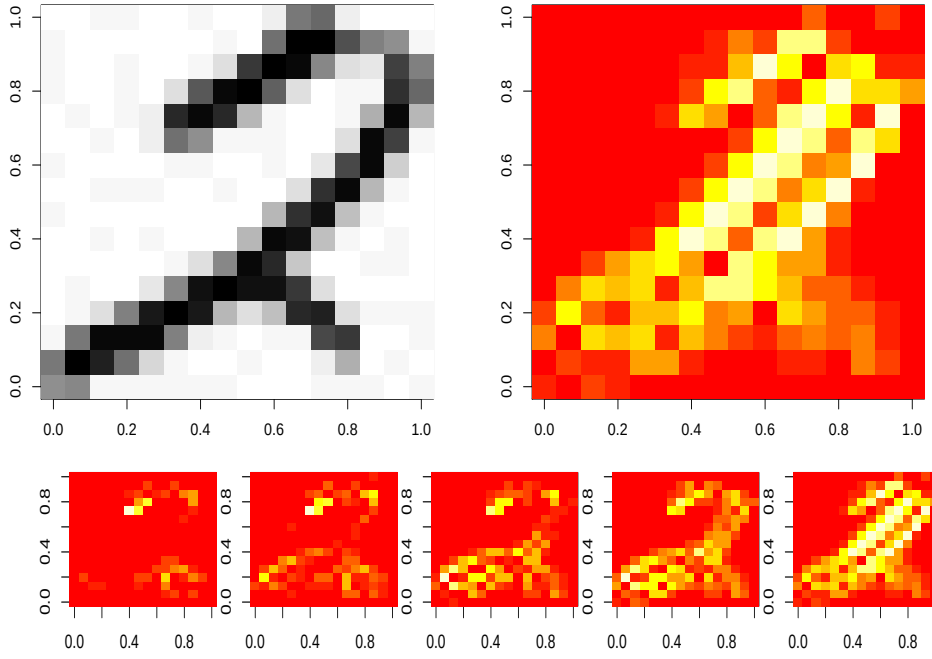


Figure 17.: density rodeo on the image data: the upper plots are the evaluation digit and the bandwidths output by the reverse rodeo. The lower subplots illustrate a series of bandwidth plots sampled at different rodeo steps: 10, 20, 40, 60, and 100

#### 3.5.4 Using Other Baseline Densities

**Example 6:** (Using normal distributions as the irrelevant dimensions) Figure 18 illustrates the output bandwidths from the semiparametric rodeo (developed in section 3.4) for both 15-dimensional and 20-dimensional synthetic datasets with  $r = 5$  relevant dimensions ( $n = 1000$ ). When using normal distributions as irrelevant dimensions, the relevant dimensions are generated as

$$X_i \sim \text{Uniform}(0,1), \quad \text{for } i = 1, \dots, 5. \quad (3.32)$$

while the irrelevant dimensions are generated as

$$X_i \sim N(0.5, (0.05i)^2), \quad \text{for } i = 6, \dots, d. \quad (3.33)$$

The evaluation point is  $x = (\frac{1}{2}, \dots, \frac{1}{2})$ . Even when normal distributions are used as irrelevant dimensions, the result is similar as before, showing that the bandwidths of the relevant dimensions shrink toward zero, while the bandwidths of the irrelevant dimensions remain large, this is just what we expected.

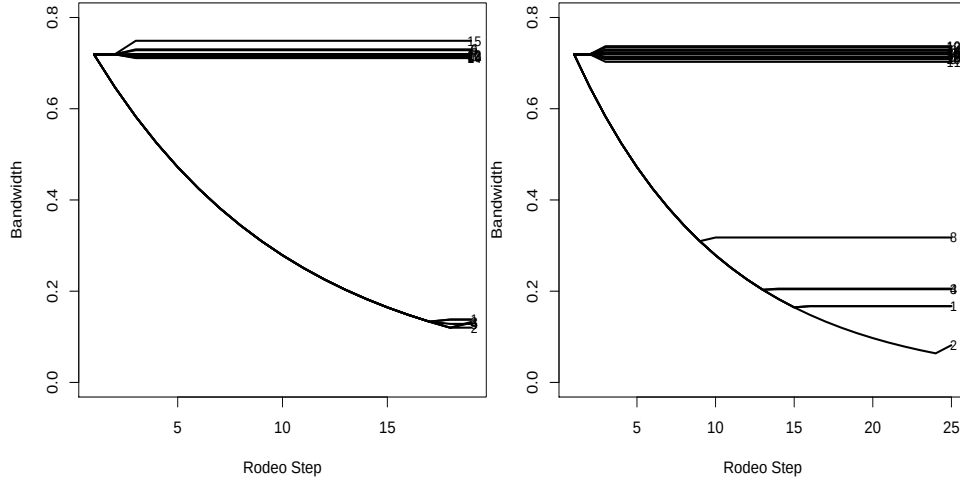


Figure 18.: The bandwidth output by the local semiparametric rodeo for a 15-dimensional synthetic dataset (Left) and a 20-dimensional synthetic dataset (Right). Using Gaussian distribution as the irrelevant dimensions

**Example 7:** ( The semiparametric density estimator for one dimensional problem ) For the illustration purpose, we also applied the semiparametric rodeo algorithm on a dimensional example. We simulated 1000 one-dimensional data points with  $X_i \sim \text{Uniform}(0, 1)$ . With  $\beta = 0.9$ , the results of the semiparametric rodeo algorithm are shown in figure 19. The first plot shows the true density function, the second plot is the estimated density function, the lower left plot illustrates the estimated bandwidths at different evaluation points, the last one is the estimated density function by the kernel density estimator with bandwidth selected by unbiased cross validation. Based on a visual examination of the results, we see that the density function estimated by the semiparametric rodeo is quite similar to that estimated by the kernel density estimator with unbiased cross validation. However, the selected bandwidths are quite small in this case ( $\approx 0.015$ ). Since the true density is uniform, smaller bandwidths are needed to correct the assumed normal density.

### 3.6 THEORETICAL PROPERTIES

Here we show the asymptotic properties of the resulting estimator when assuming the baseline component  $b(x)$  is a uniform distribution function. Our main theoretical results characterize the asymptotic running time, selected

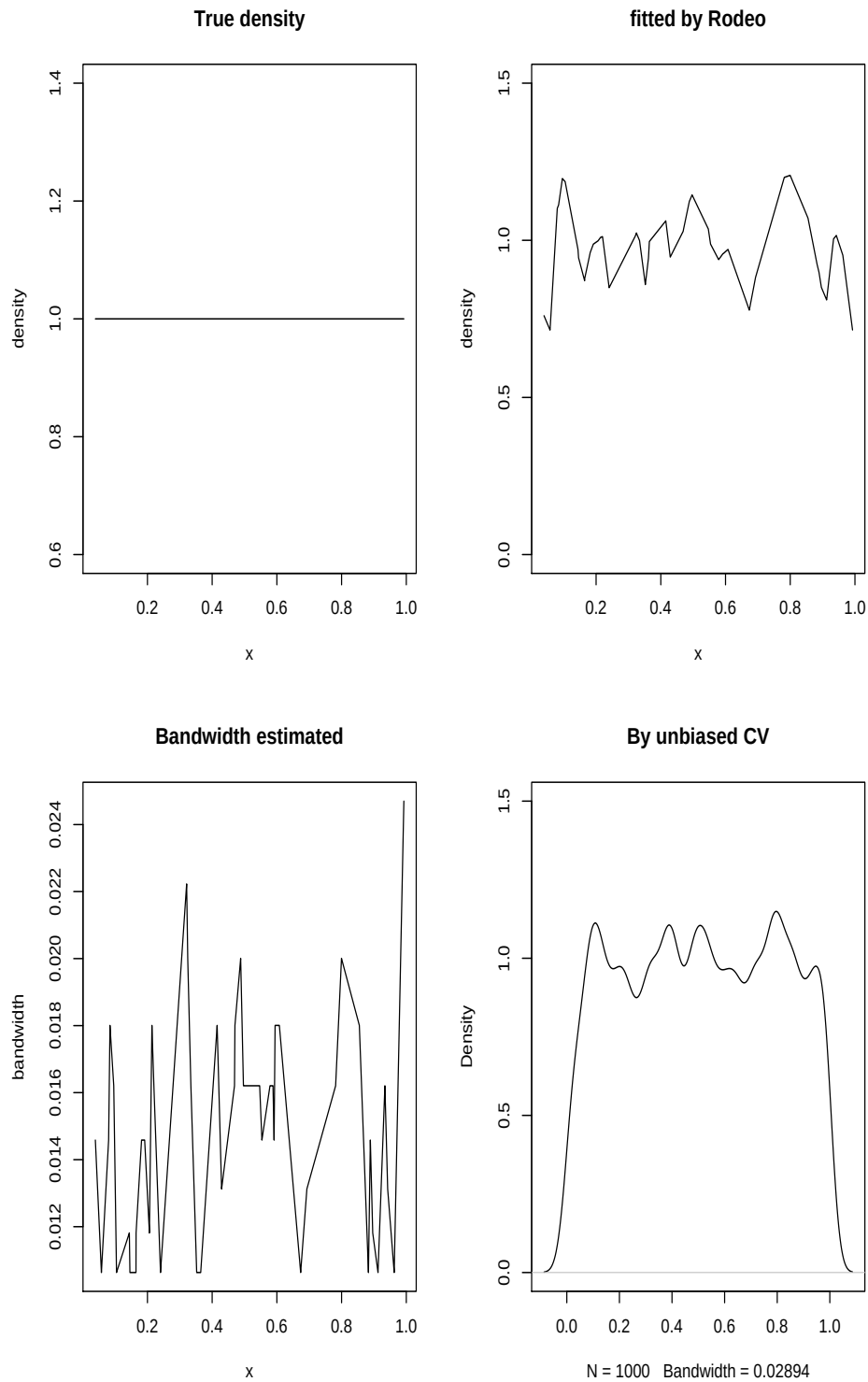


Figure 19.: The density rodeo results for fitting the uniform distribution with the semiparametric rodeo. The first plot shows the true density , the second plot is the estimated density, the lower left plot illustrates the estimated bandwidths at different evaluation points, the last one is the estimated density function by the KDE with bandwidth selected by cross validation

bandwidths and the risk of the resulting estimator. To simplify notations, in the sequel we denote the true density  $p(x)$  by  $f(x)$ .

We assume that the underlying density function  $f$  has continuous second order derivatives in a neighborhood of  $x$ . For convenience of notation, the dimensions are numbered such that the relevant variables  $x_j$  correspond to  $1 \leq j \leq r$  and the irrelevant variables  $x_j$  correspond to  $r+1 \leq j \leq d$ . We write  $Y_n = \tilde{O}_P(a_n)$  to mean that  $Y_n = O(b_n a_n)$  where  $b_n$  is logarithmic in  $n$ . As noted earlier, we write  $a_n = \Omega(b_n)$  if  $\liminf_n \left| \frac{a_n}{b_n} \right| > 0$ ; similarly  $a_n = \tilde{\Omega}(b_n)$  if  $a_n = \Omega(b_n c_n)$  where  $c_n$  is logarithmic in  $n$ . Also, let  $\mathcal{H}_f(x)$  denote the Hessian matrix of  $f(x)$ , let  $h_j^{(s)}$  denote the  $j^{\text{th}}$  bandwidth at step  $s$  and denote the bandwidth matrix by  $H^{(s)} = \text{diag}(h_1^{(s)}, \dots, h_d^{(s)})$ . In the following, we assume that the data lines in a unit cube  $[0, 1]^d$ .

We list the assumptions needed to establish the main result.

**Assumption 3.1. (A1) Kernel assumption:** assuming that  $\mathcal{K}$  is a bounded symmetric kernel, s.t.  $\int \mathcal{K}(u) du = 1$ ,  $\int u \mathcal{K}(u) du = 0_d$  while  $\mathcal{K}_H(\cdot) = \frac{1}{\det(H)} \mathcal{K}(H^{-1} \cdot)$  represents the kernel with bandwidth matrix  $H = \text{diag}(h_1, \dots, h_d)$ . then

$$\int uu^T \mathcal{K}(u) du = v_2 I_d \text{ and } v_2 < \infty \quad (3.34)$$

$$\int \mathcal{K}^2(u) du = R(\mathcal{K}) < \infty. \quad (3.35)$$

We also assume that there exist  $C_{\mathcal{K}}, C_d < \infty$  such that

$$\sup_u |\mathcal{K}(u)| < C_{\mathcal{K}} \text{ and } \sup_u \left| \frac{d \log \mathcal{K}(u)}{du} \right| < C_d.$$

**Assumption 3.2. (A2) Initial bandwidth assumption:** Let  $h_j^{(0)}$  denotes the initial bandwidth for the  $j$ -th dimension. Then,

$$h_j^{(0)} = \frac{c_0}{\log \log n} \text{ for } (j = 1, \dots, d). \quad (3.36)$$

**Assumption 3.3. (A3) Sparsity assumption:** Assuming that  $f(x)$  could be factorized into two components,  $f(x) \propto g(x_1, \dots, x_r) b(x)$ , where  $b_{jj}(x) = 0$  for  $j = 1, \dots, d$ .

In this section, we only consider the case when  $b(x) = 1$ .

**Assumption 3.4. (A4) Hessian assumption:** Let  $\mathcal{H}_R(x)$  denotes the Hessian matrix of all the relevant dimensions  $j \leq r$ .  $\text{diag}(\mathcal{H}_R(x))$  is a continuous vector and

$$\int \text{tr}(\mathcal{H}_R^T(u) \mathcal{H}_R(u)) du < \infty \quad (3.37)$$

$$\liminf_n \min_{1 \leq j \leq r} |f_{jj}(x)| > 0. \quad (3.38)$$

**Lemma 3.1.** *Under Assumptions A3.1 – A3.4, Let  $x$  be interior to the support of  $f$  and let  $\mathcal{H}_R(x)$  denote the Hessian matrix of all the relevant dimensions  $j \leq r$ . Then, over different steps in the algorithm and over  $j$ , we have*

$$\mathbb{E}\hat{f}_{H^{(s)}}(x) = f(x) + \frac{1}{2}v_2 \text{tr}((H^{(s)})^T \mathcal{H}_R^{(s)}(x)H) + o_P(\text{tr}((H^{(s)})^T H^{(s)})) \quad (3.39)$$

and

$$\text{Var}(\hat{f}_{H^{(s)}}(x)) = \frac{1}{n \det(H^{(s)})} R(\mathcal{K})f(x) + o_P\left(\frac{1}{n \det(H^{(s)})}\right). \quad (3.40)$$

where  $v_2$  and  $R(\mathcal{K})$  are as defined in A3.1.

**Lemma 3.2.** *Suppose the kernel  $\mathcal{K}_H$  is defined as in A3.1. Given a positive constant  $\beta < 1$  and an increasing sequence of constants  $t_n = \frac{1}{4+r} \log_{1/\beta}(nb_n)$ , where  $b_n = \tilde{O}(1)$ . Define the sets of bandwidth matrices*

$$\mathcal{H}_n = \{H^{(s)} : H^{(s)} = H^{(0)}\beta^s \text{ for all the nonnegative integer } s \text{ such that } s \leq t_n\}$$

Define

$$M_n(x) = \frac{(\hat{f}_H(x) - \mathbb{E}\hat{f}_H(x))}{\sqrt{\text{Var}(\hat{f}_H(x))}} \quad (3.41)$$

Then

$$\sup_{H \in \mathcal{H}_n} \sup_z |\mathbb{P}(M_n(x) \leq z) - \Phi(z)| \longrightarrow 0. \quad (3.42)$$

**Lemma 3.3.** *Under assumptions A3.1 – A3.4, suppose that  $x$  is interior to the support of  $f$  and  $K$  is a product kernel with bandwidth matrix  $H^{(s)} = \text{diag}(h_1^{(s)}, \dots, h_d^{(s)})$ . Then*

$$\mu_j^{(s)} = \frac{\partial}{\partial h_j^{(s)}} \mathbb{E}[\hat{f}_{H^{(s)}}(x) - f(x)] = o_P(h_j^{(s)}) \text{ for all } j \in R^c \quad (3.43)$$

For  $j \in R$  we have

$$\mu_j^{(s)} = \frac{\partial}{\partial h_j^{(s)}} \mathbb{E}[\hat{f}_{H^{(s)}}(x) - f(x)] = h_j^{(s)} v_2 f_{jj}(x) + o_P(h_j^{(s)}). \quad (3.44)$$

Thus, for any integer  $s > 0$ ,  $h_s = h_0\beta^s$ , each  $j > r$  satisfies  $\mu_j^{(s)} = o_P(h_j^{(s)}) = o_P(h_j^{(0)})$ .

**Lemma 3.4.** *Define*

$$C = \frac{R(\mathcal{K})f(x)}{4} \quad (3.45)$$

then, if  $h_j^{(0)}$  is defined as in A3.2.

$$(s_j^{(s)})^2 = \text{Var}(Z_j^{(s)}) = \frac{C}{n(h_j^{(s)})^2} \left( \prod_{k=1}^d \frac{1}{h_k^{(s)}} \right) (1 + o_P(1)) \quad (3.46)$$

**Lemma 3.5.** *Under assumptions A3.1 – A3.4. Let  $Z_j = \frac{1}{n} \sum_{i=1}^n Z_{ji}$  be defined as in equation(3.4), given a positive constant  $\beta < 1$  and an increasing sequence of constants  $t_n = \frac{1}{4+r} \log_{1/\beta}(nb_n)$ , where  $b_n = \tilde{O}(1)$ . Define the sets of bandwidth matrices*

$$\mathcal{H}_n = \{H^{(s)} : H^{(s)} = H^{(0)} \beta^s \text{ for all the nonnegative integer } s \text{ such that } s \leq t_n\}$$

Then

$$\sup_{H \in \mathcal{H}_n} \sup_z \left| \mathbb{P} \left( \frac{Z_j - \mathbb{E}Z_j}{\sqrt{\text{Var}(Z_j)}} \leq z \right) - \Phi(z) \right| \longrightarrow 0. \quad (3.47)$$

**Lemma 3.6.** *Let  $Z \sim N(\mu, \sigma^2)$ . If  $\lambda > 2\mu$  and  $\lambda^2 > 2\sigma^2$  then*

$$\mathbb{P}(|Z| > \lambda) \leq \frac{5\lambda}{\sigma} \exp \left\{ -\frac{\lambda^2}{8\sigma^2} \right\}. \quad (3.48)$$

Moreover, if  $\lambda \geq 5\sigma$  then

$$\mathbb{P}(|Z| > \lambda) \leq \exp \left\{ -\frac{\lambda^2}{16\sigma^2} \right\}. \quad (3.49)$$

The proof of this Lemma could be found in [Lafferty and Wasserman, 2008].

**Theorem 3.1.** *Under assumptions A3.1 – A3.4, suppose that  $A_{\min} = \min_{j \leq r} |f_{jj}(x)| = \tilde{\Omega}(1)$  and  $A_{\max} = \max_{j \leq r} |f_{jj}(x)| = \tilde{O}(1)$ . Then, the number of iterations  $T_n$  until the density Rodeo algorithm stops satisfies*

$$\mathbb{P} \left( \frac{1}{4+r} \log_{1/\beta}(na_n) \leq T_n \leq \frac{1}{4+r} \log_{1/\beta}(nb_n) \right) \longrightarrow 1 \quad (3.50)$$

where  $a_n = \tilde{\Omega}(1)$  and  $b_n = \tilde{O}(1)$ . Moreover, the algorithm outputs bandwidths  $H^* = \text{diag}(h_1^*, \dots, h_d^*)$  that satisfies

$$\mathbb{P} \left( h_j^* = h_j^{(0)} \text{ for all } j > r \right) \longrightarrow 1 \quad (3.51)$$

Also, we have

$$\mathbb{P} \left( h_j^{(0)} (nb_n)^{-1/(4+r)} \leq h_j^* \leq h_j^{(0)} (na_n)^{-1/(4+r)} \text{ for all } j \leq r \right) \longrightarrow 1 \quad (3.52)$$

assuming that  $h_j^{(0)}$  is defined as in A3.2.

**Corollary 3.1.** *Under the same condition of theorem 3.1, the risk  $\mathcal{R}_{h^*}$  of the density rodeo estimator satisfies*

$$\mathcal{R}_{H^*} = \mathbb{E} \int \left( \hat{f}_{H^*}(x) - f(x) \right)^2 dx = \tilde{O}_p \left( n^{-4/(4+r)} \right) \quad (3.53)$$

*Proof.* Since the integrand is nonnegative, the order of integration and expectation can be reversed, so that

$$\mathcal{R}_{H^*} = \mathbb{E} \int \left( \hat{f}_{H^*}(x) - f(x) \right)^2 dx = \int \mathbb{E} \left( \hat{f}_{H^*}(x) - f(x) \right)^2 dx \quad (3.54)$$

$$= \int \text{Bias}^2 \left( \hat{f}_{H^*}(x) \right) dx + \int \text{Var} \left( \hat{f}_{H^*}(x) \right) dx. \quad (3.55)$$

Given the bandwidths in expression(3.51) and expression(3.52), we have that the squared bias is given by

$$\begin{aligned} \int \text{Bias}^2 \left( \hat{f}_{H^*}(x) \right) dx &= \int \left( \sum_{j \leq r} v_2 f_{jj}(x) h_j^{*2} \right)^2 dx + o_P(\text{tr}(H^{*T} H^*)) \\ &= \int \sum_{i,j \leq r} v_2^2 f_{ii}(x) f_{jj}(x) h_i^{*2} h_j^{*2} dx + o_P(\text{tr}(H^{*T} H^*)) \\ &= \tilde{O}_P(n^{-4/(4+r)}) \end{aligned} \quad (3.56)$$

by Theorem 3.1. Similarly, by lemma 3.4, we calculate the variance as

$$\int \text{Var} \left( \hat{f}_{H^*}(x) \right) dx = \int \frac{1}{n} \prod_i \frac{1}{h_i^*} R(\mathcal{K}) f(x) (1 + o_P(1)) dx \quad (3.57)$$

$$= \tilde{O}_P(n^{-1+r/(4+r)}) \quad (3.58)$$

$$= \tilde{O}_P(n^{-4/(4+r)}). \quad (3.59)$$

The result follows from the bias-variance decomposition.  $\square$

This result shows that the optimal rates of convergence is obtained up to a logarithmic factor.

### 3.7 CONCLUSIONS

This chapter is mainly used to illustrate the generality of the rodeo framework. Under some suitably-defined sparsity condition, the previously developed nonparametric regression framework is easily adapted to perform high-dimensional density estimation. The resulting method is both computationally efficient and theoretically soundable. Empirical results show that our method is better than the built-in methods in many cases.

Current assumption requires the underlying density to be factorized into two components. Another interesting assumption is to assume that the observed high-dimensional data are lying on a low-dimensional smooth manifold. A recent result of Bickel and Li [2006] shows that local polynomial regression can adapt to the local manifold structure in the sense that it

achieves the optimal convergence rate. When assuming all dimensions use the same bandwidth  $h$ , they formalize an asymptotic irrelevance condition as

$$\begin{aligned} & \exists \epsilon (0 < \epsilon < 1), \text{ s.t.} \\ & \mathbb{E} \left[ \mathcal{K}^\gamma \left( \frac{X-x}{h} \right) w(X) \mathbb{1} \left( X \in \left( \mathcal{B}_{x,h^{1-\epsilon}}^D \cap \mathcal{X} \right)^c \right) \right] = o(h^{d+2}) \end{aligned} \quad (3.60)$$

for  $\gamma = 1, 2$  and  $|w(x)| \leq M(1 + |x|^2)$ . Under this kind of assumptions, it's interesting to design a blockwised rodeo algorithm (i.e. only on a local portion of the data, we estimate the bandwidth) which can also adapt to the local manifold structure and achieves a better risk.

### 3.8 APPENDIX: TECHNICAL PROOFS

#### 3.8.1 Derivation of the Semiparametric Rodeo Estimator

When assuming a Gaussian kernel and a Gaussian baseline distribution, the semiparametric kernel density estimator is defined as

$$\bar{f}_H(x) = \frac{\hat{b}(x) \sum_{i=1}^n \mathcal{K}_H(X^{(i)} - x)}{n \int \mathcal{K}_H(u - x) \hat{b}(u) du}. \quad (3.61)$$

Here, we ignore the hat notation for both  $\mu$  and  $\Sigma$ . Assuming that

$$K_h(u) \sim N(0, H), \quad H = \text{diag}(h_1^2, h_2^2, \dots, h_d^2) \quad (3.62)$$

$$\hat{b}(u) = N(\mu, \Sigma). \quad (3.63)$$

We get that the denominator is  $\propto N(\mu, H + \Sigma)$ , therefore

$$n \int_{-\infty}^{\infty} K_h(u - x) \hat{b}(u) du \quad (3.64)$$

$$= \frac{n}{\sqrt{2\pi|H + \Sigma|}} \exp \left\{ -\frac{(x - \mu)^T (H + \Sigma)^{-1} (x - \mu)}{2} \right\}. \quad (3.65)$$

Thus

$$\frac{\hat{b}(x)}{n \int_{-\infty}^{\infty} K_h(u - x) \hat{b}(u) du} \quad (3.66)$$

$$= \frac{\frac{1}{\sqrt{2\pi|\Sigma|}} \exp \left\{ -\frac{(x - \mu)^T (\Sigma)^{-1} (x - \mu)}{2} \right\}}{\frac{n}{\sqrt{2\pi|H + \Sigma|}} \exp \left\{ -\frac{(x - \mu)^T (H + \Sigma)^{-1} (x - \mu)}{2} \right\}} \quad (3.67)$$

$$= \frac{1}{n} \sqrt{\frac{|H + \Sigma|}{|\Sigma|}} \exp \left\{ -\frac{(x - \mu)^T (\Sigma^{-1} - (H + \Sigma)^{-1}) (x - \mu)}{2} \right\}. \quad (3.68)$$



The final estimator is

$$\begin{aligned} \bar{f}_H(x) & \\ &= \hat{f}_H(x) \sqrt{\frac{|H + \Sigma|}{|\Sigma|}} \exp \left\{ -\frac{(x - \mu)^T (\Sigma^{-1} - (H + \Sigma)^{-1}) (x - \mu)}{2} \right\} \end{aligned} \quad (3.69)$$

where  $\hat{f}_H(x)$  is the standard kernel density estimator defined in equation (3.3).

### 3.8.2 Proofs of the Main Results

For the convenience of notation, we suppress the superscripts unless necessary.

#### 3.8.2.1 Proof of Lemma 3.1

For the expectation term,

$$\mathbb{E} \hat{f}_H(x) \quad (3.70)$$

$$= \mathbb{E} \frac{1}{n \det(H)} \sum_{i=1}^n \mathcal{K}(H^{-1}(x - X^{(i)})) \quad (3.71)$$

$$= \frac{1}{\det(H)} \int \mathcal{K}(H^{-1}(u - x)) f(u) du \quad (3.72)$$

$$= \int \mathcal{K}(u) f(x + Hu) du \quad (3.73)$$

$$= \int \mathcal{K}(u) \{f(x) + u^T H^T \nabla f(x) + \frac{1}{2} u^T H^T \mathcal{H}_f(x) Hu + o_P(\text{tr}(u^T H^T Hu))\} du$$

$$= f(x) + \frac{1}{2} v_2 \text{tr}(H^T \mathcal{H}_f(x) H) + o_P(\text{tr}(H^T H)) \quad (3.74)$$

$$= f(x) + \frac{1}{2} v_2 \text{tr}(H^T \mathcal{H}_R(x) H) + o_P(\text{tr}(H^T H)). \quad (3.75)$$

Equation (3.75) follows from Assumption A3.3. While

$$\text{Var}(\hat{f}_H(x)) \quad (3.76)$$

$$= \text{Var} \left( \frac{1}{n \det(H)} \sum_{i=1}^n \mathcal{K}(H^{-1}(x - X^{(i)})) \right) \quad (3.77)$$

$$= \frac{1}{n \det(H)^2} \text{Var} \left( \mathcal{K}(H^{-1}(x - X^{(i)})) \right) \quad (3.78)$$

$$= \frac{1}{n \det(H)^2} \mathbb{E} \{ \mathcal{K}^2(H^{-1}(x - X^{(i)})) \} - \frac{1}{n \det(H)^2} \mathbb{E}^2 \{ \mathcal{K}(H^{-1}(x - X^{(i)})) \}$$

$$= \frac{1}{n \det(H)^2} \int \{ \mathcal{K}(H^{-1}(u - x)) \}^2 f(u) du - \frac{1}{n} \mathbb{E}^2 \{ \hat{f}_H(x) \} \quad (3.79)$$

$$= \frac{1}{n \det(H)} \int \{ \mathcal{K}(u) \}^2 f(u + Hu) du - \frac{1}{n} \mathbb{E}^2 \{ \hat{f}_H(x) \} \quad (3.80)$$

$$= \frac{1}{n \det(H)} R(\mathcal{K}) f(x) + o_P \left( \frac{1}{n \det(H)} \right). \quad (3.81)$$

The first equality follows from the fact that all the  $X^{(i)}$ 's are i.i.d. The last equality follows by a Taylor's expansion.

### 3.8.2.2 Proof of Lemma 3.2

We define  $\hat{f}_H(x) = \frac{1}{n} \sum_{i=1}^n J_i(x)$ , where  $J_1(x), J_2(x), \dots, J_n(x)$  are i.i.d distributed, and

$$M_n(x) = \frac{(\hat{f}_H(x) - \mathbb{E}\hat{f}_H(x))}{\sqrt{\text{Var}(\hat{f}_H(x))}}. \quad (3.82)$$

From the Berry-Esseen bound, for each fixed  $H$ , we get that

$$\sup_z |\mathbb{P}(M_n(x) \leq z) - \Phi(z)| \leq \frac{33 \mathbb{E}|J_1(x) - \mathbb{E}J_1(x)|^3}{4 \sqrt{n} \text{Var}^{3/2}(J_1(x))} \quad (3.83)$$

$$\leq \frac{33 \mathbb{E}(|J_1(x)| + |\mathbb{E}J_1(x)|)^3}{4 \sqrt{n} \text{Var}^{3/2}(J_1(x))} \quad (3.84)$$

$$\leq \frac{66|J_1(x)|^3}{\sqrt{n} \text{Var}^{3/2}(J_1(x))} \quad (3.85)$$

$$\leq \frac{66 \prod_{k=1}^d \frac{1}{h_k^3} C_{\mathcal{K}}^3}{\sqrt{n} \prod_{k=1}^d \frac{1}{h_k^{3/2}} (4C)^3} \quad (3.86)$$

$$= \frac{66(C_{\mathcal{K}}/4C)^3}{\sqrt{n} \prod_{k=1}^d h_k^{3/2}}. \quad (3.87)$$

Where  $C_{\mathcal{K}}$  and  $C$  are defined in assumption A3.4 and Lemma 3.1 respectively. Based on this, the supreme over all the bandwidths  $H \in \mathcal{H}_n$  satisfies

$$\begin{aligned} \sup_{H \in \mathcal{H}_n} \sup_z |\mathbb{P}(M_n(x) \leq z) - \Phi(z)| &\leq \frac{66(C_{\mathcal{K}}/4C)^3}{\sqrt{n} \left( \prod_{k=1}^d h_k^{(0)} \beta^{t_n} \right)^{3/2}} \quad (3.88) \\ &= O\left( \frac{b_n^{1/(4+r)} (\log \log n)^{3d/2}}{n^{(r-2)/(8r+2)}} \right) \rightarrow 0. \end{aligned}$$

The result follows directly.

### 3.8.2.3 Proof of Lemma 3.3

For  $j \in R$ , from lemma 3.2, we have

$$\mathbb{E}\hat{f}_H(x) - f(x) = \frac{1}{2} v_2 \text{tr}(H^T \mathcal{H}_R(x) H) + o_P(\text{tr}(H^T H)) \quad (3.89)$$

Also, under some regularity conditions, we have

$$\mu_j = \frac{\partial}{\partial h_j} \mathbb{E}[\hat{f}_H(x) - f(x)] = h_j v_2 f_{jj}(x) + o_P(h_j). \quad (3.90)$$

For  $j \in R^c$ , the proof proceeds by equation (3.89), when  $j \in R^c$ , the corresponding elements in the Heissen  $\mathcal{H}_f(x)$  will be zero, the result follows directly.

#### 3.8.2.4 Proof of Lemma 3.4

Assuming that  $\zeta \sim N(0, 1)$ . From lemma 3.1, we could represent the kernel density estimator  $\hat{f}_H(x)$  as

$$\begin{aligned}\hat{f}_H(x) &= \mathbb{E}\hat{f}_H(x) + \sqrt{\text{Var}(\hat{f}_H(x))} \times \zeta \\ &= f(x) + \frac{1}{2}v_2 \text{tr}(H^T \mathcal{H}_f(x) H) + o_P(\text{tr}(H^T H)) + \sqrt{\text{Var}(\hat{f}_H(x))} \times \zeta.\end{aligned}\quad (3.91)$$

Thus,

$$Z_j = \frac{\partial \hat{f}_H(x)}{\partial h_j} + \frac{\partial}{\partial h_j} \left( \frac{1}{2} v_2 \text{tr}(H^T \mathcal{H}_R(x) H) \right) + \frac{\partial}{\partial h_j} \left( \sqrt{\text{Var}(\hat{f}_H(x))} \times \zeta \right).$$

Since

$$\begin{aligned}& \frac{\partial}{\partial h_j} \left( \sqrt{\text{Var}(\hat{f}_H(x))} \right) \\ &= \frac{1}{2} \frac{1}{\sqrt{\text{Var}(\hat{f}_H(x))}} \frac{\partial}{\partial h_j} \left( \text{Var}(\hat{f}_H(x)) \right)\end{aligned}\quad (3.92)$$

$$\begin{aligned}&= -\frac{1}{2} \frac{1}{\sqrt{\text{Var}(\hat{f}_H(x))}} \left( \frac{R(\mathcal{K})f(x)}{h_j n \det(H)} \right) \left( 1 + o_P \left( \frac{1}{n \det(H) h_j} \right) \right) \\ &= -\frac{1}{2} \sqrt{\frac{R(\mathcal{K})f(x)}{h_j^2 n \det(H)}} (1 + o_P(\sqrt{h_j})).\end{aligned}\quad (3.93)$$

The second equality follows from lemma 3.1, therefore

$$s_j^2 = \text{Var}(Z_j) = \frac{C}{nh_j^2} \left( \prod_{k=1}^d \frac{1}{h_k} \right) (1 + o_P(1)). \quad (3.94)$$

## 3.8.2.5 Proof of Lemma 3.5

Since  $Z_j = \frac{1}{n} \sum_{i=1}^n Z_{ji}$ , and  $Z_{j1}, Z_{j2}, \dots, Z_{jn}$  are i.i.d distributed. Similar as in the proof of lemma 3.2, from the Berry-Esseen bound, for each fixed  $H$ , we get that

$$\sup_z \left| \mathbb{P} \left( \frac{Z_j - \mathbb{E}Z_j}{\sqrt{\text{Var}(Z_j)}} \leq z \right) - \Phi(z) \right| \leq \frac{33 \mathbb{E}|Z_{j1} - \mathbb{E}Z_{j1}|^3}{4 \sqrt{n} \text{Var}^{3/2}(Z_{j1})} \quad (3.95)$$

$$\leq \frac{66|Z_{j1}|^3}{\sqrt{n} \text{Var}^{3/2}(Z_{j1})} \quad (3.96)$$

$$\leq \frac{66 \frac{1}{h_j^9} \prod_{k=1}^d \frac{1}{h_k^3} C_{\mathcal{M}}^3}{\sqrt{n} \frac{1}{h_j^3} \prod_{k=1}^d \frac{1}{h_k^{3/2}} (C^{1/2})^3} \quad (3.97)$$

$$= \frac{66(C_{\mathcal{M}}/\sqrt{C})^3}{\sqrt{n} h_j^6 \prod_{k=1}^d h_k^{3/2}}. \quad (3.98)$$

Where  $C_{\mathcal{M}}$  is evaluated from  $C_{\mathcal{K}}$  and  $C_d$  in assumption A3.4 and  $C$  is defined in lemma 3.1. Based on the same reasoning as in lemma 3.2, the supreme over all the bandwidths  $H \in \mathcal{H}_n$  satisfies

$$\sup_{H \in \mathcal{H}_n} \sup_z \left| \mathbb{P} \left( \frac{Z_j - \mathbb{E}Z_j}{\sqrt{\text{Var}(Z_j)}} \leq z \right) - \Phi(z) \right| \longrightarrow 0. \quad (3.99)$$

## 3.8.2.6 Proof of Lemma 3.6

Without loss of generality, we assume  $\mu > 0$ . Then,

$$\mathbb{P}(|Z| > \lambda) \leq 2\mathbb{P}(Z > \lambda) \quad (3.100)$$

$$= 2\mathbb{P}\left(\frac{Z - \mu}{\sigma} > \frac{\lambda - \mu}{\sigma}\right) \quad (3.101)$$

$$\leq \frac{2\sigma}{\lambda - \mu} \exp\left\{-\frac{(\lambda - \mu)^2}{2\sigma^2}\right\} \equiv B(\mu). \quad (3.102)$$

Now  $B(\mu) = B(0) + \mu B'(\tilde{\mu})$  for some  $0 \leq \tilde{\mu} \leq \mu$  and

$$B'(\mu) = \frac{2\sigma}{\lambda - \mu} \exp\left\{-\frac{(\lambda - \mu)^2}{2\sigma^2}\right\} \left(\frac{\lambda - \mu}{\sigma^2} + \frac{1}{\lambda - \mu}\right). \quad (3.103)$$

Hence

$$B'(\mu) \leq \frac{2\sigma}{\lambda - \mu} \exp\left\{-\frac{(\lambda - \mu)^2}{2\sigma^2}\right\} \left(\frac{\lambda}{\sigma^2} + \frac{1}{\lambda - \mu}\right). \quad (3.104)$$

When  $\lambda \geq 2\mu$ ,  $1/(\lambda - \mu) \leq 2/\lambda$  and  $(\lambda - \mu)^2 \geq \lambda^2/4$  so that if  $\lambda^2 \geq 2\sigma^2$  then

$$B'(\mu) \leq \frac{4\sigma}{\lambda} \exp\left\{-\frac{\lambda^2}{8\sigma^2}\right\} \left(\frac{\lambda}{\sigma^2} + \frac{2}{\lambda}\right) \leq \frac{8}{\sigma} \exp\left\{-\frac{\lambda^2}{8\sigma^2}\right\}. \quad (3.105)$$

Thus,

$$\mathbb{P}(|Z| > \lambda) \leq \frac{2\sigma}{\lambda} \exp \left\{ -\frac{\lambda^2}{2\sigma^2} \right\} + \frac{8\mu}{\sigma} \exp \left\{ -\frac{\lambda^2}{8\sigma^2} \right\} \leq \frac{5\lambda}{\sigma} \exp \left\{ -\frac{\lambda^2}{8\sigma^2} \right\}.$$

The last statement follows since  $5xe^{-x^2/8} \leq e^{-x^2/16}$  for all  $x \geq 5$ .

### 3.8.2.7 Proof of theorem 3.1

First, we consider the case  $j > r$ . Let  $V_t = \{j > r : h_j = h_0\beta^t\}$  be the set of irrelevant dimensions that are active at stage  $t > 1$  of the algorithm. Define  $v_j = \text{Var}(Z_j)$ , from lemma 3.3 and the algorithm in figure 9, for sufficiently large  $n$ , it's obvious that  $\lambda_j \geq 2\mu_j$ ,  $\lambda_j^2 \geq 2s_j^2$ , and  $\lambda \geq 5s_j$ , and  $v_j^2/s_j^2 = 1 + o(1)$  with probability tending to 1. Assuming  $\tilde{Z}_j$  is a normal random variable with the same mean and variance as  $Z_j$ . Then

$$\mathbb{P}(|Z_j| > \lambda_j, \text{ for some } j \in V_t) \quad (3.106)$$

$$\leq \sum_{j \in V_t} \mathbb{P}(|Z_j| > \lambda_j) + o(1) \quad (3.107)$$

$$= \sum_{j \in V_t} (\mathbb{P}(|\tilde{Z}_j| > \lambda_j) + \mathbb{P}(|Z_j| > \lambda_j) - \mathbb{P}(|\tilde{Z}_j| > \lambda_j)) + o(1) \quad (3.108)$$

$$\leq d \exp \left\{ -\lambda_j^2/16v_j^2 \right\} + o(1) \quad (3.109)$$

$$= d \exp \left\{ -\lambda_j^2(1 + o(1))/16s_j^2 \right\} + o(1) \rightarrow 0 \quad (3.110)$$

Therefore, with probability tending to 1,  $h_j = h_0$  for each  $j > r$ , meaning that the bandwidth for each irrelevant dimension is frozen in the first step in the algorithm.

Now consider  $j \leq r$ . By assumption A3.4 and lemma 3.3, for sufficiently large  $n$ ,  $\mu_j \geq ch_j|f_{jj}(x)|$  for some  $c > 0$ . Without loss of generality, assume that  $ch_jf_{jj} > 0$ . We claim that in the iteration  $t$  of the algorithm, if

$$t \leq \frac{1}{4+r} \log_{1/\beta} \left( \frac{c^2 n A_{\min}^2 h_0^{4+d}}{8C \log(nc_n)} \right), \quad (3.111)$$

then

$$\mathbb{P}(h_j = h_0\beta^t, \text{ for all } j \leq r) \rightarrow 1. \quad (3.112)$$

To show this, first note that inequality (3.111) can be written as

$$\left( \frac{1}{\beta} \right)^{t(4+r)} \leq \frac{c^2 n A_{\min}^2 h_0^{4+d}}{8C \log(nc_n)} \quad (3.113)$$

except on an event of vanishing probability. We have shown above that

$$\prod_{j>r} \frac{1}{h_j} \leq \left( \frac{1}{h_0} \right)^{d-r}. \quad (3.114)$$

So on the complement of this event, if each relevant dimension is active at step  $s \leq t$ , we have

$$\frac{\lambda_j^2}{h_j^2} = \frac{2s_j^2 \log(nc_n)}{h_j^2} \quad (3.115)$$

$$= \frac{2C \log(nc_n)}{nh_j^4} \prod_i \frac{1}{h_i} \quad (3.116)$$

$$\leq \frac{2C \log(nc_n)}{nh_0^{4+d}} \left(\frac{1}{\beta}\right)^{(4+r)t} \quad (3.117)$$

$$\leq \frac{c^2 A_{\min}^2}{4} \quad (3.118)$$

$$\leq \frac{c^2 f_{jj}(x)^2}{4} \quad (3.119)$$

which implies that

$$cf_{jj}(x)h_j \geq 2\lambda_j \quad (3.120)$$

and hence

$$\frac{cf_{jj}(x)h_j - \lambda_j}{s_j} \geq \frac{\lambda_j}{s_j} = \sqrt{2 \log(nc_n)} \quad (3.121)$$

for each  $j \leq r$ . Now,

$$\mathbb{P}(\text{rodeo halts}) = \mathbb{P}(|Z_j| < \lambda_j \text{ for all } j \leq r) \quad (3.122)$$

$$\leq \mathbb{P}(|Z_j| < \lambda_j \text{ for some } j \leq r) \quad (3.123)$$

$$\leq \sum_{j \leq r} \mathbb{P}(|Z_j| < \lambda_j) \quad (3.124)$$

$$\leq \sum_{j \leq r} \mathbb{P}(Z_j < \lambda_j) \quad (3.125)$$

$$\leq \sum_{j \leq r} \mathbb{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{\mu_j - \lambda_j}{s_j}\right) \quad (3.126)$$

$$\leq \sum_{j \leq r} \mathbb{P}\left(\frac{Z_j - \mu_j}{s_j} > \frac{cf_{jj}(x)h_j - \lambda_j}{s_j}\right) \quad (3.127)$$

$$\leq \sum_{j \leq r} \mathbb{P}\left(\left|\frac{Z_j - \mu_j}{s_j}\right| > \frac{cf_{jj}(x)h_j - \lambda_j}{s_j}\right) \quad (3.128)$$

$$\approx \sum_{j \leq r} \mathbb{P}\left(\left|\frac{\tilde{Z}_j - \mu_j}{s_j}\right| > \frac{cf_{jj}(x)h_j - \lambda_j}{s_j}\right) \quad (3.129)$$

$$\leq \frac{r}{nc_n \sqrt{2 \log(nc_n)}} \quad (3.130)$$

where equation (3.129) follows the same idea as in equation (3.108). The last inequality follows from the standard Miller's inequality. Finally, summing over all iterations  $s \leq t$  gives

$$\mathbb{P} \left( \bigcup_{s \leq t} \bigcup_{j \leq r} \left\{ |Z_j^{(s)}| < \lambda_j^{(s)} \right\} \right) \leq \frac{tr}{nc_n \sqrt{2 \log(nc_n)}} \quad (3.131)$$

$$\leq \frac{\frac{r}{4+r} \log_{1/\beta} \left( \frac{c^2 n A_{\min}^2 h_0^{4+d}}{8C \log(nc_n)} \right)}{nc_n \sqrt{2 \log(nc_n)}} \longrightarrow 0 \quad (3.132)$$

by the density Rodeo's algorithm. Thus, the bandwidths  $h_j$  for  $j \leq r$  satisfy, with high probability,

$$h_j = h_0 \beta^t \leq h_0 \left( \frac{8C \log(nc_n)}{c^2 A_{\min}^2 n h_0^{4+d}} \right)^{1/(4+r)} \quad (3.133)$$

$$= n^{-1/(4+r)} \left( \frac{8C \log(nc_n)}{c^2 A_{\min}^2 h_0^{d-r}} \right)^{1/(4+r)}. \quad (3.134)$$

In particular, with probability approaching one, the algorithm runs for a number of iterations  $T_n$  bounded from below by

$$T_n \geq \frac{1}{4+r} \log_{1/\beta}(na_n) \quad (3.135)$$

where

$$a_n = \frac{c^2 A_{\min}^2 h_0^{d-r}}{8C \log(nc_n)} = \tilde{\Omega}(1). \quad (3.136)$$

We next show that the algorithm is unlikely to reach iteration  $s$ , if

$$s \geq \frac{1}{4+r} \log_{1/\beta}(nb_n) \quad (3.137)$$

where  $b_n = \tilde{\Omega}(1)$  will be defined below. From the argument above, we know that except on an event of vanishing probability, each relevant dimension  $j \leq r$  has bandwidth no larger than

$$h_j \leq h_0 \beta^{(\log_{1/\beta}(na_n))/(4+r)} = \frac{h_0}{(na_n)^{1/(4+r)}}. \quad (3.138)$$

Thus, if relevant dimension  $j$  has bandwidth  $h_j \leq h_0 \beta^s$ , then from lemma 3.4 we have that

$$\frac{s_j^2}{\mu_j^2} = \frac{s_j^2}{v_2^2 f_{jj}^2(x) h_j^2} \quad (3.139)$$

$$\geq \frac{C}{v_2^2 f_{jj}^2(x) n h_j^4 \det(H)} \quad (3.140)$$

$$\geq \frac{C}{v_2^2 f_{jj}^2(x) n h_0^4 \beta^{4s}} \frac{n^{r/(4+r)} a_n^{r/(4+r)}}{h_0^r} \frac{1}{h_0^{d-r}} \quad (3.141)$$

$$= \frac{C}{v_2^2 f_{jj}^2(x) n^{4/(4+r)}} \frac{a_n^{r/(4+r)}}{h_0^{4+d}} \frac{1}{\beta^{4s}} \quad (3.142)$$

$$\geq \frac{C}{A_{max}^2 n^{4/(4+r)}} \frac{a_n^{r/(4+r)}}{h_0^{4+d}} \frac{1}{\beta^{4s}}. \quad (3.143)$$

Therefore

$$\frac{s_j^2}{\mu_j^2} \geq \log \log n \quad (3.144)$$

in case

$$\left(\frac{1}{\beta}\right)^s \geq (nb_n)^{1/(4+r)} \geq n^{1/(4+r)} \left(\frac{A_{max}^2 h_0^{4+d} \log \log n}{C a_n^{r/(4+r)}}\right)^{1/4}, \quad (3.145)$$

which defines  $b_n = \tilde{O}(1)$ . It follows that in iteration  $s \geq \frac{1}{4+r} \log_{1/\beta}(nb_n)$ , the probability of a relevant dimension having estimated derivative  $Z_j$  above threshold is bounded by

$$\mathbb{P}(|Z_j| > \lambda_j, \text{ for some } j \leq r) \leq \sum_{j \leq r} \mathbb{P}(|Z_j| > \lambda_j) \quad (3.146)$$

$$= \sum_{j \leq r} \mathbb{P}\left(\frac{|Z_j|}{s_j} > \frac{\lambda_j}{s_j}\right) \quad (3.147)$$

$$\approx \sum_{j \leq r} \mathbb{P}\left(\frac{|\tilde{Z}_j|}{s_j} > \frac{\lambda_j}{s_j}\right) \quad (3.148)$$

$$\leq \sum_{j \leq r} \mathbb{P}\left(\frac{s_j}{\lambda_j} e^{-\lambda_j^2/(2s_j^2)} + \frac{1}{4} \frac{\mu_j^2}{s_j^2}\right) \quad (3.149)$$

$$\leq \frac{r}{nc_n \sqrt{2 \log(nc_n)}} + \frac{1}{4} \sum_{j \in V_t} \frac{\mu_j^2}{s_j^2} \quad (3.150)$$

$$\leq \frac{r}{nc_n \sqrt{2 \log(nc_n)}} + \frac{r}{4 \log \log n} \quad (3.151)$$

$$= O\left(\frac{1}{\log \log n}\right), \quad (3.152)$$

which gives the statement of the theorem.



## NONPARANORMAL: LEARNING NONPARAMETRIC UNDIRECTED GRAPHS

---

*Recent methods for estimating sparse undirected graphs for real-valued data in high dimensional problems rely heavily on the assumption of normality. We show how to use a semiparametric Gaussian copula—or “nonparanormal”—for high dimensional inference. Just as additive models extend linear models by replacing linear functions with a set of one-dimensional smooth functions, the nonparanormal extends the normal by transforming the variables by smooth functions. We derive a method for estimating the nonparanormal, study the method’s theoretical properties, and show that it works well in many examples.*

### 4.1 INTRODUCTION AND MOTIVATION

The linear model is a mainstay of statistical inference that has been extended in several important ways. An extension to high dimensions was achieved by adding a sparsity constraint, leading to the lasso [Tibshirani, 1996]. An extension to nonparametric models was achieved by replacing linear functions with smooth functions, leading to additive models [Hastie and Tibshirani, 1999]. These two ideas were recently combined, leading to an extension called sparse additive models (SpAM) [Ravikumar et al., 2007, 2009a]. In this paper we consider a similar nonparametric extension of undirected graphical models based on multivariate Gaussian distributions in the high dimensional setting. Specifically, we use a high dimensional Gaussian copula with nonparametric marginals, which we refer to as a nonparanormal distribution.

If  $X$  is a  $d$ -dimensional random vector distributed according to a multivariate Gaussian distribution with covariance matrix  $\Sigma$ , the conditional independence relations between the random variables  $X_1, X_2, \dots, X_d$  are encoded in a graph formed from the precision matrix  $\Omega = \Sigma^{-1}$ . Specifically, missing edges in the graph correspond to zeroes of  $\Omega$ . To estimate the graph from a sample of size  $n$ , it is only necessary to estimate  $\Sigma$ , which is easy if  $n$  is much larger than  $d$ . However, when  $d$  is larger than  $n$ , the problem is more challenging. Recent work has focused on the problem of estimating the graph in this high dimensional setting, which becomes feasible if  $G$  is sparse. Yuan and Lin [2007] and Banerjee et al. [2008] propose an estimator based on regularized maximum likelihood using an  $\ell_1$  constraint on the entries of

$\Omega$ , and [Friedman et al. \[2007\]](#) develop an efficient algorithm for computing the estimator using a graphical version of the lasso. The resulting estimation procedure has excellent theoretical properties, as shown recently by [Rothman et al. \[2008\]](#) and [Ravikumar et al. \[2009b\]](#).

While Gaussian graphical models can be useful, a reliance on exact normality is limiting. Our goal in this paper is to weaken this assumption. Our approach parallels the ideas behind sparse additive models for regression [[Ravikumar et al., 2007, 2009a](#)]. Specifically, we replace the Gaussian with a semiparametric Gaussian copula. This means that we replace the random variable  $X = (X_1, \dots, X_d)$  by the transformed random variable  $f(X) = (f_1(X_1), \dots, f_d(X_d))$ , and assume that  $f(X)$  is multivariate Gaussian. This semiparametric copula results in a nonparametric extension of the normal that we call the *nonparanormal* distribution. The nonparanormal depends on the functions  $\{f_j\}$ , and a mean  $\mu$  and covariance matrix  $\Sigma$ , all of which are to be estimated from data. While the resulting family of distributions is much richer than the standard parametric normal (the paranormal), the independence relations among the variables are still encoded in the precision matrix  $\Omega = \Sigma^{-1}$ . We propose a nonparametric estimator for the functions  $\{f_j\}$ , and show how the graphical lasso can be used to estimate the graph in the high dimensional setting. The relationship between linear regression models, Gaussian graphical models, and their extensions to nonparametric and high dimensional models is summarized in Figure 20.

Most theoretical results on semiparametric copulas focus on low or at least finite dimensional models [[Klaassen and Wellner, 1997](#), [Tsukahara, 2005](#)]. Models with increasing dimension require a more delicate analysis; in particular, simply plugging in the usual empirical distribution of the marginals does not lead to accurate inference. Instead we use a truncated empirical distribution. We give a theoretical analysis of this estimator, proving consistency results with respect to risk, model selection, and estimation of  $\Omega$  in the Frobenius norm.

In the following section we review the basic notion of the graph corresponding to a multivariate Gaussian, and formulate different criteria for evaluating estimators of the covariance or inverse covariance. In Section 4.3 we present the nonparanormal, and in Section 4.4 we discuss estimation of the model. We present a theoretical analysis of the estimation method in Section 4.5, with the detailed proofs collected in an appendix. In Section 4.6 we present experiments with both simulated data and gene microarray data, where the problem is to construct the isoprenoid biosynthetic pathway.

## 4.2 ESTIMATING UNDIRECTED GRAPHS

Let  $X = (X_1, \dots, X_d)$  denote a random vector with distribution  $P = N(\mu, \Sigma)$ . The undirected graph  $G = (V, E)$  corresponding to  $P$  consists of a vertex set

| Assumptions   | Dimension | Regression            | Graphical Models     |
|---------------|-----------|-----------------------|----------------------|
| parametric    | low       | linear model          | multivariate normal  |
|               | high      | lasso                 | graphical lasso      |
| nonparametric | low       | additive model        | nonparanormal        |
|               | high      | sparse additive model | $L_1$ -nonparanormal |

Figure 20.: Comparison of regression and graphical models. The nonparanormal extends additive models to the graphical model setting. Regularizing the inverse covariance leads to an extension to high dimensions, which parallels sparse additive models for regression.

$V$  and an edge set  $E$ . The set  $V$  has  $d$  elements, one for each component of  $X$ . The edge set  $E$  consists of ordered pairs  $(i, j)$  where  $(i, j) \in E$  if there is an edge between  $X_i$  and  $X_j$ . The edge between  $(i, j)$  is excluded from  $E$  if and only if  $X_i$  is independent of  $X_j$  given the other variables  $X_{\setminus\{i,j\}} \equiv (X_s : 1 \leq s \leq d, s \neq i, j)$ , written

$$X_i \perp\!\!\!\perp X_j \mid X_{\setminus\{i,j\}}. \quad (4.1)$$

It is well known that, for multivariate Gaussian distributions, (4.1) holds if and only if  $\Omega_{ij} = 0$  where  $\Omega = \Sigma^{-1}$ .

Let  $X^{(1)}, X^{(2)}, \dots, X^{(n)}$  be a random sample from  $P$ , where  $X^{(i)} \in \mathbb{R}^d$ . If  $n$  is much larger than  $d$ , then we can estimate  $\Sigma$  using maximum likelihood, leading to the estimate  $\hat{\Omega} = S^{-1}$ , where

$$S = \frac{1}{n} \sum_{i=1}^n \left( X^{(i)} - \bar{X} \right) \left( X^{(i)} - \bar{X} \right)^T$$

is the sample covariance, with  $\bar{X}$  the sample mean. The zeroes of  $\Omega$  can then be estimated by applying hypothesis testing to  $\hat{\Omega}$  [Drton and Perlman, 2007, 2008].

When  $d > n$ , maximum likelihood is no longer useful; in particular, the estimate  $\hat{\Sigma}$  is not positive definite, having rank no greater than  $n$ . Inspired by the success of the lasso for linear models, several authors have suggested estimating  $\Sigma$  by minimizing

$$-\ell(\Omega) + \lambda \sum_{j \neq k} |\Omega_{jk}|$$

where

$$\ell(\Omega) = \frac{1}{2} (\log |\Omega| - \text{tr}(\Omega S) - d \log(2\pi))$$

is the log-likelihood with  $S$  the sample covariance matrix. The estimator  $\hat{\Omega}$  can be computed efficiently using the glasso algorithm [Friedman et al., 2007],

which is a block coordinate descent algorithm that uses the standard lasso to estimate a single row and column of  $\Omega$  in each iteration. Under appropriate sparsity conditions, the resulting estimator  $\hat{\Omega}$  has been shown to have good theoretical properties [Rothman et al., 2008, Ravikumar et al., 2009b].

There are several different ways to judge the quality of an estimator  $\hat{\Sigma}$  of the covariance or  $\hat{\Omega}$  of the inverse covariance. We discuss three in this paper, persistency, norm consistency, and sparsistency. Persistency means consistency in risk, when the model is not necessarily assumed to be correct. Suppose the true distribution  $P$  has mean  $\mu_0$ , and that we use a multivariate normal  $p(x; \mu_0, \Sigma)$  for prediction; we do not assume that  $P$  is normal. We observe a new vector  $X \sim P$  and define the prediction risk to be

$$R(\Sigma) = -\mathbb{E} \log d(X; \mu_0, \Sigma) = - \int \log d(x; \mu_0, \Sigma) dP(x).$$

It follows that

$$R(\Sigma) = \frac{1}{2} \left( \text{tr}(\Sigma^{-1} \Sigma_0) + \log |\Sigma| - d \log(2\pi) \right)$$

where  $\Sigma_0$  is the covariance of  $X$  under  $P$ . If  $\mathcal{S}$  is a set of covariance matrices, the oracle is defined to be the covariance matrix  $\Sigma_*$  minimizing  $R(\Sigma)$  over  $\mathcal{S}$ :

$$\Sigma_* = \arg \min_{\Sigma \in \mathcal{S}} R(\Sigma).$$

Thus  $p(x; \mu_0, \Sigma_*)$  is the best predictor of a new observation among all distributions in  $\{p(x; \mu_0, \Sigma) : \Sigma \in \mathcal{S}\}$ . In particular, if  $\mathcal{S}$  consists of covariance matrices with sparse graphs, then  $p(x; \mu_0, \Sigma_*)$  is, in some sense, the best sparse predictor. An estimator  $\hat{\Sigma}_n$  is *persistent* if

$$R(\hat{\Sigma}_n) - R(\Sigma_*) \xrightarrow{P} 0$$

as the sample size  $n$  increases to infinity. Thus, a persistent estimator approximates the best estimator over the class  $\mathcal{S}$ , but we do not assume that the true distribution has a covariance matrix in  $\mathcal{S}$ , or even that it is Gaussian. Moreover, we allow the dimension  $d = d_n$  to increase with  $n$ . On the other hand, norm consistency and sparsistency require that the true distribution is Gaussian. In this case, let  $\Sigma_0$  denote the true covariance matrix. An estimator is *norm consistent* if

$$\|\hat{\Sigma}_n - \Sigma\| \xrightarrow{P} 0$$

where  $\|\cdot\|$  is a norm. If  $E(\Omega)$  denotes the edge set corresponding to  $\Omega$ , an estimator is *sparsistent* if

$$\mathbb{P}(E(\Omega) \neq E(\hat{\Omega}_n)) \rightarrow 0.$$

Thus, a sparsistent estimator identifies the correct graph consistently. We present our theoretical analysis on these properties of the nonparanormal in Section 4.5.

## 4.3 THE NONPARANORMAL

We say that a random vector  $X = (X_1, \dots, X_d)^T$  has a *nonparanormal* distribution if there exist functions  $\{f_j\}_{j=1}^d$  such that  $Z \equiv f(X) \sim N(\mu, \Sigma)$ , where  $f(X) = (f_1(X_1), \dots, f_d(X_d))$ . We then write

$$X \sim \text{NPN}(\mu, \Sigma, f).$$

When the  $f_j$ 's are monotone and differentiable, the joint probability density function of  $X$  is given by

$$p_X(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (f(x) - \mu)^T \Sigma^{-1} (f(x) - \mu) \right\} \prod_{j=1}^d |f'_j(x_j)|. \quad (4.2)$$

**Lemma 4.1.** *The nonparanormal distribution  $\text{NPN}(\mu, \Sigma, f)$  is a Gaussian copula when the  $f_j$ 's are monotone and differentiable.*

*Proof.* By Sklar's theorem [Sklar, 1959], any joint distribution can be written as

$$F(x_1, \dots, x_d) = C\{F_1(x_1), \dots, F_d(x_d)\}$$

where the function  $C$  is called a copula. For the nonparanormal we have

$$F(x_1, \dots, x_d) = \Phi_{\mu, \Sigma}(\Phi^{-1}(F_1(x_1)), \dots, \Phi^{-1}(F_d(x_d)))$$

where  $\Phi_{\mu, \Sigma}$  is the multivariate Gaussian cdf and  $\Phi$  is the univariate standard Gaussian cdf. Thus, the corresponding copula is

$$C(u_1, \dots, u_d) = \Phi_{\mu, \Sigma}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)).$$

This is exactly a Gaussian copula with parameters  $\mu$  and  $\Sigma$ . If each  $f_j$  is differentiable then the density of  $X$  has the same form as (4.2).  $\square$

Note that the density in (4.2) is not identifiable; to make the family identifiable we demand that  $f_j$  preserve means and variances:

$$\mu_j = \mathbb{E}(Z_j) = \mathbb{E}(X_j) \quad \text{and} \quad \sigma_j^2 \equiv \Sigma_{jj} = \text{Var}(Z_j) = \text{Var}(X_j). \quad (4.3)$$

Note that these conditions only depend on  $\text{diag}(\Sigma)$  but not the full covariance matrix.

Let  $F_j(x)$  denote the marginal distribution function of  $X_j$ . Then

$$F_j(x) = \mathbb{P}(X_j \leq x) = \mathbb{P}(Z_j \leq f_j(x)) = \Phi\left(\frac{f_j(x) - \mu_j}{\sigma_j}\right)$$

which implies that

$$f_j(x) = \mu_j + \sigma_j \Phi^{-1}(F_j(x)). \quad (4.4)$$

The following basic fact says that the independence graph of the nonparanormal is encoded in  $\Omega = \Sigma^{-1}$ , as for the parametric normal.

**Lemma 4.2.** *If  $X \sim \text{NPN}(\mu, \Sigma, f)$  is nonparanormal and each  $f_j$  is differentiable, then  $X_i \perp\!\!\!\perp X_j \mid X_{\setminus\{i,j\}}$  if and only if  $\Omega_{ij} = 0$ , where  $\Omega = \Sigma^{-1}$ .*

*Proof.* From the form of the density (4.2), it follows that the density factors with respect to the graph of  $\Omega$ , and therefore obeys the global Markov property of the graph.  $\square$

Next we show that the above is true for any choice of identification restrictions.

**Lemma 4.3.** *Define*

$$h_j(x) = \Phi^{-1}(F_j(x)) \quad (4.5)$$

*and let  $\Lambda$  be the covariance matrix of  $h(X)$ . Then  $X_j \perp\!\!\!\perp X_k \mid X_{\setminus\{j,k\}}$  if and only if  $\Lambda_{jk}^{-1} = 0$ .*

*Proof.* We can rewrite the covariance matrix as

$$\Sigma_{jk} = \text{Cov}(Z_j, Z_k) = \sigma_j \sigma_k \text{Cov}(h_j(X_j), h_k(X_k)).$$

Hence  $\Sigma = D\Lambda D$  and

$$\Sigma^{-1} = D^{-1}\Lambda^{-1}D^{-1},$$

where  $D$  is the diagonal matrix with  $\text{diag}(D) = \sigma$ . The zero pattern of  $\Lambda^{-1}$  is therefore identical to the zero pattern of  $\Sigma^{-1}$ .  $\square$

Thus, it is not necessary to estimate  $\mu$  or  $\sigma$  to estimate the graph.

Figure 21 shows three examples of 2-dimensional nonparanormal densities. In each case, the component functions  $f_j(x)$  take the form

$$f_j(x) = a_j \text{sign}(x)|x|^{\alpha_j} + b_j$$

where the constants  $a_j$  and  $b_j$  are set to enforce the identifiability constraints (4.3). The covariance in each case is  $\Sigma = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$  and the mean is  $\mu = (0, 0)$ . The exponent  $\alpha_j$  determines the nonlinearity. It can be seen how the concavity of the density changes with the exponent  $\alpha$ , and that  $\alpha > 1$  can result in multiple modes.

The assumption that  $f(X) = (f_1(X_1), \dots, f_d(X_d))$  is normal leads to a semi-parametric model where only one dimensional functions need to be estimated. But the monotonicity of the functions  $f_j$ , which map onto  $\mathbb{R}$ , enables computational tractability of the nonparanormal. For more general functions  $f$ , the normalizing constant for the density

$$p_X(x) \propto \exp \left\{ -\frac{1}{2} (f(x) - \mu)^T \Sigma^{-1} (f(x) - \mu) \right\} \quad (4.6)$$

cannot be computed in closed form.

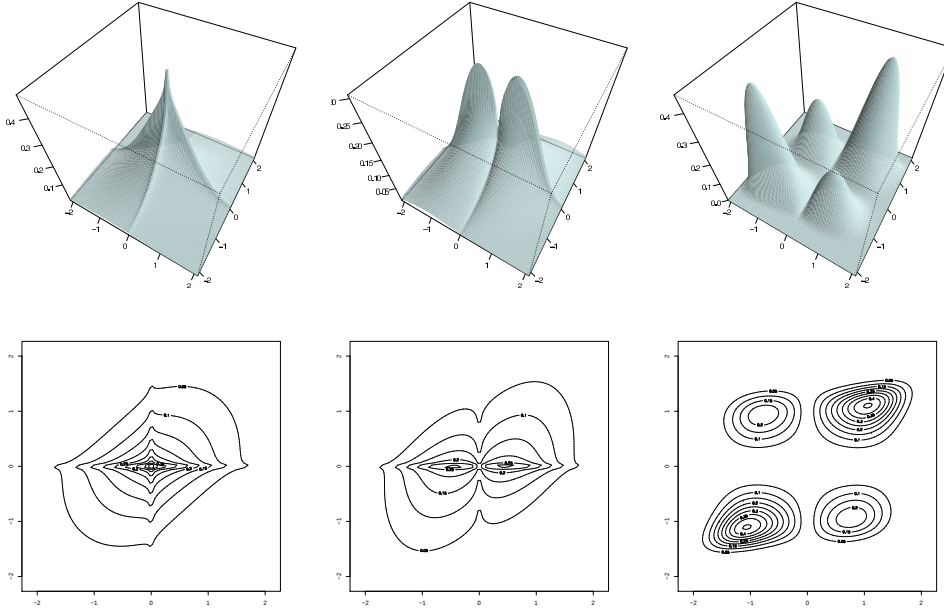


Figure 21.: Densities of three 2-dimensional nonparanormals. The component functions have the form  $f_j(x) = \text{sign}(x)|x|^{\alpha_j}$ . Left:  $\alpha_1 = 0.9, \alpha_2 = 0.8$ ; center:  $\alpha_1 = 1.2, \alpha_2 = 0.8$ ; right  $\alpha_1 = 2, \alpha_2 = 3$ . In each case  $\mu = (0, 0)$  and  $\Sigma = \begin{pmatrix} 1 & .5 \\ .5 & 1 \end{pmatrix}$ .

#### 4.4 ESTIMATION METHOD

Let  $X^{(1)}, \dots, X^{(n)}$  be a sample of size  $n$  where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T \in \mathbb{R}^d$ . In light of (4.5) we define

$$\hat{h}_j(x) = \Phi^{-1}(\tilde{F}_j(x)) \quad (4.7)$$

where  $\tilde{F}_j$  is an estimator of  $F_j$ . A natural candidate for  $\tilde{F}_j$  is the marginal empirical distribution function

$$\hat{F}_j(t) \equiv \frac{1}{n} \sum_{i=1}^n I(X_j^{(i)} \leq t).$$

Now, let  $\theta$  denote the parameters of the copula. Tsukahara (2005) suggests taking  $\hat{\theta}$  to be the solution of

$$\sum_{i=1}^n \phi\left(\tilde{F}_1(X_1^{(i)}), \dots, \tilde{F}_d(X_d^{(i)}), \theta\right) = 0$$

where  $\phi$  is an estimating equation and  $\tilde{F}_j(t) = n\hat{F}_j(t)/(n+1)$ . In our case,  $\theta$  corresponds to the covariance matrix. The resulting estimator  $\hat{\theta}$ , called a rank approximate Z-estimator, has excellent theoretical properties. However, we are interested in the high dimensional scenario where the dimension  $d$

is allowed to increase with  $n$ ; the variance of  $\hat{F}_j(t)$  is too large in this case. Instead, we use the following truncated or *Winsorized*<sup>1</sup> estimator:

$$\tilde{F}_j(x) = \begin{cases} \delta_n & \text{if } \hat{F}_j(x) < \delta_n \\ \hat{F}_j(x) & \text{if } \delta_n \leq \hat{F}_j(x) \leq 1 - \delta_n \\ (1 - \delta_n) & \text{if } \hat{F}_j(x) > 1 - \delta_n, \end{cases} \quad (4.8)$$

where  $\delta_n$  is a truncation parameter. Clearly, there is a bias-variance tradeoff in choosing  $\delta_n$ . Essentially the same estimator with  $\delta_n = 1/n$  is studied by [Klaassen and Wellner \[1997\]](#) in the case of bivariate Gaussian copula. In what follows we use

$$\delta_n \equiv \frac{1}{4n^{1/4}\sqrt{\pi \log n}}.$$

This provides the right balance so that we can achieve the desired rate of convergence in our estimate of  $\Omega$  and the associated undirected graph  $G$  in the high dimensional setting.

Given this estimate of the distribution of variable  $X_j$ , we then estimate the transformation function  $f_j$  by

$$\tilde{f}_j(x) \equiv \hat{\mu}_j + \hat{\sigma}_j \tilde{h}_j(x) \quad (4.9)$$

where

$$\tilde{h}_j(x) = \Phi^{-1}(\tilde{F}_j(x)) \quad (4.10)$$

and  $\hat{\mu}_j$  and  $\hat{\sigma}_j$  are the sample mean and the standard deviation:

$$\hat{\mu}_j \equiv \frac{1}{n} \sum_{i=1}^n X_j^{(i)} \quad \text{and} \quad \hat{\sigma}_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_j^{(i)} - \hat{\mu}_j)^2}.$$

Now, let  $S_n(\tilde{f})$  be the sample covariance matrix of  $\tilde{f}(X^{(1)}), \dots, \tilde{f}(X^{(n)})$ ; that is,

$$S_n(\tilde{f}) \equiv \frac{1}{n} \sum_{i=1}^n \left( \tilde{f}(X^{(i)}) - \mu_n(\tilde{f}) \right) \left( \tilde{f}(X^{(i)}) - \mu_n(\tilde{f}) \right)^T \quad (4.11)$$

$$\mu_n(\tilde{f}) \equiv \frac{1}{n} \sum_{i=1}^n \tilde{f}(X^{(i)}). \quad (4.12)$$

We then estimate  $\Omega$  using  $S_n(\tilde{f})$ . For instance, the maximum likelihood estimator is  $\hat{\Omega}_n^{\text{MLE}} = S_n(\tilde{f})^{-1}$ . The  $\ell_1$ -regularized estimator is

$$\hat{\Omega}_n = \arg \min_{\Omega} \{ \text{tr}(\Omega S_n(\tilde{f})) - \log |\Omega| + \lambda \|\Omega\|_1 \} \quad (4.13)$$

<sup>1</sup> After Charles P. Winsor, whom John Tukey credited with converting him from topology to statistics [Mallows 1990](#).



where  $\lambda$  is a regularization parameter, and  $\|\Omega\|_1 = \sum_{j \neq k} |\Omega_{jk}|$ . The estimated graph is then  $\hat{E}_n = \{(j, k) : \hat{\Omega}_{jk} \neq 0\}$ .

The nonparanormal is analogous to a sparse additive regression model [Ravikumar et al., 2009a], in the sense that both methods transform the variables by univariate functions. However, while sparse additive models use a regularized risk criterion to fit univariate transformations, our nonparanormal estimator uses a two-step procedure:

1. Replace the observations, for each variable, by their respective normal scores, subject to a Winsorized truncation.
2. Apply the graphical lasso to the transformed data to estimate the undirected graph.

The first step is non-iterative and computationally efficient, with no tuning parameters; it also makes the nonparanormal amenable to theoretical analysis.

Starting with the model in (4.2), another possibility would be to parametrize each  $f_j$  according to some parametric class of monotone functions such as the Box-Cox family, and then find the maximum likelihood estimates of  $(\Omega, f_1, \dots, f_d)$  in that class. This might lead to estimates of  $f_j$  that depend on  $\Omega$ , and vice versa, and the estimation problem would not in general be convex. Alternatively, due to (4.4), the marginal information could be used to estimate the parameters. Our nonparametric approach to estimating the transformations has the advantages of making few assumptions and being easy to compute. In the following section we analyze the theoretical properties of this estimator.

#### 4.5 THEORETICAL PROPERTIES

In this section we present our theoretical results on risk consistency, model selection consistency, and norm consistency of the covariance  $\Sigma$  and inverse covariance  $\Omega$ . From Lemma 4.3, the estimate of the graph does not depend on  $\sigma_j$ ,  $j \in \{1, \dots, d\}$  and  $\mu$ , so we assume that  $\sigma_j = 1$  and  $\mu = 0$ . Our key technical result is an analysis of the covariance of the Winsorized estimator defined in (4.8), (4.9), and (4.11). In particular, we show that under appropriate conditions,

$$\max_{j,k} |S_n(\tilde{f})_{jk} - S_n(f)_{jk}| = o_P(1)$$

where  $S_n(\tilde{f})_{jk}$  denotes the  $(j, k)$  entry of the matrix. This result allows us to leverage the recent analysis of Rothman et al. [2008] and Ravikumar et al. [2009b] in the Gaussian case to obtain consistency results for the nonparanormal. More precisely, our main theorem is the following.

**Theorem 4.1.** Suppose that  $d = n^\xi$  and let  $\tilde{f}$  be the Winsorized estimator defined in (4.9) with  $\delta_n = \frac{1}{4n^{1/4}\sqrt{\pi \log n}}$ . Define

$$C_M \equiv \frac{48}{\sqrt{\pi}} \left( \sqrt{2M} - 1 \right) (M + 2). \quad (4.14)$$

For some  $M \geq 2(\xi + 1)$ .

Then for any  $\epsilon \geq C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$  and sufficiently large  $n$ , we have

$$\mathbb{P} \left( \max_{jk} |S_n(\tilde{f})_{jk} - S_n(f)_{jk}| > 2\epsilon \right) \quad (4.15)$$

$$\leq \frac{1}{2\sqrt{\pi \log(nd)}} + 2 \exp \left( 2 \log d - \frac{n^{1/2} \epsilon^2}{1232 \pi^2 \log^2 n} \right) \quad (4.16)$$

$$+ 2 \exp \left( 2 \log d - \frac{n^{1/2}}{8 \pi \log n} \right) + o(1). \quad (4.17)$$

The proof of the above theorem is given in Section 4.8. The following corollary is immediate, and specifies the scaling of the dimension in terms of sample size.

**Corollary 4.1.** Let  $M \geq \max\{15\pi, 2\xi + 1\}$ . Then

$$\mathbb{P} \left( \max_{jk} |S_n(\tilde{f})_{jk} - S_n(f)_{jk}| > 2C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}} \right) = o(1). \quad (4.18)$$

Hence,

$$\max_{jk} |S_n(\tilde{f})_{jk} - S_n(f)_{jk}| = O_P \left( \sqrt{\frac{\log d \log^2 n}{n^{1/2}}} \right).$$

The following corollary yields estimation consistency in both the Frobenius norm and the  $\ell_2$ -operator norm. The proof follows the same arguments as the proof of Theorem 1 and Theorem 2 from Rothman et al. [2008], replacing their Lemma 1 with our Theorem 4.1.

For a matrix  $A = (a_{ij})$ , the Frobenius norm  $\|\cdot\|_F$  is defined as  $\|A\|_F \equiv \sqrt{\sum_{i,j} a_{ij}^2}$ . The  $\ell_2$ -operator norm  $\|\cdot\|_2$  is defined as the magnitude of the largest eigenvalue of the matrix,  $\|A\|_2 \equiv \max_{\|x\|_2=1} \|Ax\|_2$ . In the following, we write  $a_n \asymp b_n$  if there are positive constants  $c$  and  $C$  independent of  $n$  such that  $c \leq a_n/b_n \leq C$ .

**Corollary 4.2.** Suppose that the data are generated as  $X^{(i)} \sim \text{NPN}(\mu_0, \Sigma_0, f_0)$ , and let  $\Omega_0 = \Sigma_0^{-1}$ . If the regularization parameter  $\lambda_n$  is chosen as

$$\lambda_n \asymp 2C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$$

where  $C_M$  is defined in Theorem 4.1. Then the nonparanormal estimator  $\hat{\Omega}_n$  of (4.13) satisfies

$$\|\hat{\Omega}_n - \Omega_0\|_F = O_P \left( \sqrt{\frac{(s+d)(\log d \log^2 n)}{n^{1/2}}} \right) \quad (4.19)$$

and

$$\|\hat{\Omega}_n - \Omega_0\|_2 = O_P \left( \sqrt{\frac{s(\log d \log^2 n)}{n^{1/2}}} \right), \quad (4.20)$$

where

$$s \equiv \text{Card}(\{(i, j) \in \{1, \dots, d\} \times \{1, \dots, d\} \mid \Omega_0(i, j) \neq 0, i \neq j\}) \quad (4.21)$$

is the number of nonzero off-diagonal elements of the true precision matrix.

To prove the model selection consistency result, we need further assumptions. We follow Ravikumar (2009) and let the  $d^2 \times d^2$  Fisher information matrix of  $\Sigma_0$  be  $\Gamma \equiv \Sigma_0 \otimes \Sigma_0$  where  $\otimes$  is the Kronecker matrix product, and define the support set  $S$  of  $\Omega_0 = \Sigma_0^{-1}$  as

$$S \equiv \{(i, j) \in \{1, \dots, d\} \times \{1, \dots, d\} \mid \Omega_0(i, j) \neq 0\}. \quad (4.22)$$

We use  $S^c$  to denote the complement of  $S$  in the set  $\{1, \dots, d\} \times \{1, \dots, d\}$ , and for any two subsets  $T$  and  $T'$  of  $\{1, \dots, d\} \times \{1, \dots, d\}$ , we use  $\Gamma_{TT'}$  to denote the sub-matrix with rows and columns of  $\Gamma$  indexed by  $T$  and  $T'$  respectively.

**Assumption 4.1.** *There exists some  $\alpha \in (0, 1]$ , such that  $\|\Gamma_{S^c S}(\Gamma_{SS})^{-1}\|_\infty \leq 1 - \alpha$ .*

As in Ravikumar et al. [2009b], we define two quantities  $K_{\Sigma_0} \equiv \|\Sigma_0\|_\infty$  and  $K_\Gamma \equiv \|(\Gamma_{SS})^{-1}\|_\infty$ . Further, we define the maximum row degree as

$$\text{deg} \equiv \max_{i=1, \dots, d} \text{Card}(\{j \in 1, \dots, d \mid \Omega_0(i, j) \neq 0\}). \quad (4.23)$$

**Assumption 4.2.** *The quantities  $K_{\Sigma_0}$  and  $K_\Gamma$  are bounded, and there are positive constants  $C$  such that*

$$\min_{(j, k) \in S} |\Omega_0(j, k)| \geq C \sqrt{\frac{\log^3 n}{n^{1/2}}} \quad (4.24)$$

for large enough  $n$ .

The proof of the following corollary uses our Theorem 4.1 in place of Equation (12) in the analysis of Ravikumar et al. [2009b].

**Corollary 4.3.** *Suppose the regularization parameter is chosen as*

$$\lambda_n \asymp 2C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$$

where  $C(M, n, p)$  is defined in Theorem 4.1. Then the nonparanormal estimator  $\hat{\Omega}_n$  satisfies

$$\mathbb{P}(\mathcal{G}(\hat{\Omega}_n, \Omega_0)) \geq 1 - o(1) \quad (4.25)$$

where  $\mathcal{G}(\hat{\Omega}_n, \Omega_0)$  is the event

$$\{\text{sign}(\hat{\Omega}_n(j, k)) = \text{sign}(\Omega_0(j, k)), \quad \forall j, k \in S\}. \quad (4.26)$$

Our persistency (risk consistency) result parallels the persistency result for additive models given in Ravikumar et al. [2009a], and allows model dimension that grows exponentially with sample size. The definition in this theorem uses the fact (from Lemma 4.4) that  $\sup_x \Phi^{-1}(\tilde{F}_j(x)) \leq \sqrt{2 \log n}$  when  $\delta_n = 1/(4n^{1/4} \sqrt{\pi \log n})$ .

In the next theorem, we do not assume the true model is nonparanormal and define the population and sample risks as

$$R(f, \Omega) = \frac{1}{2} \left\{ \text{tr} [\Omega \mathbb{E}(f(X)f(X)^T)] - \log |\Omega| - p \log(2\pi) \right\} \quad (4.27)$$

$$\hat{R}(f, \Omega) = \frac{1}{2} \left\{ \text{tr} [\Omega S_n(f)] - \log |\Omega| - p \log(2\pi) \right\}. \quad (4.28)$$

**Theorem 4.2.** *Suppose that  $d \leq e^{n^\xi}$  for some  $\xi < 1$ , and define the classes*

$$\mathcal{M}_n = \left\{ f : \mathbb{R} \rightarrow \mathbb{R} : f \text{ is monotone with } \|f\|_\infty \leq C \sqrt{\log n} \right\} \quad (4.29)$$

$$\mathcal{C}_n = \left\{ \Omega : \|\Omega^{-1}\|_1 \leq L_n \right\}. \quad (4.30)$$

Let  $\hat{\Omega}_n$  be given by

$$\hat{\Omega}_n = \arg \min_{\Omega \in \mathcal{C}_n} \left\{ \text{tr} (\Omega S_n(\tilde{f})) - \log |\Omega| \right\}. \quad (4.31)$$

Then

$$R(\tilde{f}_n, \hat{\Omega}_n) - \inf_{(f, \Omega) \in \mathcal{M}_n^p \oplus \mathcal{C}_n} R(f, \Omega) = O_P \left( L_n \sqrt{\frac{\log n}{n^{1-\xi}}} \right).$$

Hence the Winsorized estimator of  $(f, \Omega)$  with  $\delta_n = 1/(4n^{1/4} \sqrt{\pi \log n})$  is persistent over  $\mathcal{C}_n$  when  $L_n = o(n^{(1-\xi)/2} / \sqrt{\log n})$ .

The proofs of Theorems 4.1 and 4.2 are given in Section 4.8.

## 4.6 EXPERIMENTAL RESULTS

In this section, we report experimental results on synthetic and real data sets. We mainly compare the  $\ell_1$ -regularized nonparanormal and Gaussian (paranormal) models, computed using the graphical lasso algorithm (glasso) of [Friedman et al. \[2007\]](#). The primary conclusions are: (i) When the data are multivariate Gaussian, the performance of the two methods is comparable; (ii) when the model is correct, the nonparanormal performs much better than the graphical lasso in many cases; (iii) for a particular gene microarray data set, our method behaves differently from the graphical lasso, and may support different biological conclusions.

Note that we can reuse the glasso implementation to fit a sparse nonparanormal. In particular, after computing the Winsorized sample covariance  $S_n(\tilde{f})$ , we pass this matrix to the glasso routine to carry out the optimization

$$\hat{\Omega}_n = \arg \min_{\Omega} \{ \text{tr}(\Omega S_n(\tilde{f})) - \log |\Omega| + \lambda_n \|\Omega\|_1 \}. \quad (4.32)$$

## 4.6.1 Neighborhood Graphs

We begin by describing a procedure to generate graphs as in [\[Meinshausen and Bühlmann, 2006\]](#), with respect to which several distributions can then be defined. We generate a  $d$ -dimensional sparse graph  $G \equiv (V, E)$  as follows: Let  $V = \{1, \dots, d\}$  correspond to variables  $X = (X_1, \dots, X_d)$ . We associate each index  $j$  with a point  $(Y_j^{(1)}, Y_j^{(2)}) \in [0, 1]^2$  where

$$Y_1^{(k)}, \dots, Y_n^{(k)} \sim \text{Uniform}[0, 1]$$

for  $k = 1, 2$ . Each pair of nodes  $(i, j)$  is included in the edge set  $E$  with probability

$$\mathbb{P}\left((i, j) \in E\right) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|y_i - y_j\|_n^2}{2s}\right) \quad (4.33)$$

where  $y_i \equiv (y_i^{(1)}, y_i^{(2)})$  is the observation of  $(Y_i^{(1)}, Y_i^{(2)})$  and  $\|\cdot\|_n$  represents the Euclidean distance. Here,  $s = 0.125$  is a parameter that controls the sparsity level of the generated graph. We restrict the maximum degree of the graph to be four and build the inverse covariance matrix  $\Omega_0$  according to

$$\Omega_0(i, j) = \begin{cases} 1 & \text{if } i = j \\ 0.245 & \text{if } (i, j) \in E \\ 0 & \text{otherwise,} \end{cases} \quad (4.34)$$

where the value 0.245 guarantees positive definiteness of the inverse covariance matrix.

Given  $\Omega_0$ ,  $n$  data points are sampled from

$$X^{(1)}, \dots, X^{(n)} \sim \text{NPN}(\mu_0, \Sigma_0, f_0) \quad (4.35)$$

where  $\mu_0 = (1.5, \dots, 1.5)$ ,  $\Sigma_0 = \Omega_0^{-1}$ . For simplicity, the transformation functions for all dimensions are the same,  $f_1 = \dots = f_d = f$ . To sample data from the nonparanormal distribution, we also require  $g \equiv f^{-1}$ ; two different transformations  $g$  are employed.

**Definition 4.1.** (Gaussian CDF Transformation) *Let  $g_0$  be a one-dimensional Gaussian cumulative distribution function with mean  $\mu_{g_0}$  and the standard deviation  $\sigma_{g_0}$ , that is,*

$$g_0(t) \equiv \Phi\left(\frac{t - \mu_{g_0}}{\sigma_{g_0}}\right). \quad (4.36)$$

We define the transformation function  $g_j = f_j^{-1}$  for the  $j$ -th dimension as

$$g_j(z_j) \equiv \sigma_j \left( \frac{g_0(z_j) - \int g_0(t) \phi\left(\frac{t - \mu_j}{\sigma_j}\right) dt}{\sqrt{\int \left( g_0(y) - \int g_0(t) \phi\left(\frac{t - \mu_j}{\sigma_j}\right) dt \right)^2 \phi\left(\frac{y - \mu_j}{\sigma_j}\right) dy}} \right) + \mu_j$$

where  $\sigma_j = \Sigma_0(j, j)$ .

**Definition 4.2.** (Symmetric Power Transformation) *Let  $g_0$  be the symmetric and odd transformation given by*

$$g_0(t) = \text{sign}(t)|t|^\alpha \quad (4.37)$$

where  $\alpha > 0$  is a parameter. We define the power transformation for the  $j$ -th dimension as

$$g_j(z_j) \equiv \sigma_j \left( \frac{g_0(z_j - \mu_j)}{\sqrt{\int g_0^2(t - \mu_j) \phi\left(\frac{t - \mu_j}{\sigma_j}\right) dt}} \right) + \mu_j. \quad (4.38)$$

These transformation are constructed to preserve the marginal mean and standard deviation. In the following experiments, we refer to them as the cdf transformation and the power transformation, respectively. For the cdf transformation, we set  $\mu_{g_0} = 0.05$  and  $\sigma_{g_0} = 0.4$ . For the power transformation, we set  $\alpha = 3$ .

To visualize these two transformations, we sample 5000 data points from a one-dimensional normal distribution  $N(0.5, 1.0)$  and then apply the above two transformations; the results are shown in Figure 22. It can be seen how the cdf and power transformations map a univariate normal distribution into a highly skewed and a bi-modal distribution, respectively.

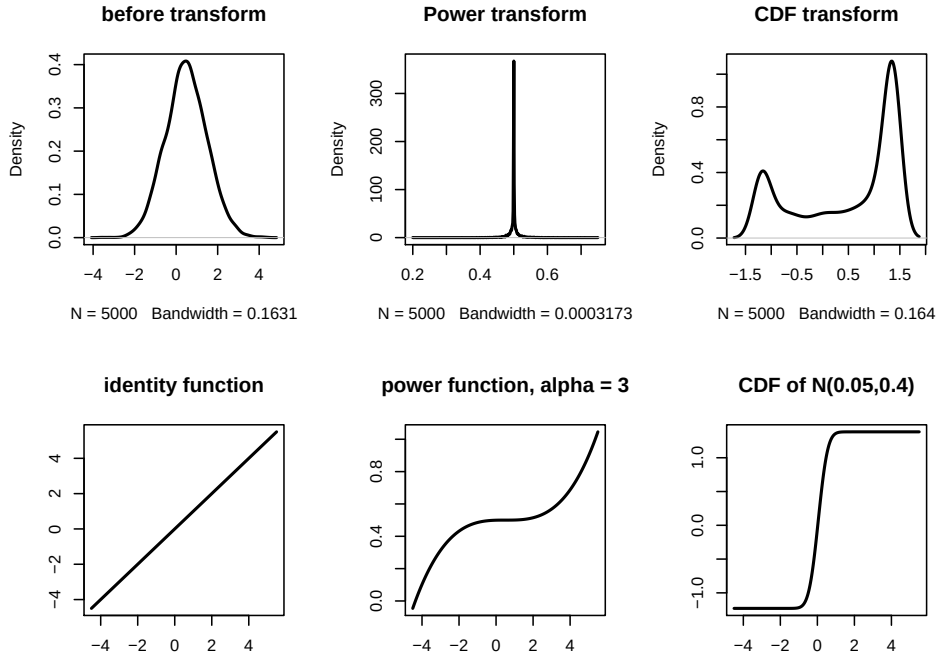


Figure 22.: The power and cdf transformations. The densities are estimated using a kernel density estimator with bandwidths selected by cross-validation.

To generate synthetic data, we set  $d = 40$ , resulting in  $\binom{40}{2} + 40 = 820$  parameters to be estimated, and vary the sample sizes from  $n = 200$  to  $n = 1000$ . Three conditions are considered, corresponding to using the cdf transform, the power transform, or no transformation. In each case, both the glasso and the nonparanormal are applied to estimate the graph.

#### 4.6.1.1 Comparison of Regularization Paths

We choose a set of regularization parameters  $\Lambda$ ; for each  $\lambda \in \Lambda$ , we obtain an estimate  $\hat{\Omega}_n$  which is a  $40 \times 40$  matrix. The upper triangular matrix has 780 parameters; we vectorize it to get a 780-dimensional parameter vector. A regularization path is the trace of these parameters over all the regularization parameters within  $\Lambda$ . The regularization paths for both methods are plotted in Figure 23. For the cdf transformation and the power transformation, the nonparanormal separates the relevant and the irrelevant dimensions very well. For the glasso, relevant variables are mixed with irrelevant variables. If no transformation is applied, the paths for both methods are almost the same.

#### 4.6.1.2 Estimated Transformations

For sample size  $n = 1000$ , we plot the estimated transformations for three of the variables in Figure 24. It is clear that Winsorization plays a significant role for the power transformation. This is intuitive due to the high skewness of the nonparanormal distribution in this case.

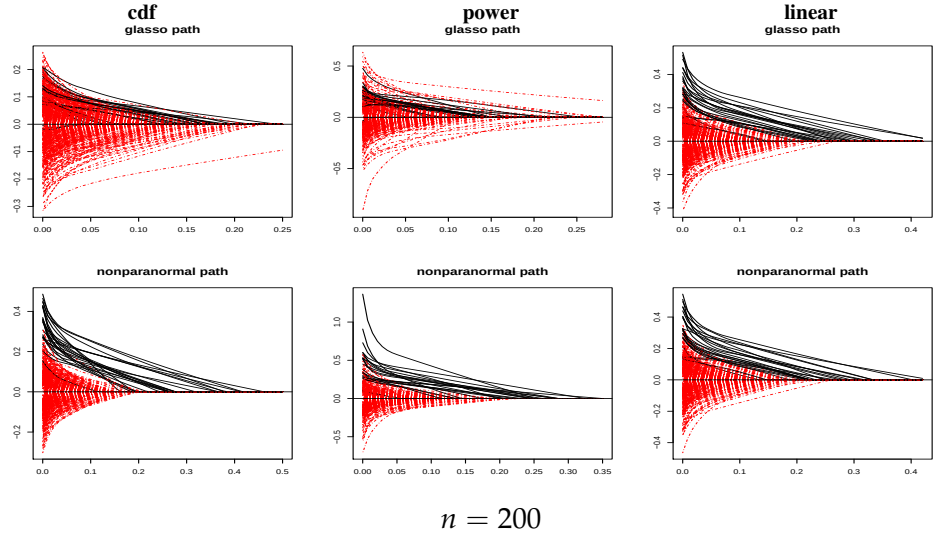
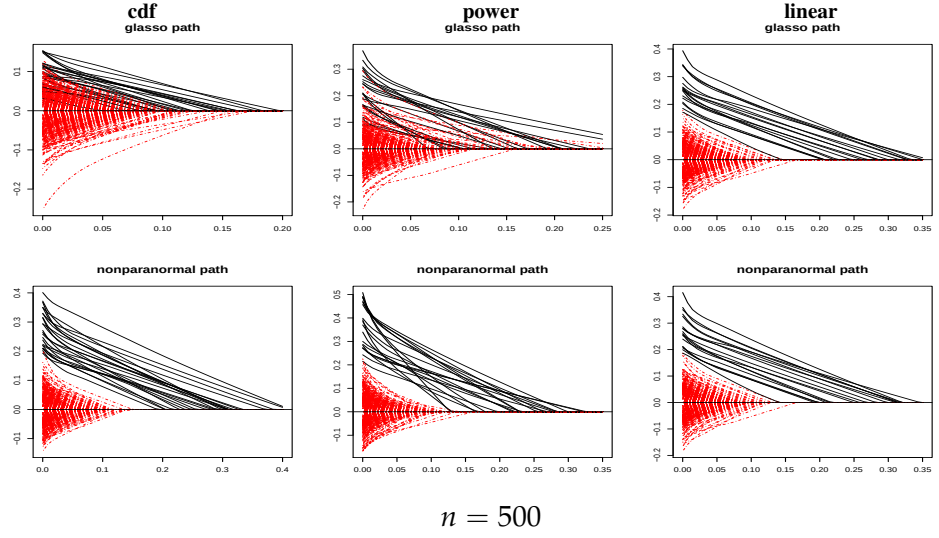


Figure 23.: Regularization paths for the glasso and nonparanormal with  $n = 500$  (top) and  $n = 200$  (bottom). The paths for the relevant variables (nonzero inverse covariance entries) are plotted as solid (black) lines; the paths for the irrelevant variables are plotted as dashed (red) lines. For non-Gaussian distributions, the nonparanormal better separates the relevant and irrelevant dimensions.



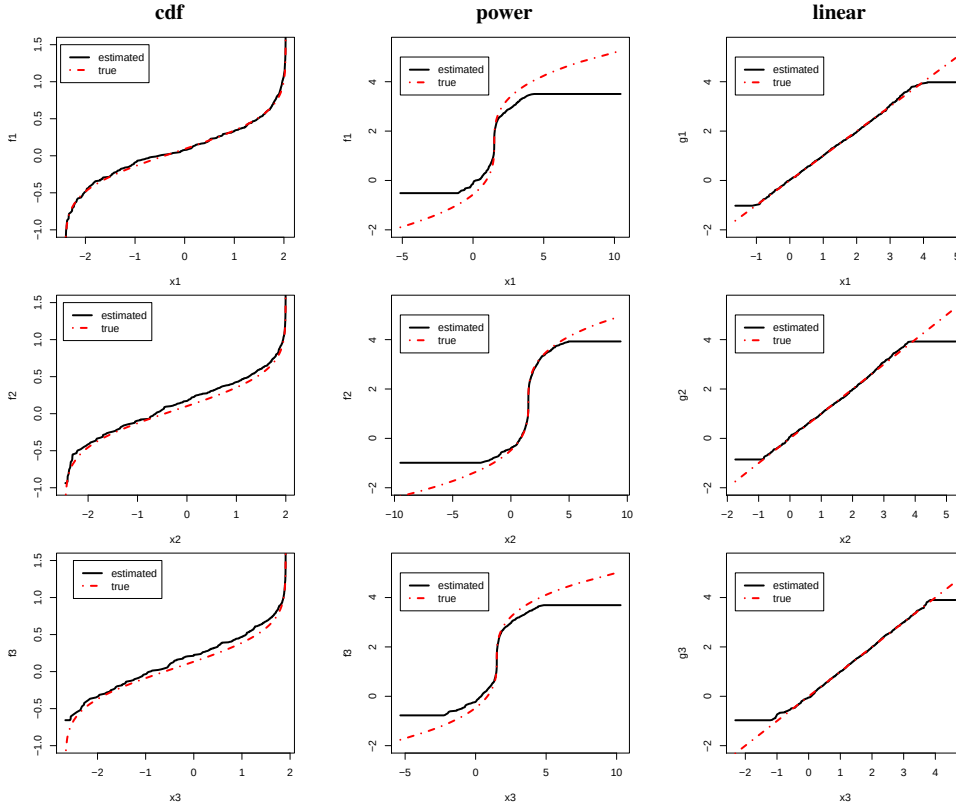


Figure 24.: Estimated transformations for the first three variables. Winsorization plays a significant role for the power transformation due to its high skewness.

#### 4.6.1.3 Quantitative Comparison

To evaluate the performance for structure estimation quantitatively, we use false positive and false negative rates. Let  $G = (V, E)$  be a  $d$ -dimensional graph (which has at most  $\binom{d}{2}$  edges) in which there are  $|E| = r$  edges, and let  $\hat{G}^\lambda = (V, \hat{E}^\lambda)$  be an estimated graph using the regularization parameter  $\lambda$ . The number of false positives at  $\lambda$  is

$$\text{FP}(\lambda) \equiv \text{number of edges in } \hat{E}^\lambda \text{ not in } E \quad (4.39)$$

The number of false negatives at  $\lambda$  is defined as

$$\text{FN}(\lambda) \equiv \text{number of edges in } E \text{ not in } \hat{E}^\lambda. \quad (4.40)$$

The oracle regularization level  $\lambda^*$  is then

$$\lambda^* = \arg \min_{\lambda \in \Lambda} \{\text{FP}(\lambda) + \text{FN}(\lambda)\}. \quad (4.41)$$

The oracle score is  $\text{FP}(\lambda^*) + \text{FN}(\lambda^*)$ . Figure 25 shows boxplots of the oracle scores for the two methods, calculated using 100 simulations.

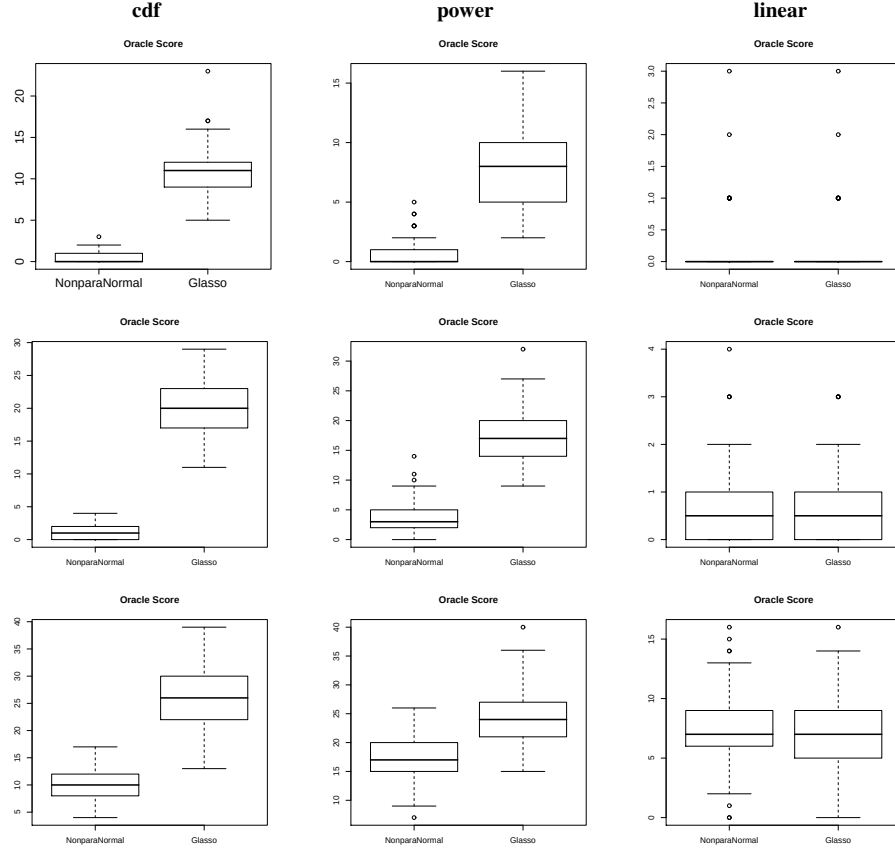


Figure 25.: Boxplots of the oracle scores for  $n = 1000, 500, 200$  (top, center, bottom).

To illustrate the overall performance of these two methods over the full paths, ROC curves are shown in Figure 26, using

$$\left(1 - \frac{\text{FN}(\lambda)}{r}, 1 - \frac{\text{FP}(\lambda)}{\binom{d}{2} - r}\right). \quad (4.42)$$

The curves clearly show how the performance of both methods improves with sample size, and that the nonparanormal is superior to the Gaussian model in most cases.

Let  $\text{FPE} \equiv \text{FP}(\lambda^*)$  and  $\text{FNE} \equiv \text{FN}(\lambda^*)$ , Tables 1, 2, and 3 provide numerical comparisons of both methods on data sets with different transformations, where we repeat the experiments 100 times and report the average FPE and FNE values with the corresponding standard deviations. It's clear from the tables that the nonparanormal achieves significantly smaller errors than the glasso if the true distribution of the data is not multivariate Gaussian and achieves performance comparable to the glasso when the true distribution is exactly multivariate Gaussian.

Figure 27 shows typical runs for the cdf and power transformations. It's clear that when the glasso estimates the graph incorrectly, the mistakes include both false positives and negatives.

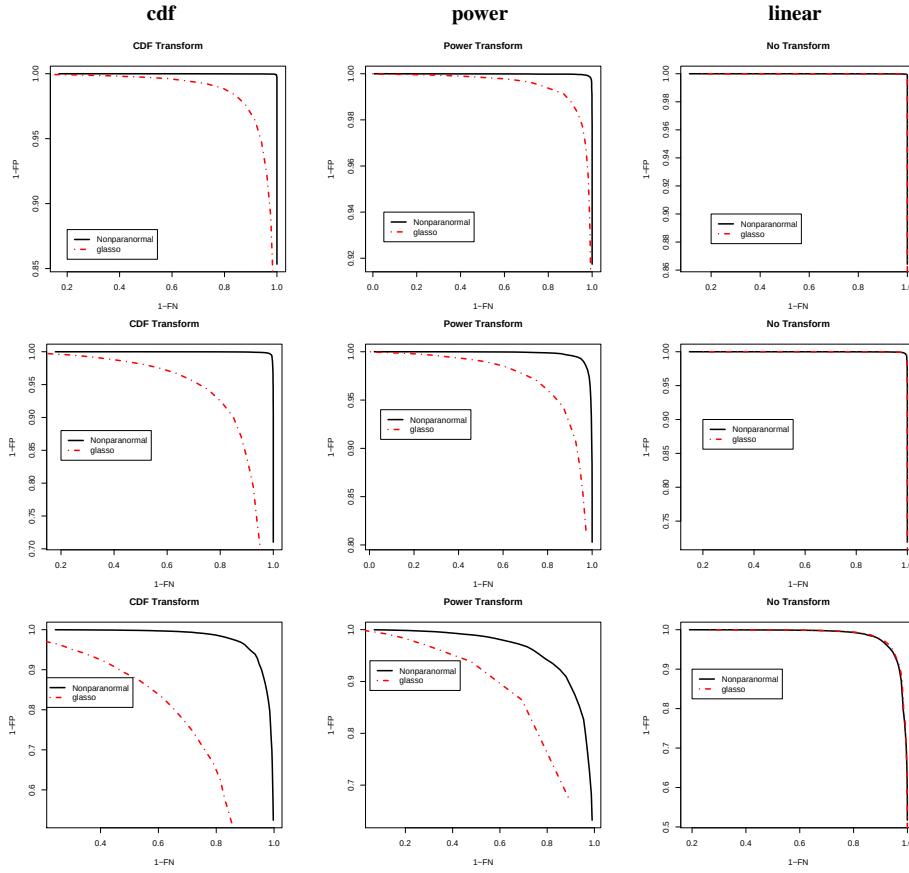


Figure 26.: ROC curves for sample sizes  $n = 1000, 500, 200$  (top, middle, bottom).

#### 4.6.1.4 Comparison in the Gaussian Case

The previous experiments indicate that the nonparanormal works almost as well as the glasso in the Gaussian case. This initially appears surprising, since a parametric method is expected to be more efficient than a nonparametric method if the parametric assumption is correct. To manifest this efficiency loss, we conducted some experiments with very small  $n$  and relatively large  $d$ . For multivariate Gaussian models, Figure 28 shows results with  $(n, d, s) = (50, 40, 1/8), (50, 100, 1/15)$  and  $(30, 100, 1/15)$ . From the mean ROC curves, we see that nonparanormal does indeed behave worse than the glasso, suggesting some efficiency loss. However, from the corresponding boxplots, the efficiency reduction is relatively insignificant.

#### 4.6.1.5 The Case When $d \gg n$

Figure 29 shows results from a simulation of the nonparanormal using cdf transformations with  $n = 200, d = 500$  and sparsity level  $s = 1/40$ . The boxplot shows that the nonparanormal outperforms the glasso. A typical

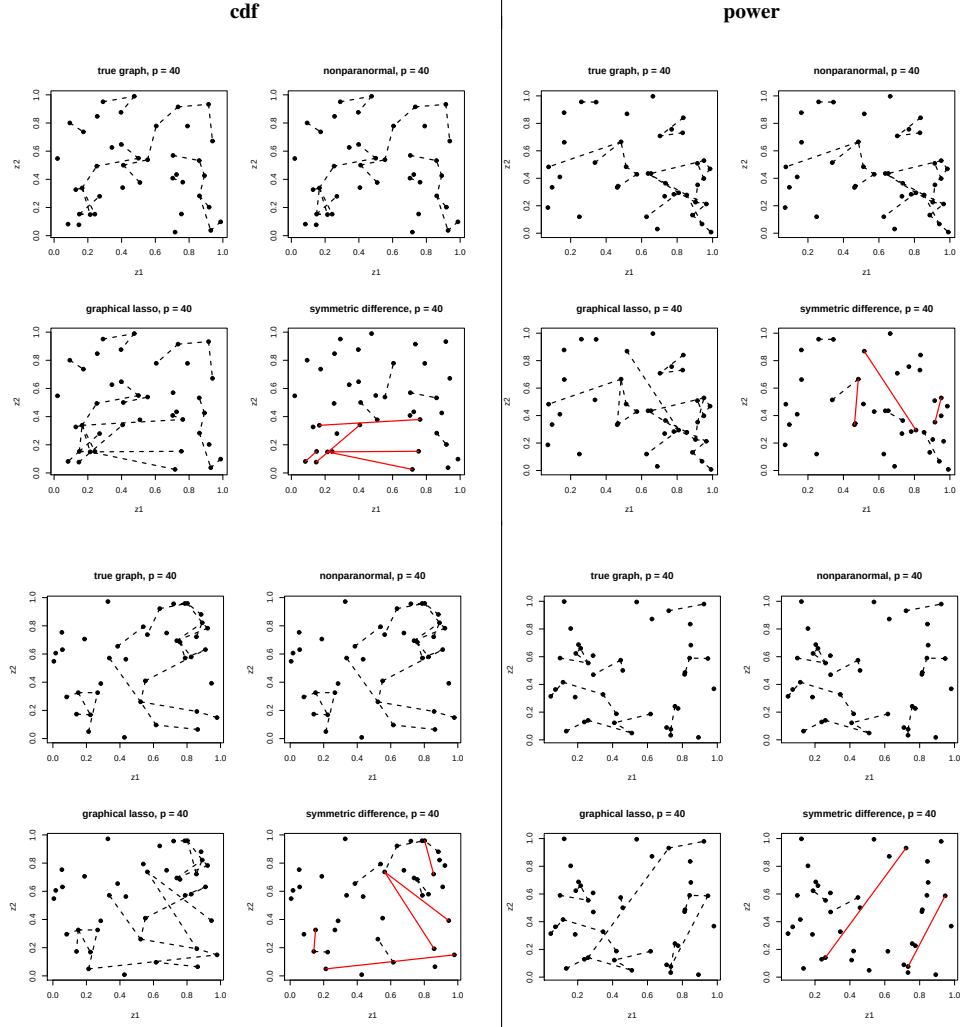


Figure 27.: Typical runs for the two methods for  $n = 1000$  using the cdf and power transformations. The dashed (black) lines in the symmetric difference plots indicate edges found by the glasso but not the nonparanormal, and vice-versa for the solid (red) lines.

| $n$  | Nonparanormal |           |      |           | glasso |           |       |           |
|------|---------------|-----------|------|-----------|--------|-----------|-------|-----------|
|      | FPE           | (sd(FPE)) | FNE  | (sd(FNE)) | FPE    | (sd(FPE)) | FNE   | (sd(FNE)) |
| 1000 | 0.10          | (0.3333)  | 0.05 | (0.2190)  | 3.73   | (2.3904)  | 7.24  | (3.2910)  |
| 900  | 0.18          | (0.5389)  | 0.16 | (0.4197)  | 3.31   | (2.4358)  | 8.94  | (3.2808)  |
| 800  | 0.16          | (0.5069)  | 0.23 | (0.5659)  | 3.80   | (2.9439)  | 9.91  | (3.4789)  |
| 700  | 0.26          | (0.6295)  | 0.43 | (0.7420)  | 3.45   | (2.5519)  | 12.26 | (3.5862)  |
| 600  | 0.33          | (0.6039)  | 0.41 | (0.6371)  | 3.31   | (2.8804)  | 14.25 | (4.0735)  |
| 500  | 0.58          | (0.9658)  | 1.10 | (1.0396)  | 3.18   | (2.9211)  | 17.54 | (4.4368)  |
| 400  | 0.71          | (1.0569)  | 1.52 | (1.2016)  | 1.58   | (2.3535)  | 21.18 | (4.9855)  |
| 300  | 1.37          | (1.4470)  | 2.97 | (2.0123)  | 0.67   | (1.6940)  | 23.14 | (5.0232)  |
| 200  | 2.03          | (1.9356)  | 7.13 | (3.4514)  | 0.01   | (0.1000)  | 24.03 | (4.9816)  |

Table 1.: Quantitative comparison on the data set using the cdf transformation. For both FPE and FNE, the nonparanormal performs much better in general.

run of the regularization paths confirms this conclusion, showing that the nonparanormal path separates the relevant and irrelevant dimensions very well. In contrast, with the glasso the relevant variables are “buried” among the irrelevant variables.

#### 4.6.2 Gene Microarray Data

In this study, we consider a data set based on Affymetrix GeneChip microarrays for the plant *Arabidopsis thaliana*, [Wille et al., 2004]. The sample size is  $n = 118$ . The expression levels for each chip are pre-processed by log-transformation and standardization. A subset of 40 genes from the isoprenoid pathway are chosen, and we study the associations among them using both the paranormal and nonparanormal models. Even though these data are generally treated as multivariate Gaussian in the previous analysis [Wille et al., 2004], our study shows that the results of the nonparanormal and the glasso are very different over a wide range of regularization parameters. This suggests the nonparanormal could support different scientific conclusions.

##### 4.6.2.1 Comparison of the Regularization Paths

We first compare the regularization paths of the two methods, in Figure 30. To generate the paths, we select 50 regularization parameters on an evenly spaced grid in the interval  $[0.16, 1.2]$ . Although the paths for the two methods

| $n$  | Nonparanormal |           |       |           | glasso |           |       |           |
|------|---------------|-----------|-------|-----------|--------|-----------|-------|-----------|
|      | FPE           | (sd(FPE)) | FNE   | (sd(FNE)) | FPE    | (sd(FPE)) | FNE   | (sd(FNE)) |
| 1000 | 0.27          | (0.7086)  | 0.35  | (0.6571)  | 2.89   | (1.9482)  | 4.97  | (2.9213)  |
| 900  | 0.38          | (0.6783)  | 0.41  | (0.6210)  | 2.98   | (2.3697)  | 5.99  | (3.0467)  |
| 800  | 0.25          | (0.5751)  | 0.73  | (0.8270)  | 4.10   | (2.7834)  | 6.39  | (3.3571)  |
| 700  | 0.69          | (0.9067)  | 0.90  | (1.0200)  | 4.42   | (2.8891)  | 8.80  | (3.9848)  |
| 600  | 0.92          | (1.2282)  | 1.59  | (1.5314)  | 4.64   | (3.3830)  | 10.58 | (4.2168)  |
| 500  | 1.17          | (1.3413)  | 2.56  | (2.3325)  | 4.00   | (2.9644)  | 13.09 | (4.4903)  |
| 400  | 1.88          | (1.6470)  | 4.97  | (2.7687)  | 3.14   | (3.4699)  | 17.87 | (4.7750)  |
| 300  | 2.97          | (2.4181)  | 7.85  | (3.5572)  | 1.36   | (2.3805)  | 21.24 | (4.7505)  |
| 200  | 2.82          | (2.6184)  | 14.53 | (4.3378)  | 0.37   | (0.9914)  | 24.01 | (5.0940)  |

Table 2.: Quantitative comparison on the data set using the power transformation. For both FPE and FNE, the nonparanormal performs much better in general.

look similar, there are some subtle differences. In particular, variables become nonzero in a different order, especially when the regularization parameter is in the range  $\lambda \in [0.2, 0.3]$ . As shown below, these subtle differences in the paths lead to different model selection behaviors.

#### 4.6.2.2 Comparison of the Estimated Graphs

Figure 31 compares the estimated graphs for the two methods at several values of the regularization parameter  $\lambda$  in the range  $[0.16, 0.37]$ . For each  $\lambda$ , we show the estimated graph from the nonparanormal in the first column. In the second column we show the graph obtained by scanning the full regularization path of the glasso fit and finding the graph having the smallest symmetric difference with the nonparanormal graph. The symmetric difference graph is shown in the third column. The closest glasso fit is different, with edges selected by the glasso not selected by the nonparanormal, and vice-versa. Several estimated transformations are plotted in Figure 32, which are nonlinear. Interestingly, several of the differences between the fitted graphs are related to these variables.

## 4.7 CONCLUDING REMARKS

In this paper we have introduced the nonparanormal, a type of Gaussian copula with nonparametric marginals that is suitable for estimating high

| $n$  | Nonparanormal |           |      |           | glasso |           |      |           |
|------|---------------|-----------|------|-----------|--------|-----------|------|-----------|
|      | FPE           | (sd(FPE)) | FNE  | (sd(FNE)) | FPE    | (sd(FPE)) | FNE  | (sd(FNE)) |
| 1000 | 0.10          | (0.3333)  | 0.05 | (0.2190)  | 0.09   | (0.3208)  | 0.06 | (0.2386)  |
| 900  | 0.24          | (0.7537)  | 0.14 | (0.4025)  | 0.22   | (0.6447)  | 0.15 | (0.4113)  |
| 800  | 0.17          | (0.4277)  | 0.16 | (0.3949)  | 0.16   | (0.4431)  | 0.19 | (0.4191)  |
| 700  | 0.25          | (0.6871)  | 0.33 | (0.8534)  | 0.29   | (0.8201)  | 0.27 | (0.7501)  |
| 600  | 0.37          | (0.7740)  | 0.36 | (0.7456)  | 0.36   | (0.7722)  | 0.37 | (0.6459)  |
| 500  | 0.28          | (0.5874)  | 0.46 | (0.7442)  | 0.25   | (0.5573)  | 0.45 | (0.6571)  |
| 400  | 0.55          | (0.8453)  | 1.37 | (1.2605)  | 0.47   | (0.7713)  | 1.35 | (1.2502)  |
| 300  | 1.24          | (1.3715)  | 3.07 | (1.7306)  | 0.98   | (1.2058)  | 3.04 | (1.8905)  |
| 200  | 1.62          | (1.7219)  | 5.89 | (2.7373)  | 1.55   | (1.6779)  | 5.62 | (2.6620)  |

Table 3.: Quantitative comparison on the data set without any transformation. The two methods behave similarly, the glasso is slightly better.

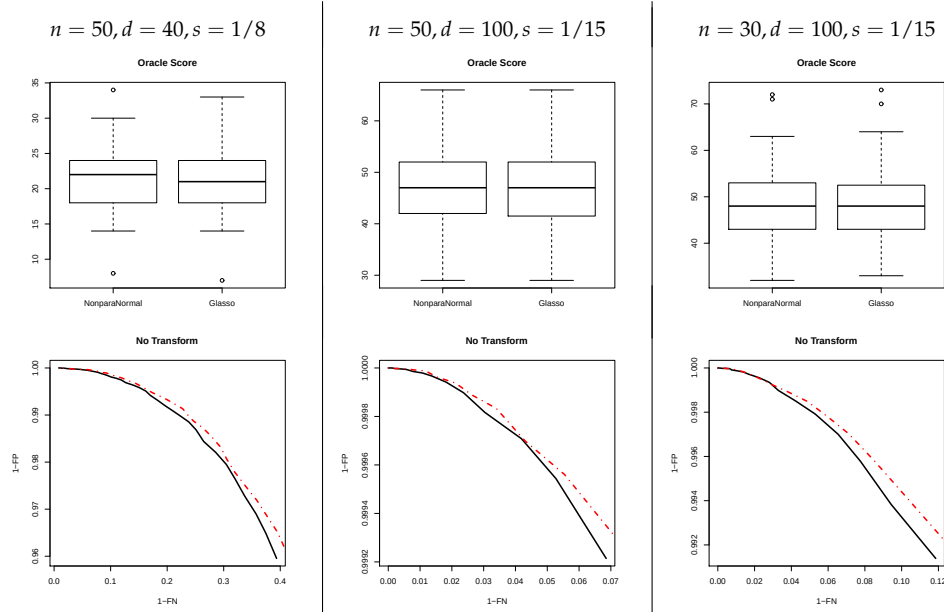


Figure 28.: For Gaussian models, comparison of boxplots of the oracle scores and ROC curves for small  $n$  and relatively large  $d$ . The ROC curves suggest some efficiency loss of the nonparanormal; however, the corresponding boxplots indicate this loss is insignificant.

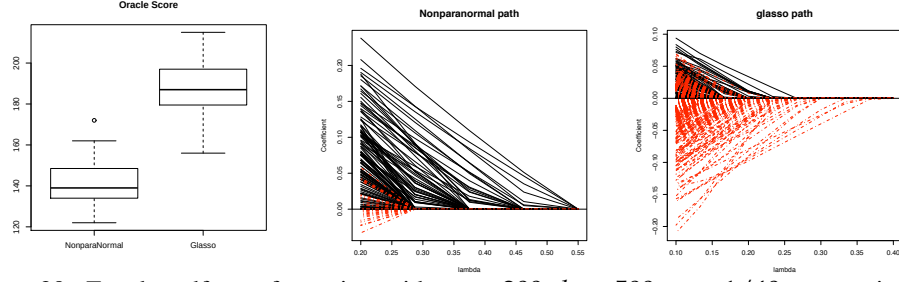


Figure 29.: For the cdf transformation with  $n = 200, d = 500, s = 1/40$ , comparison of the boxplots and a typical run of the regularization paths. The nonparanormal paths separate the relevant from the irrelevant dimensions well. For the glasso, the relevant variables are “buried” in irrelevant variables.

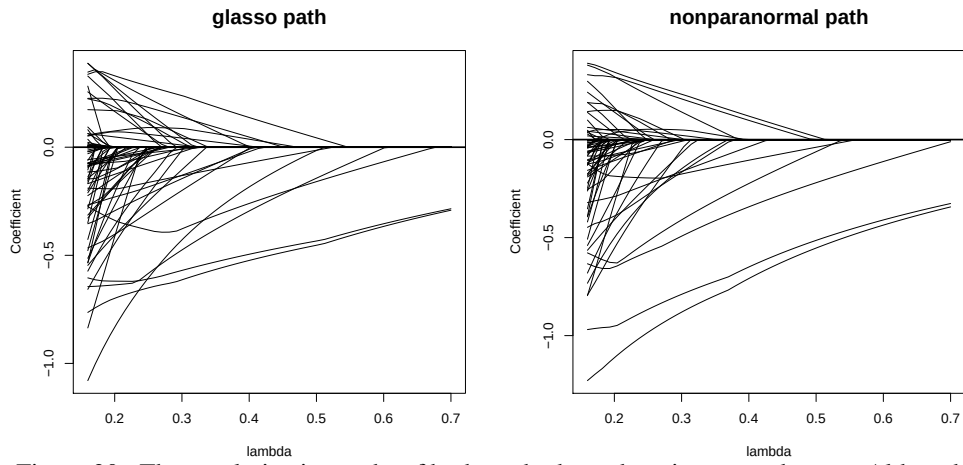


Figure 30.: The regularization paths of both methods on the microarray data set. Although the paths for the two methods look similar, there are some subtle differences.

dimensional undirected graphs. The nonparanormal can be viewed as an extension of sparse additive models to the setting of graphical models. We proposed an estimator for the component functions that is based on thresholding the tails of the empirical distribution function at appropriate levels. A theoretical analysis was given to bound the difference between the sample covariance with respect to these estimated functions and the true sample covariance. This analysis was leveraged with the recent work of [Ravikumar et al. \[2009b\]](#) and [Rothman et al. \[2008\]](#) to obtain consistency results for the nonparanormal. Computationally, fitting a high dimensional nonparanormal is no more difficult than estimating a multivariate Gaussian, and indeed one can exploit existing software for the graphical lasso. Our experimental results indicate that the sparse nonparanormal can give very different results than a sparse Gaussian graphical model. This suggests that it may be a useful tool for relaxing the normality assumption, which is often made only for convenience.



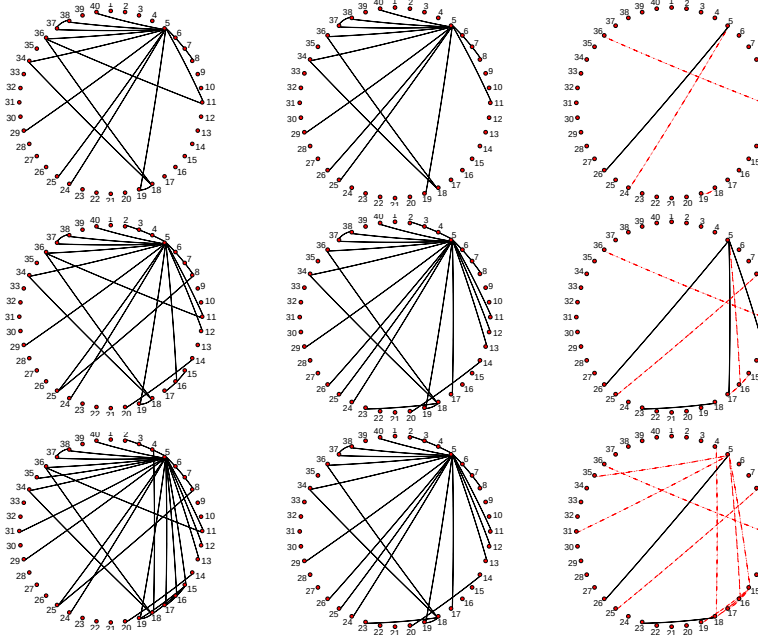


Figure 31.: The nonparanormal estimated graph for three values of  $\lambda = 0.2448, 0.2661, 0.30857$  (left column), the closest glasso estimated graph from the full path (middle) and the symmetric difference graph (right).

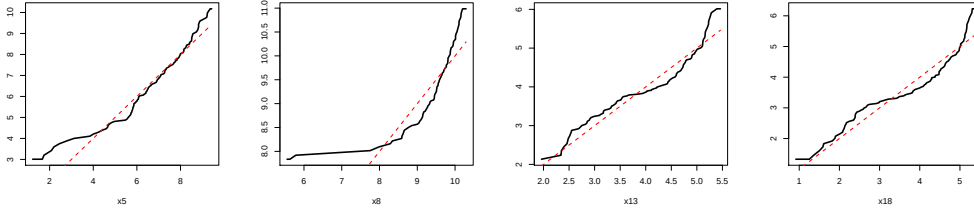


Figure 32.: Estimated transformations for the microarray data set, indicating non-Gaussian marginals. The corresponding genes are among the nodes appearing in the symmetric difference graphs above.

#### 4.8 APPENDIX: TECHNICAL PROOFS

We assume, without loss of generality from Lemma 4.3, that  $\mu_j = 0$  and  $\sigma_j = 1$  for all  $j = 1, \dots, d$ . Thus, define  $\tilde{f}_j(x) \equiv \Phi^{-1}(\tilde{F}_j(x))$  and  $f_j(x) \equiv \Phi^{-1}(F_j(x))$ , and let  $g_j \equiv f_j^{-1}$ .

##### 4.8.1 Proof of Theorem 4.1

We start with some useful lemmas; the first is from Abramovich et al. [2006].

**Lemma 4.4.** (Gaussian Distribution function vs. Quantile function) Let  $\Phi$  and  $\phi$  denote the distribution and density functions of a standard Gaussian random variable. Then

$$\frac{\phi(t)}{2t} \leq 1 - \Phi(t) \leq \frac{\phi(t)}{t} \text{ if } t \geq 1$$

and

$$(\Phi^{-1})'(\eta) = \frac{1}{\phi(\Phi^{-1}(\eta))}.$$

Also, for  $\eta \geq 0.99$ , we have

$$\Phi^{-1}(\eta) = \sqrt{2 \log \left( \frac{1}{1-\eta} \right)} - r(\eta) \quad (4.43)$$

where  $r(\eta) \in [0, 1.5]$ .

**Lemma 4.5.** (Distribution function of the transformed random variable) For any  $\alpha \in (-\infty, \infty)$

$$\Phi^{-1} \left( F_j \left( g_j(\alpha \sqrt{\log n}) \right) \right) = \alpha \sqrt{\log n}. \quad (4.44)$$

*Proof.* The statement follows from

$$F_j(t) = \mathbb{P}(X_j \leq t) = \mathbb{P}(g_j(Z_j) \leq t) \quad (4.45)$$

$$= \mathbb{P}(Z_j \leq g_j^{-1}(t)) = \Phi \left( g_j^{-1}(t) \right). \quad (4.46)$$

which holds for any  $t$ .  $\square$

**Lemma 4.6.** (Gaussian maximal inequality) Let  $W_1, \dots, W_n$  be identically distributed standard Gaussian random variables (do not have to be independent). Then for any  $\alpha > 0$

$$\mathbb{P} \left( \max_{1 \leq i \leq n} W_i > \sqrt{\alpha \log n} \right) \leq \frac{1}{n^{\alpha/2-1} \sqrt{2\pi\alpha \log n}}. \quad (4.47)$$

*Proof.* Using Mill's inequality, we have

$$\mathbb{P} \left( \max_{1 \leq i \leq n} W_i > \sqrt{\alpha \log n} \right) \quad (4.48)$$

$$\leq \sum_{i=1}^n \mathbb{P} \left( W_i > \sqrt{\alpha \log n} \right) \quad (4.49)$$

$$\leq n \frac{\phi(\sqrt{\alpha \log n})}{\sqrt{\alpha \log n}} \quad (4.50)$$

$$= \frac{1}{n^{\alpha/2-1} \sqrt{2\pi\alpha \log n}}, \quad (4.51)$$

from which the result follows.  $\square$

**Lemma 4.7.** For any  $\alpha > 0$  that satisfies  $1 - \delta_n - \Phi(\sqrt{\alpha \log n}) > 0$  for all  $n$ , we have

$$\mathbb{P} \left[ \hat{F}_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) > 1 - \delta_n \right] \quad (4.52)$$

$$\leq \exp \left\{ -2n \left( 1 - \delta_n - \Phi \left( \sqrt{\alpha \log n} \right) \right)^2 \right\}. \quad (4.53)$$

and

$$\mathbb{P} \left[ \hat{F}_j \left( g_j \left( -\sqrt{\alpha \log n} \right) \right) < \delta_n \right] \quad (4.54)$$

$$\leq \exp \left\{ -2n \left( 1 - \delta_n - \Phi \left( \sqrt{\alpha \log n} \right) \right)^2 \right\}. \quad (4.55)$$

*Proof.* Using Hoeffding's inequality,

$$\mathbb{P} \left[ \hat{F}_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) > 1 - \delta_n \right] \quad (4.56)$$

$$\begin{aligned} &= \mathbb{P} \left[ \hat{F}_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) - F_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) > 1 - \delta_n - F_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) \right] \\ &\leq \exp \left\{ -2n \left( 1 - \delta_n - F_j \left( g_j \left( \sqrt{\alpha \log n} \right) \right) \right)^2 \right\}. \end{aligned} \quad (4.57)$$

Equation (4.52) then follows from equation (4.46). The proof of equation (4.54) uses the same argument.  $\square$

Now let  $M > 2$  and set  $\beta = \frac{1}{2}$ . We split the interval

$$\left[ g_j(-\sqrt{M \log n}), g_j(\sqrt{M \log n}) \right]$$

into two parts, the middle

$$\mathcal{M}_n \equiv \left( g_j \left( -\sqrt{\beta \log n} \right), g_j \left( \sqrt{\beta \log n} \right) \right) \quad (4.58)$$

and ends

$$\mathcal{E}_n \equiv \left[ g_j \left( -\sqrt{M \log n} \right), g_j \left( -\sqrt{\beta \log n} \right) \right] \cup \left[ g_j \left( \sqrt{\beta \log n} \right), g_j \left( \sqrt{M \log n} \right) \right].$$

The behaviors of the function estimates in these two regions are different, so we first establish bounds on the probability that a sample can fall in the end region  $\mathcal{E}_n$ .

**Lemma 4.8.** Let  $A \equiv \sqrt{\frac{2}{\pi}}(\sqrt{M} - \sqrt{\beta})$ . Then

$$\mathbb{P} \left( X_j^{(1)} \in \mathcal{E}_n \right) \leq A \sqrt{\frac{\log n}{n^\beta}}, \quad \forall j \in \{1, \dots, d\}. \quad (4.59)$$

*Proof.* Using Equation (4.46) and the mean value theorem, we have

$$\mathbb{P} \left( X_j^{(1)} \in \mathcal{E}_n \right) \quad (4.60)$$

$$= \mathbb{P} \left( X_j^{(1)} \in \left[ g_j(\sqrt{\beta \log n}), g_j(\sqrt{M \log n}) \right] \right) \quad (4.61)$$

$$+ \mathbb{P} \left( X_j^{(1)} \in \left[ g_j(-\sqrt{M \log n}), g_j(-\sqrt{\beta \log n}) \right] \right) \quad (4.62)$$

$$= F_j \left( g_j(\sqrt{M \log n}) \right) - F_j \left( g_j(\sqrt{\beta \log n}) \right) \quad (4.63)$$

$$+ F_j \left( g_j(-\sqrt{\beta \log n}) \right) - F_j \left( g_j(-\sqrt{M \log n}) \right) \quad (4.64)$$

$$= 2 \left( \Phi(\sqrt{M \log n}) - \Phi(\sqrt{\beta \log n}) \right) \quad (4.65)$$

$$\leq 2\phi \left( \sqrt{\beta \log n} \right) \left( \sqrt{M \log n} - \sqrt{\beta \log n} \right). \quad (4.66)$$

The result of the lemma follows directly.  $\square$

We next bound the error of the Winsorized estimate of a component function over the end region.

**Lemma 4.9.** *For all  $n$ , we have, for all  $j \in \{1, \dots, d\}$ ,*

$$\sup_{t \in \mathcal{E}_n} \left| \Phi^{-1}(\tilde{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| < \sqrt{2(M+2) \log n}. \quad (4.67)$$

*Proof.* From Lemma 4.5 and the definition of  $\mathcal{E}_n$ , we have

$$\sup_{t \in \mathcal{E}_n} \left| \Phi^{-1}(F_j(t)) \right| \in [0, \sqrt{M \log n}].$$

Given the fact that  $\delta_n = \frac{1}{4n^{1/4} \sqrt{\pi \log n}}$ , we have  $\tilde{F}_j(t) \in \left(\frac{1}{n}, 1 - \frac{1}{n}\right)$ . Therefore, from Equation (4.43),

$$\sup_{t \in \mathcal{E}_n} \left| \Phi^{-1}(\tilde{F}_j(t)) \right| \in [0, \sqrt{2 \log n}]. \quad (4.68)$$

The result follows from the triangle inequality and  $\sqrt{M} + \sqrt{2} \leq \sqrt{2(M+2)}$ .  $\square$

Now for any  $\epsilon > 0$ , we have

$$\mathbb{P} \left( \max_{j,k} |S_n(\tilde{f})_{jk} - S_n(f)_{jk}| > 2\epsilon \right) \quad (4.69)$$

$$= \mathbb{P} \left( \max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \{ \tilde{f}_j(X_j^{(i)}) \tilde{f}_k(X_k^{(i)}) - f_j(X_j^{(i)}) f_k(X_k^{(i)}) \right. \right. \quad (4.70)$$

$$\left. - \mu_n(\tilde{f}_j) \mu_n(\tilde{f}_k) + \mu_n(f_j) \mu_n(f_k) \} \right| > 2\epsilon \right) \quad (4.71)$$

$$\leq \mathbb{P} \left( \max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \left( \tilde{f}_j(X_j^{(i)}) \tilde{f}_k(X_k^{(i)}) - f_j(X_j^{(i)}) f_k(X_k^{(i)}) \right) \right| > \epsilon \right) \quad (4.72)$$

$$+ \mathbb{P} \left( \max_{j,k} |\mu_n(\tilde{f}_j) \mu_n(\tilde{f}_k) - \mu_n(f_j) \mu_n(f_k)| > \epsilon \right). \quad (4.73)$$

We only need to analyze the rate for the first term above, since the second one is of higher order [Cai et al., 2010]. Let

$$\Delta_i(j, k) \equiv \tilde{f}_j(X_j^{(i)}) \tilde{f}_k(X_k^{(i)}) - f_j(X_j^{(i)}) f_k(X_k^{(i)}) \quad (4.74)$$

and

$$\Theta_{t,s}(j, k) \equiv \tilde{f}_j(t) \tilde{f}_k(s) - f_j(t) f_k(s). \quad (4.75)$$

We define the event  $\mathcal{A}_n$  as

$$\mathcal{A}_n \equiv \left\{ g_j \left( -\sqrt{M \log n} \right) \leq X_j^{(1)}, \dots, X_j^{(n)} \leq g_j \left( \sqrt{M \log n} \right), j = 1, \dots, d \right\}.$$

Then, by Lemma 4.6, when  $M \geq 2(\zeta + 1)$ , we have

$$\mathbb{P}(\mathcal{A}_n^c) \leq \mathbb{P}\left(\max_{i,j \in \{1, \dots, n\} \times \{1, \dots, d\}} |f_j(X_j^{(i)})| > \sqrt{2 \log(nd)}\right) \leq \frac{1}{2\sqrt{\pi \log(nd)}}.$$

Therefore

$$\mathbb{P}\left(\max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \Delta_i(j,k) \right| > \epsilon\right) \quad (4.76)$$

$$\leq \mathbb{P}\left(\max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \Delta_i(j,k) \right| > \epsilon, \mathcal{A}_n\right) \quad (4.77)$$

$$+ \mathbb{P}(\mathcal{A}_n^c) \quad (4.78)$$

$$\leq \mathbb{P}\left(\max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \Delta_i(j,k) \right| > \epsilon, \mathcal{A}_n\right) + \frac{1}{2\sqrt{\pi \log(nd)}}. \quad (4.79)$$

Thus, we only need to carry out our analysis on the event  $\mathcal{A}_n$ . On this event, we have the following decomposition:

$$\mathbb{P}\left(\max_{j,k} \left| \frac{1}{n} \sum_{i=1}^n \Delta_i(j,k) \right| > \epsilon, \mathcal{A}_n\right) \quad (4.80)$$

$$\leq \mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{M}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}\right) \quad (4.81)$$

$$+ \mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}\right) \quad (4.82)$$

$$+ 2\mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}\right). \quad (4.83)$$

We now analyze each of these terms separately.

**Lemma 4.10.** *On the event  $\mathcal{A}_n$ , let  $\beta = 1/2$  and  $\epsilon \geq C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$ , then*

$$\mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}\right) = o(1). \quad (4.84)$$

*Proof.* We define

$$\theta_1 \equiv \frac{n^{\beta/2} \epsilon}{8A\sqrt{\log n}} \quad (4.85)$$

with the same parameter  $A$  as in Lemma 4.8. Such a  $\theta_1$  guarantees that

$$\frac{n\epsilon}{4\theta_1} - nA\sqrt{\frac{\log n}{n^\beta}} = nA\sqrt{\frac{\log n}{n^\beta}} > 0. \quad (4.86)$$

By Lemma 4.8, we have

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n I\left(X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n\right) > \frac{\epsilon}{4\theta_1}\right) \quad (4.87)$$

$$\leq \mathbb{P}\left(\sum_{i=1}^n I\left(X_j^{(i)} \in \mathcal{E}_n\right) > \frac{n\epsilon}{4\theta_1}\right) \quad (4.88)$$

$$= \mathbb{P}\left(\sum_{i=1}^n \left(I\left(X_j^{(i)} \in \mathcal{E}_n\right) - \mathbb{P}\left(X_j^{(1)} \in \mathcal{E}_n\right)\right) > \frac{n\epsilon}{4\theta_1} - n\mathbb{P}\left(X_j^{(1)} \in \mathcal{E}_n\right)\right)$$

$$\leq \mathbb{P}\left(\sum_{i=1}^n \left(I\left(X_j^{(i)} \in \mathcal{E}_n\right) - \mathbb{P}\left(X_j^{(1)} \in \mathcal{E}_n\right)\right) > \frac{n\epsilon}{4\theta_1} - nA\sqrt{\frac{\log n}{n^\beta}}\right).$$

Using Bernstein's inequality, for  $\beta = \frac{1}{2}$ ,

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n I\left(X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n\right) > \frac{\epsilon}{4\theta_1}\right) \quad (4.89)$$

$$\leq \mathbb{P}\left(\sum_{i=1}^n \left(I\left(X_j^{(i)} \in \mathcal{E}_n\right) - \mathbb{P}\left(X_j^{(1)} \in \mathcal{E}_n\right)\right) > nA\sqrt{\frac{\log n}{n^\beta}}\right) \quad (4.90)$$

$$\leq \exp\left(-\frac{c_1 n^{2-\beta} \log n}{c_2 n^{1-\beta/2} \sqrt{\log n} + c_3 n^{1-\beta/2} \sqrt{\log n}}\right) = o(1), \quad (4.91)$$

where  $c_1, c_2, c_3 > 0$  are generic constants.

Using the fact that

$$\mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}\right) \quad (4.92)$$

$$= \mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}, \max_{j,k} \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| > \theta_1\right) \\ + \mathbb{P}\left(\max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4}, \max_{j,k} \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| \leq \theta_1\right),$$

we have

$$\mathbb{P} \left( \max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4} \right) \quad (4.93)$$

$$\leq \mathbb{P} \left( \max_{j,k} \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| > \theta_1 \right) \quad (4.94)$$

$$+ \mathbb{P} \left( \frac{1}{n} \sum_{i=1}^n I \left( X_j^{(i)} \in \mathcal{E}_n, X_k^{(i)} \in \mathcal{E}_n \right) > \frac{\epsilon}{4\theta_1} \right) \quad (4.95)$$

$$= \mathbb{P} \left( \max_{j,k} \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| > \theta_1 \right) + o(1). \quad (4.96)$$

Now, we analyze the first term

$$\mathbb{P} \left( \max_{j,k} \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| > \theta_1 \right) \quad (4.97)$$

$$\leq d^2 \mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\Theta_{t,s}(j,k)| > \theta_1 \right) \quad (4.98)$$

$$= d^2 \mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\tilde{f}_j(t)\tilde{f}_k(s) - f_j(t)f_k(s)| > \theta_1 \right). \quad (4.99)$$

By adding and subtracting terms  $f_j(t)$  and  $f_s(t)$ , we have

$$\mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |\tilde{f}_j(t)\tilde{f}_k(s) - f_j(t)f_k(s)| > \theta_1 \right) \quad (4.100)$$

$$\leq \mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))(\tilde{f}_k(s) - f_k(s))| > \frac{\theta_1}{3} \right) \quad (4.101)$$

$$+ \mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))| \cdot |f_k(s)| > \frac{\theta_1}{3} \right) \quad (4.102)$$

$$+ \mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |(\tilde{f}_k(s) - f_k(s))| \cdot |f_j(t)| > \frac{\theta_1}{3} \right). \quad (4.103)$$

The first term can further be decomposed to be

$$\mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))(\tilde{f}_k(s) - f_k(s))| > \frac{\theta_1}{3} \right) \quad (4.104)$$

$$\leq \mathbb{P} \left( \sup_{t \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))| > \sqrt{\frac{\theta_1}{3}} \right) \quad (4.105)$$

$$+ \mathbb{P} \left( \sup_{s \in \mathcal{E}_n} |(\tilde{f}_k(s) - f_k(s))| > \sqrt{\frac{\theta_1}{3}} \right). \quad (4.106)$$

Also, from the definition of  $\mathcal{E}_n$ , we have

$$\sup_{t \in \mathcal{E}_n} |f_j(t)| = \sup_{t \in \mathcal{E}_n} |g_j^{-1}(t)| \leq \sqrt{M \log n}. \quad (4.107)$$

Since  $\epsilon \geq C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$ , we have

$$\frac{\theta_1}{3} = \frac{n^{\beta/2} \epsilon}{24A \sqrt{\log n}} \geq \frac{C_M \sqrt{\log d \log^2 n}}{24A \sqrt{\log n}} = 2(M+2) \log n. \quad (4.108)$$

This implies that

$$\sqrt{\frac{\theta_1}{3}} \geq \sqrt{2(M+2) \log n} \quad \text{and} \quad \frac{\theta_1}{3\sqrt{M \log n}} \geq \sqrt{2(M+2) \log n}. \quad (4.109)$$

Then, from Lemma 4.9, we get

$$\mathbb{P} \left( \sup_{t \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))| > \sqrt{\frac{\theta_1}{3}} \right) = 0 \quad (4.110)$$

and

$$\mathbb{P} \left( \sup_{t \in \mathcal{E}_n, s \in \mathcal{E}_n} |(\tilde{f}_j(t) - f_j(t))| \cdot |f_k(s)| > \frac{\theta_1}{3} \right) = 0. \quad (4.111)$$

The claim of the lemma then follows directly.  $\square$

**Remark 4.1.** From the above analysis, we see that the data in the tails doesn't affect the rate. Using exactly the same argument, we can also show that

$$\mathbb{P} \left( \max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4} \right) = o(1). \quad (4.112)$$

**Lemma 4.11.** On the event  $\mathcal{A}_n$ , let  $\beta = 1/2$  and  $\epsilon \geq C_M \sqrt{\frac{\log d \log^2 n}{n^{1/2}}}$ . We have

$$\begin{aligned} & \mathbb{P} \left( \max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{M}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4} \right) \\ & \leq 2 \exp \left( 2 \log d - \frac{n^{1/2} \epsilon^2}{1232 \pi^2 \log^2 n} \right) + 2 \exp \left( 2 \log d - \frac{n^{1/2}}{8 \pi \log n} \right). \end{aligned} \quad (4.113)$$



*Proof.* We have

$$\mathbb{P} \left( \max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{M}_n} |\Delta_i(j, k)| > \frac{\epsilon}{4} \right) \quad (4.114)$$

$$\leq d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n, s \in \mathcal{M}_n} |\tilde{f}_j(t) \tilde{f}_k(s) - f_j(t) f_k(s)| > \frac{\epsilon}{4} \right) \quad (4.115)$$

$$\leq d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n, s \in \mathcal{M}_n} |(\tilde{f}_j(t) - f_j(t))(\tilde{f}_k(s) - f_k(s))| > \frac{\epsilon}{12} \right) \quad (4.116)$$

$$+ 2d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n, s \in \mathcal{M}_n} |(\tilde{f}_j(t) - f_j(t))| \cdot |f_k(s)| > \frac{\epsilon}{12} \right). \quad (4.117)$$

Further, since

$$\sup_{t \in \mathcal{M}_n} |f_j(t)| = \sup_{t \in \mathcal{M}_n} |g_j^{-1}(t)| = \sqrt{\beta \log n} \quad (4.118)$$

and

$$\sup_{t \in \mathcal{M}_n, s \in \mathcal{M}_n} |(\tilde{f}_j(t) - f_j(t))(\tilde{f}_k(s) - f_k(s))|$$

is of higher order than

$$\sup_{t \in \mathcal{M}_n, s \in \mathcal{M}_n} |(\tilde{f}_j(t) - f_j(t))| \cdot |f_k(s)|,$$

we only need to analyze the term

$$\mathbb{P} \left( \sup_{t \in \mathcal{M}_n} |(\tilde{f}_j(t) - f_j(t))| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right).$$

Since  $\delta_n = \frac{1}{4n^{\beta/2}\sqrt{2\pi\beta \log n}}$ , using Mill's inequality we have

$$2\delta_n = \frac{\phi(\sqrt{\beta \log n})}{2\sqrt{\beta \log n}} \leq 1 - \Phi(\sqrt{\beta \log n}). \quad (4.119)$$

This implies that

$$1 - \delta_n - \Phi(\sqrt{\beta \log n}) \geq \delta_n > 0. \quad (4.120)$$

Using Lemma 4.7, we have

$$d^2 \mathbb{P} \left( \hat{F}_j \left( g_j \left( \sqrt{\beta \log n} \right) \right) > 1 - \delta_n \right) \quad (4.121)$$

$$\leq d^2 \exp(-2n\delta_n^2) = \exp \left( 2 \log d - \frac{n^{1-\beta}}{(16\pi\beta \log n)} \right) \quad (4.122)$$

and

$$d^2 \mathbb{P} \left( \hat{F}_j \left( g_j \left( -\sqrt{\beta \log n} \right) \right) < \delta_n \right) \leq \exp \left( 2 \log d - \frac{n^{1-\beta}}{(16\pi\beta \log n)} \right) \quad (4.123)$$

Define an event  $\mathcal{B}_n$  as

$$\mathcal{B}_n \equiv \left\{ \delta_n \leq \hat{F}_j \left( g_j \left( \sqrt{\beta \log n} \right) \right) \leq 1 - \delta_n, j = 1, \dots, d \right\}. \quad (4.124)$$

From (4.121) and (4.123), it is easy to see that

$$\mathbb{P}(\mathcal{B}_n^c) \leq 2 \exp \left( 2 \log d - \frac{n^{1/2}}{8\pi \log n} \right). \quad (4.125)$$

From the definition of  $\tilde{F}_j$ , we have

$$d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n} |\tilde{f}_j(t) - f_j(t)| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right) \quad (4.126)$$

$$\leq d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n} \left| \Phi^{-1}(\tilde{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| > \frac{\epsilon}{12\sqrt{\beta \log n}}, \mathcal{B}_n \right) + \mathbb{P}(\mathcal{B}_n^c).$$

$$\leq d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n} \left| \Phi^{-1}(\hat{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right) \quad (4.127)$$

$$+ 2 \exp \left( 2 \log d - \frac{n^{1/2}}{8\pi \log n} \right). \quad (4.128)$$

Define

$$T_{1n} \equiv \max \left\{ F_j \left( g_j \left( \sqrt{\beta \log n} \right) \right), 1 - \delta_n \right\} \quad (4.129)$$

and

$$T_{2n} \equiv 1 - \min \left\{ F_j \left( g_j \left( -\sqrt{\beta \log n} \right) \right), \delta_n \right\}. \quad (4.130)$$

From Equation (4.46) and the fact that  $1 - \delta_n \geq \Phi(\sqrt{\beta \log n})$ , we have that

$$T_{1n} = T_{2n} = 1 - \delta_n. \quad (4.131)$$

Thus, by the mean value theorem,

$$\mathbb{P} \left( \sup_{t \in \mathcal{M}_n} \left| \Phi^{-1}(\hat{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right) \quad (4.132)$$

$$\leq \mathbb{P} \left( (\Phi^{-1})'(\max\{T_{1n}, T_{2n}\}) \sup_{t \in \mathcal{M}_n} |\hat{F}_j(t) - F_j(t)| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right)$$

$$= \mathbb{P} \left( (\Phi^{-1})'(1 - \delta_n) \sup_{t \in \mathcal{M}_n} |\hat{F}_j(t) - F_j(t)| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right). \quad (4.133)$$

Finally, using the Dvoretzky-Kiefer-Wolfowitz inequality,

$$\mathbb{P} \left( \sup_{t \in \mathcal{M}_n} \left| \Phi^{-1}(\hat{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right) \quad (4.134)$$

$$\leq \mathbb{P} \left( \sup_{t \in \mathcal{M}_n} |\hat{F}_j(t) - F_j(t)| > \frac{\epsilon}{(\Phi^{-1})'(1 - \delta_n) 12\sqrt{\beta \log n}} \right) \quad (4.135)$$

$$\leq 2 \exp \left( -2 \frac{n\epsilon^2}{144\beta \log n [(\Phi^{-1})'(1 - \delta_n)]^2} \right). \quad (4.136)$$

Furthermore, by Lemma 4.4,

$$(\Phi^{-1})'(1 - \delta_n) = \frac{1}{\phi(\Phi^{-1}(1 - \delta_n))} \quad (4.137)$$

$$\leq \frac{1}{\phi \left( \sqrt{2 \log \frac{1}{\delta_n}} \right)} \quad (4.138)$$

$$= \sqrt{2\pi} \left( \frac{1}{\delta_n} \right) \quad (4.139)$$

$$= 8\pi n^{\beta/2} \sqrt{\beta \log n}. \quad (4.140)$$

This implies that

$$d^2 \mathbb{P} \left( \sup_{t \in \mathcal{M}_n} \left| \Phi^{-1}(\hat{F}_j(t)) - \Phi^{-1}(F_j(t)) \right| > \frac{\epsilon}{12\sqrt{\beta \log n}} \right) \quad (4.141)$$

$$\leq 2 \exp \left( 2 \log d - \frac{n^{1/2} \epsilon^2}{1232 \pi^2 \log^2 n} \right). \quad (4.142)$$

In summary, we have

$$\mathbb{P} \left( \max_{j,k} \frac{1}{n} \sum_{X_j^{(i)} \in \mathcal{M}_n, X_k^{(i)} \in \mathcal{E}_n} |\Delta_i(j,k)| > \frac{\epsilon}{4} \right) \quad (4.143)$$

$$\leq 2 \exp \left( 2 \log d - \frac{n^{1/2} \epsilon^2}{1232 \pi^2 \log^2 n} \right) + 2 \exp \left( 2 \log d - \frac{n^{1/2}}{8\pi \log n} \right).$$

This finish the proof.  $\square$

The conclusion of Theorem 4.1 follows from Lemma 4.10 and Lemma 4.11.

## 4.8.2 Proof of Theorem 4.2

*Proof.* First note that the population and sample risks are

$$\begin{aligned} R(f, \Omega) &= \frac{1}{2} \left\{ \text{tr} \left[ \Omega \mathbb{E}(f(X)f(X)^T) \right] - \log |\Omega| - d \log(2\pi) \right\} \\ \hat{R}(f, \Omega) &= \frac{1}{2} \left\{ \text{tr} [\Omega S_n(f)] - \log |\Omega| - d \log(2\pi) \right\}. \end{aligned}$$

Therefore, for all  $(f, \Omega) \in \mathcal{M}_n^d \oplus \mathcal{C}_n$ , we have

$$\begin{aligned} |\hat{R}(f, \Omega) - R(f, \Omega)| &= \frac{1}{2} \left| \text{tr} \left[ \Omega \left( \mathbb{E}[ff^T] - S_n(f) \right) \right] \right| \\ &\leq \frac{1}{2} \|\Omega\|_1 \max_{jk} \sup_{f_j, f_k \in \mathcal{M}_n} |\mathbb{E}(f_j(X_j)f_k(X_k) - S_n(f)_{jk})| \\ &\leq \frac{L_n}{2} \max_{jk} \sup_{f_j, f_k \in \mathcal{M}_n} |\mathbb{E}(f_j(X_j)f_k(X_k) - S_n(f)_{jk})|. \end{aligned}$$

Now, if  $\mathcal{F}$  is a class of functions, we have

$$\mathbb{E} \left( \sup_{g \in \mathcal{F}} |\hat{\mu}(g) - \mu(g)| \right) \leq \frac{C J_{[]}(\|F\|_\infty, \mathcal{F})}{\sqrt{n}} \quad (4.144)$$

for some  $C > 0$ , where  $F(x) = \sup_{g \in \mathcal{F}} |g(x)|$ ,  $\mu(g) = \mathbb{E}(g(X))$  and  $\hat{\mu}(g) = n^{-1} \sum_{i=1}^n g(X_i)$  (see Corollary 19.35 of [van der Vaart 1998](#)). Here the bracketing integral is defined to be

$$J_{[]}(\delta, \mathcal{F}) = \int_0^\delta \sqrt{\log N_{[]}(\epsilon, \mathcal{F})} d\epsilon \quad (4.145)$$

where  $\log N_{[]}(\epsilon, \mathcal{F})$  is the bracketing entropy. For the class of one dimensional, bounded and monotone functions, the bracketing entropy satisfies

$$\log N_{[]}(\epsilon, \mathcal{M}) \leq K \left( \frac{1}{\epsilon} \right) \quad (4.146)$$

for some  $K > 0$  [[van der Vaart and Wellner, 1996](#)].

Now, let  $\mathcal{P}_{n,d}$  be the class of all functions of the form  $m(x) = f_j(x_j)f_k(x_k)$  for  $j, k \in \{1, \dots, d\}$ , where  $f_j \in \mathcal{M}_n$  for each  $j$ . Then the bracketing entropy satisfies

$$\log N_{[]}(\sqrt{\log n}, \mathcal{P}_{n,d}) \leq 2 \log d + K \left( \frac{1}{\sqrt{\log n}} \right)$$

and the bracketing integral satisfies  $J_{[]}(\sqrt{\log n}, \mathcal{P}_{n,d}) = O(\sqrt{\log n \log d})$ . It follows from (6.124) and Markov's inequality that

$$\max_{jk} \sup_{f_j, f_k \in \mathcal{M}_n} |S_n(f)_{jk} - \mathbb{E}(f_j(X_j)f_k(X_k))| = O_P \left( \sqrt{\frac{\log n \log d}{n}} \right) = O_P \left( \sqrt{\frac{\log n}{n^{1-\xi}}} \right).$$

Therefore,

$$\sup_{(f, \Omega) \in \mathcal{M}_n^d \oplus \mathcal{C}_n} |\hat{R}(f, \Omega) - R(f, \Omega)| = O_P \left( \frac{L_n \sqrt{\log n}}{n^{(1-\xi)/2}} \right).$$

As a consequence, we have

$$\begin{aligned} R(f^*, \Omega^*) &\leq R(\tilde{f}_n, \hat{\Omega}_n) \\ &\leq \hat{R}(\tilde{f}_n, \hat{\Omega}_n) + O_P \left( \frac{L_n \sqrt{\log n}}{n^{(1-\xi)/2}} \right) \\ &\leq \hat{R}(f^*, \Omega^*) + O_P \left( \frac{L_n \sqrt{\log n}}{n^{(1-\xi)/2}} \right) \\ &\leq R(f^*, \Omega^*) + O_P \left( \frac{L_n \sqrt{\log n}}{n^{(1-\xi)/2}} \right) \end{aligned}$$

and the conclusion follows.  $\square$



*In this chapter, we study high dimensional graph estimation and density estimation using a family of density estimators based on forest structured undirected graphical models. For density estimation, we do not assume the true distribution corresponds to a forest; rather, we form kernel density estimates of the bivariate and univariate marginals, and apply Kruskal's algorithm to estimate the optimal forest on held out data. We prove an oracle inequality on the excess risk of the resulting estimator relative to the risk of the best forest. Viewing the forest size as a complexity parameter, we then select a forest using data splitting, and prove bounds on excess risk and structure selection consistency of the procedure. Experiments with simulated data and microarray data indicate that the methods are a practical alternative to Gaussian graphical models.*

## 5.1 INTRODUCTION AND MOTIVATION

As we have explained in the previous chapter, one way to explore the structure of a high dimensional distribution  $P$  for a random vector  $X = (X_1, \dots, X_d)$  is to estimate its undirected graph. The undirected graph  $G$  associated with  $P$  has  $d$  vertices corresponding to the variables  $X_1, \dots, X_d$ , and omits an edge between two nodes  $X_i$  and  $X_j$  if and only if  $X_i$  and  $X_j$  are conditionally independent given the other variables. Currently, the most popular methods for estimating  $G$  assume that the distribution  $P$  is Gaussian. Finding the graphical structure in this case amounts to estimating the inverse covariance matrix  $\Omega$ ; the edge between  $X_j$  and  $X_k$  is missing if and only if  $\Omega_{jk} = 0$ . Algorithms for optimizing the  $\ell_1$ -regularized log-likelihood have recently been proposed that efficiently produce sparse estimates of the inverse covariance matrix and the underlying graph [Banerjee et al., 2008, Friedman et al., 2007].

In this chapter our goal is to relax the Gaussian assumption and to develop nonparametric methods for estimating the graph of a distribution. Of course, estimating a high dimensional distribution is impossible without making any assumptions. The approach we take here is to force the graphical structure to be a forest, where each pair of vertices is connected by at most one path. Thus, we relax the distributional assumption of normality but we restrict the family of undirected graphs that are allowed.

If the graph for  $P$  is a forest, then a simple conditioning argument shows that its density  $p$  can be written as

$$p(x) = \prod_{(i,j) \in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \prod_{k=1}^d p(x_k) \quad (5.1)$$

where  $E$  is the set of edges in the forest [Lauritzen, 1996]. Here  $p(x_i, x_j)$  is the bivariate marginal density of variables  $X_i$  and  $X_j$ , and  $p(x_k)$  is the univariate marginal density of the variable  $X_k$ . With this factorization, we see that it is only necessary to estimate the bivariate and univariate marginals. Given any distribution  $P$  with density  $p$ , there is a tree  $T$  and a density  $p_T$  whose graph is  $T$  and which is closest in Kullback-Leibler divergence to  $p$ . When  $P$  is known, then the best fitting tree distribution can be obtained by Kruskal's algorithm [Kruskal, 1956], or other algorithms for finding a maximum weight spanning tree. In the discrete case, the algorithm can be applied to the estimated probability mass function, resulting in a procedure originally proposed by Chow and Liu [1968]. Here we are concerned with continuous random variables, and we estimate the bivariate marginals with nonparametric kernel density estimators.

In high dimensions, fitting a fully connected spanning tree can be expected to overfit. We regulate the complexity of the forest by selecting the included edges using a data splitting scheme, a simple form of cross validation. In particular, we consider the family of forest structured densities that use the marginal kernel density estimates constructed on the first partition of the data, and estimate the risk of the resulting densities over a second, held out partition. The final forest optimizing the held out risk is then obtained by finding a maximum weight spanning forest for an appropriate set of edge weights.

A closely related approach is proposed by Bach and Jordan [2003], where a tree is estimated for the random vector  $Y = WX$  instead of  $X$ , where  $W$  is a linear transformation, using an algorithm that alternates between estimating  $W$  and estimating the tree  $T$ . Kernel density estimators are used, and a regularization term that is a function of the number of edges in the tree is included to bias the optimization toward smaller trees. We omit the transformation  $W$ , and we use a data splitting method rather than penalization to choose the complexity of the forest.

While tree and forest structured density estimation has been long recognized as a useful tool, there has been little theoretical analysis of the statistical properties of such density estimators. The main contribution of this paper is an analysis of the asymptotic properties of forest density estimation in high dimensions. We allow both the sample size  $n$  and dimension  $d$  to increase, and prove oracle results on the risk of the method. In particular, we assume



that the univariate and bivariate marginal densities lie in a Hölder class with exponent  $\beta$  (see Section 5.4 for details), and show that

$$R(\hat{p}_{\hat{F}}) - \min_F R(\hat{p}_F) = O_P \left( \sqrt{\log(nd)} \left( \frac{k^* + \hat{k}}{n^{\beta/(2+2\beta)}} + \frac{d}{n^{\beta/(1+2\beta)}} \right) \right) \quad (5.2)$$

where  $R$  denotes the risk, the expected negative log-likelihood,  $\hat{k}$  is the number of edges in the estimated forest  $\hat{F}$ , and  $k^*$  is the number of edges in the optimal forest  $F^*$  that can be constructed in terms of the kernel density estimates  $\hat{p}$ .

In addition to the above results on risk consistency, we establish conditions under which

$$\mathbb{P} \left( \hat{F}_d^{(k)} = F_d^{*(k)} \right) \rightarrow 1 \quad (5.3)$$

as  $n \rightarrow \infty$ , where  $F_d^{*(k)}$  is the *oracle forest*—the best forest with  $k$  edges; this result allows the dimensionality  $d$  to increase as fast as  $o \left( \exp(n^{\beta/(1+\beta)}) \right)$ , while still having consistency in the selection of the oracle forest.

Among the only other previous work analyzing tree structured graphical models is Tan et al. [2009a] and Chechetka and Guestrin [2007]. Tan et al. [2009a] analyze the error exponent in the rate of decay of the error probability for estimating the tree, in the fixed dimension setting, and Chechetka and Guestrin [2007] give a PAC analysis. An extension to the Gaussian case is given by Tan et al. [2009b].

Here is the organization of this chapter. In Section 5.2 we review some background and notation. In Section 5.3 we present a two-stage algorithm for estimating high dimensional densities supported by forests, and we provide a theoretical analysis of the algorithm in Section 5.4, with the detailed proofs collected in the appendix. In Section 5.5 we present experiments with both simulated data and gene microarray datasets, where the problem is to estimate the gene-gene association graphs.

## 5.2 PRELIMINARIES AND NOTATION

Let  $p^*(x)$  be a probability density with respect to Lebesgue measure  $\mu(\cdot)$  on  $\mathbb{R}^d$  and let  $X^{(1)}, \dots, X^{(n)}$  be  $n$  independent identically distributed  $\mathbb{R}^d$ -valued data vectors sampled from  $p^*(x)$  where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})$ . Let  $\mathcal{X}_j$  denote the range of  $X_j^{(i)}$  and let  $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_d$ .

A graph is a forest if it is acyclic. If  $F$  is a  $d$ -node undirected forest with vertex set  $V_F = \{1, \dots, d\}$  and edge set  $E(F) \subset \{1, \dots, d\} \times \{1, \dots, d\}$ , the number of edges satisfies  $|E(F)| < d$ , noting that we do not restrict the graph

to be connected. We say that a probability density function  $p(x)$  is *supported by a forest*  $F$  if the density can be written as

$$p_F(x) = \prod_{(i,j) \in E(F)} \frac{p(x_i, x_j)}{p(x_i) p(x_j)} \prod_{k \in V_F} p(x_k), \quad (5.4)$$

where each  $p(x_i, x_j)$  is a bivariate density on  $\mathcal{X}_i \times \mathcal{X}_j$ , and each  $p(x_k)$  is a univariate density on  $\mathcal{X}_k$ . More details can be found in [Lauritzen \[1996\]](#).

Let  $\mathcal{F}_d$  be the family of forests with  $d$  nodes, and let  $\mathcal{P}_d$  be the corresponding family of densities:

$$\mathcal{P}_d = \left\{ p \geq 0 : \int_{\mathcal{X}} p(x) d\mu(x) = 1, \text{ and } p(x) \text{ satisfies (5.4) for some } F \in \mathcal{F}_d \right\}. \quad (5.5)$$

To bound the number of labeled spanning forests on  $d$  nodes, note that each such forest can be obtained by forming a labeled tree on  $d + 1$  nodes, and then removing node  $d + 1$ . From Cayley's formula [[Cayley, 1889](#), [Aigner and Ziegler, 1998](#)], we then obtain the following.

**Proposition 5.1.** *The size of the collection  $\mathcal{F}_d$  of labeled forests on  $d$  nodes satisfies*

$$|\mathcal{F}_d| < (d + 1)^{d-1}. \quad (5.6)$$

Define the oracle forest density

$$q^* = \arg \min_{q \in \mathcal{P}_d} D(p^* \| q) \quad (5.7)$$

where the Kullback-Leibler divergence  $D(p \| q)$  between two densities  $p$  and  $q$  is

$$D(p \| q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx, \quad (5.8)$$

under the convention that  $0 \log(0/q) = 0$ , and  $p \log(p/0) = \infty$  for  $p \neq 0$ . The following is proved by [Bach and Jordan \[2003\]](#).

**Proposition 5.2.** *Let  $q^*$  be defined as in (5.7). There exists a forest  $F^* \in \mathcal{F}_d$ , such that*

$$q^* = p_{F^*}^* = \prod_{(i,j) \in E(F^*)} \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} \prod_{k \in V_{F^*}} p^*(x_k) \quad (5.9)$$

where  $p^*(x_i, x_j)$  and  $p^*(x_i)$  are the bivariate and univariate marginal densities of  $p^*$ .

For any density  $q(x)$ , the negative log-likelihood risk  $R(q)$  is defined as

$$R(q) = -\mathbb{E} \log q(X) = - \int_{\mathcal{X}} p^*(x) \log q(x) dx \quad (5.10)$$

where the expectation is defined with respect to the distribution of  $X$ .

It is straightforward to see that the density  $q^*$  defined in (5.7) also minimizes the negative log-likelihood loss:

$$q^* = \arg \min_{q \in \mathcal{P}_d} D(p^* \| q) = \arg \min_{q \in \mathcal{P}_d} R(q). \quad (5.11)$$

Let  $\hat{p}(x)$  be the kernel density estimate, we also define

$$\hat{R}(q) = - \int_{\mathcal{X}} \hat{p}(x) \log q(x) dx. \quad (5.12)$$

We thus define the oracle risk as  $R^* = R(q^*)$ . Using Proposition 5.2 and equation (5.4), we have

$$\begin{aligned} R^* &= R(q^*) = R(p_{F^*}^*) \\ &= - \int_{\mathcal{X}} p^*(x) \left( \sum_{(i,j) \in E(F^*)} \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} + \sum_{k \in V_{F^*}} \log (p^*(x_k)) \right) dx \\ &= - \sum_{(i,j) \in E(F^*)} \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} dx_i dx_j \end{aligned} \quad (5.13)$$

$$\begin{aligned} &- \sum_{k \in V_{F^*}} \int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k) dx_k \\ &= - \sum_{(i,j) \in E(F^*)} I(X_i; X_j) + \sum_{k \in V_{F^*}} H(X_k), \end{aligned} \quad (5.14)$$

where

$$I(X_i; X_j) = \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i) p^*(x_j)} dx_i dx_j \quad (5.15)$$

is the mutual information between the pair of variables  $X_i, X_j$  and

$$H(X_k) = - \int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k) dx_k \quad (5.16)$$

is the entropy. While the best forest will in fact be a spanning tree, the densities  $p^*(x_i, x_j)$  are in practice not known. We estimate the marginals using finite data, in terms of a kernel density estimates  $\hat{p}_{n_1}(x_i, x_j)$  over a training set of size  $n_1$ . With these estimated marginals, we consider all forest density estimates of the form

$$\hat{p}_F(x) = \prod_{(i,j) \in E(F)} \frac{\hat{p}_{n_1}(x_i, x_j)}{\hat{p}_{n_1}(x_i) \hat{p}_{n_1}(x_j)} \prod_{k \in V_F} \hat{p}_{n_1}(x_k). \quad (5.17)$$

Within this family, the best density estimate may not be supported on a full spanning tree, since a full tree will in general be subject to overfitting. Analogously, in high dimensional linear regression, the optimal regression model will generally be a full  $d$ -dimensional fit, with a nonzero parameter for each variable. However, when estimated on finite data the variance of a full model will dominate the squared bias, resulting in overfitting. In our setting of density estimation we will regulate the complexity of the forest by cross validating over a held out set.

There are several different ways to judge the quality of a forest structured density estimator. In this paper we concern ourselves with prediction and structure estimation.

**Definition 5.1** ((Risk consistency)). *For an estimator  $\hat{q}_n \in \mathcal{P}_d$ , the excess risk is defined as  $R(\hat{q}_n) - R^*$ . The estimator  $\hat{q}_n$  is risk consistent with convergence rate  $\delta_n$  if*

$$\lim_{M \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{P}(R(\hat{q}_n) - R^* \geq M\delta_n) = 0. \quad (5.18)$$

In this case we write  $R(\hat{q}_n) - R^* = O_P(\delta_n)$ .

**Definition 5.2** ((Estimation consistency)). *An estimator  $\hat{q}_n \in \mathcal{P}_d$  is estimation consistent with convergence rate  $\delta_n$ , with respect to the Kullback-Leibler divergence, if*

$$\lim_{M \rightarrow \infty} \limsup_{n \rightarrow \infty} \mathbb{P}(D(p_{F^*}^* \| \hat{q}_n) \geq M\delta_n) = 0. \quad (5.19)$$

**Definition 5.3** ((Structure selection consistency)). *An estimator  $\hat{q}_n \in \mathcal{P}_d$  supported by a forest  $\hat{F}_n$  is structure selection consistent if*

$$\mathbb{P}(E(\hat{F}_n) \neq E(F^*)) \rightarrow 0, \quad (5.20)$$

as  $n$  goes to infinity, where  $F^*$  is defined in (5.9).

Later we will show that estimation consistency is almost equivalent to risk consistency. If the true density is give, these two criteria are exactly the same; otherwise, the estimation consistency requires stronger conditions than those of the risk consistency.

It is important to note that risk consistency is an oracle property, in the sense that the true density  $p^*(x)$  is not restricted to be supported by a forest; rather, the property assesses how well a given estimator  $\hat{q}$  approximates the best forest density (the oracle) within a class.

### 5.3 KERNEL DENSITY ESTIMATION FOR FORESTS

If the true density  $p^*(x)$  were known, by Proposition 5.2, the density estimation problem would be reduced to finding the best forest structure  $F_d^*$ , satisfying

$$F_d^* = \arg \min_{F \in \mathcal{F}_d} R(p_F^*) = \arg \min_{F \in \mathcal{F}_d} D(p^* \| p_F^*). \quad (5.21)$$

The optimal forest  $F_d^*$  can be found by minimizing the right hand side of (5.14). Since the entropy term  $H(X) = \sum_k H(X_k)$  is constant across all forests, this can be recast as the problem of finding the maximum weight spanning forest for a weighted graph, where the weight of the edge connecting nodes  $i$  and  $j$  is  $I(X_i; X_j)$ . Kruskal's algorithm [Kruskal, 1956] is a greedy algorithm that is guaranteed to find a maximum weight spanning tree of a weighted graph. In the setting of density estimation, this procedure was proposed by Chow and Liu [1968] as a way of constructing a tree approximation to a distribution. At each stage the algorithm adds an edge connecting that pair of variables with maximum mutual information among all pairs not yet visited by the algorithm, if doing so does not form a cycle. When stopped early, after  $k < d - 1$  edges have been added, it yields the best  $k$ -edge weighted forest.

Of course, the above procedure is not practical since the true density  $p^*(x)$  is unknown. We replace the population mutual information  $I(X_i; X_j)$  in (5.14) by the plug-in estimate  $\hat{I}_n(X_i, X_j)$ , defined as

$$\hat{I}_n(X_i, X_j) = \int_{\mathcal{X}_i \times \mathcal{X}_j} \hat{p}_n(x_i, x_j) \log \frac{\hat{p}_n(x_i, x_j)}{\hat{p}_n(x_i) \hat{p}_n(x_j)} dx_i dx_j \quad (5.22)$$

where  $\hat{p}_n(x_i, x_j)$  and  $\hat{p}_n(x_i)$  are bivariate and univariate kernel density estimates. Given this estimated mutual information matrix  $\hat{M}_n = [\hat{I}_n(X_i, X_j)]$ , we can then apply Kruskal's algorithm (equivalently, the Chow-Liu algorithm) to find the best forest structure  $\hat{F}_n$ .

Since the number of edges of  $\hat{F}_n$  controls the number of degrees of freedom in the final density estimator, we need an automatic data-dependent way to choose it. We adopt the following two-stage procedure. First, randomly partition the data into two sets  $\mathcal{D}_1$  and  $\mathcal{D}_2$  of sizes  $n_1$  and  $n_2$ ; then, apply the following steps:

1. Using  $\mathcal{D}_1$ , construct kernel density estimates of the univariate and bivariate marginals and calculate  $\hat{I}_{n_1}(X_i, X_j)$  for  $i, j \in \{1, \dots, d\}$  with  $i \neq j$ . Construct a full tree  $\hat{F}_{n_1}^{(d-1)}$  with  $d - 1$  edges, using the Chow-Liu algorithm.
2. Using  $\mathcal{D}_2$ , prune the tree  $\hat{F}_{n_1}^{(d-1)}$  to find a forest  $\hat{F}_{n_1}^{(\hat{k})}$  with  $\hat{k}$  edges, for  $0 \leq \hat{k} \leq d - 1$ .

Once  $\hat{F}_{n_1}^{(\hat{k})}$  is obtained in Step 2, we can calculate  $\hat{p}_{\hat{F}_{n_1}^{(\hat{k})}}$  according to (5.4), using the kernel density estimates constructed in Step 1.

### 5.3.1 Step 1: Estimating the marginals

Step 1 is carried out on the dataset  $\mathcal{D}_1$ . Let  $K(\cdot)$  be a univariate kernel function. Given an evaluation point  $(x_i, x_j)$ , the bivariate kernel density estimate for  $(X_i, X_j)$  based on the observations  $\{X_i^{(s)}, X_j^{(s)}\}_{s \in \mathcal{D}_1}$  is defined as

$$\hat{p}_{n_1}(x_i, x_j) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_2^2} K\left(\frac{X_i^{(s)} - x_i}{h_2}\right) K\left(\frac{X_j^{(s)} - x_j}{h_2}\right), \quad (5.23)$$

where we use a product kernel with  $h_2 > 0$  be the bandwidth parameter. The univariate kernel density estimate  $\hat{p}_{n_1}(x_k)$  for  $X_k$  is

$$\hat{p}_{n_1}(x_k) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_1} K\left(\frac{X_k^{(s)} - x_k}{h_1}\right), \quad (5.24)$$

where  $h_1 > 0$  is the univariate bandwidth. Detailed specifications for  $K(\cdot)$  and  $h_1, h_2$  will be discussed in the next section.

We assume that the data lie in a  $d$ -dimensional unit cube  $\mathcal{X} = [0, 1]^d$ . To calculate the empirical mutual information  $\hat{I}_{n_1}(X_i, X_j)$ , we need to numerically evaluate a two-dimensional integral. To do so, we calculate the kernel density estimates on a grid of points. We choose  $m$  evaluation points on each dimension,  $x_{1i} < x_{2i} < \dots < x_{mi}$  for the  $i$ th variable. The mutual information  $\hat{I}_{n_1}(X_i, X_j)$  is then approximated as

$$\hat{I}_{n_1}(X_i, X_j) = \frac{1}{m^2} \sum_{k=1}^m \sum_{\ell=1}^m \hat{p}_{n_1}(x_{ki}, x_{\ell j}) \log \frac{\hat{p}_{n_1}(x_{ki}, x_{\ell j})}{\hat{p}_{n_1}(x_{ki}) \hat{p}_{n_1}(x_{\ell j})}. \quad (5.25)$$

The approximation error can be made arbitrarily small by choosing  $m$  sufficiently large. As a practical concern, care needs to be taken that the factors  $\hat{p}_{n_1}(x_{ki})$  and  $\hat{p}_{n_1}(x_{\ell j})$  in the denominator are not too small; a truncation procedure can be used to ensure this. Once the  $d \times d$  mutual information matrix  $\hat{M}_{n_1} = [\hat{I}_{n_1}(X_i, X_j)]$  is obtained, we can apply the Chow-Liu (Kruskal) algorithm to find a maximum weight spanning tree.

### 5.3.2 Step 2: Optimizing the forest

The full tree  $\hat{F}_{n_1}^{(d-1)}$  obtained in Step 1 might have high variance when the dimension  $d$  is large, leading to overfitting in the density estimate. In order to reduce the variance, we prune the tree; that is, we choose forest with  $k \leq d - 1$  edges. The number of edges  $k$  is a tuning parameter that induces a bias-variance tradeoff.

In order to choose  $k$ , note that in stage  $k$  of the Chow-Liu algorithm we have an edge set  $E^{(k)}$  which corresponds to a forest  $\hat{F}_{n_1}^{(k)}$  with  $k$  edges, where

**Algorithm 5.3.1** Chow-Liu (Kruskal)

- 
- 1: **Input** data  $\mathcal{D}_1 = \{X^{(1)}, \dots, X^{(n_1)}\}$ .
  - 2: Calculate  $\hat{M}_{n_1}$ , according to (5.23), (5.24), and (5.25).
  - 3: Initialize  $E^{(0)} = \emptyset$
  - 4: **for**  $k = 1, \dots, d - 1$  **do**
  - 5:    $(i^{(k)}, j^{(k)}) \leftarrow \arg \max_{(i,j)} \hat{M}_{n_1}(i, j)$  such that  $E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$  does not contain a cycle
  - 6:    $E^{(k)} \leftarrow E^{(k-1)} \cup \{(i^{(k)}, j^{(k)})\}$
  - 7: **Output** tree  $\hat{F}_{n_1}^{(d-1)}$  with edge set  $E^{(d-1)}$ .
- 

$\hat{F}_{n_1}^{(0)}$  is the union of  $d$  disconnected nodes. To select  $k$ , we choose among the  $d$  trees  $\hat{F}_{n_1}^{(0)}, \hat{F}_{n_1}^{(1)}, \dots, \hat{F}_{n_1}^{(d-1)}$ .

Let  $\hat{p}_{n_2}(x_i, x_j)$  and  $\hat{p}_{n_2}(x_k)$  be defined as in (5.23) and (5.24), but now evaluated solely based on the held-out data in  $\mathcal{D}_2$ . For a density  $p_F$  that is supported by a forest  $F$ , we define the held-out negative log-likelihood risk as

$$\hat{R}_{n_2}(p_F) \quad (5.26)$$

$$= - \sum_{(i,j) \in E_F} \int_{\mathcal{X}_i \times \mathcal{X}_j} \hat{p}_{n_2}(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i) p(x_j)} dx_i dx_j \quad (5.27)$$

$$- \sum_{k \in V_F} \int_{\mathcal{X}_k} \hat{p}_{n_2}(x_k) \log p(x_k) dx_k. \quad (5.28)$$

The selected forest is then  $\hat{F}_{n_1}^{(\hat{k})}$  where

$$\hat{k} = \arg \min_{k \in \{0, \dots, d-1\}} \hat{R}_{n_2} \left( \hat{p}_{\hat{F}_{n_1}^{(k)}} \right) \quad (5.29)$$

and where  $\hat{p}_{\hat{F}_{n_1}^{(k)}}$  is computed using the density estimate  $\hat{p}_{n_1}$  constructed on  $\mathcal{D}_1$ .

For computational simplicity, we can also estimate  $\hat{k}$  as

$$\begin{aligned} \hat{k} &= \arg \max_{k \in \{0, \dots, d-1\}} \frac{1}{n_2} \sum_{s \in \mathcal{D}_2} \log \left( \prod_{(i,j) \in E^{(k)}} \frac{\hat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\hat{p}_{n_1}(X_i^{(s)}) \hat{p}_{n_1}(X_j^{(s)})} \prod_{k \in V_{\hat{F}_{n_1}^{(k)}}} \hat{p}_{n_1}(X_k^{(s)}) \right) \\ &= \arg \max_{k \in \{0, \dots, d-1\}} \frac{1}{n_2} \sum_{s \in \mathcal{D}_2} \log \left( \prod_{(i,j) \in E^{(k)}} \frac{\hat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\hat{p}_{n_1}(X_i^{(s)}) \hat{p}_{n_1}(X_j^{(s)})} \right). \end{aligned} \quad (5.30)$$

This minimization can be efficiently carried out by iterating over the  $d - 1$  edges in  $\hat{F}_{n_1}^{(d-1)}$ .

Once  $\hat{k}$  is obtained, the final forest density estimate is given by

$$\hat{p}_n(x) = \prod_{(i,j) \in E(\hat{k})} \frac{\hat{p}_{n_1}(x_i, x_j)}{\hat{p}_{n_1}(x_i) \hat{p}_{n_1}(x_j)} \prod_k \hat{p}_{n_1}(x_k). \quad (5.31)$$

**Remark 5.1.** For computational efficiency, Step 1 can be carried out simultaneously with Step 2. In particular, during the Chow-Liu iteration, whenever an edge is added to  $E^{(k)}$ , the log-likelihood of the resulting density estimator on  $\mathcal{D}_2$  can be immediately computed. A more efficient algorithm to speed up the computation of the mutual information matrix is discussed in the later Appendix section.

### 5.3.3 Building a forest on held-out data

Another approach to estimating the forest structure is to estimate the marginal densities on the training set, but only build graphs on the held-out data. To do so, we first estimate the univariate and bivariate kernel density estimates using  $\mathcal{D}_1$ , denoted by  $\hat{p}_{n_1}(x_i)$  and  $\hat{p}_{n_1}(x_i, x_j)$ . We also construct a new set of univariate and bivariate kernel density estimates using  $\mathcal{D}_2$ ,  $\hat{p}_{n_2}(x_i)$  and  $\hat{p}_{n_2}(x_i, x_j)$ . We then estimate the “cross-entropies” of the kernel density estimates  $\hat{p}_{n_1}$  for each pair of variables by computing

$$\hat{I}_{n_2, n_1}(X_i, X_j) = \int \hat{p}_{n_2}(x_i, x_j) \log \frac{\hat{p}_{n_1}(x_i, x_j)}{\hat{p}_{n_1}(x_i) \hat{p}_{n_1}(x_j)} dx_i dx_j \quad (5.32)$$

$$\approx \frac{1}{m^2} \sum_{k=1}^m \sum_{\ell=1}^m \hat{p}_{n_2}(x_{ki}, x_{\ell j}) \log \frac{\hat{p}_{n_1}(x_{ki}, x_{\ell j})}{\hat{p}_{n_1}(x_{ki}) \hat{p}_{n_1}(x_{\ell j})}. \quad (5.33)$$

Our method is to use  $\hat{I}_{n_2, n_1}(X_i, X_j)$  as edge weights on a full graph and run Kruskal’s algorithm until we encounter edges with negative weight. Let  $\mathcal{F}$  be the set of all forests and  $\hat{w}_{n_2}(i, j) = \hat{I}_{n_2, n_1}(X_i, X_j)$ . The final forest is then

$$\hat{F}_{n_2} = \arg \max_{F \in \mathcal{F}} \hat{w}_{n_2}(F) = \arg \min_{F \in \mathcal{F}} \hat{R}_{n_2}(\hat{p}_F) \quad (5.34)$$

By building a forest on held-out data, we directly cross-validate over *all* forests.

## 5.4 STATISTICAL PROPERTIES

In this section we present our theoretical results on risk consistency, structure selection consistency, and estimation consistency of the forest density estimate  $\hat{p}_n = \hat{p}_{\hat{F}_d^{(k)}}$ .

To establish some notation, we write  $a_n = \Omega(b_n)$  if there exists a constant  $c$  such that  $a_n \geq cb_n$  for sufficiently large  $n$ . We also write  $a_n \asymp b_n$  if there exists a constant  $c$  such that  $a_n \leq cb_n$  and  $b_n \leq ca_n$  for sufficiently large  $n$ . Given a



$d$ -dimensional function  $f$  on the domain  $\mathcal{X}$ , we denote its  $L_2(P)$ -norm and sup-norm as

$$\|f\|_{L_2(P)} = \sqrt{\int_{\mathcal{X}} f^2(x) dP_X(x)}, \quad \|f\|_{\infty} = \sup_{x \in \mathcal{X}} |f(x)| \quad (5.35)$$

where  $P_X$  is the probability measure induced by  $X$ . Throughout this section, all constants are treated as generic values, and as a result they can change from line to line.

In our use of a data splitting scheme, we always adopt equally sized splits for simplicity, so that  $n_1 = n_2 = n/2$ , noting that this does not affect the final rate of convergence.

#### 5.4.1 Assumptions on the density

Fix  $\beta > 0$ . For any  $d$ -tuple  $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$  and  $x = (x_1, \dots, x_d) \in \mathcal{X}$ , we define  $x^\alpha = \prod_{j=1}^d x_j^{\alpha_j}$ . Let  $D^\alpha$  denote the differential operator

$$D^\alpha = \frac{\partial^{\alpha_1 + \dots + \alpha_d}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}. \quad (5.36)$$

For any real-valued  $d$ -dimensional function  $f$  on  $\mathcal{X}$  that is  $\lfloor \beta \rfloor$ -times continuously differentiable at point  $x_0 \in \mathcal{X}$ , let  $P_{f, x_0}^{(\beta)}(x)$  be its Taylor polynomial of degree  $\lfloor \beta \rfloor$  at point  $x_0$ :

$$P_{f, x_0}^{(\beta)}(x) = \sum_{\alpha_1 + \dots + \alpha_d \leq \lfloor \beta \rfloor} \frac{(x - x_0)^\alpha}{\alpha_1! \dots \alpha_d!} D^\alpha f(x_0). \quad (5.37)$$

Fix  $L > 0$ , and denote by  $\Sigma(\beta, L, r, x_0)$  the set of functions  $f : \mathcal{X} \rightarrow \mathbb{R}$  that are  $\lfloor \beta \rfloor$ -times continuously differentiable at  $x_0$  and satisfy

$$\left| f(x) - P_{f, x_0}^{(\beta)}(x) \right| \leq L \|x - x_0\|_2^\beta, \quad \forall x \in \mathcal{B}(x_0, r) \quad (5.38)$$

where  $\mathcal{B}(x_0, r) = \{x : \|x - x_0\|_2 \leq r\}$  is the  $L_2$ -ball of radius  $r$  centered at  $x_0$ . The set  $\Sigma(\beta, L, r, x_0)$  is called the  $(\beta, L, r, x_0)$ -locally Hölder class of functions. Given a set  $A$ , we define

$$\Sigma(\beta, L, r, A) = \bigcap_{x_0 \in A} \Sigma(\beta, L, r, x_0). \quad (5.39)$$

The following are the regularity assumptions we make on the true density function  $p^*(x)$ .

**Assumption 5.1.** For any  $1 \leq i < j \leq d$ , we assume

(D1) there exist  $L_1 > 0$  and  $L_2 > 0$  such that for any  $c > 0$  the true bivariate and univariate densities satisfy

$$p^*(x_i, x_j) \in \Sigma\left(\beta, L_2, c (\log n/n)^{\frac{1}{2\beta+2}}, \mathcal{X}_i \times \mathcal{X}_j\right) \quad (5.40)$$

and

$$p^*(x_i) \in \Sigma \left( \beta, L_1, c (\log n / n)^{\frac{1}{2\beta+1}}, \mathcal{X}_i \right); \quad (5.41)$$

(D2) there exists two constants  $c_1$  and  $c_2$  such that

$$c_1 \leq \inf_{x_i, x_j \in \mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \leq \sup_{x_i, x_j \in \mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \leq c_2 \quad (5.42)$$

$\mu$ -almost surely.

These assumptions are mild, in the sense that instead of adding constraints on the joint density  $p^*(x)$ , we only add regularity conditions on the bivariate and univariate marginals.

#### 5.4.2 Assumptions on the kernel

An important ingredient in our analysis is an exponential concentration result for the kernel density estimate, due to [Giné and Guillou \[2002\]](#). We first specify the requirements on the kernel function  $K(\cdot)$ .

Let  $(\Omega, \mathcal{A})$  be a measurable space and let  $\mathcal{F}$  be a uniformly bounded collection of measurable functions.

**Definition 5.4.**  $\mathcal{F}$  is a bounded measurable VC class of functions with characteristics  $A$  and  $v$  if it is separable and for every probability measure  $P$  on  $(\Omega, \mathcal{A})$  and any  $0 < \epsilon < 1$ ,

$$N \left( \epsilon \|F\|_{L_2(P)}, \mathcal{F}, \|\cdot\|_{L_2(P)} \right) \leq \left( \frac{A}{\epsilon} \right)^v, \quad (5.43)$$

where  $F(x) = \sup_{f \in \mathcal{F}} |f(x)|$  and  $N(\epsilon, \mathcal{F}, \|\cdot\|_{L_2(P)})$  denotes the  $\epsilon$ -covering number of the metric space  $(\Omega, \|\cdot\|_{L_2(P)})$ ; that is, the smallest number of balls of radius no larger than  $\epsilon$  (in the norm  $\|\cdot\|_{L_2(P)}$ ) needed to cover  $\mathcal{F}$ .

The one-dimensional density estimates are constructed using a kernel  $K$ , and the two-dimensional estimates are constructed using the product kernel

$$K_2(x, y) = K(x) \cdot K(y). \quad (5.44)$$

**Assumption 5.2.** The kernel  $K$  satisfies the following properties.

(K1)  $\int_{-\infty}^{\infty} K(u) du = 1$ ,  $\int_{-\infty}^{\infty} K^2(u) du < \infty$  and  $\sup_{u \in \mathbb{R}} K(u) \leq c$  for some constant  $c$ .

(K2)  $K$  is a finite linear combination of functions  $g$  whose epigraphs  $\text{epi}(g) = \{(s, u) : g(s) \geq u\}$ , can be represented as a finite number of Boolean operations (union and intersection) among sets of the form  $\{(s, u) : Q(s, u) \geq \phi(u)\}$ , where  $Q$  is a polynomial on  $\mathbb{R} \times \mathbb{R}$  and  $\phi$  is an arbitrary real function.

(K3)  $K$  has a compact support and for any  $\ell \geq 1$  and  $1 \leq \ell' \leq \lfloor \beta \rfloor$

$$\int |t|^\beta |K(t)| dt < \infty, \text{ and } \int |K(t)|^\ell dt < \infty, \int t^{\ell'} K(t) dt = 0. \quad (5.45)$$

Assumptions (K1), (K2) and (K3) are mild. As pointed out by [Nolan and Pollard \[1987\]](#), both the pyramid (truncated or not) kernel and the boxcar kernel satisfy them. It follows from (K2) that the classes of functions

$$\mathcal{F}_1 = \left\{ \frac{1}{h_1} K\left(\frac{u - \cdot}{h_1}\right) : u \in \mathbb{R}, h_1 > 0 \right\} \quad (5.46)$$

$$\mathcal{F}_2 = \left\{ \frac{1}{h_2^2} K\left(\frac{u - \cdot}{h_2}\right) K\left(\frac{t - \cdot}{h_2}\right) : u, t \in \mathbb{R}, h_2 > 0 \right\} \quad (5.47)$$

are bounded VC classes, in the sense of Definition 5.4. Assumption (K3) essentially says that the kernel  $K(\cdot)$  should be  $\beta$ -valid; see [Tsybakov \[2008\]](#) and Definition 6.1 in [Rigollet and Vert \[2009\]](#) for further details about this assumption.

We choose the bandwidths  $h_1$  and  $h_2$  used in the one-dimensional and two-dimensional kernel density estimates to satisfy

$$h_1 \asymp \left( \frac{\log n}{n} \right)^{\frac{1}{1+2\beta}} \quad (5.48)$$

$$h_2 \asymp \left( \frac{\log n}{n} \right)^{\frac{1}{2+2\beta}}. \quad (5.49)$$

This choice of bandwidths ensures the optimal rate of convergence.

### 5.4.3 Risk consistency

Given the above assumptions, we first present a key lemma that establishes the rates of convergence of bivariate and univariate kernel density estimates in the sup norm. The proof of this and our other technical results are provided in the later appendix sections.

**Lemma 5.1.** *Under Assumptions 5.1 and 5.2, and choosing bandwidths satisfying (5.48) and (5.49), the bivariate and univariate kernel density estimates  $\hat{p}(x_i, x_j)$  and  $\hat{p}(x_k)$  in (5.23) and (5.24) satisfy*

$$\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |\hat{p}(x_i, x_j) - p^*(x_i, x_j)| \quad (5.50)$$

$$= O_P \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} \right) \quad (5.51)$$

and

$$\max_{k \in \{1, \dots, d\}} \sup_{x_k \in \mathcal{X}_k} |\hat{p}(x_k) - p^*(x_k)| = O_P \left( \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right). \quad (5.52)$$

To describe the risk consistency result, let  $\mathcal{P}_d^{(d-1)} = \mathcal{P}_d$  be the family of densities that are supported by forests with at most  $d - 1$  edges, as already defined in (5.5). For  $0 \leq k \leq d - 1$ , we define  $\mathcal{P}_d^{(k)}$  as the family of  $d$ -dimensional densities that are supported by forests with at most  $k$  edges. Then

$$\mathcal{P}_d^{(0)} \subset \mathcal{P}_d^{(1)} \subset \dots \subset \mathcal{P}_d^{(d-1)}. \quad (5.53)$$

Now, due to the nesting property (5.53), we have

$$\inf_{q_F \in \mathcal{P}_d^{(0)}} R(q_F) \geq \inf_{q_F \in \mathcal{P}_d^{(1)}} R(q_F) \geq \dots \geq \inf_{q_F \in \mathcal{P}_d^{(d-1)}} R(q_F). \quad (5.54)$$

We first analyze the forest density estimator obtained using a fixed number of edges  $k < d$ ; specifically, consider stopping the Chow-Liu algorithm in Stage 1 after  $k$  iterations. This is in contrast to the algorithm described in 5.3.2, where the pruned tree size is automatically determined on the held out data. While this is not very realistic in applications, since the tuning parameter  $k$  is generally hard to choose, the analysis in this case is simpler, and can be directly exploited to analyze the more complicated data-dependent method.

**Theorem 5.1** (Risk consistency). *Let  $\hat{p}_{\hat{F}_d^{(k)}}$  be the forest density estimate with*

$$|E(\hat{F}_d^{(k)})| = k$$

*obtained after the first  $k$  iterations of the Chow-Liu algorithm, for some  $k \in \{0, \dots, d - 1\}$ . Under Assumptions 5.1 and 5.2, we have*

$$R(\hat{p}_{\hat{F}_d^{(k)}}) - \inf_{q_F \in \mathcal{P}_d^{(k)}} R(q_F) = O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right). \quad (5.55)$$

Note that this result allows the dimension  $d$  to increase at a rate

$$d = o \left( \sqrt{n^{2\beta/(1+2\beta)} / \log n} \right)$$

and the number of edges  $k$  to increase at a rate  $o \left( \sqrt{n^{\beta/(1+\beta)} / \log n} \right)$ , with the excess risk still decreasing to zero asymptotically.

The above results can be used to prove a risk consistency result for the data-dependent pruning method using the data-splitting scheme described in Section 5.3.2.

**Theorem 5.2.** *Let  $\hat{p}_{\hat{F}_d^{(\hat{k})}}$  be the forest density estimate using the data-dependent pruning method in Section 5.3.2, and let  $\hat{p}_{\hat{F}_d^{(k)}}$  be the estimate with  $|E(\hat{F}_d^{(k)})| = k$  obtained*

after the first  $k$  iterations of the Chow-Liu algorithm. Under Assumptions 5.1 and 5.2, we have

$$R(\hat{p}_{\hat{F}_d^{(k)}}) - \min_{0 \leq k \leq d-1} R(\hat{p}_{\hat{F}_d^{(k)}}) \quad (5.56)$$

$$= O_P \left( (k^* + \hat{k}) \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right) \quad (5.57)$$

where  $k^* = \arg \min_{0 \leq k \leq d-1} R(\hat{p}_{\hat{F}_d^{(k)}})$ .

The proof of this theorem is given in the appendix. A parallel result can be obtained for the method described in Section 5.3.3, which builds the forest by running Kruskal's algorithm on the heldout data.

**Theorem 5.3.** Let  $\hat{F}_{n_2}$  be the forest obtained using Kruskal's algorithm on held-out data, and let  $\hat{k} = |\hat{F}_{n_2}|$  be the number of edges in  $\hat{F}_{n_2}$ . Then

$$R(\hat{p}_{\hat{F}_{n_2}}) - \min_{F \in \mathcal{F}} R(\hat{p}_F) \quad (5.58)$$

$$= O_P \left( (k^* + \hat{k}) \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right) \quad (5.59)$$

where  $k^* = |F^*|$  is the number of edges in the optimal forest  $F^* = \arg \min_{F \in \mathcal{F}} R(\hat{p}_F)$ .

#### 5.4.4 Structure selection consistency

In this section, we provide conditions guaranteeing that the procedure is structure selection consistent. Again, we do not assume the true density  $p^*(x)$  is consistent with a forest; rather, we are interested in comparing the estimated forest structure to the oracle forest which minimizes the risk. In this way our result differs from that in Tan et al. [2009a], although there are similarities in the analysis.

By Proposition 5.2, we can define

$$p_{F_d^{(k)}}^* = \arg \min_{q_F \in \mathcal{P}_d^{(k)}} R(q_F). \quad (5.60)$$

Thus  $F_d^{(k)}$  is the optimal forest within  $\mathcal{P}_d^{(k)}$  that minimizes the negative log-likelihood loss. Let  $\hat{F}_d^{(k)}$  be the estimated forest structure, fixing the number of edges at  $k$ ; we want to study conditions under which

$$\mathbb{P} \left( \hat{F}_d^{(k)} = F_d^{(k)} \right) \rightarrow 1. \quad (5.61)$$

Let's first consider the population version of the algorithm—if the algorithm cannot recover the best forest  $F_d^{(k)}$  in this ideal case, there is no hope for stable recovery in the data version. The key observation is that the graph selected

by the Chow-Liu algorithm only depends on the relative order of the edges with respect to mutual information, not on the specific mutual information values. Let

$$\mathcal{E} = \left\{ \{(i, j), (k, \ell)\} : i < j \text{ and } k < \ell, j \neq \ell \text{ and } i, j, k, \ell \in \{1, \dots, d\} \right\}. \quad (5.62)$$

The cardinality of  $\mathcal{E}$  is

$$|\mathcal{E}| = O(d^4). \quad (5.63)$$

Let  $e = (i, j)$  be an edge; the corresponding mutual information associated with  $e$  is denoted as  $I_e$ . If for all  $(e, e') \in \mathcal{E}$ , we have  $I_e \neq I_{e'}$ , the population version of the Chow-Liu algorithm will always obtain the unique solution  $F_d^{(k)}$ . However, this condition is, in a sense, both too weak and too strong. It is too weak because the sample estimates of the mutual information values will only approximate the population values, and could change the relative ordering of some edges. However, the assumption is too strong because, in fact, the relative order of many edge pairs might be changed without affecting the graph selected by the algorithm. For instance, when  $k \geq 2$  and  $I_e$  and  $I_{e'}$  are the largest two mutual information values, it's guaranteed that  $e$  and  $e'$  will both be included in the learned forest  $F_d^{(k)}$  whether  $I_e > I_{e'}$  or  $I_e < I_{e'}$ .

Define the *crucial set*  $\mathcal{J} \subset \mathcal{E}$  to be a set of pairs of edges  $(e, e')$  such that  $I_e \neq I_{e'}$  and flipping the relative order of  $I_e$  and  $I_{e'}$  changes the learned forest structure in the population Chow-Liu algorithm, with positive probability. Here, we assume that the Chow-Liu algorithm randomly selects an edge when a tie occurs.

The cardinality  $|\mathcal{J}|$  of the crucial set is a function of the true density  $p^*(x)$ , and we can expect  $|\mathcal{J}| \ll |\mathcal{E}|$ . The next assumption provides a sufficient condition for the two-stage procedure to be structure selection consistent.

**Assumption 5.3.** *Let the crucial set  $\mathcal{J}$  be defined as before. Suppose that*

$$\min_{((i,j),(k,\ell)) \in \mathcal{J}} |I(X_i; X_j) - I(X_k; X_\ell)| \geq 2L_n \quad (5.64)$$

$$\text{where } L_n = \Omega \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} \right).$$

This assumption is satisfied in many cases. For example, in a graph with population mutual informations differing by a constant, the assumption holds. Assumption 5.3 is trivially satisfied if  $\frac{n^{\beta/(1+\beta)}}{\log n + \log d} \rightarrow \infty$ . However, if two pairs of edges belonging  $\mathcal{J}$  have the same marginal distributions, the assumption may fail.

**Theorem 5.4** (Structure selection consistency). *Let  $F_d^{(k)}$  be the optimal forest within  $\mathcal{P}_d^{(k)}$  that minimizes the negative log-likelihood loss. Let  $\widehat{F}_d^{(k)}$  be the estimated forest with  $|\widehat{F}_d^{(k)}| = k$ . Under Assumptions 5.1, 5.2, and 5.3, we have*

$$\mathbb{P}\left(\widehat{F}_d^{(k)} = F_d^{(k)}\right) \rightarrow 1 \quad (5.65)$$

as  $n \rightarrow \infty$ .

The proof shows that our method is structure selection consistent as long as the dimension increases as  $d = o\left(\exp(n^{\beta/(1+\beta)})\right)$ ; in this case the error decreases at the rate  $o\left(\exp\left(4\log d - c(\log n)^{\frac{1}{1+\beta}} \log d\right)\right)$ .

#### 5.4.5 Estimation consistency

Estimation consistency can be easily established using the structure selection consistency result above. Define the event  $\mathcal{M}_k = \{\widehat{F}_d^{(k)} = F_d^{(k)}\}$ . Theorem 5.4 shows that  $\mathbb{P}(\mathcal{M}_k^c) \rightarrow 0$  as  $n$  goes to infinity.

**Lemma 5.2.** *Let  $\widehat{p}_{\widehat{F}_d^{(k)}}$  be the forest-based kernel density estimate for some fixed  $k \in \{0, \dots, d-1\}$ , and let*

$$p_{F_d^{(k)}}^* = \arg \min_{q_F \in \mathcal{P}_d^{(k)}} R(q_F). \quad (5.66)$$

Under the assumptions of Theorem 5.4,

$$D(p_{F_d^{(k)}}^* \| \widehat{p}_{\widehat{F}_d^{(k)}}) = R(\widehat{p}_{\widehat{F}_d^{(k)}}) - R(p_{F_d^{(k)}}^*) \quad (5.67)$$

on the event  $\mathcal{M}_k$ .

*Proof.* According to [Bach and Jordan \[2003\]](#), for a given forest  $F$  and a target distribution  $p^*(x)$ ,

$$D(p^* \| q_F) = D(p^* \| p_F^*) + D(p_F^* \| q_F) \quad (5.68)$$

for all distributions  $q_F$  that are supported by  $F$ . We further have

$$D(p^* \| q) = \int_{\mathcal{X}} p^*(x) \log p^*(x) - \int_{\mathcal{X}} p^*(x) \log q(x) dx \quad (5.69)$$

$$= \int_{\mathcal{X}} p^*(x) \log p^*(x) dx + R(q) \quad (5.70)$$

for any distribution  $q$ . Using (5.68) and (5.70), and conditioning on the event  $\mathcal{M}_k$ , we have

$$D(p_{F_d}^* \| \hat{p}_{\hat{F}_d^{(k)}}) \quad (5.71)$$

$$= D(p^* \| \hat{p}_{\hat{F}_d^{(k)}}) - D(p^* \| p_{F_d}^*) \quad (5.72)$$

$$\begin{aligned} &= \int_{\mathcal{X}} p^*(x) \log p^*(x) dx + R(\hat{p}_{\hat{F}_d^{(k)}}) - \int_{\mathcal{X}} p^*(x) \log p^*(x) dx - R(p_{F_d}^*) \\ &= R(\hat{p}_{\hat{F}_d^{(k)}}) - R(p_{F_d}^*), \end{aligned}$$

which gives the desired result.  $\square$

The above lemma combined with Theorem 5.1 allows us to obtain the following estimation consistency result, the proof of which is omitted.

**Corollary 5.1** (Estimation consistency). *Under Assumptions 5.1, 5.2, and 5.3, we have*

$$D(p_{F_d}^* \| \hat{p}_{\hat{F}_d^{(k)}}) = O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right). \quad (5.73)$$

## 5.5 EXPERIMENTAL RESULTS

In this section, we report numerical results on both synthetic datasets and microarray data. We mainly compare the forest density estimator with sparse Gaussian graphical models, fitting a multivariate Gaussian with a sparse inverse covariance matrix. The sparse Gaussian models are estimated using the graphical lasso algorithm (glasso) of Friedman et al. [2007], which is a refined version of an algorithm first derived by Banerjee et al. [2008]. Since the glasso typically results in a large parameter bias as a consequence of the  $\ell_1$  regularization, we also compare with a method that we call the *refit glasso*, which is a two-step procedure—in the first step, a sparse inverse covariance matrix is obtained by the glasso; in the second step, a Gaussian model is refit without  $\ell_1$  regularization, but enforcing the sparsity pattern obtained in the first step.

To quantitatively compare the performance of these estimators, we calculate the log-likelihood of all methods on a held-out dataset  $\mathcal{D}_2$ . With  $\hat{\mu}_{n_1}$  and  $\hat{\Omega}_{n_1}$  denoting the estimates from the Gaussian model, the held-out log-likelihood can be explicitly evaluated as

$$\ell_{\text{gauss}} = -\frac{1}{n_2} \sum_{s \in \mathcal{D}_2} \left\{ \frac{1}{2} (X^{(s)} - \hat{\mu}_{n_1})^T \hat{\Omega}_{n_1} (X^{(s)} - \hat{\mu}_{n_1}) + \frac{1}{2} \log \left( \frac{|\hat{\Omega}_{n_1}|}{(2\pi)^d} \right) \right\}.$$



For a given tree structure  $\hat{F}$ , the held-out log-likelihood for the forest density estimator is

$$\ell_{\text{fde}} = \frac{1}{n_2} \sum_{s \in \mathcal{D}_2} \log \left( \prod_{(i,j) \in E(\hat{F})} \frac{\hat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\hat{p}_{n_1}(X_i^{(s)})\hat{p}_{n_1}(X_j^{(s)})} \prod_{k \in V_{\hat{F}}} \hat{p}_{n_1}(X_k^{(s)}) \right), \quad (5.74)$$

where  $\hat{p}_{n_1}(\cdot)$  are the corresponding kernel density estimates using the plug-in bandwidths.

Since the held-out log-likelihood of the forest density estimator is indexed by the number of edges included in the tree, while the held-out log-likelihoods of the glasso and the refit glasso are indexed by a continuously varying regularization parameter, we need to find a way to calibrate them. To address this issue, we plot the held-out log-likelihood of the forest density estimator as a step function indexed by the tree size. We then run the full path of the glasso and discretize it according to the corresponding sparsity level, i.e., how many edges are selected for each value of the regularization parameter. The size of the forest density estimator and the sparsity level of the glasso (and the refit glasso) can then be aligned for a fair comparison.

### 5.5.1 Synthetic data

We use a procedure to generate high dimensional Gaussian and non-Gaussian data which are consistent with an undirected graph. We generate high dimensional graphs that contain cycles, and so are not forests. In dimension  $d = 100$ , we sample  $n_1 = n_2 = 400$  data points from a multivariate Gaussian distribution with mean vector  $\mu = (0.5, \dots, 0.5)$  and inverse covariance matrix  $\Omega$ . The diagonal elements of  $\Omega$  are all 62. We then randomly generate many connected subgraphs containing no more than eight nodes each, and set the corresponding non-diagonal elements in  $\Omega$  at random, drawing values uniformly from  $-30$  to  $-10$ . To obtain non-Gaussian data, we simply transform each dimension of the data by its empirical distribution function; such a transformation preserves the graph structure but the joint distribution is no longer Gaussian (see [Liu et al. \[2009a\]](#)).

To calculate the pairwise mutual information  $\hat{I}(X_i; X_j)$ , we need to numerically evaluate two-dimensional integrals. We first rescale the data into  $[0, 1]^d$  and calculate the kernel density estimates on a grid of points; we choose  $m = 128$  evaluation points  $x_i^{(1)} < x_i^{(2)} < \dots < x_i^{(m)}$  for each dimension  $i$ , and then evaluate the bivariate and the univariate kernel density estimates on this grid.

There are three different kernel density estimates that we use—the bivariate kde, the univariate kde, and the marginalized bivariate kde. Specifically,

the bivariate kernel density estimate on  $x_i, x_j$  based on the observations  $\{X_i^{(s)}, X_j^{(s)}\}_{s \in \mathcal{D}_1}$  is defined as

$$\hat{p}(x_i, x_j) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_{2i}h_{2j}} K\left(\frac{X_i^{(s)} - x_i}{h_{2i}}\right) K\left(\frac{X_j^{(s)} - x_j}{h_{2j}}\right), \quad (5.75)$$

using a product kernel. The bandwidths  $h_{2i}, h_{2j}$  are chosen as

$$h_{2k} = 1.06 \cdot \min \left\{ \hat{\sigma}_k, \frac{\hat{q}_{k,0.75} - \hat{q}_{k,0.25}}{1.34} \right\} \cdot n^{-1/(2\beta+2)}, \quad (5.76)$$

where  $\hat{\sigma}_k$  is the sample standard deviation of  $\{X_k^{(s)}\}_{s \in \mathcal{D}_1}$  and  $\hat{q}_{k,0.75}, \hat{q}_{k,0.25}$  are the 75% and 25% sample quantiles of  $\{X_k^{(s)}\}_{s \in \mathcal{D}_1}$ .

In all the experiments, we set  $\beta = 2$ , such a choice of  $\beta$  and the “plug-in” bandwidth  $h_{2k}$  (and  $h_{1k}$  in the following) is a very common practice in nonparametric Statistics. For more details, see [Fan and Gijbels \[1996\]](#) and [Tsybakov \[2008\]](#).

Given an evaluation point  $x_k$ , the univariate kernel density estimate  $\hat{p}(x_k)$  based on the observations  $\{X_k^{(s)}\}_{s \in \mathcal{D}_1}$  is defined as

$$\hat{p}(x_k) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_{1k}} K\left(\frac{X_k^{(s)} - x_k}{h_{1k}}\right), \quad (5.77)$$

where  $h_{1k} > 0$  is defined as

$$h_{1k} = 1.06 \cdot \min \left\{ \hat{\sigma}_k, \frac{\hat{q}_{k,0.75} - \hat{q}_{k,0.25}}{1.34} \right\} \cdot n^{-1/(2\beta+1)}. \quad (5.78)$$

Finally, the marginal univariate kernel density estimate  $\hat{p}_M(x_k)$  based on the observations  $\{X_k^{(s)}\}_{s \in \mathcal{D}_1}$  is defined by integrating the irrelevant dimension out of the bivariate kernel density estimates  $\hat{p}(x_j, x_k)$  on the unit square  $[0, 1]^2$ . Thus,

$$\hat{p}_M(x_k) = \frac{1}{m-1} \sum_{\ell=1}^m \hat{p}(x_j^{(\ell)}, x_k). \quad (5.79)$$

With the above definitions of the bivariate and univariate kernel density estimates, we consider estimating the mutual information  $I(X_i; X_j)$  in three different ways, depending on which estimates for the univariate densities are employed.

$$\hat{I}_{\text{fast}}(X_i, X_j) \quad (5.80)$$

$$= \frac{1}{(m-1)^2} \sum_{k'=1}^m \sum_{\ell'=1}^m \hat{p}(x_i^{(k')}, x_j^{(\ell')}) \log \hat{p}(x_i^{(k')}, x_j^{(\ell')}) - \quad (5.81)$$

$$\frac{1}{m-1} \sum_{k'=1}^m \hat{p}(x_i^{(k')}) \log \hat{p}(x_i^{(k')}) - \frac{1}{m-1} \sum_{\ell'=1}^m \hat{p}(x_j^{(\ell')}) \log \hat{p}(x_j^{(\ell')})$$

$$\hat{I}_{\text{medium}}(X_i, X_j) \quad (5.82)$$

$$= \frac{1}{(m-1)^2} \sum_{k'=1}^m \sum_{\ell'=1}^m \hat{p}(x_i^{(k')}, x_j^{(\ell')}) \log \frac{\hat{p}(x_i^{(k')}, x_j^{(\ell')})}{\hat{p}(x_i^{(k')}) \hat{p}(x_j^{(\ell')})}. \quad (5.83)$$

$$\hat{I}_{\text{slow}}(X_i, X_j) \quad (5.84)$$

$$= \frac{1}{(m-1)^2} \sum_{k'=1}^m \sum_{\ell'=1}^m \hat{p}(x_i^{(k')}, x_j^{(\ell')}) \log \hat{p}(x_i^{(k')}, x_j^{(\ell')}) - \quad (5.85)$$

$$\frac{1}{m-1} \sum_{k'=1}^m \hat{p}_M(x_i^{(k')}) \log \hat{p}_M(x_i^{(k')}) - \frac{1}{m-1} \sum_{\ell'=1}^m \hat{p}_M(x_j^{(\ell')}) \log \hat{p}_M(x_j^{(\ell')}).$$

The terms “fast,” “medium” and “slow” refer to the theoretical statistical rates of convergence of the estimators. The “fast” estimate uses one-dimensional univariate kernel density estimators wherever possible. The “medium” estimate uses the one-dimensional kernel density estimates in the denominator of  $p(x_i, x_j) / (p(x_i)p(x_j))$ , but averages with respect to the bivariate density. Finally, the “slow” estimate marginalizes the bivariate densities to estimate the univariate densities. While the rate of convergence is the two-dimensional rate, the “slow” estimate ensures the consistency of the bivariate and univariate densities.

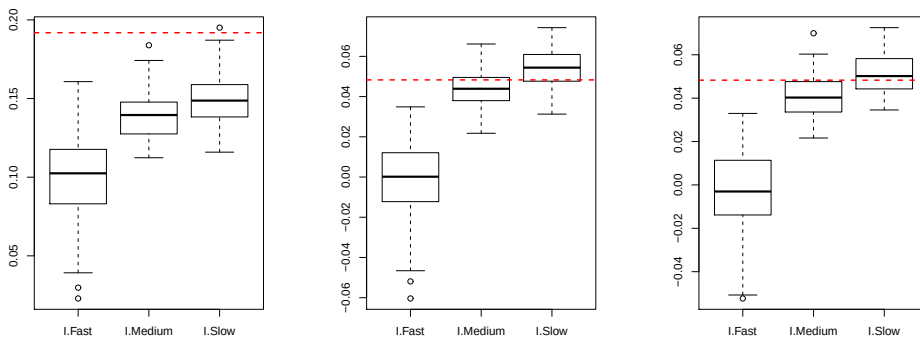


Figure 33.: (Gaussian example) Boxplots of  $\hat{I}_{\text{fast}}$ ,  $\hat{I}_{\text{medium}}$ , and  $\hat{I}_{\text{slow}}$  on three different pairs of variables. The red-dashed horizontal lines represent the population values.

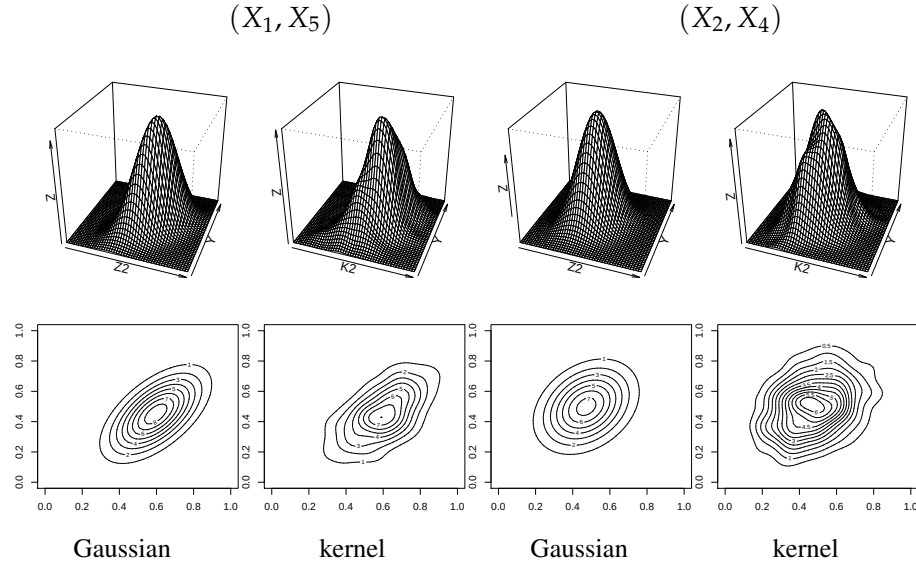


Figure 34.: Perspective and contour plots of the bivariate Gaussian fits vs. the kernel density estimates for two edges of a Gaussian graphical model.

Figure 33 compares  $\hat{I}_{\text{fast}}$ ,  $\hat{I}_{\text{medium}}$ , and  $\hat{I}_{\text{slow}}$  on different pairs of variables. The boxplots are based on 100 trials. Compared to the ground truth, which can be computed exactly in the Gaussian case, we see that the performance of  $\hat{I}_{\text{medium}}$  and  $\hat{I}_{\text{slow}}$  is better than that of  $\hat{I}_{\text{fast}}$ . This is due to the fact that simply replacing the population density with a “plug-in” version can lead to biased estimates; in fact,  $\hat{I}_{\text{fast}}$  is not even guaranteed to be non-negative. In what follows, we employ  $\hat{I}_{\text{medium}}$  for all the calculations, due to its ease of computation and good finite sample performance. Figure 34 compares the bivariate fits of the kernel density estimates and the Gaussian models over four edges. For the Gaussian fits of each edge, we directly calculate the bivariate sample covariance and sample mean and plug them into the bivariate Gaussian density function. From the perspective and contour plots, we see that the bivariate kernel density estimates provide reasonable fits for these bivariate components.

A typical run showing the held-out log-likelihood and estimated graphs is provided in Figure 35. We see that for the Gaussian data, the refit glasso has a higher held-out log-likelihood than the forest density estimator and the glasso. This is expected, since the Gaussian model is correct. For very sparse models, however, the performance of the glasso is worse than that of the forest density estimator, due to the large parameter bias resulting from the  $\ell_1$  regularization. We also observe an efficiency loss in the nonparametric forest density estimator, compared to the refit glasso. The graphs are automatically

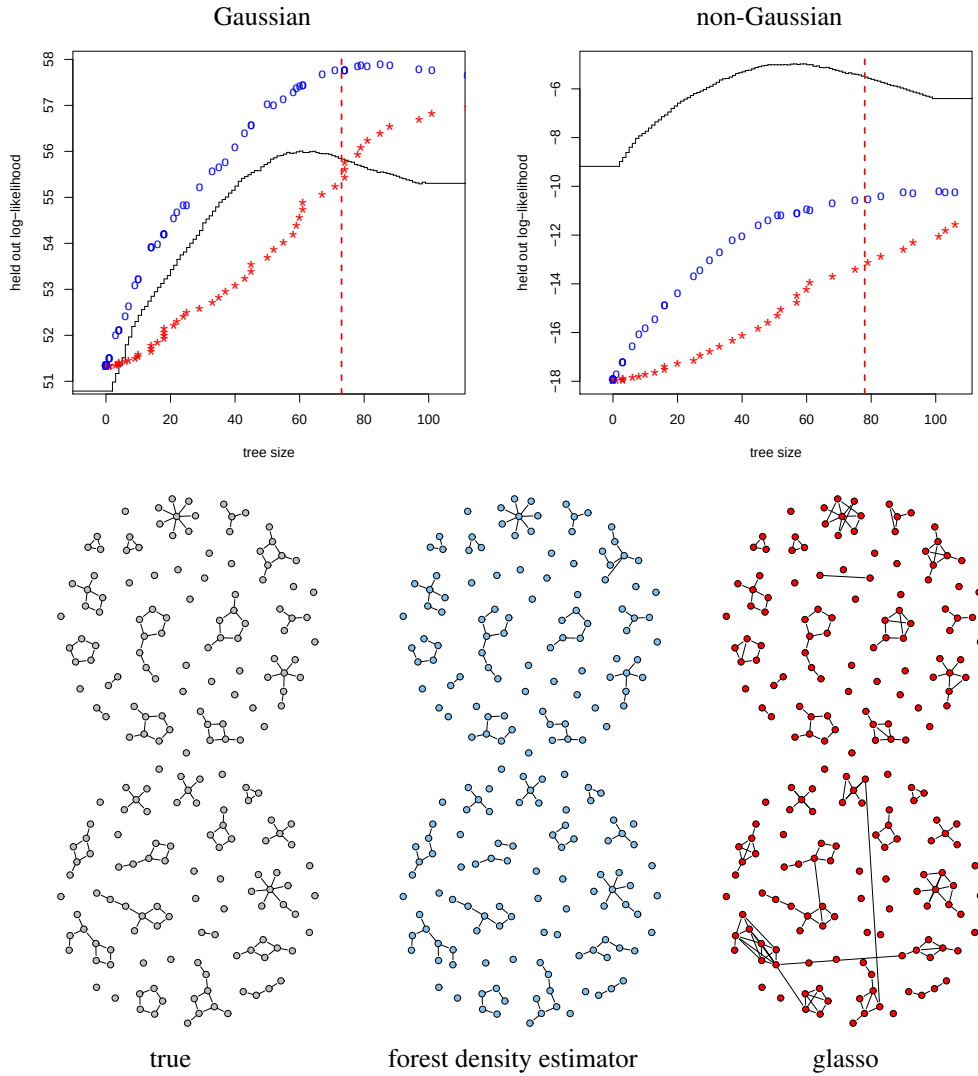


Figure 35.: Synthetic data. Top-left Gaussian, and top-right non-Gaussian: Held-out log-likelihood plots of the forest density estimator (black step function), glasso (red stars), and refit glasso (blue circles), the vertical dashed red line indicates the size of the true graph. Bottom plots show the true and estimated graphs for the Gaussian (second row) and non-Gaussian data (third row).

selected using the held-out log-likelihood, and we see that the nonparametric forest-based kernel density estimator tends to select a sparser model, while the parametric Gaussian models tend to overselect. This observation is new and is quite typical in our simulations. Another observation is that the held-out log-likelihood curve of the glasso becomes flat for less sparse models but never goes down. This suggests that the held-out log-likelihood is not a good model selection criterion for the glasso. For the non-Gaussian data, even though the refit glasso results in a reasonable graph, the forest density estimator performs much better in terms of held-out log-likelihood risk and graph estimation accuracy.

### 5.5.2 *Microarray data*

#### 5.5.2.1 *Arabidopsis thaliana Data*

In this example, we consider a dataset based on Affymetrix GeneChip microarrays for the plant *Arabidopsis thaliana*, [Wille et al., 2004]. The sample size is  $n = 118$ . The expression levels for each chip are pre-processed by a log-transformation and standardization. A subset of 40 genes from the isoprenoid pathway are chosen, and we study the associations among them using the glasso, the refit glasso, and the tree-based kernel density estimator.

From the held-out log-likelihood curves in Figure 36, we see that the tree-based kernel density estimator has a better generalization performance than the glasso and the refit glasso. This is not surprising, given that the true distribution of the data is not Gaussian. Another observation is that for the tree-based kernel density estimator, the held-out log-likelihood curve achieves a maximum when there are only 35 edges in the model. In contrast, the held-out log-likelihood curves of the glasso and refit glasso achieve maxima when there are around 280 edges and 100 edges respectively, while their predictive estimates are still inferior to those of the tree-based kernel density estimator.

Figure 36 also shows the estimated graphs for the tree-based kernel density estimator and the glasso. The graphs are automatically selected based on held-out log-likelihood. The two graphs are clearly different; it appears that the nonparametric tree-based kernel density estimator has the potential to provide different biological insights than the parametric Gaussian graphical model.

#### 5.5.2.2 *HapMap Data*

This dataset comes from Nayak et al. [2009]. The dataset contains Affymetrix chip measured expression levels of 4238 genes for 295 normal subjects in the *Centre d'Etude du Polymorphisme Humain* (CEPH) and the International HapMap collections. The 295 subjects come from four different groups: 148 unrelated grandparents in the CEPH-Utah pedigrees, 43 Han Chinese in

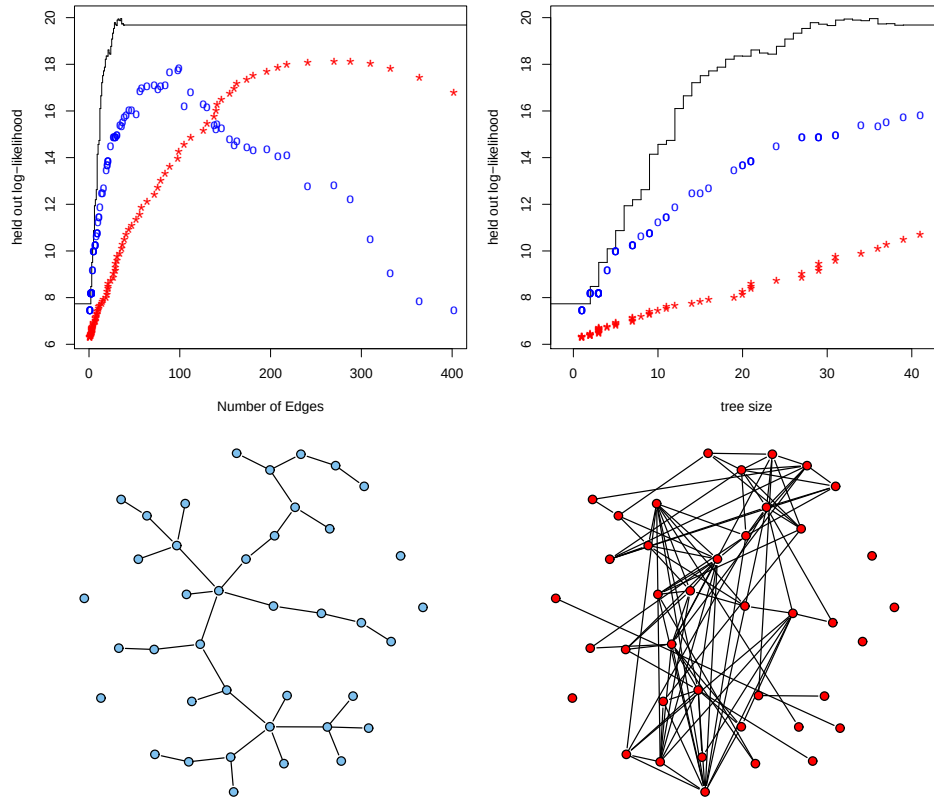


Figure 36.: Results on microarray data. Top: held-out log-likelihood (left) and its zoom-in (right) of the tree-based kernel density estimator (black step function), glasso (red stars), and refit glasso (blue circles). Bottom: estimated graphs using the tree-based estimator (left) and glasso (right).

Beijing, 44 Japanese in Tokyo, and 60 Yoruba in Ibadan, Nigeria. Since we want to find common network patterns across different groups of subjects, we pooled the data together into a  $n = 295$  by  $d = 4238$  numerical matrix.

We estimate the full 4238 node graph using both the forest density estimator (described in Section 5.3.1 and 5.3.2) and the Meinshausen-Bühlmann neighborhood search method as proposed in [Meinshausen and Bühlmann, 2006] with regularization parameter chosen to give it about same number as edges as the forest graph.

To construct the kernel density estimates  $\hat{p}(x_i, x_j)$  we used an array of Nvidia graphical processing units (GPU) to parallelize the computation over the pairs of variables  $X_i$  and  $X_j$ . We discretise the domain of  $(X_i, X_j)$  into a  $128 \times 128$  grid, and correspondingly employ  $128 \times 128$  parallel cells in the GPU array, taking advantage of shared memory in CUDA. Parallelizing in this way increases the total performance by approximately a factor of 50, allowing the experiment to complete in a day.

The forest density estimated graph reveals one strongly connected component of more than 3000 genes and various isolated genes; this is consistent with the analysis in Nayak et al. [2009] and is realistic for the regulatory system of humans. The Gaussian graph contains similar component structure, but the set of edges differs significantly. We also ran the  $t$ -restricted forest algorithm for  $t = 2000$  and it successfully separates the giant component into three smaller components. For visualization purposes, in Figure 37, we show only a 934 gene subgraph of the strongly connected component among the full 4238 node graphs we estimated. More detailed analysis of the biological implications of this work will left as a future study.

## 5.6 CONCLUSION

We have studied forest density estimation for high dimensional data. Forest density estimation skirts the curse of dimensionality by restricting to undirected graphs without cycles, while allowing fully nonparametric marginal densities. The method is computationally simple, and the optimal size of the forest can be robustly selected by a data-splitting scheme. We have established oracle properties and rates of convergence for function estimation in this setting. Our experimental results compared the forest density estimator to the sparse Gaussian graphical model in terms of both predictive risk and the qualitative properties of the estimated graphs for human gene expression array data. Together, these results indicate that forest density estimation can be a useful tool for relaxing the normality assumption in graphical modeling.



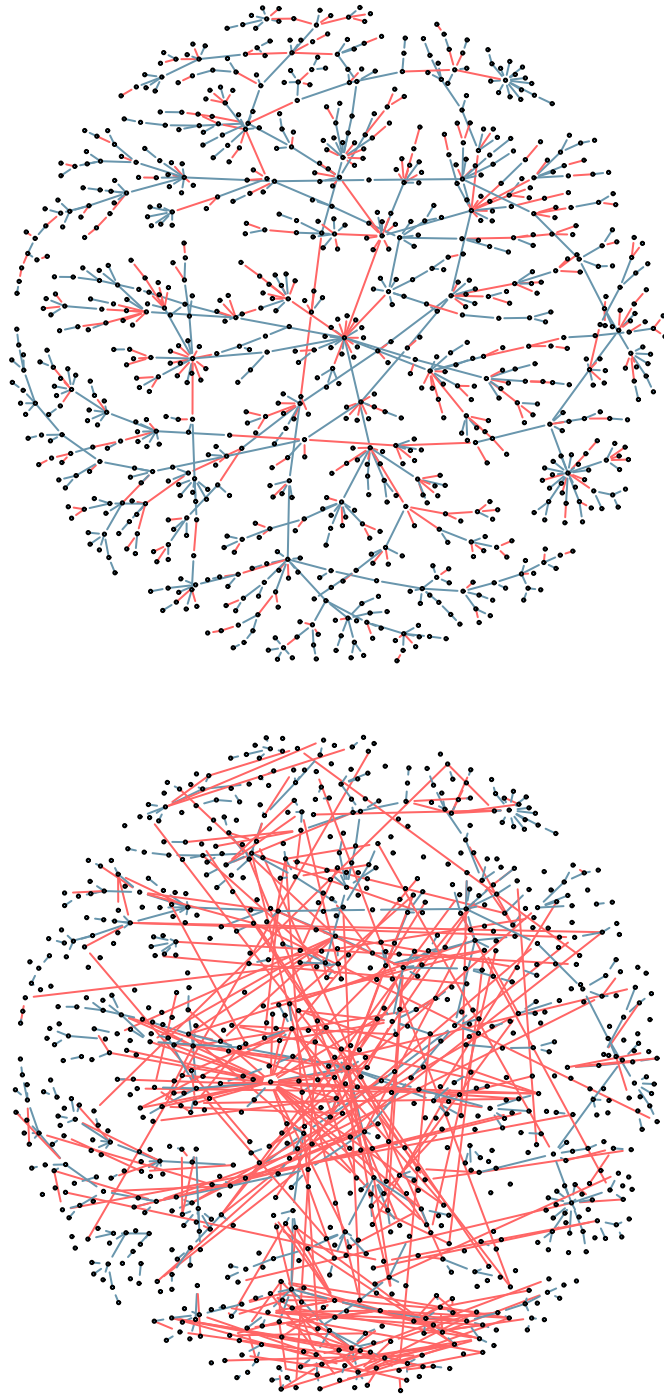


Figure 37.: A 934 gene subgraph of the full estimated 4238 gene network. Upper: estimated forest graph. Lower: estimated Gaussian graph. Red edges in the forest graph are missing from the Gaussian graph and vice versa; the blue edges are shared by both graphs. Note that the layout of the genes is the same for both graphs.

## 5.7 APPENDIX: TECHNICAL PROOFS

## 5.7.0.3 Proof of Lemma 8.3

We only need to consider the more complicated bivariate case (5.51); the result in (5.52) follows from the same line of proof. First, given the assumptions, the following lemma can be obtained by an application of Corollary 2.2 of [Giné and Guillou \[2002\]](#). For a detailed proof, see [Rinaldo and Wasserman \[2009b\]](#).

**Lemma 5.3.** [[Giné and Guillou, 2002](#)] *Let  $\hat{p}$  be a bivariate kernel density estimate using a kernel  $K(\cdot)$  for which Assumption 5.2 holds and suppose that*

$$\sup_{t \in \mathcal{X}^2} \sup_{h_2 > 0} \int_{\mathcal{X}^2} K_2^2(u) p^*(t - uh_2) du \leq D < \infty. \quad (5.86)$$

1. *Let the bandwidth  $h_2$  be fixed. Then there exist constants  $L > 0$  and  $C > 0$ , which depend only on the VC characteristics of  $\mathcal{F}_2$  in (5.47), such that for any  $c_1 \geq C$  and  $0 < \epsilon \leq c_1 D / \|K_2\|_\infty$ , there exists  $n_0 > 0$  which depends on  $\epsilon$ ,  $D$ ,  $\|K_2\|_\infty$  and the VC characteristics of  $K_2$ , such that for all  $n \geq n_0$ ,*

$$\mathbb{P} \left( \sup_{u \in \mathcal{X}^2} |\hat{p}(u) - \mathbb{E}\hat{p}(u)| > 2\epsilon \right) \quad (5.87)$$

$$\leq L \exp \left\{ -\frac{1}{L} \frac{\log(1 + c_1/(4L))}{c_1} \frac{nh_2^2 \epsilon^2}{D} \right\}. \quad (5.88)$$

2. *Let  $h_2 \rightarrow 0$  in such a way that  $nh_2^2/\log h_2 \rightarrow \infty$ , and let  $\epsilon \rightarrow 0$  so that*

$$\epsilon = \Omega \left( \sqrt{\frac{\log r_n}{nh_2^2}} \right), \quad (5.89)$$

where  $r_n = \Omega(h_2^{-1})$ . Then (5.88) holds for sufficiently large  $n$ .

From (D2) in Assumption 5.1 and (K1) in Assumption 5.2, it's easy to see that (5.86) is satisfied. Also, since

$$h_2 \asymp \left( \frac{\log n}{n} \right)^{\frac{1}{2+2\beta}}, \quad (5.90)$$

it's clear that  $nh_2^2/\log h_2 \rightarrow \infty$ . Part 2 of Lemma 5.3 shows that there exist  $c_2$  and  $c_3$  such that

$$\mathbb{P} \left( \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |\hat{p}(x_i, x_j) - \mathbb{E}\hat{p}(x_i, x_j)| \geq \frac{\epsilon}{2} \right) \quad (5.91)$$

$$\leq c_2 \exp \left( -c_3 n^{\frac{\beta}{1+\beta}} (\log n)^{\frac{1}{1+\beta}} \epsilon^2 \right) \quad (5.92)$$

for all  $\epsilon$  satisfying (5.89).

This shows that for any  $i, j \in \{1, \dots, d\}$  with  $i \neq j$ , the bivariate kernel density estimate  $\hat{p}(x_i, x_j)$  is uniformly close to  $\mathbb{E}\hat{p}(x_i, x_j)$ . Note that  $\mathbb{E}\hat{p}(x_i, x_j)$  can be written as

$$\mathbb{E}\hat{p}(x_i, x_j) = \int \frac{1}{h_2^2} K\left(\frac{u_i - x_i}{h_2}\right) K\left(\frac{v_j - x_j}{h_2}\right) p^*(u_i, v_j) du_i dv_j. \quad (5.93)$$

The next lemma, from [Rigollet and Vert \[2009\]](#), provides a uniform deviation bound on the bias term  $\mathbb{E}\hat{p}(x_i, x_j) - p^*(x_i, x_j)$ .

**Lemma 5.4.** [[Rigollet and Vert, 2009](#)] *Under (D1) in Assumption 5.1 and (K3) in Assumption 5.2, we have*

$$\sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |\mathbb{E}\hat{p}(x_i, x_j) - p^*(x_i, x_j)| \leq L_1 h_2^\beta \int_{\mathcal{X}^2} (u^2 + v^2)^{\beta/2} K(u) K(v) dudv. \quad (5.94)$$

where  $L$  is defined in (D1) of Assumption 5.1.

Let  $c_4 = L_1 \int_{\mathcal{X}^2} (u^2 + v^2)^{\beta/2} K(u) K(v) dudv$ . From the discussion of Example 6.1 in [Rigollet and Vert \[2009\]](#) and (K1) in Assumption 5.2, we know that  $c_4 < \infty$  and only depends on  $K$  and  $\beta$ . Therefore

$$\mathbb{P} \left( \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |p^*(x_i, x_j) - \mathbb{E}\hat{p}(x_i, x_j)| \geq \frac{\epsilon}{2} \right) = 0 \quad (5.95)$$

for  $\epsilon \geq 4c_4 h_2^\beta$ .

The desired result in Lemma 8.3 is an exponential probability inequality showing that  $\hat{p}(x_i, x_j)$  is close to  $p^*(x_i, x_j)$ . To obtain this, we use a union bound:

$$\begin{aligned} \mathbb{P} \left( \max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |\hat{p}(x_i, x_j) - p^*(x_i, x_j)| \geq \epsilon \right) \\ \leq d^2 \mathbb{P} \left( \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |\hat{p}(x_i, x_j) - \mathbb{E}\hat{p}(x_i, x_j)| \geq \frac{\epsilon}{2} \right) \\ + d^2 \mathbb{P} \left( \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |p^*(x_i, x_j) - \mathbb{E}\hat{p}(x_i, x_j)| \geq \frac{\epsilon}{2} \right). \end{aligned} \quad (5.96)$$

Choosing

$$\epsilon = \Omega \left( 4c_4 \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} \right), \quad (5.97)$$

the result directly follows by combining (5.92) and (5.95)

## 5.7.0.4 Proof of Theorem 5.1

First, from (D2) in Assumption 5.1 and Lemma 8.3, we have for any  $i \neq j$ ,

$$\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} \left( \frac{\hat{p}(x_i, x_j)}{p^*(x_i, x_j)} - 1 \right) \quad (5.98)$$

$$= O_P \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \right). \quad (5.99)$$

The next lemma bounds the deviation of  $\hat{R}(\hat{p}_F)$  from  $R(p_F^*)$  over different choices of  $F \in \mathcal{F}_d$  with  $|E(F)| \leq k$ . In the following, we let

$$\mathcal{F}_d^{(k)} = \{F \in \mathcal{F}_d : |E(F)| \leq k\} \quad (5.100)$$

denote the family of  $d$ -node forests with no more than  $k$  edges.

**Lemma 5.5.** *Under the assumptions of Theorem 5.1, we have*

$$\sup_{F \in \mathcal{F}_d^{(k)}} |\hat{R}(\hat{p}_F) - R(p_F^*)| = O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right).$$

*Proof.* For any  $F \in \mathcal{F}_d^{(k)}$ , we have

$$|\hat{R}(\hat{p}_F) - R(p_F^*)| \leq A_1 + A_2 \quad (5.101)$$

where

$$A_1 = \left| \sum_{(i,j) \in E(F)} \left( \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i)p^*(x_j)} dx_i dx_j \right. \right. \quad (5.102)$$

$$\left. - \int_{\mathcal{X}_i \times \mathcal{X}_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} dx_i dx_j \right), \quad (5.103)$$

and

$$A_2 = \left| \sum_{k \in V_F} \left( \int_{\mathcal{X}_k} p^*(x_k) \log p^*(x_k) dx_k - \int_{\mathcal{X}_k} \hat{p}(x_k) \log \hat{p}(x_k) dx_k \right) \right|. \quad (5.104)$$

Defining  $p_{ij}^* = p^*(x_i, x_j)$  and  $\hat{p}_{ij} = \hat{p}(x_i, x_j)$ , we further have It now suffices to show that

$$A_1 = O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \right) \quad (5.105)$$

and

$$A_2 = O_P \left( d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right). \quad (5.106)$$

In the sequel, we only prove (5.105); (5.106) follows in the same way. We will also exploit the fact that the univariate and bivariate densities are assumed to have the same

smoothness parameter  $\beta$ , therefore the univariate terms are of higher order, and so can be safely ignored.

To show (5.105), using the fact that  $\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} p^*(x_i, x_j) \leq c_2$ , it's sufficient to prove that

$$\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |p^*(x_i, x_j) - \hat{p}(x_i, x_j)| \quad (5.107)$$

$$= O_P \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \right) \quad (5.108)$$

and

$$\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} D(p_{ij}^* \| \hat{p}_{ij}) = O_P \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \right). \quad (5.109)$$

Equation (5.108) directly follows from (5.51) in Lemma 8.3, while (5.109) follows from the fact that, for any densities  $p$  and  $q$ , where  $q$  is strictly positive,

$$D(p \| q) = \int \frac{p(x)}{q(x)} \log \frac{p(x)}{q(x)} q(x) dx. \quad (5.110)$$

By a Taylor expansion, for  $x \approx 1$ ,

$$x \log x = (x - 1) + o(x - 1) \quad (5.111)$$

and we then have

$$D(p_{ij}^* \| \hat{p}_{ij}) = O_P \left( \sup_{(x_i, x_j) \in \mathcal{X}_i \times \mathcal{X}_j} |p^*(x_i, x_j) - \hat{p}(x_i, x_j)| \right). \quad (5.112)$$

The desired result follows by combining (5.108) and (5.109).  $\square$

The next auxiliary lemma is also needed to obtain the main result. It shows that  $\hat{R}(\hat{p}_F)$  does not deviate much from  $R(\hat{p}_F)$  uniformly over different choices of  $F \in \mathcal{F}_d^{(k)}$ .

**Lemma 5.6.** *Under the assumptions of Theorem 5.1, we have*

$$\sup_{F \in \mathcal{F}_d^{(k)}} |R(\hat{p}_F) - \hat{R}(\hat{p}_F)| = O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right).$$

*Proof.* Following the same line of argument as in Lemma 5.5, we have for all  $F \in \mathcal{F}_d^{(k)}$ ,

$$|R(\hat{p}_F) - \hat{R}(\hat{p}_F)| \quad (5.113)$$

$$\begin{aligned} &\leq \left| \sum_{(i,j) \in E(F)} \left( \int_{\mathcal{X}_i \times \mathcal{X}_j} p^*(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} dx_i dx_j \right. \right. \\ &\quad \left. \left. - \int_{\mathcal{X}_i \times \mathcal{X}_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} dx_i dx_j \right) \right| \\ &\quad + \left| \sum_{k \in V_F} \left( \int_{\mathcal{X}_k} p^*(x_k) \log \hat{p}(x_k) dx_k - \int_{\mathcal{X}_k} \hat{p}(x_k) \log \hat{p}(x_k) dx_k \right) \right| \\ &= O_P \left( \sum_{(i,j) \in E(F)} \left| \sup_{(x_i, x_j)} |p^*(x_i, x_j) - \hat{p}(x_i, x_j)| \int \log \hat{p}(x_i, x_j) dx_i dx_j \right| \right) \\ &\quad + \left| \sum_{k \in V_F} \left( \int_{\mathcal{X}_k} p^*(x_k) \log \hat{p}(x_k) dx_k - \int_{\mathcal{X}_k} \hat{p}(x_k) \log \hat{p}(x_k) dx_k \right) \right|. \end{aligned} \quad (5.114)$$

From (5.98), we get that

$$\max_{(i,j) \in \{1, \dots, d\} \times \{1, \dots, d\}} \log |\hat{p}(x_i, x_j)| < \max\{|\log c_2|, |\log c_1|\} + 1 \quad (5.115)$$

for large enough  $n$ . The result then directly follows from (5.51) and (5.52) in Lemma 8.3.  $\square$

The proof of the main theorem follows by repeatedly applying the previous two lemmas. As in Proposition 5.2, with

$$p_{F_d}^* = \arg \min_{q_F \in \mathcal{P}_d^{(k)}} R(q_F), \quad (5.116)$$

Let  $\psi(n, d, \beta) = k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}$ . We have

$$\begin{aligned} &R(\hat{p}_{\hat{F}_d^{(k)}}) - R(p_{F_d}^*) \\ &= R(\hat{p}_{\hat{F}_d^{(k)}}) - \hat{R}(\hat{p}_{\hat{F}_d^{(k)}}) + \hat{R}(\hat{p}_{\hat{F}_d^{(k)}}) - R(p_{F_d}^*) \end{aligned} \quad (5.117)$$

$$= \hat{R}(\hat{p}_{\hat{F}_d^{(k)}}) - R(p_{F_d}^*) + O_P(\psi(n, d, \beta)) \quad (5.118)$$

$$\leq \hat{R}(\hat{p}_{F_d^{(k)}}) - R(p_{F_d}^*) + O_P(\psi(n, d, \beta)) \quad (5.119)$$

$$= R(p_{F_d}^*) - R(p_{F_d}^*) + O_P(\psi(n, d, \beta)) \quad (5.120)$$

$$= O_P \left( k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} + d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \right). \quad (5.121)$$

where (5.118) follows from Lemma 5.6, (5.119) follows from the fact that  $\hat{p}_{\hat{F}_d^{(k)}}$  is the minimizer of  $\hat{R}(\cdot)$ , and (5.120) follows from Lemma 5.5.

## 5.7.0.5 Proof of Theorem 5.2

To simplify notation, we denote

$$\phi_n(k) = k \sqrt{\frac{\log n + \log d}{n^{\beta/(\beta+1)}}} \quad (5.122)$$

$$\psi_n(d) = d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}}. \quad (5.123)$$

Following the same proof as Lemma 5.6, we obtain the following.

**Lemma 5.7.** *Under the assumptions of Theorem 5.1, we have*

$$\sup_{F \in \mathcal{F}_d^{(k)}} |R(\hat{p}_F) - \hat{R}_{n_2}(\hat{p}_F)| = O_P\left(\phi_n(k) + \psi_n(d)\right). \quad (5.124)$$

where  $\hat{R}_{n_2}$  is the held out risk.

To prove Theorem 5.2, we now have

$$R(\hat{p}_{\hat{F}_d^{(\hat{k})}}) - R(\hat{p}_{\hat{F}_d^{(k^*)}}) \quad (5.125)$$

$$= R(\hat{p}_{\hat{F}_d^{(\hat{k})}}) - \hat{R}_{n_2}(\hat{p}_{\hat{F}_d^{(\hat{k})}}) + \hat{R}_{n_2}(\hat{p}_{\hat{F}_d^{(\hat{k})}}) - R(\hat{p}_{\hat{F}_d^{(k^*)}}) \quad (5.126)$$

$$= O_P(\phi_n(\hat{k}) + \psi_n(d)) + \hat{R}_{n_2}(\hat{p}_{\hat{F}_d^{(\hat{k})}}) - R(\hat{p}_{\hat{F}_d^{(k^*)}}) \quad (5.127)$$

$$\leq O_P(\phi_n(\hat{k}) + \psi_n(d)) + \hat{R}_{n_2}(\hat{p}_{\hat{F}_d^{(k^*)}}) - R(\hat{p}_{\hat{F}_d^{(k^*)}}) \quad (5.128)$$

$$= O_P\left(\phi_n(\hat{k}) + \phi_n(k^*) + \psi_n(d)\right). \quad (5.129)$$

where (5.128) follows from the fact that  $\hat{k}$  is the minimizer of  $\hat{R}_{n_2}(\cdot)$ .

## 5.7.0.6 Proof of Theorem 5.3

Using the shorthand

$$\phi_n(k) = k \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} \quad (5.130)$$

$$\psi_n(d) = d \sqrt{\frac{\log n + \log d}{n^{2\beta/(1+2\beta)}}} \quad (5.131)$$

We have that

$$R(\hat{p}_{\hat{F}_{n_2}}) - R(\hat{p}_{F^*}) = R(\hat{p}_{\hat{F}_{n_2}}) - \hat{R}_{n_2}(\hat{p}_{\hat{F}_{n_2}}) + \hat{R}_{n_2}(\hat{p}_{\hat{F}_{n_2}}) - R(\hat{p}_{F^*}) \quad (5.132)$$

$$= O_P(\phi_n(\hat{k}) + \psi_n(d)) + \hat{R}_{n_2}(\hat{p}_{\hat{F}_{n_2}}) - R(\hat{p}_{F^*}) \quad (5.133)$$

$$\leq O_P(\phi_n(\hat{k}) + \psi_n(d)) + \hat{R}_{n_2}(\hat{p}_{F^*}) - R(\hat{p}_{F^*}) \quad (5.134)$$

$$= O_P(\phi_n(\hat{k}) + \phi_n(k^*) + \psi_n(d)) \quad (5.135)$$

$$(5.136)$$

where line 5.134 follows because  $\hat{F}_{n_2}$  is the minimizer of  $\hat{R}_{n_2}(\cdot)$ .

## 5.7.0.7 Proof of Theorem 5.4

We begin by showing an exponential probability inequality on the difference between the empirical and population mutual informations.

**Lemma 5.8.** *Under Assumptions 5.1, 5.2, there exist generic constants  $c_5$  and  $c_6$  satisfying*

$$\mathbb{P} \left( |I(X_i; X_j) - \hat{I}(X_i; X_j)| > \epsilon \right) \leq c_5 \exp \left( -c_6 n^{\frac{\beta}{1+\beta}} (\log n)^{\frac{1}{1+\beta}} \epsilon^2 \right). \quad (5.137)$$

for arbitrary  $i, j \in \{1, \dots, d\}$  with  $i \neq j$ , and  $\epsilon \rightarrow 0$  so that

$$\epsilon = \Omega \left( \sqrt{\frac{\log r_n}{nh_2^2}} \right), \quad (5.138)$$

where  $r_n = \Omega(h_2^{-1})$ .

*Proof.* For any  $\epsilon = \Omega \left( \sqrt{\frac{\log r_n}{nh_2^2}} \right)$ , we have

$$\begin{aligned} & \mathbb{P} \left( |I(X_i; X_j) - \hat{I}(X_i; X_j)| > \epsilon \right) \\ &= \mathbb{P} \left( \left| \int p^*(x_i, x_j) \log \frac{p^*(x_i, x_j)}{p^*(x_i)p^*(x_j)} dx_i dx_j \right. \right. \end{aligned} \quad (5.139)$$

$$\left. - \int \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)} dx_i dx_j \right| > \epsilon \right)$$

$$\leq \mathbb{P} \left( \left| \int (p^*(x_i, x_j) \log p^*(x_i, x_j) - \hat{p}(x_i, x_j) \log \hat{p}(x_i, x_j)) dx_i dx_j \right| > \frac{\epsilon}{2} \right)$$

$$\begin{aligned} &+ \mathbb{P} \left( \left| \int (p^*(x_i, x_j) \log p^*(x_i)p^*(x_j) \right. \right. \\ &\quad \left. \left. - \hat{p}(x_i, x_j) \log \hat{p}(x_i)\hat{p}(x_j)) dx_i dx_j \right| > \frac{\epsilon}{2} \right) \end{aligned} \quad (5.140)$$

Since the second term of (5.140) only involves univariate kernel density estimates, this term is dominated by the first term, and we only need to analyze

$$\mathbb{P} \left( \left| \int_{\mathcal{X}_i \times \mathcal{X}_j} (p^*(x_i, x_j) \log p^*(x_i, x_j) - \hat{p}(x_i, x_j) \log \hat{p}(x_i, x_j)) dx_i dx_j \right| > \frac{\epsilon}{2} \right).$$

The desired result then follows from the same analysis as in Lemma 5.5.  $\square$

Let

$$L_n = \Omega \left( \sqrt{\frac{\log n + \log d}{n^{\beta/(1+\beta)}}} \right) \quad (5.141)$$



be defined as in Assumption 5.3. To prove the main theorem, we see the event  $\widehat{F}_d^{(k)} \neq F_d^{(k)}$  implies that there must be at least exist two pairs of edges  $(i, j)$  and  $(k, \ell)$ , such that

$$\text{sign}\left(I(X_i, X_j) - I(X_k, X_\ell)\right) \neq \text{sign}\left(\widehat{I}(X_i, X_j) - \widehat{I}(X_k, X_\ell)\right). \quad (5.142)$$

Therefore, we have

$$\begin{aligned} & \mathbb{P}\left(\widehat{F}_d^{(k)} \neq F_d^{(k)}\right) \\ & \leq \mathbb{P}\left(\left(I(X_i, X_j) - I(X_k, X_\ell)\right) \cdot \left(\widehat{I}(X_i, X_j) - \widehat{I}(X_k, X_\ell)\right) \leq 0, \exists (i, j), (k, \ell)\right). \end{aligned} \quad (5.143)$$

With  $d$  nodes, there can be no more than  $d^4/2$  pairs of edges; thus, applying a union bound yields

$$\begin{aligned} & \mathbb{P}\left(\left(I(X_i, X_j) - I(X_k, X_\ell)\right) \cdot \left(\widehat{I}(X_i, X_j) - \widehat{I}(X_k, X_\ell)\right) \leq 0, \text{ for some } (i, j), (k, \ell)\right) \\ & \leq \frac{d^4}{2} \max_{((i, j), (k, \ell)) \in \mathcal{J}} \mathbb{P}\left(\left(I(X_i, X_j) - I(X_k, X_\ell)\right) \cdot \left(\widehat{I}(X_i, X_j) - \widehat{I}(X_k, X_\ell)\right) \leq 0\right). \end{aligned}$$

Assumption 5.3 specifies that

$$\min_{((i, j), (k, \ell)) \in \mathcal{J}} |I(X_i, X_j) - I(X_k, X_\ell)| > 2L_n. \quad (5.144)$$

Therefore, in order for (5.142) hold, there must exist an edge  $(i, j) \in \mathcal{J}$  such that

$$|I(X_i, X_j) - \widehat{I}(X_i, X_j)| > L_n. \quad (5.145)$$

Thus, we have

$$\begin{aligned} & \max_{((i, j), (k, \ell)) \in \mathcal{J}} \mathbb{P}\left(\left(I(X_i, X_j) - I(X_k, X_\ell)\right) \cdot \left(\widehat{I}(X_i, X_j) - \widehat{I}(X_k, X_\ell)\right) \leq 0\right) \\ & \leq \max_{i, j \in \{1, \dots, d\}, i \neq j} \mathbb{P}\left(|I(X_i, X_j) - \widehat{I}(X_i, X_j)| > L_n\right) \end{aligned} \quad (5.146)$$

$$\leq c_5 \exp\left(-c_6 n^{\frac{\beta}{1+\beta}} (\log n)^{\frac{1}{1+\beta}} L_n^2\right). \quad (5.147)$$

where (5.147) follows from Lemma 5.8.

Chaining together the above arguments, we obtain

$$\mathbb{P}\left(\widehat{F}_d^{(k)} \neq F_d^{(k)}\right) \quad (5.148)$$

$$\leq d^4 \max_{i, j \in \{1, \dots, d\}, i \neq j} \mathbb{P}\left(|I(X_i, X_j) - \widehat{I}(X_i, X_j)| > L_n\right) \quad (5.149)$$

$$\leq d^4 c_5 \exp\left(-c_6 n^{\frac{\beta}{1+\beta}} (\log n)^{\frac{1}{1+\beta}} L_n^2\right) \quad (5.150)$$

$$= o\left(c_5 \exp\left(4 \log d - c_6 (\log n)^{\frac{1}{1+\beta}} \log d\right)\right) \quad (5.151)$$

$$= o(1). \quad (5.152)$$

The conclusion of the theorem now directly follows.

## 5.7.1 Computation of the Mutual Information Matrix

In this appendix we explain different methods for computing the mutual information matrix, and making the tree estimation more efficient. One way to evaluate the empirical mutual information is to use

$$\hat{I}(X_i; X_j) = \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \log \frac{\hat{p}_{n_1}(X_i^{(s)}, X_j^{(s)})}{\hat{p}_{n_1}(X_i^{(s)}) \hat{p}_{n_1}(X_j^{(s)})}. \quad (5.153)$$

Compared with our proposed method

$$\hat{I}_{n_1}(X_i, X_j) = \frac{1}{m^2} \sum_{k=1}^m \sum_{\ell=1}^m \hat{p}_{n_1}(x_{ki}, x_{\ell j}) \log \frac{\hat{p}_{n_1}(x_{ki}, x_{\ell j})}{\hat{p}_{n_1}(x_{ki}) \hat{p}_{n_1}(x_{\ell j})}, \quad (5.154)$$

(5.153) is somewhat easier to calculate. However, if the sample size in  $\mathcal{D}_1$  is small, the approximation error can be large. A different analysis is needed to provide justification of the method based on (5.153), which would be more difficult since  $\hat{p}_{n_1}(\cdot)$  is dependent on  $\mathcal{D}_1$ . For these reasons we use the method in (5.154).

Also, note that instead of using the grid based method to evaluate the numerical integral, one could use sampling. If we can obtain  $m_1$  i.i.d. samples from the bivariate density  $\hat{p}(X_i, X_j)$ ,

$$\left\{ (X_i^{(s)}, X_j^{(s)}) \right\}_{s=1}^{m_1} \stackrel{\text{i.i.d.}}{\sim} \hat{p}_{n_1}(x_i, x_j), \quad (5.155)$$

then the empirical mutual information can be evaluated as

$$\hat{I}(X_i; X_j) = \frac{1}{m_1} \sum_{s=1}^{m_1} \log \frac{\hat{p}(X_i^{(s)}, X_j^{(s)})}{\hat{p}(X_i^{(s)}) \hat{p}(X_j^{(s)})}. \quad (5.156)$$

Compared with (5.153), the main advantage of this approach is that the estimate can be arbitrarily close to (5.25) for large enough  $m_1$  and  $m$ . Also, the computation can be easier compared to the brutal-force algorithm

Let  $\hat{p}_{n_1}(X_i, X_j)$  be the bivariate kernel density estimator on  $\mathcal{D}_1$ . To sample a point from  $\hat{p}_{n_1}(X_i, X_j)$ , we first random draw a sample  $(X_i^{(k')}, X_j^{(\ell')})$  from  $\mathcal{D}_1$ , and then sample a point  $(X, Y)$  from the bivariate distribution

$$(X, Y) \sim \frac{1}{h_2^2} K\left(\frac{X_i^{(k')} - \cdot}{h_2}\right) K\left(\frac{X_j^{(\ell')} - \cdot}{h_2}\right). \quad (5.157)$$

Though this sampling strategy is superior to the brutal-force algorithm, it requires evaluation of the bivariate kernel density estimates on many random points, which is time consuming; the grid-based method is preferred.

In our two-stage procedure, the stage requires calculation of the empirical mutual information  $\hat{I}(X_i; X_j)$  for  $\binom{d}{2}$  entries. Each requires  $O(m^2 n_1)$  work to evaluate the bivariate and univariate kernel density estimates on the  $m \times m$  grid, in a naive implementation. Therefore, the total time to calculate the empirical mutual information matrix  $M$  is  $O(m^2 n_1 d^2)$ . In the second stage, the time complexity of the Chow-Liu algorithm is dominated by that of the first step. Therefore the total time complexity is  $O(m^2 n_1 d^2)$ . The first stage requires  $O(d^2)$  space to store the matrix  $M$  and  $O(m^2 n_1)$  space to evaluate the kernel density estimates on  $\mathcal{D}_1$ . The space complexity for the Chow-Liu algorithm is  $O(d^2)$ , and thus the total space complexity is  $O(d^2 + m^2 n_1)$ .

---

**Algorithm 5.7.1** More efficient calculation of the mutual information matrix  $M$ .

---

```

1: Initialize  $M = \mathbf{0}_{d \times d}$  and  $H^{(i)} = \mathbf{0}_{n_1 \times m}$  for  $i = 1, \dots, d$ .
2: % calculate and pre-store the univariate KDE
3: for  $k = 1, \dots, d$  do
4:   for  $k' = 1, \dots, m$  do
5:      $\hat{p}(x_k^{(k')}) \leftarrow \frac{1}{n_1} \sum_{s \in \mathcal{D}_1} \frac{1}{h_1} K \left( \frac{X_k^{(s)} - x_k^{(k')}}{h_1} \right)$ 
6:   for  $k' = 1, \dots, m$  do
7:     % calculate the components used for the bivariate KDE
8:     for  $i' = 1, \dots, n_1$  do
9:       for  $i = 1, \dots, d$  do
10:         $H^{(i)}(i', k') \leftarrow \frac{1}{h_2} K \left( \frac{X_i^{i'} - x_i^{(k')}}{h_2} \right)$ 
11:     % calculate the mutual information matrix
12:     for  $\ell' = 1, \dots, m$  do
13:       for  $i = 1, \dots, d - 1$  do
14:         for  $j = i + 1, \dots, d$  do
15:            $\hat{p}(x_i^{(k')}, x_j^{(\ell')}) \leftarrow 0$ 
16:           for  $i' = 1, \dots, n_1$  do
17:              $\hat{p}(x_i^{(k')}, x_j^{(\ell')}) \leftarrow \hat{p}(x_i^{(k')}, x_j^{(\ell')}) + H^{(i)}(i', k') \cdot H^{(j)}(i', \ell')$ 
18:            $\hat{p}(x_i^{(k')}, x_j^{(\ell')}) \leftarrow \hat{p}(x_i^{(k')}, x_j^{(\ell')}) / n_1$ 
19:            $M(i, j) \leftarrow M(i, j) + \frac{1}{m^2} \hat{p}(x_i^{(k')}, x_j^{(\ell')}) \cdot \log \frac{\hat{p}(x_i^{(k')}, x_j^{(\ell')})}{\hat{p}(x_i^{(k')}) \cdot \hat{p}(x_j^{(\ell')})}$ 

```

---

The quadratic time and space complexity in the number of variables  $d$  is acceptable for many practical applications but can be prohibitive when the dimension  $d$  is large. The main bottleneck is to calculate the empirical mutual information matrix  $M$ . Due to the utilization of the kernel density estimate, the time complexity is  $O(d^2 m^2 n_1)$ . The straightforward implementation of the

brutal-form algorithm is conceptually easy but computationally inefficient, due to many redundant operations. For example, in the nested for loop, many components of the bivariate and univariate kernel density estimates are repeatedly evaluated. Here we suggest an alternative method which can significantly reduce such redundancy at the price of increased but still affordable space complexity.

The main technique used in the speed up algorithm is to change the order of the multiple nested for loops, combined with some pre-calculation. This algorithm can significantly boost the empirical performance, although the worst case time complexity remains the same. An alternative suggested by [Bach and Jordan \[2003\]](#) is to approximate the mutual information, although this would require further analysis and justification.

## Part IV

# SUPERVISED LEARNING



In this chapter, we present a new class of methods for nonparametric multi-task regression and multi-class classification called sparse additive models. Our models, named MT-SpAM, combine ideas from sparse linear modeling and additive nonparametric regression. Especially, we utilize a regularization method that enforces common sparsity patterns across different function components in a nonparametric additive model. We derive an algorithm for fitting the models that is practical and effective even when the number of predictors is larger than the sample size. The algorithms employ a coordinate descent approach that is based on a functional soft-thresholding operator. The framework yields several new models, including multi-task sparse additive models, multi-response sparse additive models, and sparse additive multi-category logistic regression. These methods have good theoretical properties and perform well on both synthetic and real data. We also present some newest insights on the sparse backfitting algorithms.

## 6.1 INTRODUCTION AND MOTIVATION

Substantial progress has been made recently on the problem of fitting high dimensional linear regression models of the form

$$Y = \beta_0 + \sum_{j=1}^d \beta_j X_j + \epsilon.$$

Here  $Y$  is a real-valued response,  $X_j$  is a predictor (or covariate) and  $\epsilon$  is a mean zero error term. Finding an estimate of  $\beta = (\beta_1, \dots, \beta_d)^T$  when  $d > n$  that is both statistically well-behaved and computationally efficient has proved challenging; however, under the assumption that the vector  $\beta$  is sparse, the lasso estimator (Tibshirani [1996]) has been remarkably successful.

Let  $(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})$  be observed data points where

$$X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T \in \mathbb{R}^d$$

be the  $d$ -dimensional covariate. The lasso estimator  $\hat{\beta}$  minimizes the  $\ell_1$ -penalized sum of squares

$$\hat{\beta} = \arg \min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^n (Y^{(i)} - \beta_0 + \sum_{j=1}^d \beta_j X_j^{(i)})^2 + \lambda \sum_{j=1}^d |\beta_j| \right\} \quad (6.1)$$

with the  $\ell_1$  penalty  $\|\beta\|_1$  encouraging sparse solutions, where many components  $\hat{\beta}_j$  are zero. The good empirical success of this estimator has been recently backed up by results confirming that it has strong theoretical properties; see [Bunea et al., 2007, Greenshtein and Ritov, 2004, Zhao and Yu, 2007, Meinshausen and Yu, 2009, Wainwright, 2006].

Though these high dimensional parametric models are much better understood now. Their finite-dimensional parametric assumptions may restrict their applications. In contrast, the nonparametric regression model  $Y^{(i)} = m(X^{(i)}) + \epsilon^{(i)}$ , where  $m$  is a general smooth function, relaxes the strong assumptions made by a linear model, but is much more challenging in high dimensions. Hastie and Tibshirani [1999] introduced the class of additive models of the form

$$Y^{(i)} = \sum_{j=1}^d f_j(X_j^{(i)}) + \epsilon^{(i)}. \quad (6.2)$$

This additive combination of univariate functions—one for each covariate  $X_j$ —is less general than joint multivariate nonparametric models, but can be more interpretable and easier to fit; in particular, an additive model can be estimated using a coordinate descent Gauss-Seidel procedure, called backfitting. Unfortunately, additive models only have good statistical and computational behavior when the number of variables  $d$  is not large relative to the sample size  $n$ , so their usefulness is limited in the high dimensional setting. For this, we investigate sparse additive models (SpAM), which extend the advantages of sparse linear models to the additive, nonparametric setting. The underlying model is the same as in (6.2), but we impose a sparsity constraint on the index set  $\{j : f_j \not\equiv 0\}$  of functions  $f_j$  that are not identically zero. Lin and Zhang [2006] have proposed COSSO, an extension of lasso to this setting, for the case where the component functions  $f_j$  belong to a reproducing kernel Hilbert space (RKHS). They penalize the sum of the RKHS norms of the component functions. Yuan [2007] proposed an extension of the non-negative garrote to this setting. As with the parametric non-negative garrote, the success of this method depends on the initial estimates of component functions  $f_j$ .

The above models have a real-valued response  $Y$ . In applications, many learning problems can be naturally formulated in terms of multi-category classification or multi-task regression. In a multi-category classification problem, it is required to discriminate between the different categories using a set of high-dimensional feature vectors—for instance, classifying the type of tumor



in a cancer patient from gene expression data. In a multi-task regression problem, it is of interest to form several regression estimators for related data sets that share common types of covariates—for instance, predicting test scores across different school districts. In other areas, such as multi-channel signal processing, it is of interest to simultaneously decompose multiple signals in terms of a large common overcomplete dictionary, which is a multi-response regression problem. In each case, while the details of the estimators vary from instance to instance, across categories, or tasks, they may share a common sparsity pattern of relevant variables selected from a high-dimensional space. How to find this common sparsity pattern is an interesting learning task.

In the parametric setting, progress has been recently made on such problems using regularization based on the sum of supremum norms [Turlach et al., 2005, Tropp et al., 2006, Zhang, 2006]. For example, let

$$\mathcal{D}_n = \left\{ (X^{(i),(k)}, Y^{(i),(k)})_{i=1}^{n_k} \right\}_{k=1}^K \quad (6.3)$$

be the observed data points for  $K$  tasks. We consider the  $K$ -task linear regression problem

$$Y^{(i),(k)} = \beta_0^{(k)} + \sum_{j=1}^d \beta_j^{(k)} X_j^{(i),(k)} + \epsilon_i^{(k)}$$

where the superscript  $k$  indexes the tasks, and the subscript  $i = 1, \dots, n_k$  indexes the instances within a task. Using quadratic loss, Zhang [2006] suggests to estimate  $\hat{\beta}$  by minimizing the following objective function

$$\sum_{k=1}^K \left[ \frac{1}{2n_k} \sum_{i=1}^{n_k} \left( Y^{(i),(k)} - \beta_0^{(k)} - \sum_{j=1}^d \beta_j^{(k)} X_j^{(i),(k)} \right)^2 \right] + \lambda \sum_{j=1}^d \max_k |\beta_j^{(k)}| \quad (6.4)$$

where  $\max_k |\beta_j^{(k)}| = \|\beta_j\|_\infty$  is the sup-norm of the vector  $\beta_j \equiv (\beta_j^{(1)}, \dots, \beta_j^{(K)})^T$  of coefficients for the  $j^{\text{th}}$  feature across different tasks. The sum of sup-norms regularization has the effect of “grouping” the elements in  $\beta_j$  such that they can be shrunk towards zero simultaneously. The problems of multi-response (or multivariate) regression and multi-category classification can be viewed as a special case of the multi-task regression problem where tasks share the same design matrix. Turlach et al. [2005] and Fornasier and Rauhut [2008] propose the same sum of sup-norms regularization as in (6.121) for such problems in the linear model setting. In related work, Zhang et al. [2008] propose the sup-norm support vector machine, demonstrating its effectiveness on gene data.

In this chapter we develop new methods for nonparametric estimation for such multi-task and multi-category regression and classification problems. Rather than fitting a linear model, we instead estimate smooth functions of the data, and formulate a regularization framework that encourages joint functional sparsity, where the component functions can be different across

tasks while sharing a common sparsity pattern. Building on a recently proposed method called sparse additive models, or “SpAM” [Ravikumar et al., 2007], we propose a convex regularization functional that can be viewed as a nonparametric analog of the sum of sup-norms regularization for linear models. Based on this regularization functional, we develop new models for nonparametric multi-task regression and classification, including multi-task sparse additive models (MT-SpAM), multi-response sparse additive models (MR-SpAM), and sparse multi-category additive logistic regression (SMALR).

The main contributions of this work include (1) an efficient iterative algorithm based on a functional soft-thresholding operator derived from sub-differential calculus, leading to the multi-task and multi-response SpAM procedures, (2) a penalized local scoring algorithm that corresponds to fitting a sequence of multi-response SpAM estimates for sparse multi-category additive logistic regression, and (3) the successful application of this methodology to multi-category tumor classification and biomarker discovery from gene microarray data. In the sequel, we first present some background materials on single-task sparse additive models due to its notational simplicity. We then show how to generalize the SpAM to multi-task settings. Thorough experimental results on both simulated and real-world datasets are also provided.

## 6.2 BACKGROUND MATERIALS ON SINGLE-TASK SPARSE ADDITIVE MODELS

In this section, we explain some backgrounds on single-task sparse additive models. The results of this section have appeared in Ravikumar et al. [2009a] (with Han Liu as a co-author) and the thesis of Pradeep Ravikumar. Since the notation and key ideas are highly related to the remaining contents, we present here for completeness. However, we do not treat the materials of this section as novel contribution of this thesis.

Our main results include the formulation of a convex optimization problem for estimating a sparse additive model, an efficient backfitting algorithm for constructing the estimator, and theoretical results that analyze the effectiveness of the estimator in the high dimensional setting. Our theoretical results are of two different types. First, we show that, under suitable choices of the design parameters, the SpAM backfitting algorithm recovers the correct sparsity pattern asymptotically; this is a property we call *sparsistency*, as a shorthand for “sparsity pattern consistency.” Second, we show that the estimator is *persistent*, in the sense of Greenshtein and Ritov [2004], which is a form of risk consistency.

6.2.0.1 *Main Ideas*

Let  $(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})$  be observed data points where

$$X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T \in \mathbb{R}^d$$

be the  $d$ -dimensional covariate. We form an additive model

$$Y^{(i)} = \alpha + \sum_{j=1}^d \beta_j g_j(X_j^{(i)}) + \epsilon^{(i)} \quad (6.5)$$

with the identifiability conditions that the component functions have mean zero and norm one:  $\int g_j(x_j) dP(x_j) = 0$  and  $\int g_j^2(x_j) dP(x_j) = 1$ . Further, we impose the sparsity condition  $\sum_{j=1}^d |\beta_j| \leq L_n$  and the smoothness condition  $g_j \in \mathcal{T}_j$  where  $\mathcal{T}_j$  is some class of smooth functions. While this problem is not convex, it makes clear the way in which sparsity is encouraged, through the  $\ell_1$  penalty  $\sum_{j=1}^d |\beta_j| \leq L_n$  which induces sparsity. Below, we derive an alternative formulation that is convex. This approach is closely related to the COSSO, introduced by [Lin and Zhang \[2006\]](#), in which a regression function  $m(x)$  is assumed to be a sparse linear combination of smooth functions. However, [Lin and Zhang \[2006\]](#) put a sparsity constraint on the second derivatives of the  $g_j$ . Our formulation of sparse additive models allows the use of general smoothing operators, not only smoothing splines. As we explain later, SpAM can also be thought of as a functional version of the grouped lasso [[Yuan and Lin, 2006](#)].

6.2.0.2 *Notation and Assumptions*

Given a general nonparametric regression model:

$$Y^{(i)} = m(X^{(i)}) + \epsilon^{(i)}, \quad (6.6)$$

we assume that  $\epsilon^{(i)} \sim N(0, \sigma^2)$  independent of  $X^{(i)}$  and

$$m(x) = \sum_{j=1}^d f_j(x_j). \quad (6.7)$$

Let  $\mu$  denote the distribution of  $X$ , and let  $\mu_j$  denote the marginal distribution of  $X_j$  for each  $j = 1, \dots, d$ . For a function  $f_j$  on  $[0, 1]$  denote its  $L_2(\mu_j)$  norm by

$$\|f_j\|_{\mu_j} = \sqrt{\int_0^1 f_j^2(x) d\mu_j(x)} = \sqrt{\mathbb{E}(f_j(X_j)^2)}. \quad (6.8)$$

When the variable  $X_j$  is clear from the context, we remove the dependence on  $\mu_j$  in the notation  $\|\cdot\|_{\mu_j}$  and simply write  $\|f_j\|$ .

For  $j \in \{1, \dots, d\}$ , let  $\mathcal{H}_j$  denote the Hilbert subspace  $L_2(\mu_j)$  of measurable functions  $f_j(x_j)$  of the single scalar variable  $x_j$  with zero mean,  $\mathbb{E}(f_j(X_j)) = 0$ . Thus,  $\mathcal{H}_j$  has the inner product

$$\langle f_j, f'_j \rangle = \mathbb{E} \left( f_j(X_j) f'_j(X_j) \right) \quad (6.9)$$

and  $\|f_j\| = \sqrt{\mathbb{E}(f_j(X_j)^2)} < \infty$ . Let  $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \dots \oplus \mathcal{H}_d$  denote the Hilbert space of functions of  $(x_1, \dots, x_d)$  that have the additive form:  $m(x) = \sum_j f_j(x_j)$ , with  $f_j \in \mathcal{H}_j, j = 1, \dots, d$ .

Let  $\{\psi_{jk}, k = 0, 1, \dots\}$  denote a uniformly bounded, orthonormal basis with respect to  $L^2[0, 1]$ . Unless stated otherwise, we assume that  $f_j \in \mathcal{T}_j$  where

$$\mathcal{T}_j = \left\{ f_j \in \mathcal{H}_j : f_j(x_j) = \sum_{k=0}^{\infty} \beta_{jk} \psi_{jk}(x_j), \quad \sum_{k=0}^{\infty} \beta_{jk}^2 k^{2\nu_j} \leq C^2 \right\} \quad (6.10)$$

for some  $0 < C < \infty$ . We shall take  $\nu_j = 2$  although the extension to other levels of smoothness is straightforward. It is also possible to adapt to  $\nu_j$  although we do not pursue that direction here.

Let  $\Lambda_{\min}(A)$  and  $\Lambda_{\max}(A)$  denote the minimum and maximum eigenvalues of a square matrix  $A$ . If  $v = (v_1, \dots, v_k)^T$  is a vector, we use the norms

$$\|v\| = \sqrt{\sum_{j=1}^k v_j^2}, \quad \|v\|_1 = \sum_{j=1}^k |v_j|, \quad \|v\|_{\infty} = \max_j |v_j|. \quad (6.11)$$

### 6.2.0.3 A Convex Formulation and the Algorithm

The outline of the derivation of our algorithm is as follows. We first formulate a population level optimization problem, and show that the minimizing functions can be obtained by iterating through a series of soft-thresholded univariate conditional expectations. We then plug in smoothed estimates of these univariate conditional expectations, to derive our sparse backfitting algorithm.

*Population SpAM.* For simplicity, assume that  $\mathbb{E}(Y^{(i)}) = 0$ . The standard additive model optimization problem in  $L_2(\mu)$  (the population setting) is

$$\min_{f_j \in \mathcal{H}_j, 1 \leq j \leq d} \mathbb{E} \left( Y - \sum_{j=1}^d f_j(X_j) \right)^2 \quad (6.12)$$

where the expectation is taken with respect to  $X$  and the noise  $\epsilon$ . Now consider the following modification of this problem that introduces a scaling parameter for each function, and that imposes additional constraints:

$$\min_{\beta \in \mathbb{R}^d, g_j \in \mathcal{H}_j} \mathbb{E} \left( Y - \sum_{j=1}^d \beta_j g_j(X_j) \right)^2 \quad (6.13)$$

$$\text{subject to: } \sum_{j=1}^d |\beta_j| \leq L, \quad (6.14)$$

$$\mathbb{E} (g_j^2) = 1, \quad j = 1, \dots, d. \quad (6.15)$$

noting that  $g_j$  is a function while  $\beta = (\beta_1, \dots, \beta_d)^T$  is a vector. The constraint that  $\beta$  lies in the  $\ell_1$ -ball  $\{\beta : \|\beta\|_1 \leq L\}$  encourages sparsity of the estimated  $\beta$ , just as for the parametric lasso [Tibshirani, 1996]. It is convenient to absorb the scaling constants  $\beta_j$  into the functions  $f_j$ , and re-express the minimization in the following equivalent Lagrangian form:

$$\mathcal{L}(f, \lambda) = \frac{1}{2} \mathbb{E} \left( Y - \sum_{j=1}^d f_j(X_j) \right)^2 + \lambda \sum_{j=1}^d \sqrt{\mathbb{E}(f_j^2(X_j))}. \quad (6.16)$$

**Theorem 6.1.** *The minimizers  $f_j \in \mathcal{H}_j$  of (6.16) satisfy*

$$f_j = \left[ 1 - \frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}} \right]_+ P_j \quad \text{a.s.} \quad (6.17)$$

where  $[\cdot]_+$  denotes the positive part, and  $P_j = \mathbb{E}[R_j | X_j]$  denotes the projection of the residual  $R_j = Y - \sum_{k \neq j} f_k(X_k)$  onto  $\mathcal{H}_j$ .

*Proof of Theorem 6.1.* Consider the minimization of the Lagrangian

$$\min_{\{f_j \in \mathcal{H}_j\}} \mathcal{L}(f, \lambda) \equiv \frac{1}{2} \mathbb{E} \left( Y - \sum_{j=1}^d f_j(X_j) \right)^2 + \lambda \sum_{j=1}^d \sqrt{\mathbb{E}(f_j(X_j)^2)} \quad (6.18)$$

with respect to  $f_j \in \mathcal{H}_j$ , holding the other components  $\{f_k, k \neq j\}$  fixed. The stationary condition is obtained by setting the Fréchet derivative to zero. Denote by  $\partial_j \mathcal{L}(f, \lambda; \eta_j)$  the directional derivative with respect to  $f_j$  in the direction  $\eta_j(X_j) \in \mathcal{H}_j$  ( $\mathbb{E}(\eta_j) = 0$ ,  $\mathbb{E}(\eta_j^2) < \infty$ ). Then the stationary condition can be formulated as

$$\partial_j \mathcal{L}(f, \lambda; \eta_j) = \frac{1}{2} \mathbb{E} [(f_j - R_j + \lambda v_j) \eta_j] = 0 \quad (6.19)$$

where  $R_j = Y - \sum_{k \neq j} f_k$  is the residual for  $f_j$ , and  $v_j \in \mathcal{H}_j$  is an element of the subgradient  $\partial \sqrt{\mathbb{E}(f_j^2)}$ , satisfying  $v_j = f_j / \sqrt{\mathbb{E}(f_j^2)}$  if  $\mathbb{E}(f_j^2) \neq 0$  and  $v_j \in \{u_j \in \mathcal{H}_j | \mathbb{E}(u_j^2) \leq 1\}$  otherwise.

Using iterated expectations, the above condition can be rewritten as

$$\mathbb{E}[(f_j + \lambda v_j - \mathbb{E}(R_j|X_j)) \eta_j] = 0. \quad (6.20)$$

But since  $f_j - \mathbb{E}(R_j|X_j) + \lambda v_j \in \mathcal{H}_j$ , we can compute the derivative in the direction  $\eta_j = f_j - \mathbb{E}(R_j|X_j) + \lambda v_j \in \mathcal{H}_j$ , implying that

$$\mathbb{E}[(f_j(x_j) - \mathbb{E}(R_j|X_j = x_j) + \lambda v_j(x_j))^2] = 0; \quad (6.21)$$

that is,

$$f_j + \lambda v_j = \mathbb{E}(R_j|X_j) \quad \text{a.s.} \quad (6.22)$$

Denote the conditional expectation  $\mathbb{E}(R_j|X_j)$ —also the projection of the residual  $R_j$  onto  $\mathcal{H}_j$ —by  $P_j$ . Now if  $\mathbb{E}(f_j^2) \neq 0$ , then  $v_j = \frac{f_j}{\sqrt{\mathbb{E}(f_j^2)}}$ , which from condition (6.22) implies

$$\sqrt{\mathbb{E}(P_j^2)} = \sqrt{\mathbb{E}\left[\left(f_j + \lambda f_j / \sqrt{\mathbb{E}(f_j^2)}\right)^2\right]} \quad (6.23)$$

$$= \left(1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}}\right) \sqrt{\mathbb{E}(f_j^2)} \quad (6.24)$$

$$= \sqrt{\mathbb{E}(f_j^2)} + \lambda \quad (6.25)$$

$$\geq \lambda. \quad (6.26)$$

If  $\mathbb{E}(f_j^2) = 0$ , then  $f_j = 0$  a.e., and  $\sqrt{\mathbb{E}(v_j^2)} \leq 1$ . Equation (6.22) then implies that

$$\sqrt{\mathbb{E}(P_j^2)} \leq \lambda. \quad (6.27)$$

We thus obtain the equivalence

$$\sqrt{\mathbb{E}(P_j^2)} \leq \lambda \iff f_j = 0 \quad \text{a.e.} \quad (6.28)$$

Rewriting equation (6.22) in light of (6.28), we obtain

$$\begin{aligned} \left(1 + \frac{\lambda}{\sqrt{\mathbb{E}(f_j^2)}}\right) f_j &= P_j && \text{if } \sqrt{\mathbb{E}(P_j^2)} > \lambda \\ f_j &= 0 && \text{otherwise.} \end{aligned}$$

Using (6.25), we thus arrive at the soft thresholding update for  $f_j$ :

$$f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}(P_j^2)}}\right]_+ P_j \quad (6.29)$$

where  $[\cdot]_+$  denotes the positive part and  $P_j = \mathbb{E}[R_j | X_j]$ .  $\square$

At the population level, the  $f_j$ 's can be found by a coordinate descent procedure that fixes  $(f_k : k \neq j)$  and fits  $f_j$  by equation (6.17), then iterates over  $j$ .

*Data version of SpAM.* To obtain a sample version of the population solution, we insert sample estimates into the population algorithm, as in standard backfitting [Hastie and Tibshirani, 1999]. Thus, we estimate the projection  $P_j = \mathbb{E}(R_j | X_j)$  by smoothing the residuals:

$$\hat{P}_j = \mathcal{S}_j R_j \quad (6.30)$$

where  $\mathcal{S}_j$  is a linear smoother, such as a local linear or kernel smoother. Let

$$\hat{s}_j = \frac{1}{\sqrt{n}} \|\hat{P}_j\| = \sqrt{\text{mean}(\hat{P}_j^2)} \quad (6.31)$$

be the estimate of  $\sqrt{\mathbb{E}(P_j^2)}$ . Using these plug-in estimates in the coordinate descent procedure yields the SpAM backfitting algorithm given in Figure 42.

This algorithm can be seen as a functional version of the coordinate descent algorithm for solving the lasso. In particular, if we solve the lasso by iteratively minimizing with respect to a single coordinate, each iteration is given by soft thresholding; see Figure 39. Convergence properties of variants of this simple algorithm have been recently treated by Daubechies et al. [2004, 2007]. Our sparse backfitting algorithm is a direct generalization of this algorithm, and it reduces to it in case where the smoothers are local linear smoothers with large bandwidths. That is, as the bandwidth approaches infinity, the local linear smoother approaches a global linear fit, yielding the estimator  $\hat{P}_j(i) = \hat{\beta}_j X_j^{(i)}$ .

When the variables are standardized,  $\hat{s}_j = \sqrt{\frac{1}{n} \sum_{i=1}^n \hat{\beta}_j^2 X_j^{(i)2}} = |\hat{\beta}_j|$  so that the soft thresholding in step (4) of the SpAM backfitting algorithm is the same as the soft thresholding in step (3) in the coordinate descent lasso algorithm.

As an alternative to estimating the conditional expectations in (6.16) by smoothing, we can define estimators by minimizing a sample version of the problem. Thus, we would minimize

$$\frac{1}{n} \sum_{i=1}^n (Y^{(i)} - \sum_{j=1}^d f_j(X_j^{(i)}))^2 \quad (6.32)$$

subject to  $f_j \in \mathcal{T}_j$ , and

$$\sum_{j=1}^d \sqrt{\frac{1}{n} \sum_{i=1}^n f_j^2(X_j^{(i)})} \leq L, \quad \frac{1}{n} \sum_{i=1}^n f_j(X_j^{(i)}) = 0, \quad j = 1, \dots, d. \quad (6.33)$$

Note that disregarding the functional constraints  $f_j \in \mathcal{T}_j$ , and optimizing only over the  $nd$  values  $f_j(X_j^{(i)})$  leads to a finite dimensional convex optimization problem.

## SPARSE BACKFITTING ALGORITHM

---

*Input:* Data  $(X^{(i)}, Y^{(i)})$ , regularization parameter  $\lambda$ .

*Initialize*  $\hat{f}_j = 0$ , for  $j = 1, \dots, d$ .

*Iterate* until convergence:

*For each*  $j = 1, \dots, d$ :

- (1) Compute the residual:  $R_j = Y - \sum_{k \neq j} \hat{f}_k(X_k)$ ;
- (2) Estimate  $P_j = \mathbb{E}[R_j | X_j]$  by smoothing:  $\hat{P}_j = \mathcal{S}_j R_j$ ;
- (3) Estimate norm:  $\hat{s}_j^2 = \frac{1}{n} \sum_{i=1}^n \hat{P}_j^2(i)$ ;
- (4) Soft-threshold:  $\hat{f}_j = [1 - \lambda/\hat{s}_j]_+ \hat{P}_j$ ;
- (5) Center:  $\hat{f}_j \leftarrow \hat{f}_j - \text{mean}(\hat{f}_j)$ .

*Output:* Component functions  $\hat{f}_j$  and estimator  $\hat{m}(X^{(i)}) = \sum_j \hat{f}_j(X_j^{(i)})$ .

---

Figure 38.: The sparse backfitting algorithm. The first two steps in the iterative algorithm are the usual backfitting procedure; the remaining steps carry out functional soft thresholding.

*Basis Functions.* It is useful to express the model in terms of basis functions. Recall that  $B_j = (\psi_{jk} : k = 1, 2, \dots)$  is an orthonormal basis for  $\mathcal{T}_j$  and that  $\sup_x |\psi_{jk}(x)| \leq B$  for some  $B$ . Then

$$f_j(x_j) = \sum_{k=1}^{\infty} \beta_{jk} \psi_{jk}(x_j) \quad (6.34)$$

where  $\beta_{jk} = \int f_j(x_j) \psi_{jk}(x_j) dx_j$ .

Let us also define

$$\tilde{f}_j(x_j) = \sum_{k=1}^q \beta_{jk} \psi_{jk}(x_j) \quad (6.35)$$

where  $q = q_n$  is a truncation parameter. For the Sobolev space  $\mathcal{T}_j$  of order two we have that  $\|f_j - \tilde{f}_j\|^2 = O(1/q^4)$ . Let  $S = \{j : f_j \neq 0\}$ . Assuming the sparsity condition  $|S| = O(1)$  it follows that  $\|m - \tilde{m}\|^2 = O(1/q^4)$  where  $\tilde{m} = \sum_j \tilde{f}_j$ . The usual choice is  $d \asymp n^{1/5}$  yielding truncation bias  $\|m - \tilde{m}\|^2 = O(n^{-4/5})$ .

In this setting, the smoother can be taken to be the least squares projection onto the truncated set of basis functions  $\{\psi_{j1}, \dots, \psi_{jq}\}$ ; this is also called orthogonal series smoothing. Let  $\Psi_j$  denote the  $n \times q_n$  matrix given by  $\Psi_j(i, \ell) =$



## COORDINATE DESCENT LASSO

---

*Input:* Data  $(X^{(i)}, Y^{(i)})$ , regularization parameter  $\lambda$ .

*Initialize*  $\hat{\beta}_j = 0$ , for  $j = 1, \dots, d$ .

*Iterate* until convergence:

*For each*  $j = 1, \dots, d$ :

        (1) Compute the residual:  $R_j = Y - \sum_{k \neq j} \hat{\beta}_k X_k$ ;

        (2) Project residual onto  $X_j$ :  $P_j = X_j^T R_j$

        (3) Soft-threshold:  $\hat{\beta}_j = [1 - \lambda/|P_j|]_+ P_j$ ;

*Output:* Estimator  $\hat{m}(X^{(i)}) = \sum_j \hat{\beta}_j X_j^{(i)}$ .

---

Figure 39.: The SpAM backfitting algorithm is a functional version of the coordinate descent algorithm for the lasso, which computes  $\hat{\beta} = \arg \min \frac{1}{2} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$ .

$\psi_{j,\ell}(X_j^{(i)})$ . The smoothing matrix is the projection matrix  $S_j = \Psi_j(\Psi_j^T \Psi_j)^{-1} \Psi_j^T$ . In this case, the backfitting algorithm in Figure 42 is a coordinate descent algorithm for minimizing

$$\frac{1}{2n} \left\| Y - \sum_{j=1}^d \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^d \sqrt{\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j}$$

which is the sample version of (6.16). This is the Lagrangian of a second-order cone program (SOCP), and standard convexity theory implies existence of a minimizer. In Section 6.2.0.5 we prove theoretical properties of SpAM assuming that this particular smoother is being used.

*Connection with the Grouped Lasso.* The SpAM model can be thought of as a functional version of the grouped lasso [Yuan and Lin, 2006] as we now explain. Consider the following linear regression model with multiple factors,

$$Y = \sum_{j=1}^d X_j \beta_j + \epsilon = X\beta + \epsilon, \quad (6.36)$$

where  $Y$  is an  $n \times 1$  response vector,  $\epsilon$  is an  $n \times 1$  vector of iid mean zero noise,  $X_j$  is an  $n \times d_j$  matrix corresponding to the  $j$ -th factor, and  $\beta_j$  is the corresponding  $d_j \times 1$  coefficient vector. Assume for convenience (in this subsection only) that each  $X_j$  is orthogonal, so that  $X_j^T X_j = I_{d_j}$ , where  $I_{d_j}$  is the

$d_j \times d_j$  identity matrix. We use  $X = (X_1, \dots, X_d)$  to denote the full design matrix and use  $\beta = (\beta_1^T, \dots, \beta_d^T)^T$  to denote the parameter.

The *grouped lasso* estimator is defined as the solution of the following convex optimization problem:

$$\hat{\beta}(\lambda_n) = \arg \min_{\beta} \|Y - X\beta\|_2^2 + \lambda_n \sum_{j=1}^d \sqrt{d_j} \|\beta_j\| \quad (6.37)$$

where  $\sqrt{d_j}$  scales the  $j$ th term to compensate for different group sizes.

It is obvious that when  $d_j = 1$  for  $j = 1, \dots, d$ , the grouped lasso becomes the standard lasso. From the KKT optimality conditions, a necessary and sufficient condition for  $\hat{\beta} = (\hat{\beta}_1^T, \dots, \hat{\beta}_d^T)^T$  to be the grouped lasso solution is

$$\begin{aligned} -X_j^T(Y - X\hat{\beta}) + \frac{\lambda \sqrt{d_j} \hat{\beta}_j}{\|\hat{\beta}_j\|} &= \mathbf{0}, \quad \forall \hat{\beta}_j \neq \mathbf{0}, \\ \|X_j^T(Y - X\hat{\beta})\| &\leq \lambda \sqrt{d_j}, \quad \forall \hat{\beta}_j = \mathbf{0}. \end{aligned} \quad (6.38)$$

Based on this stationary condition, an iterative blockwise coordinate descent algorithm can be derived; as shown by [Yuan and Lin \[2006\]](#), a solution to (6.38) satisfies

$$\hat{\beta}_j = \left[ 1 - \frac{\lambda \sqrt{d_j}}{\|S_j\|} \right]_+ S_j \quad (6.39)$$

where  $S_j = X_j^T(Y - X\beta_{\setminus j})$ , with  $\beta_{\setminus j} = (\beta_1^T, \dots, \beta_{j-1}^T, \mathbf{0}^T, \beta_{j+1}^T, \dots, \beta_d^T)$ . By iteratively applying (6.39), the grouped lasso solution can be obtained.

As discussed in the introduction, the COSSO model of [Lin and Zhang \[2006\]](#) replaces the lasso constraint on  $\sum_j |\beta_j|$  with a RKHS constraint. The advantage of our formulation is that it decouples smoothness ( $g_j \in \mathcal{T}_j$ ) and sparsity ( $\sum_j |\beta_j| \leq L$ ). This leads to a simple algorithm that can be carried out with any nonparametric smoother and scales easily to high dimensions.

#### 6.2.0.4 Choosing the Regularization Parameter

We choose  $\lambda$  by minimizing an estimate of the risk. Let  $\nu_j$  be the effective degrees of freedom for the smoother on the  $j$ th variable, that is,  $\nu_j = \text{trace}(\mathcal{S}_j)$  where  $\mathcal{S}_j$  is the smoothing matrix for the  $j$ -th dimension. Also let  $\hat{\sigma}^2$  be an estimate of the variance. Define the total effective degrees of freedom as

$$\text{df}(\lambda) = \sum_j \nu_j I(\|\hat{f}_j\| \neq 0). \quad (6.40)$$

Two estimates of risk are

$$C_p = \frac{1}{n} \sum_{i=1}^n \left( Y^{(i)} - \sum_{j=1}^d \hat{f}_j(X_j) \right)^2 + \frac{2\hat{\sigma}^2}{n} \text{df}(\lambda) \quad (6.41)$$

and

$$\text{GCV}(\lambda) = \frac{\frac{1}{n} \sum_{i=1}^n (Y^{(i)} - \sum_j \hat{f}_j(X_j^{(i)}))^2}{(1 - \text{df}(\lambda)/n)^2}. \quad (6.42)$$

The first is  $C_p$  and the second is generalized cross validation but with degrees of freedom defined by  $\text{df}(\lambda)$ . A proof that these are valid estimates of risk is not currently available; thus, these should be regarded as heuristics.

Based on the results in Wasserman and Roeder [2009] about the lasso, it seems likely that choosing  $\lambda$  by risk estimation can lead to overfitting. One can further clean the estimate by testing  $H_0 : f_j = 0$  for all  $j$  such that  $\hat{f}_j \neq 0$ . For example, the tests in Fan and Jiang [2005] could be used.

#### 6.2.0.5 Sparsistency

In the case of linear regression, with  $f_j(X_j) = \beta_j^{*T} X_j$ , several authors have shown that, under certain conditions on  $n, d$ , the number of relevant variables  $s = |\text{supp}(\beta^*)|$ , and the design matrix  $X$ , the lasso recovers the sparsity pattern asymptotically; that is, the lasso estimator  $\hat{\beta}_n$  is *sparsistent*:

$$\mathbb{P} \left( \text{supp}(\beta^*) = \text{supp}(\hat{\beta}_n) \right) \rightarrow 1. \quad (6.43)$$

Here,  $\text{supp}(\beta) = \{j : \beta_j \neq 0\}$ . References include Wainwright [2006], Meinshausen and Bühlmann [2006], Zou [2005], Fan and Li [2001], and Zhao and Yu [2007]. We show a similar result for sparse additive models under orthogonal function regression.

In terms of an orthogonal basis  $\psi$ , we can write

$$Y^{(i)} = \sum_{j=1}^d \sum_{k=1}^{\infty} \beta_{jk}^* \psi_{jk}(X_j^{(i)}) + \epsilon^{(i)}. \quad (6.44)$$

To simplify notation, let  $\beta_j$  be the  $q_n$  dimensional vector  $\{\beta_{jk}, k = 1, \dots, q_n\}$  and let  $\Psi_j$  be the  $n \times q_n$  matrix  $\Psi_j[i, k] = \psi_{jk}(X_j^{(i)})$ . If  $A \subset \{1, \dots, d\}$ , we denote by  $\Psi_A$  the  $n \times q|A|$  matrix where for each  $j \in A$ ,  $\Psi_j$  appears as a submatrix in the natural way.

We now analyze the sparse backfitting algorithm of Figure 42 assuming an orthogonal series smoother is used to estimate the conditional expectation in its Step (2). As noted earlier, an orthogonal series smoother for a predictor  $X_j$  is the least squares projection onto a truncated set of basis functions  $\{\psi_{j1}, \dots, \psi_{jq}\}$ . Our optimization problem in this setting is

$$\min_{\beta} \frac{1}{2n} \left\| Y - \sum_{j=1}^d \Psi_j \beta_j \right\|_2^2 + \lambda \sum_{j=1}^d \sqrt{\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j}. \quad (6.45)$$

Combined with the soft-thresholding step, the update for  $f_j$  in algorithm of Figure 42 can thus be seen to solve the following problem,

$$\min_{\beta} \frac{1}{2n} \|R_j - \Psi_j \beta_j\|_2^2 + \lambda_n \sqrt{\frac{1}{n} \beta_j^T \Psi_j^T \Psi_j \beta_j}$$

where  $\|v\|_2^2$  denotes  $\sum_{i=1}^n v_i^2$  and  $R_j = Y - \sum_{l \neq j} \Psi_l \beta_l$  is the residual for  $f_j$ . The sparse backfitting algorithm thus solves

$$\begin{aligned} & \min_{\beta} \{ \mathcal{Q}_n(\beta) + \lambda_n \Omega(\beta) \} \\ & = \min_{\beta} \frac{1}{2n} \left\| Y - \sum_{j=1}^d \Psi_j \beta_j \right\|_2^2 + \lambda_n \sum_{j=1}^d \left\| \frac{1}{\sqrt{n}} \Psi_j \beta_j \right\|_2 \end{aligned} \quad (6.46)$$

where  $\mathcal{Q}_n$  denotes the squared error term and  $\Omega$  denotes the regularization term, and each  $\beta_j$  is a  $q_n$ -dimensional vector. Let  $S$  denote the true set of variables  $\{j : f_j \neq 0\}$ , with  $s = |S|$ , and let  $S^c$  denote its complement. Let  $\hat{S}_n = \{j : \hat{\beta}_j \neq 0\}$  denote the estimated set of variables from the minimizer  $\hat{\beta}_n$ , with corresponding function estimates  $\hat{f}_j(x_j) = \sum_{k=1}^{q_n} \hat{\beta}_{jk} \psi_{jk}(x_j)$ . For the results in this section, we will treat the covariates as fixed. A preliminary version of the following result is stated here without proof, for details, see in [Ravikumar et al. \[2007\]](#).

**Theorem 6.2.** *Suppose that the following conditions hold on the design matrix  $X$  in the orthogonal basis  $\psi$ :*

$$\Lambda_{\max} \left( \frac{1}{n} \Psi_S^T \Psi_S \right) \leq C_{\max} < \infty \quad (6.47)$$

$$\Lambda_{\min} \left( \frac{1}{n} \Psi_S^T \Psi_S \right) \geq C_{\min} > 0 \quad (6.48)$$

$$\max_{j \in S^c} \left\| \left( \frac{1}{n} \Psi_j^T \Psi_S \right) \left( \frac{1}{n} \Psi_S^T \Psi_S \right)^{-1} \right\| \leq \sqrt{\frac{C_{\min}}{C_{\max}}} \frac{1 - \delta}{\sqrt{s}}, \text{ for some } 0 < \delta \leq 1. \quad (6.49)$$

Assume that the truncation dimension  $q_n$  satisfies  $q_n \rightarrow \infty$  and  $q_n = o(n)$ . Furthermore, suppose the following conditions, which relate the regularization parameter  $\lambda_n$  to the design parameters  $n, p$ , the number of relevant variables  $s$ , and the truncation size  $q_n$ :

$$\frac{s}{q_n \lambda_n} \rightarrow 0 \quad (6.50)$$

$$\frac{q_n \log(q_n(d-s))}{n \lambda_n^2} \rightarrow 0 \quad (6.51)$$

$$\frac{1}{\rho_n^*} \left( \sqrt{\frac{\log(sq_n)}{n}} + \frac{s^{3/2}}{q_n} + \lambda_n \sqrt{s q_n} \right) \rightarrow 0 \quad (6.52)$$

where  $\rho_n^* = \min_{j \in S} \|\beta_j^*\|_\infty$ . Then the solution  $\hat{\beta}_n$  to (6.45) is unique and satisfies  $\hat{S}_n = S$  with probability approaching one.

This result parallels the theorem of Wainwright [2006] on model selection consistency of the lasso; however, technical subtleties arise because of the truncation dimension  $q_n$  which is increasing with sample size, and the matrix  $\Psi_j^T \Psi$  which appears in the regularization of  $\beta_j$ . As a result, the operator norm rather than the sup-norm appears in the incoherence condition (6.49). Note, however, that condition (6.49) implies that

$$\left\| \Psi_{S^c}^T \Psi_S \left( \Psi_S^T \Psi_S \right)^{-1} \right\|_\infty = \max_{j \in S^c} \left\| \Psi_j^T \Psi_S \left( \Psi_S^T \Psi_S \right)^{-1} \right\|_\infty \quad (6.53)$$

$$\leq \sqrt{\frac{C_{\min} q_n}{C_{\max}}} (1 - \delta) \quad (6.54)$$

since  $\frac{1}{\sqrt{n}} \|A\|_\infty \leq \|A\| \leq \sqrt{m} \|A\|_\infty$  for an  $m \times n$  matrix  $A$ . This relates it to the more standard incoherence conditions that have been used for sparsistency in the case of the lasso.

The following corollary, which imposes the additional condition that the number of relevant variables is bounded, follows directly. It makes explicit how to choose the design parameters  $q_n$  and  $\lambda_n$ , and implies a condition on the fastest rate at which the minimum norm  $\rho_n^*$  can approach zero.

**Corollary 6.1.** *Suppose that  $s = O(1)$ , and assume the design conditions (6.47), (6.48) and (6.49) hold. If the truncation dimension  $q_n$ , regularization parameter  $\lambda_n$ , and minimum norm  $\rho_n^*$  satisfy*

$$q_n \asymp n^{1/3} \quad (6.55)$$

$$\lambda_n \asymp \frac{\log nd}{n^{1/3}} \quad (6.56)$$

$$\frac{1}{\rho_n^*} = o\left(\frac{n^{1/6}}{\log nd}\right) \quad (6.57)$$

then  $\mathbb{P}(\hat{S}_n = S) \rightarrow 1$ .

The following proposition clarifies the implications of condition (6.57), by relating the sup-norm  $\|\beta_j\|_\infty$  to the function norm  $\|f_j\|_2$ .

**Proposition 6.1.** *Suppose that  $f(x) = \sum_k \beta_k \psi_k(x)$  is in the Sobolev space of order  $\nu > 1/2$ , so that  $\sum_{i=1}^\infty \beta_i^2 i^{2\nu} \leq C^2$  for some constant  $C$ . Then*

$$\|f\|_2 = \|\beta\|_2 \leq c \|\beta\|_\infty^{\frac{2\nu}{2\nu+1}} \quad (6.58)$$

for some constant  $c$ .

For instance, the result of Corollary 6.1 allows the norms of the coefficients  $\beta_j$  to decrease as  $\|\beta_j\|_\infty = \log^2(nd)/n^{1/6}$ . In the case  $\nu = 2$ , this would allow the norms  $\|f_j\|_2$  of the relevant functions to approach zero at the rate  $\log^{8/5}(nd)/n^{2/15}$ .

## 6.2.0.6 Persistence

The previous assumptions are very strong. They can be weakened at the expense of getting weaker results. In particular, in the section we do not assume that the true regression function is additive. We use arguments like those in Juditsky and Nemirovski [2000] and Greenshtein and Ritov [2004] in the context of linear models. In this section we treat  $X$  as random and we use triangular array asymptotics, that is, the joint distribution for the data can change with  $n$ . Let  $(X, Y)$  denote a new pair (independent of the observed data) and define the predictive risk when predicting  $Y$  with  $v(X)$  by

$$R(v) = \mathbb{E}(Y - v(X))^2. \quad (6.59)$$

When  $v(x) = \sum_j \beta_j g_j(x_j)$  we also write the risk as  $R(\beta, g)$  where  $\beta = (\beta_1, \dots, \beta_d)$  and  $g = (g_1, \dots, g_d)$ . Following Greenshtein and Ritov [2004] we say that an estimator  $\hat{m}_n$  is persistent (risk consistent) relative to a class of functions  $\mathcal{M}_n$ , if

$$R(\hat{m}_n) - R(m_n^*) \xrightarrow{P} 0 \quad (6.60)$$

where

$$m_n^* = \arg \min_{v \in \mathcal{M}_n} R(v) \quad (6.61)$$

is the predictive oracle. Greenshtein and Ritov [2004] show that the lasso is persistent for  $\mathcal{M}_n = \{\ell(x) = x^T \beta : \|\beta\|_1 \leq L_n\}$  and  $L_n = o((n/\log n)^{1/4})$ . Note that  $m_n^*$  is the best linear approximation (in prediction risk) in  $\mathcal{M}_n$  but the true regression function is not assumed to be linear. Here we show a similar result for SpAM.

In this section, we assume that the SpAM estimator  $\hat{m}_n$  is chosen to minimize

$$\frac{1}{n} \sum_{i=1}^n (Y^{(i)} - \sum_j \beta_j g_j(X_j^{(i)}))^2 \quad (6.62)$$

subject to  $\|\beta\|_1 \leq L_n$  and  $g_j \in \mathcal{T}_j$ . We make no assumptions about the design matrix. Let  $\mathcal{M}_n \equiv \mathcal{M}_n(L_n)$  be defined by

$$\mathcal{M}_n = \left\{ m : m(x) = \sum_{j=1}^d \beta_j g_j(x_j) : \mathbb{E}(g_j) = 0, \mathbb{E}(g_j^2) = 1, \sum_j |\beta_j| \leq L_n \right\}$$

and let  $m_n^* = \arg \min_{v \in \mathcal{M}_n} R(v)$ .

**Theorem 6.3.** Suppose that  $d \leq e^{n^\xi}$  for some  $\xi < 1$ . Then,

$$R(\hat{m}_n) - R(m_n^*) = O_P \left( \frac{L_n^2}{n^{(1-\xi)/2}} \right) \quad (6.63)$$

and hence, if  $L_n = o(n^{(1-\xi)/4})$  then SpAM is persistent.

### 6.3 MUTI-TASK SPARSE ADDITIVE MODELS

We now present multi-task extension of the sparse additive models. We start by recalling some notation. If  $X$  has distribution  $\mu_X$ , and  $f$  is a function of  $x$ , its  $L_2(\mu_X)$  norm is denoted by

$$\|f\|^2 = \int_{\mathcal{X}} f^2(x) d\mu_X = \mathbb{E}(f^2).$$

If  $v = (v_1, \dots, v_n)^T$  is a vector, define

$$\|v\|_n^2 = \frac{1}{n} \sum_{j=1}^n v_j^2 \text{ and } \|v\|_\infty = \max_j |v_j|.$$

For a  $d$ -dimensional random vector  $(X_1, \dots, X_d)$ , let  $\mathcal{H}_j$  denote the Hilbert subspace  $L_2(\mu_{X_j})$  of  $\mu_{X_j}$ -measurable functions  $f_j(x_j)$  of the single scalar variable  $X_j$  with zero mean, i.e.  $\mathbb{E}[f_j(X_j)] = 0$ . The inner product on this space is defined as  $\langle f_j, g_j \rangle = \mathbb{E}[f_j(X_j)g_j(X_j)]$ . In this paper, we mainly study multivariate functions  $f(x_1, \dots, x_p)$  that have an additive form, i.e.,  $f(x_1, \dots, x_p) = \alpha + \sum_j f_j(x_j)$ , with  $f_j \in \mathcal{H}_j$  for  $j = 1, \dots, d$ . With

$$\mathcal{H} \equiv \{1\} \oplus \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus \dots \oplus \mathcal{H}_d$$

denoting the direct sum Hilbert space, we have that  $f \in \mathcal{H}$ .

#### 6.3.1 Multi-task/Multi-response Sparse Additive Regression

In a  $K$ -task regression problem, we have observations

$$\{(X^{(i),(k)}, Y^{(i),(k)}), i = 1, \dots, n_k, k = 1, \dots, K\},$$

where  $X^{(i),(k)} = (X_1^{(i),(k)}, \dots, X_d^{(i),(k)})^T$  is a  $d$ -dimensional covariate vector, the superscript  $k$  indexes tasks and  $i$  indexes the i.i.d. samples for each task. In the following, for notational simplicity, we assume that  $n_1 = \dots = n_K = n$ . We also assume different tasks are comparable and each  $Y^{(k)}$  and  $X_j^{(k)}$  has been standardized, i.e., has mean zero and variance one. This is not really a restriction of the model since a straightforward weighting scheme can be adopted to extend our approach to handle noncomparable tasks. We assume the true model is

$$\mathbb{E}(Y^{(k)} | X^{(k)} = x^{(k)}) = f^{(k)}(x^{(k)}) \equiv \sum_{j=1}^d f_j^{(k)}(x_j^{(k)})$$

for  $k = 1, \dots, K$ , where, for simplicity, we take all intercepts  $\alpha^{(k)}$  to be zero. Let  $\mathcal{Q}_{f^{(k)}}(x, y) = (y - f^{(k)}(x))^2$  denote the quadratic loss. To encourage com-

mon sparsity patterns across different function components, we define the regularization functional  $\Phi_K(f)$  by

$$\Phi_K(f) = \sum_{j=1}^d \max_{k=1,\dots,K} \|f_j^{(k)}\|. \quad (6.64)$$

The regularization functional  $\Phi_K(f)$  naturally combines the idea of the sum of sup-norms penalty for parametric joint sparsity and the regularization idea of SpAM for nonparametric functional sparsity; if  $K = 1$ , then  $\Phi_1(f)$  is just the regularization term introduced for (single-task) sparse additive models by Ravikumar et al. [2009a]. If each  $f_j^{(k)}$  is a linear function, then  $\Phi_K(f)$  reduces to the sum of sup-norms regularization term as in (6.121). We shall employ  $\Phi_K(f)$  to induce joint functional sparsity in nonparametric multi-task inference.

Using this regularization functional, the multi-task sparse additive model (MT-SpAM) is formulated as a penalized M-estimator, by framing the following optimization problem

$$\hat{f}^{(1)}, \dots, \hat{f}^{(K)} = \arg \min_{f^{(1)}, \dots, f^{(K)}} \left\{ \frac{1}{2n} \sum_{i=1}^n \sum_{k=1}^K \mathcal{Q}_{f^{(k)}}(X^{(i),(k)}, Y^{(i),(k)}) + \lambda \Phi_K(f) \right\} \quad (6.65)$$

where  $f_j^{(k)} \in \mathcal{H}_j^{(k)}$  for  $j = 1, \dots, p$  and  $k = 1, \dots, K$ , and  $\lambda > 0$  is a regularization parameter. The multi-response sparse additive model (MR-SpAM) has exactly the same formulation as in (6.65) except that a common design matrix is used across the  $K$  different tasks.

### 6.3.2 Sparse Multi-Category Additive Logistic Regression

In a  $K$ -category classification problem, we are given

$$\{(X^{(i),(k)}, Y^{(i),(k)}), i = 1, \dots, n_k, k = 1, \dots, K\},$$

where  $X^{(i),(k)} = (X_1^{(i),(k)}, \dots, X_d^{(i),(k)})^T$  is a  $d$ -dimensional predictor vector and

$$Y^{(i)} = (Y^{(i),(1)}, \dots, Y^{(i),(K-1)})^T$$

is a  $(K-1)$ -dimensional response vector in which at most one element can be one, with all the others being zero. Here, we adopt the common “1-of- $K$ ” labeling convention where  $Y^{(i),(k)} = 1$  if  $X^{(i)}$  has category  $k$  and  $Y^{(i),(k)} = 0$  otherwise; if all elements of  $Y^{(i)}$  are zero, then  $X^{(i)}$  is assigned the  $K$ -th category.

The multi-category additive logistic regression model is

$$\mathbb{P}(Y^{(k)} = 1 | X = x) = \frac{\exp(f^{(k)}(x))}{1 + \sum_{k'=1}^{K-1} \exp(f^{(k')}(x))}, \quad k = 1, \dots, K-1 \quad (6.66)$$



where

$$f^{(k)}(x) = \alpha^{(k)} + \sum_{j=1}^d f_j^{(k)}(x_j)$$

has an additive form. We define  $f = (f^{(1)}, \dots, f^{(K-1)})$  to be a discriminant function and

$$p_f^{(k)}(x) = \mathbb{P}(Y^{(k)} = 1 \mid X = x)$$

to be the conditional probability of category  $k$  given  $X = x$ . The logistic regression classifier  $h_f(\cdot)$  induced by  $f$ , which is a mapping from the sample space to the category labels, is simply given by

$$h_f(x) = \arg \max_{k=1, \dots, K} p_f^{(k)}(x).$$

If a variable  $X_j$  is irrelevant, then all of the component functions  $f_j^{(k)}$  are identically zero, for each  $k = 1, 2, \dots, K-1$ . This motivates the use of the regularization functional  $\Phi_{K-1}(f)$  to zero out entire vectors  $f_j = (f_j^{(1)}, \dots, f_j^{(K-1)})$ .

Denoting

$$\ell_f(x, y) = \sum_{k=1}^{K-1} y^{(k)} f^{(k)}(x) - \log \left( 1 + \sum_{k'=1}^{K-1} \exp f^{(k')}(x) \right)$$

as the multinomial log-loss, the sparse multi-category additive logistic regression estimator (SMALR) is thus formulated as the solution to the optimization problem

$$\widehat{f}^{(1)}, \dots, \widehat{f}^{(K-1)} = \arg \min_{f^{(1)}, \dots, f^{(K-1)}} \left\{ -\frac{1}{n} \sum_{i=1}^n \ell_f(X^{(i)}, Y^{(i)}) + \lambda \Phi_{K-1}(f) \right\} \quad (6.67)$$

where  $f_j^{(k)} \in \mathcal{H}_j^{(k)}$  for  $j = 1, \dots, d$  and  $k = 1, \dots, K-1$ .

### 6.3.3 Simultaneous Sparse Backfitting

We use a blockwise coordinate descent algorithm to minimize the functional defined in (6.65). We first formulate the population version of the problem by replacing sample averages by their expectations. We then derive stationary conditions for the optimum and obtain a population version algorithm for computing the solution by a series of soft-thresholded univariate conditional expectations. Finally, a finite sample version of the algorithm can be derived by plugging in nonparametric smoothers for these conditional expectations.

For the  $j^{\text{th}}$  block of component functions  $f_j^{(1)}, \dots, f_j^{(K)}$ , let  $R_j^{(k)} = Y^{(k)} - \sum_{l \neq j} f_l^{(k)}(X_l^{(k)})$  denote the partial residuals. Assuming all but the functions in the  $j^{\text{th}}$  block to be fixed, the optimization problem is reduced to

$$\begin{aligned} & \hat{f}_j^{(1)}, \dots, \hat{f}_j^{(K)} \\ &= \arg \min_{f_j^{(1)}, \dots, f_j^{(K)}} \left\{ \frac{1}{2} \mathbb{E} \left[ \sum_{k=1}^K \left( R_j^{(k)} - f_j^{(k)}(X_j^{(k)}) \right)^2 \right] + \lambda \max_{k=1, \dots, K} \|f_j^{(k)}\| \right\}. \end{aligned} \quad (6.68)$$

The following result characterizes the solution to (6.68).

**Theorem 6.4.** Let  $P_j^{(k)} = \mathbb{E} \left( R_j^{(k)} \mid X_j^{(k)} \right)$  and  $s_j^{(k)} = \|P_j^{(k)}\|$ , and order the indices according to  $s_j^{(k_1)} \geq s_j^{(k_2)} \geq \dots \geq s_j^{(k_K)}$ . Then the solution to (6.68) is given by

$$f_j^{(k_i)} = \begin{cases} P_j^{(k_i)} & \text{for } i > m^* \\ \frac{1}{m^*} \left[ \sum_{i'=1}^{m^*} s_j^{(k_{i'})} - \lambda \right]_+ \frac{P_j^{(k_i)}}{s_j^{(k_i)}} & \text{for } i \leq m^*. \end{cases} \quad (6.69)$$

where  $m^* = \arg \max_m \frac{1}{m} \left( \sum_{i'=1}^m s_j^{(k_{i'})} - \lambda \right)$  and  $[\cdot]_+$  denotes the positive part.

Therefore, the optimization problem in (6.68) is solved by a soft-thresholding operator, given in equation (6.69), which we shall denote as

$$(f_j^{(1)}, \dots, f_j^{(K)}) = \text{Soft}_\lambda^{(\infty)}[R_j^{(1)}, \dots, R_j^{(K)}]. \quad (6.70)$$

While the proof of this result is lengthy, we sketch the key steps below, which are a functional extension of the subdifferential calculus approach of Fornasier and Rauhut [2008] in the linear setting. First, we formulate an optimality condition in terms of the Gâteaux derivative as follows.

**Lemma 6.1.** The functions  $f_j^{(k)}$  are solutions to (6.68) if and only if  $f_j^{(k)} - P_j^{(k)} + \lambda u_k v_k = 0$  (almost surely), for  $k = 1, \dots, K$ , where  $u_k$  are scalars and  $v_k$  are measurable functions of  $X_j^{(k)}$ , with

$$(u_1, \dots, u_K)^T \in \partial \|\cdot\|_\infty \Big|_{\left( \|f_j^{(1)}\|, \dots, \|f_j^{(K)}\| \right)^T} \text{ and } v_k \in \partial \|f_j^{(k)}\|, \quad k = 1, \dots, K.$$

Here the former one denotes the subdifferential of the convex functional  $\|\cdot\|_\infty$  evaluated at  $(\|f_j^{(1)}\|, \dots, \|f_j^{(K)}\|)^T$ , it lies in a  $K$ -dimensional Euclidean space. And the latter denotes the subdifferential of  $\|f_j^{(k)}\|$ , which is a set of functions. Next, the following proposition from Rockafellar and Wets [1998] is used to characterize the subdifferential of sup-norms.

**Lemma 6.2.** *The subdifferential of  $\|\cdot\|_\infty$  on  $\mathbb{R}^K$  is*

$$\partial\|\cdot\|_\infty|_x = \begin{cases} B^1(1) & \text{if } x = \mathbf{0} \\ \text{conv}\{\text{sign}(x_k)e_k : |x_k| = \|x\|_\infty\} & \text{otherwise.} \end{cases}$$

where  $B^1(1)$  denotes the  $\ell_1$  ball of radius one,  $\text{conv}(A)$  denotes the convex hull of set  $A$ , and  $e_k$  is the  $k$ -th canonical unit vector in  $\mathbb{R}^K$ .

Using Lemma 6.1 and Lemma 6.2, the proof of Theorem 6.4 proceeds by considering three cases for the sup-norm subdifferential evaluated at  $(\|f_j^{(1)}\|, \dots, \|f_j^{(K)}\|)^T$ :

- (1)  $\|f_j^{(k)}\| = 0$  for  $k = 1, \dots, K$ ;
- (2) there exists a unique  $k$ , such that  $\|f_j^{(k)}\| = \max_{k'=1, \dots, K} \|f_j^{(k')}\| \neq 0$ ;
- (3) there exists at least two  $k \neq k'$ , such that

$$\|f_j^{(k)}\| = \|f_j^{(k')}\| = \max_{m=1, \dots, K} \|f_j^{(m)}\| \neq 0.$$

The derivations for cases (1) and (2) are relatively straightforward, but for case (3) we prove the following.

**Lemma 6.3.** *The sup-norm is attained precisely at  $m > 1$  entries if only if  $m$  is the largest number such that*

$$s_j^{(k_m)} \geq \frac{1}{m-1} \left( \sum_{i'=1}^{m-1} s_j^{(k_{i'})} - \lambda \right).$$

The proof of Theorem 6.4 then follows from the above lemmas and some calculus. Based on this result, the data version of the soft-thresholding operator is obtained by replacing the conditional expectation  $P_j^{(k)} = \mathbb{E}(R_j^{(k)} | X_j^{(k)})$  by  $\mathcal{S}_j^{(k)} R_j^{(k)}$ , where  $\mathcal{S}_j^{(k)}$  is a nonparametric smoother for variable  $X_j^{(k)}$ , e.g., a local linear or spline smoother; see Figure 40. The resulting simultaneous sparse backfitting algorithm for multi-task and multi-response sparse additive models (MT-SpAM and MR-SpAM) is shown in Figure 41. The algorithm for the multi-response case (MR-SpAM) has  $\mathcal{S}_j^{(1)} = \dots = \mathcal{S}_j^{(K)}$  since there is only a common design matrix.

#### 6.3.4 Penalized Local Scoring Algorithm for SMALR

We now derive a penalized local scoring algorithm for sparse multi-category additive logistic regression (SMALR), which can be viewed as a variant of

SOFT-THRESHOLDING OPERATOR  $\text{SOFT}_\lambda^{(\infty)}[R_j^{(1)}, \dots, R_j^{(K)}; \mathcal{S}_j^{(1)}, \dots, \mathcal{S}_j^{(K)}]$ : DATA VERSION

---

*Input:* Smoothing matrices  $\mathcal{S}_j^{(k)}$ , residuals  $R_j^{(k)}$  for  $k = 1, \dots, K$ , regularization parameter  $\lambda$ .

(1) Estimate  $P_j^{(k)} = \mathbb{E}[R_j^{(k)} | X_j^{(k)}]$  by smoothing:  $\hat{P}_j^{(k)} = \mathcal{S}_j^{(k)} R_j^{(k)}$ ;

(2) Estimate norm:  $\hat{s}_j^{(k)} = \|\hat{P}_j^{(k)}\|_n$  and order the indices according to

$$\hat{s}_j^{(k_1)} \geq \hat{s}_j^{(k_2)} \geq \dots \geq \hat{s}_j^{(k_K)};$$

(3) Find  $m^* = \arg \max_m \frac{1}{m} \left( \sum_{i'=1}^m \hat{s}_j^{(k_{i'})} - \lambda \right)$  and calculate

$$\hat{f}_j^{(k_i)} = \begin{cases} \hat{P}_j^{(k_i)} & \text{for } i > m^* \\ \frac{1}{m^*} \left[ \sum_{i'=1}^{m^*} \hat{s}_j^{(k_{i'})} - \lambda \right]_+ \frac{\hat{P}_j^{(k_i)}}{\hat{s}_j^{(k_i)}} & \text{for } i \leq m^*; \end{cases}$$

(4) Center  $\hat{f}_j^{(k)} \leftarrow \hat{f}_j^{(k)} - \text{mean}(\hat{f}_j^{(k)})$  for  $k = 1, \dots, K$ .

*Output:* Functions  $\hat{f}_j^{(k)}$  for  $k = 1, \dots, K$ .

---

Figure 40.: Data version of the soft-thresholding operator.

Newton's method in function space. At each iteration, a quadratic approximation to the loss is used as a surrogate functional with the regularization term added to induce joint functional sparsity. However, a technical difficulty is that the approximate quadratic problem in each iteration is weighted by a non-diagonal matrix in function space, thus a trivial extension of the algorithm in [Ravikumar et al., 2007] for sparse binary nonparametric logistic regression does not apply. To tackle this problem, we use an auxiliary function to lower bound the log-loss, as in [Krishnapuram et al., 2005].

The population version of the log-loss is  $L(f) = \mathbb{E}[\ell_f(X, Y)]$  with

$$f = (f^{(1)}, \dots, f^{(K-1)}).$$

A second-order Lagrange form Taylor expansion to  $L(f)$  at  $\hat{f}$  is then

$$L(f) = L(\hat{f}) + \mathbb{E} \left[ \nabla L(\hat{f})^T (f - \hat{f}) \right] + \frac{1}{2} \mathbb{E} \left[ (f - \hat{f})^T H(\tilde{f}) (f - \hat{f}) \right] \quad (6.71)$$

for some function  $\tilde{f}$ , where the gradient is  $\nabla L(\hat{f}) = Y - p_{\hat{f}}(X)$  with

$$p_{\hat{f}}(X) = (p_{\hat{f}}(Y^{(1)} = 1 | X), \dots, p_{\hat{f}}(Y^{(K-1)} = 1 | X))^T,$$

## MULTI-TASK AND MULTI-RESPONSE SPAM

---

*Input:* Data  $(X^{(i),(k)}, Y^{(i),(k)}), i = 1, \dots, n, k = 1, \dots, K$  and regularization parameter  $\lambda$ .

*Initialize:* Set  $\hat{f}_j^{(k)} = 0$  and compute smoothers  $\mathcal{S}_j^{(k)}$  for  $j = 1, \dots, p$  and  $k = 1, \dots, K$ ;

*Iterate* until convergence:

*For each*  $j = 1, \dots, d$ :

(1) Compute residuals:  $R_j^{(k)} = Y^{(k)} - \sum_{k' \neq j} \hat{f}_{k'}^{(k)}$  for  $k = 1, \dots, K$ ;

(2) Threshold:  $\hat{f}_j^{(1)}, \dots, \hat{f}_j^{(K)} \leftarrow \text{Soft}_{\lambda}^{(\infty)}[R_j^{(1)}, \dots, R_j^{(K)}; \mathcal{S}_j^{(1)}, \dots, \mathcal{S}_j^{(K)}]$ .

*Output:* Functions  $\hat{f}^{(k)}$  for  $k = 1, \dots, K$ .

---

Figure 41.: The simultaneous sparse backfitting algorithm for MT-SpAM or MR-SpAM. For the multi-response case, the same smoothing matrices are used for each  $k$ .

and the Hessian is

$$H(\tilde{f}) = -\text{diag}\left(p_{\tilde{f}}(X)\right) + p_{\tilde{f}}(X)p_{\tilde{f}}(X)^T.$$

Defining  $B = -(1/4)I_{K-1}$ , it is straightforward to show that  $B \preceq H(\tilde{f})$ , i.e.,  $H(\tilde{f}) - B$  is positive-definite. Therefore, we have that

$$L(f) \geq L(\hat{f}) + \mathbb{E}\left[\nabla L(\hat{f})^T(f - \hat{f})\right] + \frac{1}{2}\mathbb{E}\left[(f - \hat{f})^T B(f - \hat{f})\right]. \quad (6.72)$$

The following lemma results from straightforward calculation.

**Lemma 6.4.** *The solution  $f$  that maximizes the righthand side of (6.72) is equivalent to the solution that minimizes  $\frac{1}{2}\mathbb{E}(\|Z - Af\|_n^2)$  where  $A = (-B)^{1/2}$  and  $Z = A^{-1}(Y - p_{\hat{f}}) + A\hat{f}$ .*

Recalling that  $f^{(k)} = \alpha^{(k)} + \sum_{j=1}^d f_j^{(k)}$ , equation (6.71) and Lemma 6.4 then justify the use of the auxiliary functional

$$\frac{1}{2} \sum_{k=1}^{K-1} \mathbb{E} \left[ \left( Z'^{(k)} - \sum_{j=1}^d f_j^{(k)}(X_j) \right)^2 \right] + \lambda' \Phi_{K-1}(f) \quad (6.73)$$

where

$$Z'^{(k)} = 4 \left( Y^{(k)} - \mathbb{P}_{\hat{f}}(Y^{(k)} = 1 | X) \right) + \hat{\alpha}^{(k)} + \sum_{j=1}^d \hat{f}_j^{(k)}(X_j)$$

and  $\lambda' = \sqrt{2}\lambda$ . This is precisely in the form of a multi-response SpAM optimization problem in equation (6.65). The resulting algorithm, in the finite sample case, is shown in Figure 42.

## SMALR: SPARSE MULTI-CATEGORY ADDITIVE LOGISTIC REGRESSION

---

*Input:* Data  $(X^{(i)}, Y^{(i)}), i = 1, \dots, n$  and regularization parameter  $\lambda$ .

*Initialize:*  $\hat{f}_j^{(k)} = 0$  and

$$\hat{\alpha}^{(k)} = \log \left( \frac{\sum_{i=1}^n Y^{(i),(k)}}{\left( n - \sum_{i=1}^n \sum_{k'=1}^{K-1} Y^{(i),(k')} \right)} \right)$$

for  $k = 1, \dots, K - 1$

*Iterate until convergence:*

(1) Compute

$$p_{\hat{f}}^{(k)}(X^{(i)}) \equiv \mathbb{P}(Y^{(k)} = 1 \mid X = X^{(i)})$$

as in (6.66) for  $k = 1, \dots, K - 1$ ;

(2) Calculate the transformed responses

$$Z_i^{(k)} = 4 \left( Y^{(i),(k)} - p_{\hat{f}}^{(k)}(X^{(i)}) \right) + \hat{\alpha}^{(k)} + \sum_{j=1}^d \hat{f}_j^{(k)}(X_j^{(i)})$$

for  $k = 1, \dots, K - 1$  and  $i = 1, \dots, n$ ;

(3) Call subroutines

$$(\hat{f}^{(1)}, \dots, \hat{f}^{(K-1)}) \leftarrow \text{MR-SpAM} \left( (X^{(i)}, Z_i^{(k)})_{i=1}^n, \sqrt{2}\lambda \right);$$

(4) Adjust the intercepts:  $\alpha^{(k)} \leftarrow \frac{1}{n} \sum_{i=1}^n Z_i^{(k)}$ ;

*Output:* Functions  $\hat{f}^{(k)}$  and intercepts  $\hat{\alpha}^{(k)}$  for  $k = 1, \dots, K - 1$ .

---

Figure 42.: The penalized local scoring algorithm for SMALR.

#### 6.4 THEORETICAL PROPERTIES OF THE MULTI-TASK SPAM

The theory of the Multi-task SpAM is a straightforward extension of that of SpAM. In particular, under the same assumptions as those in Theorems 6.2 and 6.3, we can show that the multi-task SpAM is also sparsistent and consistent. The main techniques on generalizing the sparsistency and persistency conditions of single-task SpAM to multi-task SpAM have been shown in Liu and Zhang [2008]. Since the analysis of the multi-task SpAM is a trivial generalization of the single-task SpAM, we only present the single-task SpAM analysis in the appendix of this chapter. Note that these proofs have already appeared in the thesis of Pradeep Ravikumar and in Ravikumar et al. [2009a]

(with Han Liu as a co-author). We present these proofs in the appendix for completeness.

## 6.5 NEW INSIGHTS ON THE SMOOTH SPARSE BACKFITTING ALGORITHM

The results presented in this chapter show how many of the recently established theoretical properties of  $\ell_1$  regularization for linear models extend to sparse additive models (SpAM). The sparse additive models (SpAM) are a new class of methods for high-dimensional nonparametric regression and classification. An efficient algorithm called sparse backfitting has been developed to fit these models even when the number of covariates is larger than the sample size. This algorithm is motivated as a coordinate descent procedure to minimize a population version optimization problem. Such a procedure is not implementable due to the involvement of unknown population distributions. To solve this problem, the sparse backfitting algorithm uses the empirical distributions to approximate the population counterparts. The algorithm is intuitive and does convergence well in most real-world applications. However, such a “plug-in” type procedure makes the analysis difficult. We even do not know under what conditions it converges or under what conditions the solution is unique. Our theoretical analyses have made use of a particular form of smoothing, using a truncated orthogonal basis. An important problem is thus to extend the theory to cover more general classes of smoothing operators, especially kernel smoothers. Convergence properties of the SpAM backfitting algorithm should also be investigated; convergence of special cases of standard backfitting is studied by [Buja et al. \[1989\]](#).

In this section, we show that when using the popular kernel smoothers, a version of the sparse backfitting algorithm is exactly the coordinate descent procedure of a data-version of and infinite-dimensional optimization problem. The key trick is to build a larger Hilbert space which includes all the observed data vector and the  $n$ -tuple smooth functions as its elements. A Stochastic bilinear form and its induced random (semi)-norm can then be defined using kernels. The infinite-dimensional optimization problems are formulated using these newly defined random norms.

### 6.5.1 Population Version of the Sparse Backfitting Algorithm

Let  $Y, X$  be random variables of dimension 1 and  $d$  respectively and let  $\{(Y^{(i)}, X^{(i)})\}_{i=1}^n$  be a random sample drawn from  $(Y, X)$ . Here, we suppose the covariates  $X_j$  take values in a bounded interval  $I_j$  and define the product space  $I = I_1 \times \dots \times I_d$ . Without loss of generality, we simply assume  $I = [0, 1]^d$ .

Recall that the SpAM assumes an additive decomposition of the mean function, i.e., for all  $i = 1, \dots, n$ :

$$Y^{(i)} = m(X^{(i)}) + \epsilon^i = \sum_{j=1}^d f_j(X_j^{(i)}) + \epsilon^i \text{ where } \epsilon^i \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2). \quad (6.74)$$

where we assume each  $f_j$  is a smooth unknown function lies in a second-order Sobolev ball with finite radius. To make the model identifiable, we also assume

$$\mathbb{E}Y = 0, \quad \forall j = 1, \dots, d, \quad \mathbb{E}f_j(X_j) = 0, \quad (6.75)$$

where the expectation is taken with respect to the probability measures induced by  $Y, \{X_j\}_{j=1}^d$ .

The population version of the SpAM solves the following optimization problem, where a population version  $L_2(\mu)$ -norm is applied to regularize the model.

$$\hat{f}_1, \dots, \hat{f}_p = \arg \min_{f_1, \dots, f_d} \left\{ \frac{1}{2} \mathbb{E} \left( Y - \sum_{j=1}^d f_j(X_j) \right)^2 + \lambda \sum_{j=1}^d \sqrt{\mathbb{E}f_j^2(X_j)} \right\} \quad (6.76)$$

subject to the identifiability constraints  $\mathbb{E}f_j(X_j) = 0$  for  $j = 1, \dots, d$ . This problem is not solvable since the distribution of  $(Y, X)$  is unknown. In the following, we consider a data-version algorithm using the kernel smoothers.

### 6.5.2 Function Space and Semi-Norms

We adopt the general framework from [Mammen et al., 1999], which views the smoothing procedure as a projection of the data, with respect to appropriate norms in a suitably defined vector space. Such a normed vector space contains both the space of data vector and the space of candidate regression functions. These two subspaces contain all the information relevant to the smoothing problems and reflect the full structure of smoothing. In particular, this vector space include both the data vector  $Y = (Y^{(1)}, \dots, Y^{(n)})$  and the candidate smooths  $f(x)$  is a product space containing  $n$ -tuples of functions,

$$\mathcal{F} = \left\{ (f^i : i = 1, \dots, n) : \text{Here, } f^i \text{ are functions from } \mathbb{R}^d \text{ to } \mathbb{R} \right\}. \quad (6.77)$$

The data vector  $Y$  is an element of  $\mathcal{F}$  simply by setting  $f^i \equiv Y^{(i)}, i = 1, \dots, n$ . The subspace of such  $n$ -tuples of constant functions are called data subspace, denoted as  $\mathcal{F}^Y$ . For a candidate smooth function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , we write  $f$  for the  $n$ -tuple where every entry is  $f(x)$ , i.e.

$$f^i(x) \equiv f(x), i = 1, \dots, n. \quad (6.78)$$



Then, such a subspace is called smoothing function space, denoted as  $\mathcal{F}_{\text{full}}$ , a sub-subspace which only contains additive functions is called  $\mathcal{F}_{\text{add}}$ :

$$\begin{aligned}\mathcal{F}_{\text{full}} &= \left\{ f \in \mathcal{F} : f^i \text{ does not depend on } i \right\} \\ \mathcal{F}_{\text{add}} &= \left\{ f \in \mathcal{F}_{\text{full}} : f^i(x) = f_1(x_1) + \dots + f_d(x_d) \text{ where } f_j : \mathbb{R} \rightarrow \mathbb{R} \right\}\end{aligned}$$

For two  $n$ -tuple elements  $\mathbf{f} = (f^1, \dots, f^n)$  and  $\mathbf{g} = (g^1, \dots, g^n)$  on the space  $\mathcal{F}$ , we define an inner product (strictly speaking, bilinear form) as

$$\langle \mathbf{f}, \mathbf{g} \rangle_* = \frac{1}{n} \sum_{i=1}^n \int f^i(x) g^i(x) \prod_{j=1}^d K_h(X_j^{(i)} - x_j) dx \quad (6.79)$$

where  $K_h(\cdot)$  is a nicely defined kernel function having compact support.

The corresponding induced norm (strictly speaking, semi-norm) as

$$\|\mathbf{f}\|_* = \sqrt{\int \frac{1}{n} \sum_{i=1}^n |f^i(x)|^2 \prod_{j=1}^d K_h(X_j^{(i)} - x_j) dx}. \quad (6.80)$$

Let

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^d K_h(X_j^{(i)} - x_j) \quad (6.81)$$

be the kernel density estimate of the design density using a product kernel. Therefore, for  $n$ -tuple functions  $\mathbf{f} = (f^1, \dots, f^n)^T$  in  $\mathcal{F}_{\text{Full}}$  (i.e.  $f^1 = \dots = f^n = f$ ), we have

$$\|\mathbf{f}\|_* = \sqrt{\int f^2(x) \hat{p}(x) dx}. \quad (6.82)$$

In the following, we simply use  $\|f\|_*$  to denote  $\|\mathbf{f}\|_*$  if  $\mathbf{f} \in \mathcal{F}_{\text{full}}$ .

Under this framework and using the fact that

$$\mathbb{E} (Y - f(X))^2 = \int \mathbb{E} \left[ (Y - f(X))^2 \mid X = x \right] p(x) dx, \quad (6.83)$$

the data-version SpAM is formulated by approximating the the population version expectation and norm by

$$\mathbb{E} (Y - f(X))^2 \quad (6.84)$$

$$\approx \int \frac{\sum_{i=1}^n \prod_{j=1}^d K_h(X_j^{(i)} - x_j) \left( Y^{(i)} - f(x) \right)^2}{\sum_{i=1}^n \prod_{j=1}^d K_h(X_j^{(i)} - x_j)} \hat{p}(x) dx \quad (6.85)$$

$$= \|Y - f\|_*^2 \quad (6.86)$$

and

$$\sqrt{\mathbb{E} f^2(X)} = \sqrt{\int f^2(x) \hat{p}(x) dx} \approx \|f\|_*. \quad (6.87)$$

Therefore, the data-version SpAM can be formulated as the following optimization problem:

$$\hat{f}_1, \dots, \hat{f}_d = \arg \min_{f_1, \dots, f_d} \left\{ \|Y - \sum_{j=1}^d f_j\|_*^2 + \lambda \sum_{j=1}^d \|f_j\|_* \right\}. \quad (6.88)$$

Note that here  $f_1, \dots, f_d$  are still in the function space. (6.88) can be rewritten as

$$\begin{aligned} \hat{f}_1, \dots, \hat{f}_d = \arg \min_{f_1, \dots, f_d} & \frac{1}{2n} \sum_{i=1}^n \int_I \left( Y^{(i)} - \sum_{j=1}^d f_j(x_j) \right)^2 \prod_{j=1}^d K_h(X_j^{(i)} - x_j) dx \\ & + \lambda \sum_{j=1}^d \sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j}. \end{aligned} \quad (6.89)$$

This is a penalized least squares problem in an uncountable infinite-dimensional space with a non-differentiable penalty. We derive a smoothed sparse backfitting algorithm to solve it.

### 6.5.3 Smooth Sparse Backfitting Using Kernel Smoothers

We start with a simple lemma which characterizes the solution of an unconstrained convex optimization problem in the infinite-dimensional space:

Assuming  $X$  is a Banach space with its topological dual denoted by  $X^*$ , let  $f : X \rightarrow \mathbb{R}$  be a convex functional on  $X$ .

**Definition 6.1.** (Subgradient) An element  $x^* \in X^*$  is a subgradient of the convex functional  $f$  at  $x$  if and only if it satisfies

$$f(y) \geq f(x) + x^*(y - x) \quad \forall y \in X. \quad (6.90)$$

Let  $\partial f(x)$  be the set of all subgradients at  $x$ , if  $\partial f(x)$  is not empty, we call  $f$  subdifferentiable at  $x$ . If  $f$  is subdifferentiable for all  $x \in X$ , it's called subdifferentiable.

**Proposition 6.2.** (Necessary and sufficient condition) Let  $X$  be a Banach space and  $f : X \rightarrow \mathbb{R}$  be a convex functional. Then  $x \in X$  is a global minimizer of  $f$  if and only if

$$0 \in \partial f(x). \quad (6.91)$$

*Proof.* (Sufficiency): If  $0 \in \partial f(x)$ , then

$$f(y) \geq f(x) + 0(y - x) = f(x) \quad \forall y \in X. \quad (6.92)$$

which implies that  $x$  is a global minimizer.

(Necessity): Assume  $0 \notin \partial f(x)$ , there must exists  $z \in X$  such that

$$f(z) < f(x) + 0(z - x) = f(x), \quad (6.93)$$

therefore  $x$  can not be a global minimizer of  $f$ .  $\square$

Given the above proposition, we have the next lemma which characterizes the solution of the smooth sparse backfitting:

**Lemma 6.5.** (*Optimality condition of the smooth sparse backfitting*) Let

$$\hat{p}_j(x_j) \equiv \frac{1}{n} \sum_{i=1}^n K_h(X_j^{(i)} - x_j)$$

and  $I_{-j} = I_1 \times \cdots \times I_{j-1} \times I_{j+1} \times \cdots \times I_p$  for  $\forall j = 1, \dots, d$ ,  $\hat{f} = \sum_{j=1}^d \hat{f}_j$  is the solution to the problem in (6.88) if and only if there exists  $\eta_1, \dots, \eta_d$ , such that

$$\frac{1}{n} \sum_{i=1}^n \int_{I_{-j}} \left( Y^{(i)} - \sum_{k \neq j} \hat{f}_k(x_k) - \hat{f}_j(x_j) \right) \frac{\prod_{\ell=1}^d K_h(X_\ell^{(i)} - x_\ell)}{\hat{p}_j(x_j)} dx_{-j} = \lambda \eta_j \text{ a.s.}$$

where  $\|\eta_j\|_* \leq 1$  if  $f_j(x_j) = 0$ , otherwise

$$\eta_j = \frac{\hat{f}_j(x_j)}{\|\hat{f}_j\|_*}. \quad (6.94)$$

*Proof.* First, since  $\hat{p}_j(x_j) \equiv \frac{1}{n} \sum_{i=1}^n K_h(X_j^{(i)} - x_j)$ , (6.103) can be re-written as

$$\begin{aligned} \int_{I_j} \frac{1}{2n} \sum_{i=1}^n \int_{I_{-j}} \left( Y^{(i)} - \sum_{k=1}^d f_k(x_k) \right)^2 \frac{\prod_{\ell=1}^d K_h(X_\ell^{(i)} - x_\ell)}{\hat{p}_j(x_j)} dx_{-j} \hat{p}_j(x_j) dx_j \\ + \lambda \sum_{k=1}^d \sqrt{\int_{I_j} f_j^2(x_j) \hat{p}_j(x_j) dx_j}. \end{aligned} \quad (6.95)$$

From Proposition 6.2, evaluate the subdifferential with respect to  $f_j$ , and set it to zero. We have

$$\frac{1}{n} \sum_{i=1}^n \int_{I_{-j}} \left( Y^{(i)} - \sum_{k \neq j} f_k(x_k) - f_j(x_j) \right) \frac{\prod_{\ell=1}^d K_h(X_\ell^{(i)} - x_\ell)}{\hat{p}_j(x_j)} dx_{-j} = \lambda \eta_j \text{ a.s.}$$

where  $\eta_j \in \partial \sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j}$ , which satisfies  $\|\eta_j\|_* \leq 1$  if  $f_j(x_j) = 0$ , otherwise

$$\eta_j = \frac{f_j(x_j)}{\sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j}} = \frac{f_j(x_j)}{\|f_j\|_*} \quad (6.96)$$

□

From the optimality conditions, for the case  $f_j(x_j) \neq 0$ , we have

$$\frac{\sum_{i=1}^n Y^{(i)} K_h(X_j^{(i)} - x_j)}{\sum_{i=1}^n K_h(X_j^{(i)} - x_j)} - \sum_{k \neq j} \int_{I_k} f_k(x_k) \frac{\hat{p}_{j,k}(x_j, x_k)}{\hat{p}_j(x_j)} dx_k = f_j(x_j) \left( 1 + \frac{\lambda}{\|f_j\|_*} \right)$$

where

$$\hat{p}_{j,k}(x_j, x_k) = \frac{1}{n} \sum_{i=1}^n K_h(X_j^{(i)} - x_j) K_h(X_k^{(i)} - x_k). \quad (6.97)$$

Let  $f_{0j}(x_j) \equiv \frac{\sum_{i=1}^n Y^{(i)} K_h(X_j^{(i)} - x_j)}{\sum_{i=1}^n K_h(X_j^{(i)} - x_j)}$  be the one-dimensional univariate kernel smoother, and define

$$P_j(x_j) \equiv f_{0j}(x_j) - \sum_{k \neq j} \int_{I_k} f_k(x_k) \frac{\hat{p}_{j,k}(x_j, x_k)}{\hat{p}_j(x_j)} dx_k. \quad (6.98)$$

We then obtain

$$P_j(x_j) = f_j(x_j) \left( 1 + \frac{\lambda}{\|f_j\|_*} \right) \quad (6.99)$$

which implies that

$$f_j(x_j) = \left( 1 - \frac{\lambda}{\|P_j\|_*} \right) P_j(x_j) \quad (6.100)$$

Therefore, the final updating rule is

$$f_j(x_j) \leftarrow \left[ 1 - \frac{\lambda}{\sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} P_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j}} \right]_+ P_j(x_j) \quad (6.101)$$

where

$$P_j(x_j) = f_{0j}(x_j) - \sum_{k \neq j} \int_{I_k} f_k(x_k) \frac{\hat{p}_{j,k}(x_j, x_k)}{\hat{p}_j(x_j)} dx_k. \quad (6.102)$$

In which only the one-dimensional and two-dimensional marginal kernel density estimators are involved in. For the computation, we need to evaluate the function values on a grid so that the integrals can be easily evaluated.

## 6.5.4 Existence and Uniqueness of the Solution

Define

$$F(f_1, \dots, f_d) = \frac{1}{2n} \sum_{i=1}^n \int_I \left( Y^{(i)} - \sum_{j=1}^d f_j(x_j) \right)^2 \prod_{j=1}^d K_h(X_j^{(i)} - x_j) dx \\ + \lambda \sum_{j=1}^d \sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j}.$$

where  $\lambda > 0$  and  $n > 2$ .

We consider the following optimization problem:

$$\hat{f}_1, \dots, \hat{f}_d = \arg \min_{f_1, \dots, f_d} F(f_1, \dots, f_d). \quad (6.103)$$

This is a penalized least squares problem in an uncountable infinite-dimensional space with a non-smooth penalty. We want to show that the solution exists and is unique.

Define

$$c = \inf_{f_1, \dots, f_d} F(f_1, \dots, f_d). \quad (6.104)$$

We know that  $0 \leq c < \infty$ .

We define a measure  $\mu_j(\cdot)$

$$\mu_j(E) = \frac{1}{n} \sum_{i=1}^n \int_E K_h(X_j^{(i)} - x_j) dx_j. \quad (6.105)$$

for arbitrary measurable subset  $E \subset I_j$ .

Denote

$$\mathcal{F}_j = \left\{ f_j : I_j \rightarrow \mathbb{R} \mid \int_{I_j} f_j^2(x_j) \mu_j(dx_j) < \infty \right\} \quad (6.106)$$

to be the Hilbert space equipped with the inner product  $\langle \cdot, \cdot \rangle_j$ :

$$\langle f_j, g_j \rangle_j = \frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j(x_j) g_j(x_j) K_h(X_j^{(i)} - x_j) dx_j.$$

Let

$$\mathcal{F} = \mathcal{F}_1 \times \dots \times \mathcal{F}_d$$

be the  $d$ -fold direct-sum Hilbert space equipped with the inner product  $\langle \cdot, \cdot \rangle$ :

$$\langle (f_1, \dots, f_d), (g_1, \dots, g_d) \rangle = \sum_{j=1}^d \langle f_j, g_j \rangle_j \quad (6.107)$$

where  $(f_1, \dots, f_d), (g_1, \dots, g_d) \in \mathcal{F}$ ,

**Proposition 6.3.**  $\mathcal{F}$  is a Hilbert space equipped with an inner product defined in (6.107).

*Proof.* For any  $(f_1, \dots, f_d) \in \mathcal{F}$ , we have

$$\sqrt{\langle (f_1, \dots, f_d), (f_1, \dots, f_d) \rangle} = \sqrt{\sum_{j=1}^d \langle f_j, f_j \rangle_j} < \infty. \quad (6.108)$$

□

**Proposition 6.4.**  $F(f_1, \dots, f_d)$  is a lower semi-continuous function with respect to weak convergence.

*Proof.* It's obvious that  $F(\cdot)$  is a (strongly) continuous function, which implies (strongly) lower semi-continuity. Further, since  $\mathcal{F}$  is a Hilbert space (which is a reflexive Banach space), the result follows from the convexity of  $F(\cdot)$ . □

**Theorem 6.5.** (Existence and Uniqueness) There exists a unique  $(f_1^*, \dots, f_d^*) \in \mathcal{C}$ , such that

$$F(f_1^*, \dots, f_d^*) = c. \quad (6.109)$$

*Proof.* (Existence) First, we show the existence of a minimizer.

Let  $\{(f_1^{(k)}, \dots, f_d^{(k)})\}_{k=1}^\infty \in \mathcal{F}$  be a minimizing sequence of  $F$ , i.e.

$$\lim_{k \rightarrow \infty} F(f_1^{(k)}, \dots, f_d^{(k)}) = c. \quad (6.110)$$

Therefore, for large enough  $k$ , we have

$$0 \leq F(f_1^{(k)}, \dots, f_d^{(k)}) \leq c + 1. \quad (6.111)$$

Therefore,

$$\mathcal{C} = \left\{ \sqrt{\langle (f_1^{(k)}, \dots, f_d^{(k)}), (f_1^{(k)}, \dots, f_d^{(k)}) \rangle} \leq c + 1 \right\}$$

is bounded convex set and is also (strongly) convex. This implies that  $\mathcal{C}$  must be weakly closed.

Since  $\mathcal{C}$  is bounded and weakly closed, it follows that  $\{(f_1^{(k)}, \dots, f_d^{(k)})\}_{k=1}^\infty$  has a weakly convergent subsequence

$$\{(f_1^{(k_\ell)}, \dots, f_d^{(k_\ell)})\}_{\ell=1}^\infty \xrightarrow{w} (f_1^*, \dots, f_d^*). \quad (6.112)$$

Next, since  $F(\cdot)$  is lower semi-continuous with respect to weak convergence, we have

$$c \leq F(f_1^*, \dots, f_d^*) \quad (6.113)$$

$$\leq \liminf_{\ell \rightarrow \infty} F(f_1^{(k_\ell)}, \dots, f_d^{(k_\ell)}) \quad (6.114)$$

$$= \lim_{\ell \rightarrow \infty} F(f_1^{(k_\ell)}, \dots, f_d^{(k_\ell)}) \quad (6.115)$$

$$= c. \quad (6.116)$$

This proves the existence result.

(Uniqueness) To show the uniqueness, it suffices to prove that  $F(\cdot)$  is strictly convex. This trivially follows from the fact that

$$\sqrt{\frac{1}{n} \sum_{i=1}^n \int_{I_j} f_j^2(x_j) K_h(X_j^{(i)} - x_j) dx_j} \quad (6.117)$$

is a strictly convex functional of  $f_j \in \mathcal{F}_j$ .  $\square$

## 6.6 EXPERIMENTAL RESULTS

In this section, we first use simulated data to investigate the performance of the MT-SpAM simultaneous sparse backfitting algorithm. We then apply SMALR to a tumor classification and biomarker identification problem. In the next section, we present an application in which we use the MT-SpAM to predictive brain activity patterns and feature selection.

In all experiments, the data are rescaled to lie in the  $d$ -dimensional cube  $[0, 1]^d$ . We use local linear smoothing with a Gaussian kernel. To choose the regularization parameter  $\lambda$ , we simply use  $J$ -fold cross-validation or the GCV score from [Ravikumar et al., 2007] extended to the multi-task setting:

$$\text{GCV}(\lambda) = \sum_{i=1}^n \sum_{k=1}^K \frac{\mathcal{Q}_{\hat{f}^{(k)}}(X^{(i),(k)}, Y^{(i),(k)})}{(n^2 K^2 - (nK))} \text{df}(\lambda))^2$$

where  $\text{df}(\lambda) = \sum_{k=1}^K \sum_{j=1}^d v_j^{(k)} I(\|\hat{f}_j^{(k)}\|_n \neq 0)$ , and  $v_j^{(k)} = \text{trace}(\mathcal{S}_j^{(k)})$  is the effective degrees of freedom for the univariate local linear smoother on the  $j^{\text{th}}$  variable.

### 6.6.1 Synthetic Data

We generated  $n = 100$  observations from a 10-dimensional three-task additive model with four relevant variables:

$$Y^{(i),(k)} = \sum_{j=1}^4 f_j^{(k)}(x_j^{(i),(k)}) + \epsilon_i^{(k)}, k = 1, 2, 3,$$

where  $\epsilon_i^{(k)} \sim N(0, 1)$ ; the component functions  $f_j^{(k)}$  are plotted in Figure 43. The 10-dimensional covariates are generated as

$$X_j^{(k)} = \frac{(W_j^{(k)} + tU^{(k)})}{1 + t}, j = 1, \dots, 10$$

where  $W_1^{(k)}, \dots, W_{10}^{(k)}$  and  $U^{(k)}$  are i.i.d. sampled from  $\text{Uniform}(-2.5, 2.5)$ . Thus, the correlation between  $X_j$  and  $X_{j'}$  is  $t^2/(1 + t^2)$  for  $j \neq j'$ .

The results of applying MT-SpAM with the bandwidths  $h = (0.08, \dots, 0.08)$  and regularization parameter  $\lambda = 0.25$  are summarized in Figure 43. The upper 12 figures show the 12 relevant component functions for the three tasks; the estimated function components are plotted as solid black lines and the true function components are plotted using dashed red lines. For all the other variables (from dimension 5 to dimension 10), both the true and estimated components are zero. The middle three figures show regularization paths as the parameter  $\lambda$  varies; each curve is a plot of the maximum empirical  $L_1$  norm of the component functions for each variable, with the red vertical line representing the selected model using the GCV score. As the correlation increases ( $t$  increases), the separation between the relevant dimensions and the irrelevant dimensions becomes smaller. Using the same setup but with one common design matrix, we also compare the quantitative performance of MR-SpAM with MARS [Friedman, 1991], which is a popular method for multi-response additive regression. Using 100 simulations, the table illustrates the number of times the models are correctly identified and the mean squared error with the standard deviation in the parentheses. (The MARS simulations are carried out in R, using the default options of the `mars` function in the `mda` library.)

#### 6.6.2 Gene Microarray Data

Here we apply the sparse multi-category additive logistic regression model to a microarray dataset for small round blue cell tumors (SRBCT) [Khan et al., 2001]. The data consist of expression profiles of 2,308 genes [Khan et al., 2001] with tumors classified into 4 categories: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS). The dataset includes a training set of size 63 and a test set of size 20. These data have been analyzed by different groups. The main purpose is to identify important biomarkers, which are a small set of genes that can accurately predict the type of tumor of a patient. To achieve 100% accuracy on the test data, Khan et al. [2001] use an artificial neural network approach to identify 96 genes. Tibshirani et al. [2002] identify a set of only 43 genes, using a method called nearest shrunken centroids. Zhang et al. [2008] identify 53 genes using the sup-norm support vector machine.

In our experiment, SMALR achieves 100% prediction accuracy on the test data with only 20 genes, which is a much smaller set of predictors than identified in the previous approaches. We follow the same procedure as in [Zhang et al., 2008], and use a very simple screening step based on the marginal correlation to first reduce the number of genes to 500. The SMALR model is then trained using a plugin bandwidth  $h_0 = 0.08$ , and the regularization parameter  $\lambda$  is tuned using 4-fold cross validation. The results are tabulated in Figure 44. In the left figure, we show a “heat map” of the



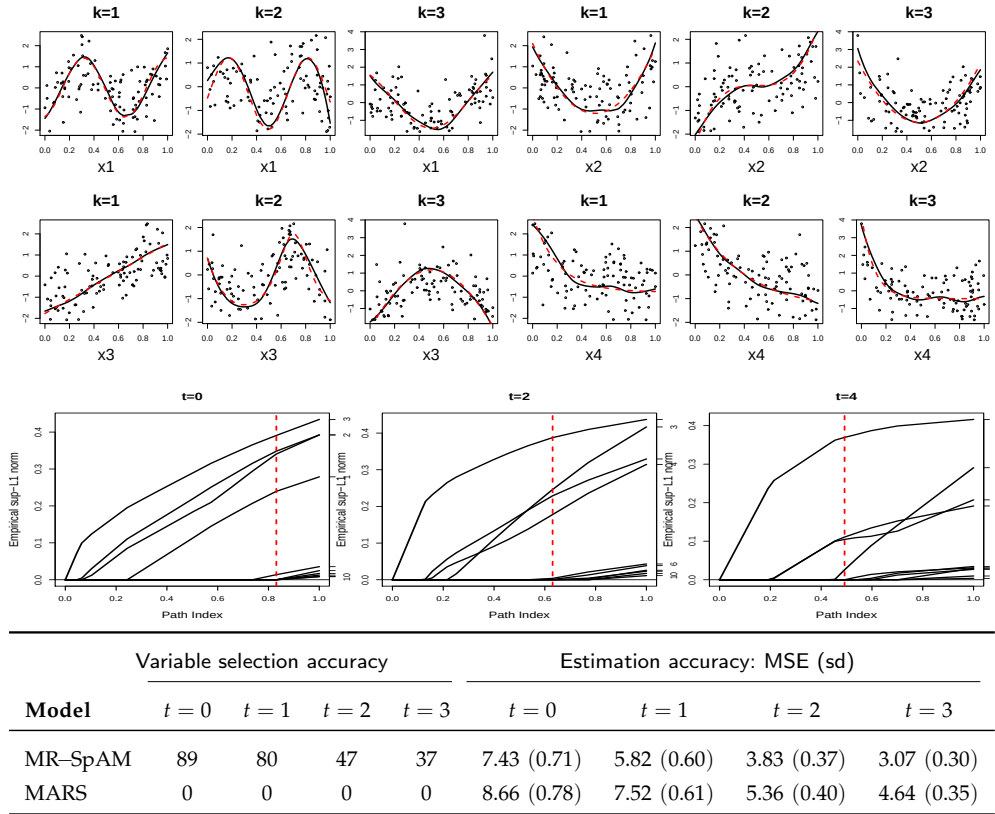


Figure 43.: (Top) Estimated vs. true functions from MT-SpAM; (Middle) Regularization paths using MT-SpAM. (Bottom) Quantitative comparison between MR-SpAM and MARS

selected variables on the training set. The rows represent the selected genes, with their cDNA chip image id. The patients are grouped into four categories according to the corresponding tumors, as illustrated in the vertical groupings. The genes are ordered by hierarchical clustering of their expression profiles. The heatmap clearly shows four block structures for the four tumor categories. This suggests visually that the 20 genes selected are highly informative of the tumor type. In the middle of Figure 44, we plot the fitted discriminant functions of different genes, with their image ids listed on the plot. The values  $k = 1, 2, 3$  under each subfigure indicate the discriminant function the plot represents. We see that the fitted functions are nonlinear. The right subfigure illustrates the total number of misclassified samples using 4-fold cross validation, the  $\lambda$  values 0.3, 0.4 are both zero, for the purpose of a sparser biomarker set, we choose  $\lambda = 0.4$ . Interestingly, only 10 of the 20 identified genes from our method are among the 43 genes selected using the shrunken centroids approach of Tibshirani et al. [2002]. 16 of them are among the 96 genes selected by neural network approach of Khan et al. [2001]. This non-overlap may suggest some further investigation.

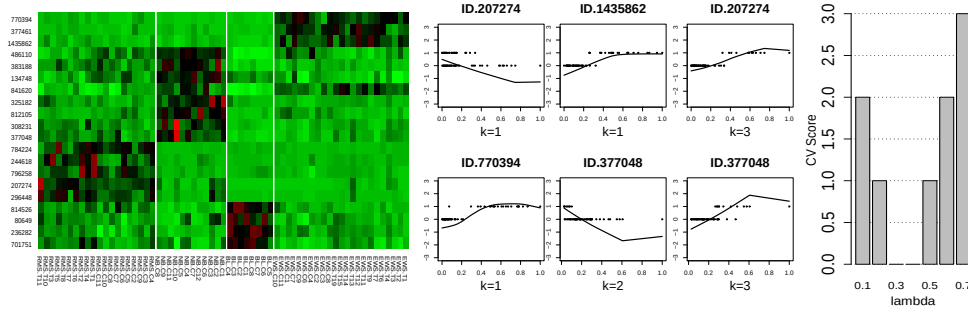


Figure 44.: SMALR results on gene data: heat map (left), marginal fits (center), and CV score (right).

## 6.7 A CASE STUDY OF MT-SPAM: NEURAL SEMANTIC BASIS DISCOVERY

In this section, we present a case study of the MT-SpAM by applying it to a problem in cognitive neuroscience. Specifically, we consider the task of predicting a person’s neural activity in response to an arbitrary word in English as described in [Mitchell et al. \[2008\]](#), [Liu et al. \[2009b\]](#). Their goal is to predict the neural image recorded using *functional magnetic resonance imaging* (fMRI) when a person thinks about a given word. To achieve this goal, they adopt a two-stage procedure as presented in [Figure 45](#). Given a stimulus word  $w$ , the first step encodes the meaning of  $w$  in terms of intermediate semantic features whose values are computed from the co-occurrences of  $w$  with a *semantic basis* in a large text corpus. The second step predicts the neural fMRI activation at each voxel of the brain, as a sum of neural activations contributed by each of the intermediate semantic features. A voxel represents a  $1\text{-}3\text{ mm}^3$  volume in the brain and is the basic spatial unit of measurement in fMRI. The training process use a small number of words to learn a linear model that maps the intermediate semantic features to neural activation images while a person is thinking about those training words.

In [Mitchell et al. \[2008\]](#), 25 sensory-action verbs are selected as the semantic basis as shown in [Table 4](#). These 25 verbs are hand-crafted based on domain knowledge from the cognitive neuroscience literature. For example, words related to foods such as apples and oranges have frequent co-occurrences with the word eat but few co-occurrences with the word wear. Conversely, words related to clothes such as shirt or dress co-occur frequently with the word wear, but not the word eat. Thus eat and wear are example *basis words* used to encode relationships of a broad set of other words. A natural question is: *What is the optimal basis of words to represent semantic meaning across many concepts?*

Rather than relying on models that require manual selection of a set of words, MT-SpAM leads to models that will perform *variable selection* to automatically learn a semantic basis of word meaning. In this way, we not only want to predict neural activity well, but also give insights into how the brain

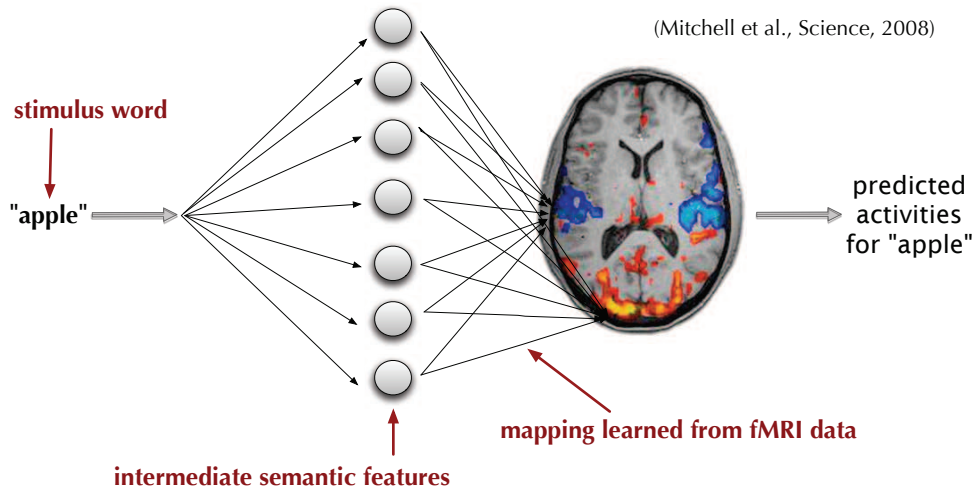


Figure 45.: Model for predicting fMRI activation for a stimuli

|        |            |      |      |       |
|--------|------------|------|------|-------|
| See    | Eat        | Run  | Say  | Enter |
| Hear   | Touch      | Push | Fear | Drive |
| Listen | Rub        | Fill | Open | Wear  |
| Taste  | Approach   | Move | Lift | Break |
| Smell  | Manipulate | Ride | Near | Clean |

Table 4.: The semantic basis used in Mitchell et al. (2008)

represents the meaning of different concepts. *The hope is that learning directly from data could lead to new semantic discovery in cognitive neuroscience.*

#### 6.7.1 Datasets

For our study, we utilize the two datasets described in [Mitchell et al. \[2008\]](#). The first dataset is collected using fMRI. First, we select 60 words as shown in Table 5 as stimulus words. The 60 words are composed of nouns from 12 categories with 5 exemplars per category. For example, a *bodypart* category includes Arm, Eye, Foot, Hand, Leg, a *tools* category includes the words Chisel, Hammer, Pliers, Saw, Screwdriver, and a *furniture* category includes Bed, Chair, Dresser, Desk, Table, etc. Then nine participants are presented with 60 different words and are asked to think about each word for several seconds while their neural activities were recorded. So that there are altogether  $n = 60$  fMRI

images taken for each participant<sup>1</sup>. A typical fMRI image contains activities in over 20,000 voxels. We select the top  $K = 500$  voxel responses using the stability criterion score described in [Mitchell et al. \[2008\]](#).

|           |         |              |           |             |
|-----------|---------|--------------|-----------|-------------|
| bear      | cat     | cow          | dog       | horse       |
| arm       | eye     | foot         | hand      | leg         |
| apartment | barn    | church       | house     | igloo       |
| arch      | chimney | closet       | door      | window      |
| coat      | dress   | pants        | shirt     | skirt       |
| bed       | chair   | desk         | dresser   | table       |
| ant       | bee     | beetle       | butterfly | fly         |
| bottle    | cup     | glass        | knife     | spoon       |
| bell      | key     | refrigerator | telephone | watch       |
| chisel    | hammer  | pliers       | saw       | screwdriver |
| carrot    | celery  | corn         | lettuce   | tomato      |
| airplane  | bicycle | car          | train     | truck       |

Table 5.: The 60 stimulus words presented during the fMRI studies. Each row represents a category

The second dataset is a symmetric matrix of text co-occurrences between the 5,000 most frequent words in English. These co-occurrences are derived from the Google Trillion Word Corpus<sup>2</sup>. The meaning of a given stimulus word is represented by a 5,000 dimensional feature vector of co-occurrences (normalized to unit length row norm). Note that the dimension of feature vector, i.e. 5000, is too high for our problem, considering that only 60 training samples are available. Typically, a smaller representation is desired such as hand-crafted 25-verb basis as described above. However, including merely 25-verb basis is too biased. As a compromise, we take the following steps to reduce the size of semantic basis. We remove the stop words, i.e. meaningless frequent words like “the”. If both single and plural form of a noun appear in our 5,000 words, we only keep one of them. Similarly, if original, gerundial and past tense of a verb co-exist, only one of them will be left. Moreover, we choose those words with higher number of hits via Google search and try to balance the number of nouns, verbs, adjectives, adverbs, etc. Finally, 250 words are selected as our semantic basis.

<sup>1</sup> Each image is actually the average of 6 different recordings of each word.

<sup>2</sup> <http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html>

Now, we show that the problem of learning a semantic basis can be formulated into a MT-SpAM problem. The goal of learning a semantic basis is to find a small common set of predictor variables that will predict the neural response well across multiple voxels, where each predictor variable is the text co-occurrences with a particular word from the our selected words. Therefore, for each participant, let the response vector  $\{Y^{(k)}\}_{k=1}^K$  represent the neural activations at a single voxel  $k$ , where each voxel indicates a task and  $K = 500$  is the number of tasks. All these tasks share the common design columns  $\{X_j\}_{j=1}^d \in \mathbb{R}^n$ , representing the co-occurrences of  $n = 60$  training words with  $d = 250$  other common English words in the Google Corpus. Therefore, this is a multitask sparse learning problem with a  $K = 500$  tasks and  $d = 250$  features. More precisely, given a new stimulus word  $w$  and its co-occurrence with semantic basis  $\{X_j\}_{j=1}^d$ , the predicted neural activation  $\hat{Y}^{(k)}$  at voxel  $k$  takes the following additive form:

$$\hat{Y}^{(k)} = \sum_{j=1}^d \hat{f}_j^{(k)}(X_j), \quad (6.118)$$

where  $\hat{f}_j^{(k)}$  are learned functions by MT-SpAM and note that the model is sparse in the sense that many blocks of competent functions  $\hat{f}$  are zeros.

After learning the multi-task sparse additive model using (6.65), given a new stimulus word  $w$ , the predicted neural activation  $Y^{(k)}$  at voxel  $k$  takes the following form:

$$Y^{(k)} = \sum_{j=1}^d \hat{f}_j^{(k)}(X_j(w)), \quad (6.119)$$

where  $X_j(w)$  is the co-currences between word  $w$  and the  $j$ -th dictionary word.  $\hat{f}_j^{(k)}$  is the learned component function indicating the amount of contribution of  $j$ th intermediate semantic feature in activation of voxel  $k$ .

### 6.7.2 Results

To evaluate our methods and compare them to existing results, we use exactly the same experimental protocols described in Mitchell et al. [2008]. For a fair comparison, instead of using the hand-crafted 25 verbs, we use MT-SpAM to select a 25 words' semantic basis. To reduce the large estimation bias introduced by sparse model, after finding semantic basis, we adopt backfitting procedure to perform function estimation only on the 25 words in semantic basis. The evaluation is based on the *leave-two-out-cross-validation* procedure:

We repeat this experiment for each of the nine different participants in the fMRI study and use the same method in Mitchell et al. [2008] to ensure consistency while testing various semantic features.

- 
- a Create a  $60 \times 250$  design matrix of semantic features using co-occurrences between the 60 training words and our selected 250 most common words.
  - b Select 2 words out of 60 for testing and use the other 58 words for training. Using (6.121), learn the function  $f_j^{(k)}$  by setting each  $\{X_j\}_{j=1}^{250}$  to be the  $58 \times 1$  vector of co-occurrences for each of the 250 basis words and each  $Y^{(k)}$  to be the  $58 \times 1$  column vector for each of the top  $K = 500$  voxel responses. In the language of MT-SpAM, this problem corresponds to the scale  $n = 58, d = 250, K = 500$ . The regularization parameter here can be set to choose the desired number of non-zero blocks of competent functions. Each non-zero block corresponds to a word from the original set of 250. We train the full path and pick regularization parameter to yield 25 non-zero blocks as the new semantic basis so that the model is easier to interpret and compare to the existing results.
  - c Create a new matrix of semantic features of the size  $58 \times 25$ , where 25 is the size of new semantic basis. Train a additive model using backfitting to predict each of the 500 voxels from the semantic feature basis.
  - d For each of the two test examples, predict the neural response of the 500 selected voxels. Compute the cosine similarity of each prediction with each of the held out images. Based on the combined similarity scores, choose which prediction goes with each held out image. Test if the joint labeling was correct. This leads to an output of 0 or 1. For more details, see [Mitchell et al. \[2008\]](#).
  - e Repeat steps b-d for all  $\binom{60}{2}$  possible pairs of words (1,770 total). Count the number of incorrect labelings in step e to determine the accuracy of the basis set.
- 

Figure 46.: The leave-two-out-cross-validation protocols

We compare MT-SpAM with several other methods. The first method directly use the hand-crafted 25 words in Table 7 as the semantic basis instead of adopting MT-SpAM to perform word selection in semantic basis. It assumes additive model as in (6.118) and the component functions  $\hat{f}_j^{(k)}$  are learned by backfitting. The second and third methods are based on linear model. The second one is proposed in [Liu et al. \[2009b\]](#) which adopts the same protocols as in Figure 46 but replaces MT-SpAM by multi-task Lasso (MT-Lasso) in step b and backfitting by ridge regression in step c. More precisely, [Liu et al. \[2009b\]](#) assumes that given a new stimulus word and its co-occurrence with

the semantic basis  $\{X_j\}_{j=1}^p$ , the predicted neural activation  $\hat{Y}^{(k)}$  at voxel  $k$  takes the following linear form:

$$\hat{Y}^{(k)} = \sum_{j=1}^d \hat{\beta}_j^{(k)} X_j, \quad (6.120)$$

where  $\hat{\beta}_j^{(k)}$  are learned by the following MT-Lasso optimization problem:

$$\hat{\beta} = \min_{\beta} \left\{ \frac{1}{2} \sum_{k=1}^K \left\| Y^{(k)} - \sum_{j=1}^d \beta_j^{(k)} X_j^{(k)} \right\|_2^2 + \lambda \sum_{j=1}^d \max_k |\beta_j^{(k)}| \right\}. \quad (6.121)$$

The last one is proposed in [Mitchell et al. \[2008\]](#) which directly use the hand-crafted 25 words in Table 7 as semantic basis and train a linear model as in (6.120) by ridge regression.

Our experiments simply use the univariate kernel smoothers with Gaussian kernels to fit MT-SpAM. We conduct sparse backfitting with plug-in bandwidths according to Scott's rule [[Scott, 1992](#), p.152]. More precisely, we adopt diagonal bandwidth matrix  $H = \text{diag}(h_1, \dots, h_d)$  with  $h_j = cn^{-1/5} \hat{\sigma}_j$ , where  $c$  is predefined constant and  $\hat{\sigma}_j$  is the estimated standard deviation for  $X_j$ . According to our experience, we prefer smaller bandwidth for variable selection and larger bandwidth for function estimation in fMRI study. Therefore, we set  $c = 1$  for MT-SpAM and  $c = 5$  for backfitting procedure.

The comparison result are presented in Table 6 and Figure 47.

| Participant Index    | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      |
|----------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| MT-SpAM              | 0.8689 | 0.8260 | 0.8345 | 0.7695 | 0.8068 | 0.7599 | 0.7379 | 0.8429 | 0.8034 |
| Additive (Handcraft) | 0.8232 | 0.7492 | 0.7785 | 0.7299 | 0.8175 | 0.7989 | 0.7718 | 0.7554 | 0.8458 |
| MT-Lasso             | 0.7576 | 0.7785 | 0.7390 | 0.6723 | 0.7429 | 0.7130 | 0.7401 | 0.7780 | 0.6215 |
| Linear (Handcraft)   | 0.8090 | 0.7599 | 0.7610 | 0.6825 | 0.7712 | 0.8181 | 0.7057 | 0.6667 | 0.7972 |

Table 6.: Accuracies for 9 fMRI participants

The experimental result shows that the nonparametric methods (MT-SpAM and Additive(Handcraft)) provide much higher accuracy than linear model based methods (MT-Lasso and Linear(Handcraft)). It shows that *brain activity should NOT be modeled based on linear assumption*. Nonparametric methods might lead to better results in neuroscience study. A typical *nonlinear* estimated function components for a single task is plotted in Figure 48.

Comparing the MT-SpAM to the additive model with hand-crafted features, we see that on participants 1, 2, 3, 4, 8, MT-SpAM outperforms the hand-crafted features; on participant 5, the accuracy is comparable between two methods. From the box plot, the average accuracy of MT-SpAM is slightly

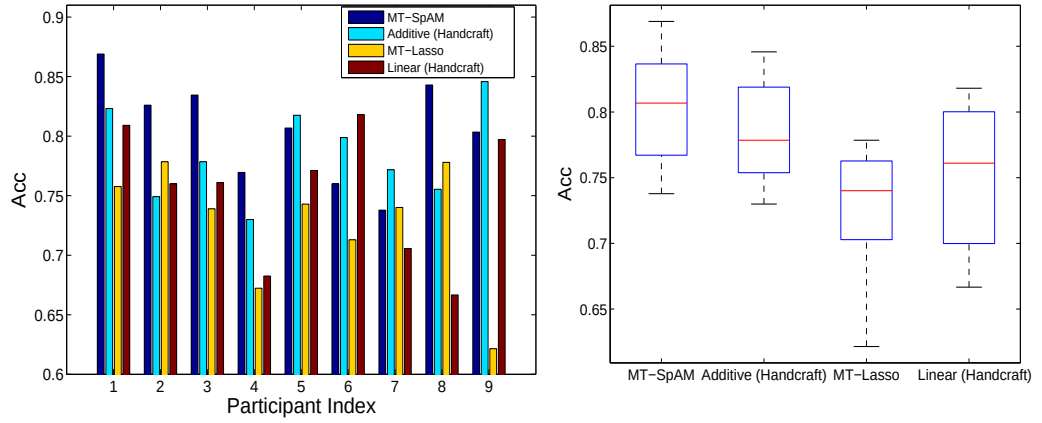


Figure 47.: Bar and Box plots for accuracies for 9 fMRI participants

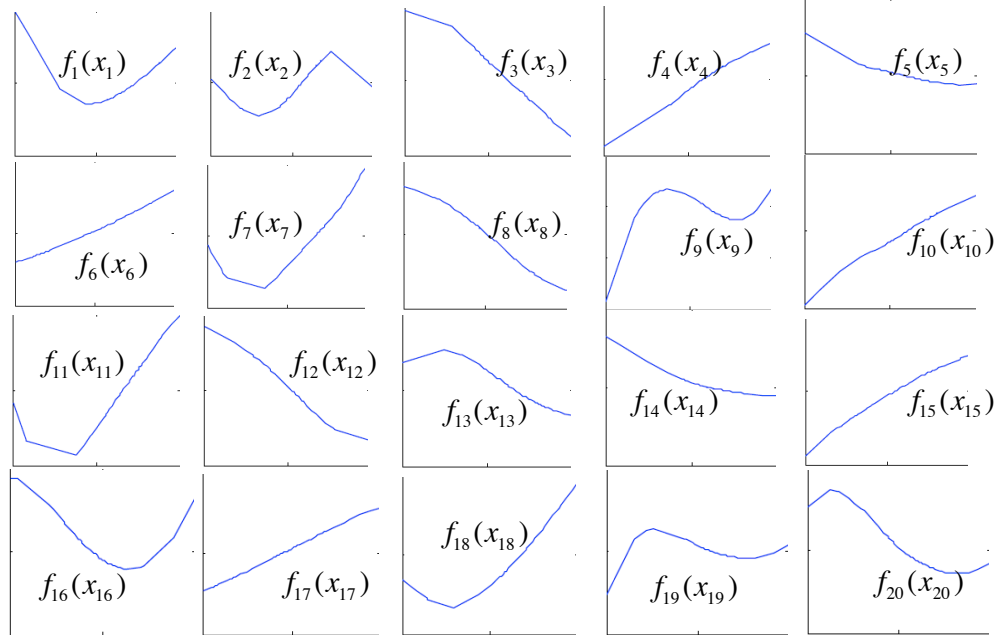


Figure 48.: The 25 estimated component functions using the MT-SpAM

higher than hand-crafted features. It is exciting that the MT-SpAM can often meet or exceed the performance of the hand-crafted features *using far fewer assumptions about neuroscience*.

Moreover, we show one sample of 25 basis words learned by MT-SpAM in Table 7. It is easy to see relationships between many of the words in the basis set and the 60 stimulus words in Table 5. For example, the model learned hotel and accommodation as basis words which are closely related to the *building* category (apartment, house, etc.) in 60 stimulus words. The basis word floor is



a part of building as stimulus words (door, window, etc.); basis word facilities is a general concept of the words (hammer, pilers, etc.); basis word shipping is highly correlated with the *transportation* category (airplane, train, etc.); basis word bedroom clearly refers to words in the *furniture* category (bed, dresser, etc.).

|               |         |         |             |             |
|---------------|---------|---------|-------------|-------------|
| accommodation | areas   | bedroom | bits        | built       |
| chairs        | checked | cut     | eye         | facilities  |
| floor         | garage  | green   | hotel       | maintenance |
| metal         | oil     | orange  | residential | shipping    |
| soft          | spaces  | stick   | thin        | usually     |

Table 7.: An example of 25 learned semantic basis words.

## 6.8 CONCLUSIONS

We have presented new approaches to fitting sparse nonparametric multi-task regression models and sparse multi-category classification models. The usefulness of these methods have been demonstrated on applications from genomics and cognitive neuroscience. A possible direction for future work is to develop procedures for automatic bandwidth selection in each dimension. We have used plug-in bandwidths and truncation dimensions  $q_n$  in our experiments and theory. It is of particular interest to develop procedures that are adaptive to different levels of smoothness in different dimensions. It would also be of interest is to consider more general penalties of the form  $p_\lambda(\|f_j\|)$ , as in [Fan and Li \[2001\]](#).

Finally, we note that while we have considered basic additive models that allow functions of individual variables, it is natural to consider interactions, as in the functional ANOVA model. One challenge is to formulate suitable incoherence conditions on the functions that enable regularization based procedures or greedy algorithms to recover the correct interaction graph. In the parametric setting, one result in this direction is [Wainwright et al. \[2007\]](#).

## 6.9 APPENDIX: TECHNICAL PROOFS

The proof of Theorem [6.2](#) has appeared in [Ravikumar et al. \[2009a\]](#), we omit it in this thesis.

*Proof of Theorem 6.3.* We begin with some notation. If  $\mathcal{M}$  is a class of functions then the  $L_\infty$  bracketing number  $N_{[]}(\epsilon, \mathcal{M})$  is defined as the smallest number of pairs  $B = \{(\ell_1, u_1), \dots, (\ell_k, u_k)\}$  such that  $\|u_j - \ell_j\|_\infty \leq \epsilon$ ,  $1 \leq j \leq k$ , and such that for every  $m \in \mathcal{M}$  there exists  $(\ell, u) \in B$  such that  $\ell \leq m \leq u$ . For the Sobolev space  $\mathcal{T}_j$ ,

$$\log N_{[]}(\epsilon, \mathcal{T}_j) \leq K \left( \frac{1}{\epsilon} \right)^{1/2} \quad (6.122)$$

for some  $K > 0$ . The bracketing integral is defined to be

$$J_{[]}(\delta, \mathcal{M}) = \int_0^\delta \sqrt{\log N_{[]}(\epsilon, \mathcal{M})} d\epsilon. \quad (6.123)$$

From Corollary 19.35 of [van der Vaart \[1998\]](#),

$$\mathbb{E} \left( \sup_{g \in \mathcal{M}} |\hat{\mu}(g) - \mu(g)| \right) \leq \frac{C J_{[]}(\|F\|_\infty, \mathcal{M})}{\sqrt{n}} \quad (6.124)$$

for some  $C > 0$ , where  $F(x) = \sup_{g \in \mathcal{M}} |g(x)|$ ,  $\mu(g) = \mathbb{E}(g(X))$  and  $\hat{\mu}(g) = n^{-1} \sum_{i=1}^n g(X_i)$ .

Set  $Z \equiv (Z_0, \dots, Z_d) = (Y, X_1, \dots, X_d)$  and note that

$$R(\beta, g) = \sum_{j=0}^d \sum_{k=0}^d \beta_j \beta_k \mathbb{E}(g_j(Z_j) g_k(Z_k)) \quad (6.125)$$

where we define  $g_0(z_0) = z_0$  and  $\beta_0 = -1$ . Also define

$$\hat{R}(\beta, g) = \frac{1}{n} \sum_{i=1}^n \sum_{j=0}^d \sum_{k=0}^d \beta_j \beta_k g_j(Z_j^{(i)}) g_k(Z_k^{(i)}). \quad (6.126)$$

Hence  $\hat{m}_n$  is the minimizer of  $\hat{R}(\beta, g)$  subject to the constraint  $\sum_j \beta_j g_j(x_j) \in \mathcal{M}_n(L_n)$  and  $g_j \in \mathcal{T}_j$ . For all  $(\beta, g)$ ,

$$|\hat{R}(\beta, g) - R(\beta, g)| \leq \|\beta\|_1^2 \max_{jk} \sup_{g_j \in \mathcal{T}_j, g_k \in \mathcal{T}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| \quad (6.127)$$

where  $\hat{\mu}_{jk}(g) = n^{-1} \sum_{i=1}^n \sum_{jk} g_j(Z_j^{(i)}) g_k(Z_k^{(i)})$  and  $\mu_{jk}(g) = \mathbb{E}(g_j(Z_j) g_k(Z_k))$ . From (6.122) it follows that

$$\log N_{[]}(\epsilon, \mathcal{M}_n) \leq 2 \log d_n + K \left( \frac{1}{\epsilon} \right)^{1/2}. \quad (6.128)$$

Hence,  $J_{[]}(\epsilon, \mathcal{M}_n) = O(\sqrt{\log d_n})$  and it follows from (6.124) and Markov's inequality that

$$\max_{jk} \sup_{g_j \in \mathcal{T}_j, g_k \in \mathcal{T}_k} |\hat{\mu}_{jk}(g) - \mu_{jk}(g)| = O_P \left( \sqrt{\frac{\log d_n}{n}} \right) = O_P \left( \frac{1}{n^{(1-\xi)/2}} \right). \quad (6.129)$$

We conclude that

$$\sup_{g \in \mathcal{M}} |\hat{R}(g) - R(g)| = O_P \left( \frac{L_n^2}{n^{(1-\xi)/2}} \right). \quad (6.130)$$

Therefore,

$$\begin{aligned} R(m^*) &\leq R(\hat{m}_n) \leq \hat{R}(\hat{m}_n) + O_P \left( \frac{L_n^2}{n^{(1-\xi)/2}} \right) \\ &\leq \hat{R}(m^*) + O_P \left( \frac{L_n^2}{n^{(1-\xi)/2}} \right) \leq R(m^*) + O_P \left( \frac{L_n^2}{n^{(1-\xi)/2}} \right) \end{aligned}$$

and the conclusion follows.  $\square$



This chapter studies the forward greedy strategy in sparse nonparametric regression. For additive models, we propose an algorithm called additive forward regression; for general multivariate models, we propose an algorithm called generalized forward regression. Both algorithms simultaneously conduct estimation and variable selection in nonparametric settings for the high dimensional sparse learning problem. Our main emphasis is empirical: on both simulated and real data, these two simple greedy methods can clearly outperform several state-of-the-art competitors, including LASSO, the sparse additive model (SpAM) we introduced in the previous chapter, and a recently proposed adaptive parametric forward-backward algorithm called Foba. We also provide some theoretical justifications of specific versions of the additive forward regression.

## 7.1 INTRODUCTION

At present, there are two major approaches to fit sparse linear models: *convex regularization* and *greedy pursuit*. The convex regularization approach regularizes the model by adding a sparsity constraint, leading to methods like LASSO [Tibshirani, 1996, Chen et al., 1998] or the Dantzig selector [Candes and Tao, 2007]. The greedy pursuit approach regularizes the model by iteratively selecting the current optimal approximation according to some criteria, leading to methods like the matching pursuit [Mallat and Zhang, 1993] or orthogonal matching pursuit (OMP) [Tropp, 2004].

As we have explained in the previous chapter, substantial progress has been made recently on applying the convex regularization idea to fit sparse additive models. For splines, Lin and Zhang [2006] propose a method called COSSO, which uses the sum of reproducing kernel Hilbert space norms as a sparsity inducing penalty, and can simultaneously conduct estimation and variable selection; Ravikumar et al. [2007, 2009a] develop a method called SpAM. The population version of SpAM can be viewed as a least squares problem penalized by the sum of  $L_2(P)$ -norms; Meier et al. [2009] develop a similar method using a different sparsity-smoothness penalty, which guarantees the solution to be a spline. All these methods can be viewed as different nonparametric variants of LASSO. They have similar drawbacks: (i) it is hard

to extend them to handle general multivariate regression where the mean functions are no longer additive; (ii) due to the large bias induced by the regularization penalty, the model estimation is suboptimal. One way to avoid this is to resort to two-stage procedures as in [Liu and Zhang \[2009\]](#), but the method becomes less robust due to the inclusion of an extra tuning parameter in the first stage.

In contrast to the convex regularization methods, greedy pursuit approaches do not suffer from such problems. Instead of trying to formulate the whole learning task into a global convex optimization, the greedy pursuit approaches adopt iterative algorithms with a local view. During each iteration, only a small number of variables are actually involved in the model fitting so that the whole inference only involves low dimensional models. Thus they naturally extend to the general multivariate regression and do not induce large estimation bias, which makes them especially suitable for high dimensional nonparametric inference. However, the greedy pursuit approaches do not attract as much attention as the convex regularization approaches in the nonparametric literature. For additive models, the only work we know of are the sparse boosting [[Bühlmann and Yu, 2006](#)] and multivariate adaptive regression splines (MARS) [[Friedman, 1991](#)]. These methods mainly target on additive models or lower-order functional ANOVA models, but without much theoretical analysis. For general multivariate regression, the only available method we are aware of is rodeo [[Lafferty and Wasserman, 2008](#)]. However, rodeo requires the total number of variables to be no larger than a double-logarithmic of the data sample size, and does not explicitly identify relevant variables.

In this chapter, we propose two greedy algorithms for sparse nonparametric learning in high dimensions. By extending the idea of the orthogonal matching pursuit to nonparametric settings, the main contributions of our work include: (i) we formulate two greedy nonparametric algorithms: additive forward regression (AFR) for sparse additive models and generalized forward regression (GFR) for general multivariate regression models. Both of them can simultaneously conduct estimation and variable selection in high dimensions. Additive forward regression can be viewed as a slight variant of the sparse boosting method of [Bühlmann and Yu \[2006\]](#). (ii) We present theoretical results for AFR using specific smoothers. (iii) We report thorough numerical results on both simulated and real-world datasets to demonstrate the superior performance of these two methods over the state-of-the-art competitors, including LASSO, SpAM, and an adaptive parametric forward-backward algorithm called Foba [[Zhang, 2008](#)].

The rest of this chapter is organized as follows: in the next section we review the basic problem formulation and notations. In [Section 7.3](#) we present the AFR algorithm, in [section 7.4](#), we present the GFR algorithm. Some theoretical

results are given in Section 7.5. In Section 7.6 we present numerical results on both simulated and real datasets, followed by a concluding section at the end.

## 7.2 SPARSE NONPARAMETRIC LEARNING IN HIGH DIMENSIONS

We begin by introducing some notation. Assuming  $n$  data points

$$\mathcal{D}_n = \left\{ (X^{(i)}, Y^{(i)}) \right\}_{i=1}^n$$

are observed from a high dimensional regression model

$$Y^{(i)} = m(X^{(i)}) + \epsilon^{(i)}, \quad \epsilon^{(i)} \sim N(0, \sigma^2) \quad i = 1, \dots, n, \quad (7.1)$$

where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T \in \mathbb{R}^d$  is a  $d$ -dimensional design point,  $m : \mathbb{R}^d \rightarrow \mathbb{R}$  is an unknown smooth mean function. Here we assume  $m$  lies in a  $d$ -dimensional second order Sobolev ball with finite radius. In the sequel, we denote the response vector  $(Y^{(1)}, \dots, Y^{(n)})^T$  by  $Y$  and the vector  $(X_j^{(1)}, \dots, X_j^{(n)})^T$  by  $X_j$  for  $1 \leq j \leq d$ .

We assume  $m$  is *functional sparse*, i.e. there exists an index set  $S \subset \{1, \dots, d\}$ , such that

$$(\text{General}) \quad m(x) = m(x_S), \quad (7.2)$$

where  $|S| = r \ll d$  and  $x_S$  denotes the sub-vector of  $x$  with elements indexed by  $S$ .

Sometimes, the function  $m$  can be assumed to have more structures to obtain a better estimation result. The most popular one is additivity assumption [Hastie and Tibshirani \[1999\]](#). In this case,  $m$  decomposes into the sum of  $r$  univariate functions  $\{m_j\}_{j \in S}$ :

$$(\text{Additive}) \quad m(x) = \alpha + \sum_{j \in S} m_j(x_j), \quad (7.3)$$

where each component function  $m_j$  is assumed to lie in a second order Sobolev ball with finite radius so that each element in the space is smooth enough. For the sake of identifiability, we also assume  $\mathbb{E}m_j(X_j) = 0$  for  $j = 1, \dots, d$ , where the expectation is taken with respect to the marginal distribution of  $X_j$ .

Given the models in (7.2) or (7.3), we have two tasks: *function estimation* and *variable selection*. For the first task, we try to find an estimate  $\hat{m}$ , such that  $\|\hat{m} - m\| \rightarrow 0$  as  $n$  goes to infinity, where  $\|\cdot\|$  is some function norm. For the second task, we try to find an estimate  $\hat{S}$ , which is an index set of variables, such that  $\mathbb{P}(S \subset \hat{S}) \rightarrow 1$  as  $n$  goes to infinity.

## 7.3 ADDITIVE FORWARD REGRESSION

In this section, we assume the true model is additive, i.e.  $m(x) = \alpha + \sum_{j \in S} m_j(x_j)$ . In general, if the true index set for the relevant variables is known, the backfitting algorithm can be directly applied to estimate  $\hat{m}$  [Hastie and Tibshirani, 1999]. It is essentially a Gauss-Seidel iteration for solving a set of normal equations in a function space. Within each iteration, it only estimates the smooth univariate function for one variable while holding all the others fixed, then cycles through the next variable. In particular, we denote the estimates on the  $j$ th variable  $X_j$  to be  $\hat{m}_j \equiv (\hat{m}_j(X_j^{(1)}), \dots, \hat{m}_j(X_j^{(n)}))^T \in \mathbb{R}^n$ . Then  $\hat{m}_j$  can be estimated by regressing the partial residual vector  $R_j = Y - \alpha - \sum_{k \neq j} \hat{m}_k$  on the variable  $X_j$ . This can be calculated by  $\hat{m}_j = \mathcal{S}_j R_j$ , where  $\mathcal{S}_j : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a smoothing matrix, which only depends on  $X^{(1)}, \dots, X^{(n)}$  but not on  $Y$ . Once  $\hat{m}_j$  is updated, the algorithm holds it fixed and repeats this process by cycling through each variable until convergence. Under mild conditions on the smoothing matrices  $\mathcal{S}_1, \dots, \mathcal{S}_d$ , the backfitting algorithm is a first order algorithm that guarantees to converge [Buja et al., 1989] and achieves the minimax rate of convergence as if only estimating a univariate function. However, for sparse learning problems, since the true index set is unknown, the backfitting algorithm no longer works due to the uncontrolled estimation variance.

By extending the idea of the orthogonal matching pursuit to sparse additive models, we design a forward greedy algorithm called the *additive forward regression* (AFR), which only involves a few variables in each iteration. Under this framework, we only need to conduct the backfitting algorithm on a small set of variables. Thus the variance can be well controlled. The algorithm is described in Figure 49, where we use  $\langle \cdot, \cdot \rangle_n$  to denote the inner product of two vectors.

The algorithm uses an active set  $\mathcal{A}$  to index the variables included in the model during each iteration and then performs a full optimization over all “active” variables via the backfitting algorithm. The main advantage of this algorithm is that during each iteration, the model inference is conducted in low dimensions and thus avoids the curse of dimensionality. The stopping criterion is controlled by a predefined parameter  $\eta$  which is equivalent to the regularization tuning parameter in convex regularization methods. Other stopping criteria, such as the maximum number of steps, can also be adopted. In practice, we always recommend to use data-dependent technique, such as cross-validation, to automatically tune this parameter.

Moreover, the smoothing matrix  $\mathcal{S}_j$  can be fairly general, e.g. univariate local linear smoothers as described below, kernel smoothers or spline smoothers [Wahba, 1990], etc.



**Input:**  $\{(X^{(i)}, Y^{(i)})\}_{i=1}^n$  and  $\eta > 0$   
 let  $\mathcal{A}^{(0)} = \emptyset$ ,  $\alpha = \sum_{i=1}^n Y^{(i)} / n$  and the residual  $R^{(0)} = Y - \alpha$   
**for**  $k = 1, 2, 3, \dots$   
   for each  $j \notin \mathcal{A}^{(k-1)}$ , estimate  $\hat{m}_j$  by smoothing:  $\hat{m}_j = \mathcal{S}_j R^{(k-1)}$   
   let  $j^{(k)} = \arg \max_{j \notin \mathcal{A}^{(k-1)}} |\langle \hat{m}_j, R^{(k-1)} \rangle_n|$   
   let  $\mathcal{A}^{(k)} = \mathcal{A}^{(k-1)} \cup j^{(k)}$   
   estimate  $\mathcal{M}^{(k)} = \{m_j : j \in \mathcal{A}^{(k)}\}$  by the backfitting algorithm  
   compute the residual  $R^{(k)} = Y - \alpha - \sum_{m_j \in \mathcal{M}^{(k)}} m_j(X_j)$   
   **if**  $(\|R^{(k-1)}\|_2^2 - \|R^{(k)}\|_2^2) / n \leq \eta$   
      $k = k - 1$   
     **break**  
   **end if**  
**end for**  
**Output:** selected variables  $\mathcal{A}^{(k)}$  and estimated component functions

$$\mathcal{M}^{(k)} = \{m_j : j \in \mathcal{A}^{(k)}\}.$$

Figure 49.: The Additive Forward Regression Algorithm

#### 7.4 GENERALIZED FORWARD REGRESSION

This section only assume  $m(x)$  to be functional sparse, i.e.  $m(x) = m(x_S)$ , without restricting the model to be additive. In this case, to find a good estimate  $\hat{m}$  becomes more challenging.

To estimate the general multivariate mean function  $m(x)$ , one of the most popular methods is the local linear regression: given an evaluation point  $x = (x_1, \dots, x_d)^T$ , the estimate  $\hat{m}(x)$  is the solution  $\hat{\alpha}_x$  to the following locally kernel weighted least squares problem:

$$\min_{\alpha_x, \beta_x} \sum_{i=1}^n \left\{ Y^{(i)} - \alpha_x - \beta_x^T (X^{(i)} - x) \right\}^2 \prod_{j=1}^d K_{h_j}(X_j^{(i)} - x_j), \quad (7.4)$$

where  $K(\cdot)$  is a one dimensional kernel function and the kernel weight function in (7.4) is taken as a product kernel with the diagonal bandwidth matrix  $H^{1/2} = \text{diag}\{h_1, \dots, h_d\}$ . Such a problem can be re-casted as a standard

**Input:**  $\{(X^{(i)}, Y^{(i)})\}_{i=1}^n$  and  $\eta > 0$   
 let  $\mathcal{A}^{(0)} = \emptyset, \alpha = \sum_{i=1}^n Y^{(i)} / n$  and  $\delta^{(0)} = \sum_{i=1}^n (Y^{(i)} - \alpha)^2 / n$   
**for**  $k = 1, 2, 3, \dots$   
   let  $j^{(k)} = \arg \min_{j \notin \mathcal{A}^{(k-1)}} \sum_{i=1}^n \left( Y^{(i)} - \mathcal{S}(\mathcal{A}^{(k-1)} \cup j)_{X^{(i)}} Y \right)^2 / n$   
   let  $\mathcal{A}^{(k)} = \mathcal{A}^{(k-1)} \cup j^{(k)}$   
   let  $\delta^{(k)} = \sum_{i=1}^n \left( Y^{(i)} - \mathcal{S}(\mathcal{A}^{(k)})_{X^{(i)}} Y \right)^2 / n$   
   **if**  $(\delta^{(k-1)} - \delta^{(k)}) \leq \eta$   
      $k = k - 1$   
     **break**  
   **end if**  
**end for**  
**Output:** selected variables  $\mathcal{A}^{(k)}$  and local linear estimates  
 $(\mathcal{S}(\mathcal{A}^{(k)})_{X^{(1)}} Y, \dots, \mathcal{S}(\mathcal{A}^{(k)})_{X^{(n)}} Y).$

Figure 50.: The Generalized Forward Regression Algorithm

weighted least squares regression. Therefore a closed-form solution to the the local linear estimate can be explicitly given by

$$\hat{\alpha}_x = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x Y = \mathcal{S}_x Y,$$

where  $e_1 = (1, 0, \dots, 0)^T$  is the first canonical vector in  $\mathbb{R}^{d+1}$  and

$$W_x = \text{diag} \left\{ \prod_{j=1}^d K_{h_j}(X_j^{(1)} - x_j), \dots, \prod_{j=1}^d K_{h_j}(X_j^{(n)} - x_j) \right\}$$

and

$$X_x = \begin{pmatrix} 1 & (X^{(1)} - x)^T \\ \vdots & \vdots \\ 1 & (X^{(n)} - x)^T \end{pmatrix}. \quad (7.5)$$

Here,  $\mathcal{S}_x$  is the local linear smoothing matrix. Note that if we constrain  $\beta_x = 0$ , then the local linear estimate reduces to the kernel estimate. The pointwise rate of convergence of such an estimate has been characterized in [Fan and](#)

Gijbels [1996]:  $|\hat{m}(x) - m(x)|^2 = O_P(n^{-4/(4+d)})$ , which is extremely slow when  $d > 10$ .

To handle the large  $d$  case, we again extend the idea of the orthogonal matching pursuit to this setting. For an index subset  $\mathcal{A} \subset \{1, \dots, d\}$  and the evaluation point  $x$ , the local linear smoother restricted on  $\mathcal{A}$  is denoted as  $S(\mathcal{A})$  and

$$S(\mathcal{A})_x = e_1^T \left( X(\mathcal{A})_x^T W(\mathcal{A})_x X(\mathcal{A})_x \right)^{-1} X(\mathcal{A})_x^T W(\mathcal{A})_x,$$

where  $W(\mathcal{A})_x$  is a diagonal matrix whose diagonal entries are the product of univariate kernels over the set  $\mathcal{A}$  and  $X(\mathcal{A})_x$  is a submatrix of  $X_x$  that only contains the columns indexed by  $\mathcal{A}$ .

Given these definitions, the *generalized forward regression* (GFR) algorithm is described in Figure 50. Similar to AFR, GFR also uses an active set  $\mathcal{A}$  to index the variables included in the model. Such mechanism allows all the statistical inference to be conducted only in low-dimensional spaces. The GFR algorithm using the multivariate local linear smoother can be computationally heavy for very high dimensional problems. However, GFR is a generic framework and can be equipped with arbitrary multivariate smoothers, e.g. kernel/Nearest Neighbor/spline smoothers. These smoothers lead to much better scalability. The only reason we use the local linear smoother as an illustrative example in this paper is due to its popularity and potential advantage on correcting the boundary bias.

## 7.5 THEORETICAL PROPERTIES

In this section, we provide the theoretical properties of the additive forward regression estimates using the spline smoother. Due to the asymptotic equivalence of the spline smoother and the local linear smoother [Silverman, 1984], we deduce that these results should also hold for the local linear smoother. Our main result in Theorem 7.1 says when using the spline smoother with certain truncation rate to implement AFR algorithm, the resulting estimator is consistent with a certain rate. When the underlying true component functions do not go to zeroes too fast, we also achieve variable selection consistency. Our analysis relies heavily on Barron et al. [2008].

**Theorem 7.1.** *Assuming there exists some  $\xi > 0$  which can be arbitrarily large, such that  $p = O(n^\xi)$ . For  $\forall j \in \{1, \dots, d\}$ , we assume  $m_j$  lies in a second-order Sobolev ball with finite radius, and  $m = \alpha + \sum_{j=1}^d m_j$ . For the additive forward regression algorithm using the spline smoother with a truncation rate at  $n^{1/4}$ , after  $(n/\log n)^{1/2}$  steps, we obtain that*

$$\|m - \hat{m}\|^2 = O_P \left( \sqrt{\frac{\log n}{n}} \right). \quad (7.6)$$

Furthermore, if we also assume

$$\min_{j \in S} \|m_j\| = \Omega \left( \left( \frac{\log n}{n} \right)^{1/4} \right),$$

then  $\mathbb{P}(S \subset \widehat{S}) \rightarrow 1$  as  $n$  goes to infinity. Here,  $\widehat{S}$  is the index set for nonzero component functions in  $\widehat{m}$ .

The rate for  $\|\widehat{m} - m\|^2$  obtained from Theorem 7.1 is only  $O(n^{-1/2})$ , which is slower than the minimax rate  $O(n^{-4/5})$ . This is mainly an artifact of our analysis instead of a drawback of the additive forward regression algorithm. In fact, if we perform a basis expansion for each component function to first cast the problem to be a finite dimensional linear model with group structure, under some more stringent smallest eigenvalue conditions on the augmented design as in Zhang [2009], we can show that AFR using spline smoothers can actually achieves the minimax rate  $O(n^{-4/5})$  up to a logarithmic factor. A detailed treatment of this issue is beyond the scope of this chapter.

*Proof.* We first describe an algorithm named *group orthogonal greedy algorithm* (GOGA), which solves a noiseless function approximation problem in a direct-sum Hilbert space. AFR can then be viewed as an empirical realization of such an “ideal” algorithm.

GOGA is a group extension of the orthogonal greedy algorithm (OGA) in Barron et al. [2008]. For  $j = 1, \dots, d$ , let  $\mathcal{H}_j$  be a Hilbert space of continuous functions with a Hamel basis  $\mathcal{D}_j$ . Then for a function  $m$  in the direct-sum Hilbert space  $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2 + \dots + \mathcal{H}_d$ , we want to approximate  $m$  using the union of many truncated bases  $\mathcal{D} = \mathcal{D}'_1 \cup \dots \cup \mathcal{D}'_d$ , where for all  $j$ ,  $\mathcal{D}'_j \subset \mathcal{D}_j$ .

We equip an inner product  $\langle \cdot, \cdot \rangle$  on  $\mathcal{H}$ :  $\forall f, g \in \mathcal{H}$ ,  $\langle f, g \rangle = \int f(X)g(X)dP_X$  where  $P_X$  is the marginal distribution for  $X$ . Let  $\|\cdot\|$  be the norm induced by the inner product  $\langle \cdot, \cdot \rangle$  on  $\mathcal{H}$ . GOGA begins by setting  $m^{(0)} = 0$ , and then recursively defines the approximant  $m^{(k)}$  based on  $m^{(k-1)}$  and its residual  $r^{(k-1)} \equiv m - m^{(k-1)}$ . More specifically: we proceed as the following: define  $f_j^{(k)}$  to be the projection of  $r^{(k-1)}$  onto the truncated basis  $\mathcal{D}'_j$ , i.e.  $f_j^{(k)} = \arg \min_{g \in \mathcal{D}'_j} \|r^{(k-1)} - g\|^2$ . We calculate  $j^{(k)}$  as

$$j_*^{(k)} = \arg \max_j |\langle r^{(k-1)}, f_j^{(k)} \rangle| \quad (7.7)$$

$m^{(k)}$  can then be calculated by projecting  $m$  onto the additive function space generated by  $\mathcal{A}^{(k)} = \mathcal{D}'_{j^{(1)}} + \dots + \mathcal{D}'_{j^{(k)}}$ :

$$\widehat{m}^{(k)} = \arg \min_{g \in \text{span}(\mathcal{A}^{(k)})} \|m - g\|^2. \quad (7.8)$$

AFR using regression splines is exactly GOGA when there is no noise. For noisy samples, we replace the unknown function  $m$  by its  $n$ -dimensional output vector  $Y$ , and replace the inner product  $\langle \cdot, \cdot \rangle$  by  $\langle \cdot, \cdot \rangle_n$ , which is defined as

$$\langle f, g \rangle_n = \frac{1}{n} \sum_{i=1}^n f(X^{(i)})g(X^{(i)}). \quad (7.9)$$

The projection of the current residual vector onto each dictionary  $\mathcal{D}_j'$  is replaced by the corresponding nonparametric smoothers.

Considering any function  $m \in \mathcal{H}$ , we proceed in the same way as in [Barron et al. \[2008\]](#), but replacing the OGA arguments in their analysis by those of GOGA. The desired results of the theorem follow from a simple argument on bounding the random random covering number of spline spaces.  $\square$

## 7.6 EXPERIMENTAL RESULTS

In this section, we present numerical results for AFR and GFR applied to both synthetic and real data. The main conclusion is that, in many cases, their performance on both function estimation and variable selection can clearly outperform those of LASSO, Foba, and SpAM. For all the reported experiments, we use local linear smoothers to implement AFR and GFR. The results for other smoothers, such as smoothing splines, are similar. Note that different bandwidth parameters will have big effects on the performances of local linear smoothers. Our experiments simply use the plug-in bandwidths according to [Fan and Gijbels \[1996\]](#) and set the bandwidth for each variable to be the same. For AFR, the bandwidth  $h$  is set to be  $1.06n^{-1/5}$  and for GFR, the bandwidth is varying over each iteration such that  $h = 1.06n^{-1/(4+|\mathcal{A}|)}$ , where  $|\mathcal{A}|$  is the size of the current active set.

For an estimate  $\hat{m}$ , the estimation performance for the synthetic data is measured by the mean square error (MSE), which is defined as

$$\text{MSE}(\hat{m}) = \frac{1}{n} \sum_{i=1}^n \left( m(X^{(i)}) - \hat{m}(X^{(i)}) \right)^2.$$

For the real data, since we do not know the true function  $m(x)$ , we approximate the mean squared error using 5-fold cross-validation scores.

### 7.6.1 The Synthetic Data

For the synthetic data experiments, we consider the *compound symmetry* covariance structure of the design matrix  $X \in \mathbb{R}^{n \times p}$  with  $n = 400$  and  $d = 20$ . Each dimension  $X_j$  is generated according to

$$X_j = \frac{W_j + tU}{1+t}, \quad j = 1, \dots, d,$$

where  $W_1, \dots, W_d$  and  $U$  are i.i.d. sampled from  $\text{Uniform}(0,1)$ . Therefore the correlation between  $X_j$  and  $X_k$  is  $t^2/(1+t^2)$  for  $j \neq k$ . We assume the true regression functions have  $r = 4$  relevant variables:

$$Y = m(X) + \epsilon = m(X_1, \dots, X_4) + \epsilon. \quad (7.10)$$

To evaluate the variable selection performance of different methods, we generate 50 designs and 50 trials for each design. For each trial, we run the greedy forward algorithm  $r$  steps. If all the relevant variables are included in, the variable selection task for this trial is said to be successful. We report the mean and standard deviation of the success rate in variable selection for various correlation between covariates by varying the values of  $t$ .

We adopt some synthetic examples as in [Lin and Zhang \[2006\]](#) and define the following four functions:  $g_1(x) = x$ ,  $g_2(x) = (2x - 1)^2$ ,  $g_3(x) = \sin(2\pi x)/(2 - \sin(2\pi x))$ , and

$$g_4(x) = 0.1 \sin(2\pi x) + 0.2 \cos(2\pi x) + 0.3 \sin^2(2\pi x) + 0.4 \cos^3(2\pi x) + 0.5 \sin^3(2\pi x). \quad (7.11)$$

The following four regression models are studied. The first model is linear; the second is additive; the third and forth are more complicated nonlinear models with at least two way interactions:

$$\begin{aligned} (\text{Model1}) : \quad & Y^{(i)} = 2X_1^{(i)} + 3X_2^{(i)} + 4X_3^{(i)} + 5X_4^{(i)} + 2N(0,1), \quad \text{with } t = 1 ; \\ (\text{Model2}) : \quad & Y^{(i)} = 5g_1(X_1^{(i)}) + 3g_2(X_2^{(i)}) + 4g_3(X_3^{(i)}) + 6g_4(X_4^{(i)}) + 4N(0,1), \\ & \quad \text{with } t = 1 ; \\ (\text{Model3}) : \quad & Y^{(i)} = \exp(2X_1^{(i)}X_2^{(i)} + X_3^{(i)}) + 2X_4^{(i)} + N(0,1), \quad \text{with } t = 0.5 ; \\ (\text{Model4}) : \quad & Y^{(i)} = \sum_{j=1}^4 g_j(X_j^{(i)}) + g_1(X_3^{(i)}X_4^{(i)}) + g_2((X_1^{(i)} + X_3^{(i)})/2) \\ & \quad + g_3(X_1^{(i)}X_2^{(i)}) + N(0,1), \quad \text{with } t = 0.5. \end{aligned}$$

Compared with LASSO, Foba, and SpAM, the estimation performance using MSE as evaluation criterion is presented in Figure 51. And Table 8 shows the rate of success for variable selection of these models with different correlations controlled by  $t$ .

From Figure 51, we see that AFR and GFR methods provide very good estimates for the underlying true regression functions as compared to others. Firstly, LASSO and SpAM perform very poorly when the selected model is very sparse. This is because they are convex regularization based approaches: to obtain a very sparse model, they induce very large estimation bias. On the other hand, the greedy pursuit based methods like Foba, AFR and GFR do not suffer from such a problem. Secondly, when the true model is linear, all methods perform similarly. For the nonlinear true regression function, AFR, GFR and SpAM outperform LASSO and Foba. It is expectable since

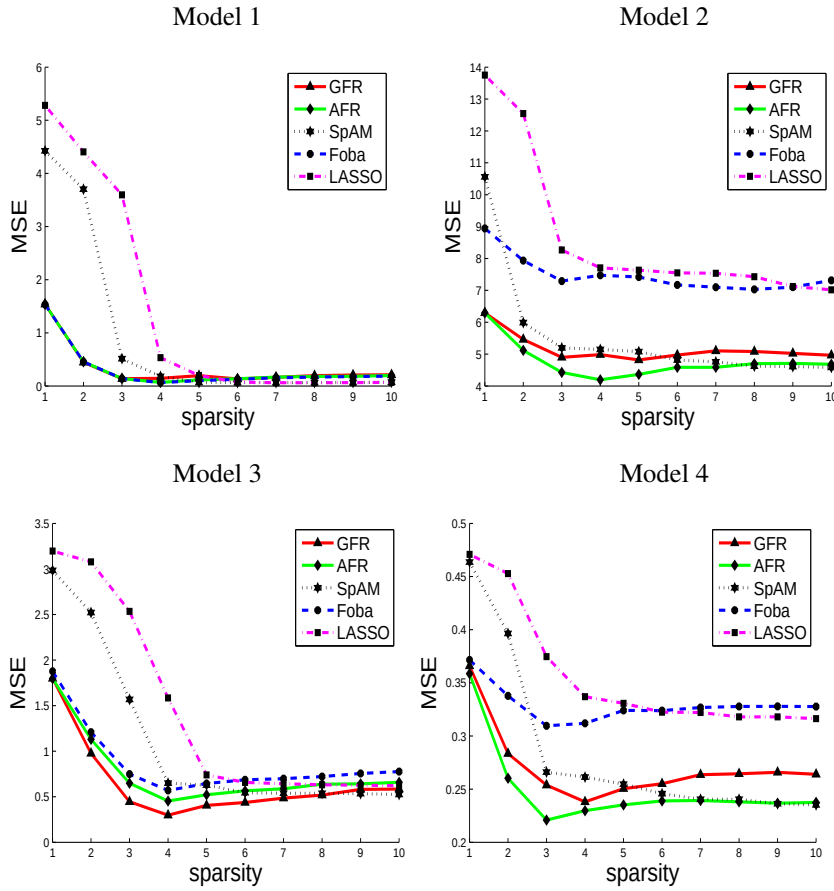


Figure 51.: Performance of the different algorithms on synthetic data: MSE versus sparsity level

LASSO and Foba are based on linear assumptions. Furthermore, we notice that when the true model is additive (Model 2) or nearly additive (Model 4), AFR performs the best. However, for the non-additive general multivariate regression function (Model 3), GFR performs the best. For all examples, when more and more irrelevant variables are included in the model, SpAM has a better generalization performance due to the regularization effect.

The variable selection performances of different methods in Table 8 are very similar to their estimation performances. We observe that, when correlation parameter  $t$  becomes larger, the performances of all methods decrease. But SpAM is most sensitive to the correlation increase. In all models, the performance of SpAM can decrease more than 70% for the larger  $t$ ; in contrast, AFR and GFR are more robust to the increased correlation between different covariates. Another interesting observation is on model 4. From the previous discussion, on this model, AFR achieves a better estimation performance. However, when comparing the variable selection performance, GFR is the best. This suggests that for nonparametric inference, the goals of estimation

Table 8.: Comparison of variable selection

| Model 1 LASSO(sd) | Foba           | SpAM           | AFR            | GFR            |                |
|-------------------|----------------|----------------|----------------|----------------|----------------|
| $t = 0$           | 1.000 (0.0000) | 1.000 (0.0000) | 0.999 (0.0028) | 0.999 (0.0039) | 0.990 (0.0229) |
| $t = 1$           | 0.879 (0.0667) | 0.882 (0.0557) | 0.683 (0.1805) | 0.879 (0.0525) | 0.839 (0.0707) |
| $t = 2$           | 0.559 (0.0913) | 0.553 (0.0777) | 0.190 (0.1815) | 0.564 (0.0739) | 0.515 (0.0869) |
| Model 2 LASSO(sd) | Foba           | SpAM           | AFR            | GFR            |                |
| $t = 0$           | 0.062 (0.0711) | 0.069 (0.0774) | 0.842 (0.1128) | 0.998 (0.0055) | 0.769 (0.1751) |
| $t = 1$           | 0.056 (0.0551) | 0.060 (0.0550) | 0.118 (0.0872) | 0.819 (0.1293) | 0.199 (0.2102) |
| $t = 2$           | 0.004 (0.0106) | 0.029 (0.0548) | 0.008 (0.0056) | 0.260 (0.1439) | 0.021 (0.0364) |
| Model 3 LASSO(sd) | Foba           | SpAM           | AFR            | GFR            |                |
| $t = 0$           | 0.997 (0.0080) | 0.999 (0.0039) | 0.980 (0.1400) | 1.000 (0.0000) | 1.000 (0.0000) |
| $t = 1$           | 0.818 (0.1137) | 0.802 (0.1006) | 0.934 (0.1799) | 1.000 (0.0000) | 0.995 (0.0103) |
| $t = 2$           | 0.522 (0.1520) | 0.391 (0.1577) | 0.395 (0.3107) | 0.902 (0.1009) | 0.845 (0.1623) |
| Model 4 LASSO(sd) | Foba           | SpAM           | AFR            | GFR            |                |
| $t = 0$           | 0.043 (0.0482) | 0.043 (0.0437) | 0.553 (0.1864) | 0.732 (0.1234) | 0.967 (0.0365) |
| $t = 0.5$         | 0.083 (0.0823) | 0.049 (0.0511) | 0.157 (0.1232) | 0.126 (0.0688) | 0.708 (0.1453) |
| $t = 1$           | 0.048 (0.0456) | 0.085 (0.0690) | 0.095 (0.0754) | 0.192 (0.0679) | 0.171 (0.1067) |

consistency and variable selection consistency might not be always coherent. Some tradeoffs might be needed to balance them.

### 7.6.2 The real data

In this subsection, we compare five methods on three real datasets: *Boston Housing*, *AutoMPG*, and *Ionosphere* data set <sup>1</sup>. *Boston Housing* contains 556 data points, with 13 features; *AutoMPG* 392 data points (we delete those with missing values), with 7 features and *Ionosphere* 351 data points, with 34 features and the binary output. We treat *Ionosphere* as a regression problem although the response is binary. We run 10 times 5-fold cross validation

<sup>1</sup> Available from *UCI Machine Learning Database Repository*: <http://archive.ics.uci.edu/ml>.



on each dataset and plot the mean and standard deviation of MSE versus different sparsity levels in Figure 52.

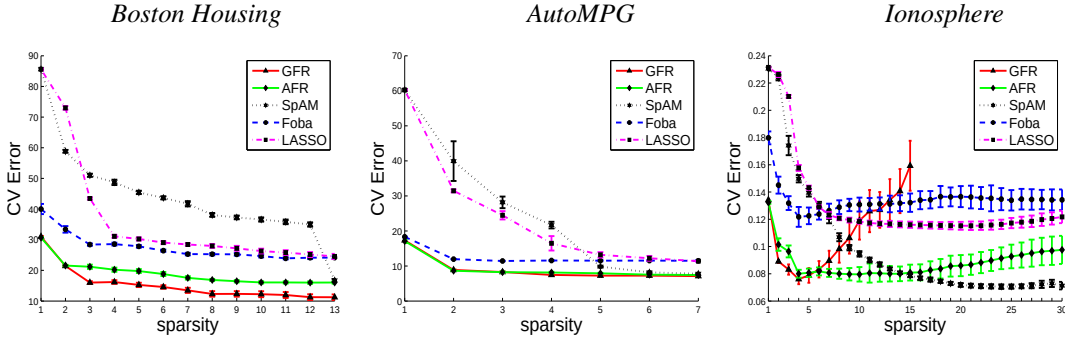


Figure 52.: Performance of the different algorithms on real datasets: CV error versus sparsity level

From Figure 52, since all the error bars are tiny, we deem all the results significant. On the Boston Housing and AutoMPG datasets, the generalization performances of AFR and GFR are clearly better than LASSO, Foba, and SpAM. For all these datasets, if we prefer very sparse models, the performance of the greedy methods are much better than the convex regularization methods due to the much less bias being induced. On the Ionosphere data, we only need to run GFR up to 15 selected variables, since the generalization performance with 15 variables is already worse than the null model due to the curse of dimensionality. Both AFR and GFR on this dataset achieve the best performances when there are no more than 10 variables included; while SpAM achieves the best CV score with 25 variables. However, this is not to say that the true model is not sparse. The main reason that SpAM can achieve good generalization performance when many variables included is due to its regularization effect. We think the true model should be sparse but not additive. Similar trend among different methods has also appeared in Model 4 of previous synthetic datasets.

## 7.7 CONCLUSIONS AND DISCUSSIONS

We presented two new greedy algorithms for nonparametric regression with either additive mean functions or general multivariate regression functions. Both methods utilize the iterative forward stepwise strategy, which guarantees the model inference is always conducted in low dimensions in each iteration. These algorithms are very easy to implement and have good empirical performance on both simulated and real datasets.

One thing worthy to note is: people sometimes criticize the forward greedy algorithms since they can never have the chance to correct the errors made in the early steps. This is especially true for high dimensional linear models,

which motivates the outcome of some adaptive forward-backward procedures such as Foba [Zhang, 2008]. We addressed a similar question: Whether a forward-backward procedure also helps in the nonparametric settings? AFR and GFR can be trivially extended to be forward-backward procedures using the same way as in Zhang [2008]. We conducted a comparative study to see whether the backward steps help or not. However, the backward step happens very rarely and the empirical performance is almost the same as the purely forward algorithm. This is very different from the linear model cases, where the backward step can be crucial. In summary, in the nonparametric settings, the backward ingredients will cost much more computational efforts with very tiny performance improvement.

A very recent research strand is to learn nonlinear models by the multiple kernel learning machinery Bach [2008a,b], another future work is to compare our methods with the multiple kernel learning approach from both theoretical and computational perspectives.

In this chapter, we propose a new nonparametric learning method based on multivariate dyadic regression trees (MDRTs). Unlike traditional dyadic decision trees (DDTs) or classification and regression trees (CARTs), MDRTs are constructed using penalized empirical risk minimization with a novel sparsity-inducing penalty. Theoretically, we show that MDRTs can simultaneously adapt to the unknown sparsity and smoothness of the true regression functions, and achieve the nearly optimal rates of convergence (in a minimax sense) for the class of  $(\alpha, C)$ -smooth functions. Empirically, MDRTs can simultaneously conduct function estimation and variable selection in high dimensions. To make MDRTs applicable for large-scale learning problems, we propose a greedy heuristic algorithm and a more effective randomization scheme. The superior performance of MDRTs are demonstrated on both synthetic and real datasets.

## 8.1 INTRODUCTION

Many application problems need to simultaneously predict several quantities using a common set of variables, e.g. predicting multi-channel signals within a time frame, predicting concentrations of several chemical constituents using the mass spectra of a sample, or predicting expression levels of many genes using a common set of phenotype variables. These problems can be naturally formulated in terms of multivariate regression.

In particular, let  $\{(X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)})\}$  be  $n$  independent and identically distributed pairs of data with  $X^{(i)} \in \mathcal{X} \subset \mathbb{R}^d$  and  $Y^{(i)} \in \mathcal{Y} \subset \mathbb{R}^p$  for  $i = 1, \dots, n$ . Moreover, we denote the  $j$ th dimension of  $Y$  by  $Y_j = (Y_j^{(1)}, \dots, Y_j^{(n)})^T$  and  $k$ th dimension of  $X$  by  $X_k = (X_k^{(1)}, \dots, X_k^{(n)})^T$ . Without loss of generality, we assume  $\mathcal{X} = [0, 1]^d$  and the true model on  $Y_j$  is :

$$Y_j^{(i)} = f_j(X^{(i)}) + \epsilon_j^{(i)}, \quad i = 1, \dots, n, \quad (8.1)$$

where  $f_j : \mathbb{R}^d \rightarrow \mathbb{R}$  is a smooth function. In the sequel, let  $f = (f_1, \dots, f_p)$ , where  $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$  is a  $p$ -valued smooth function. The vector form of (9.20) then becomes  $Y^{(i)} = f(X^{(i)}) + \epsilon^{(i)}$ ,  $i = 1, \dots, n$ . We also assume that the

noise terms  $\{\epsilon_j^{(i)}\}_{i,j}$  are independently distributed and bounded almost surely. This is a general setting of the nonparametric multivariate regression. From the minimax theory, we know that estimating  $f$  in high dimensions is very challenging. For example, when  $f_1, \dots, f_p$  lie in a  $d$ -dimensional Sobolev ball with order  $\alpha$  and radius  $C$ , denoted as  $W(\alpha, C)$ , the minimax risk is

$$\inf_{\hat{f}} \sup_{f_1, \dots, f_p \in W(\alpha, C)} R(\hat{f}, f), \quad (8.2)$$

where

$$R(\hat{f}, f) = \mathbb{E} \sum_{j=1}^p \int_{\mathcal{X}} |\hat{f}_j(X) - f(X)|^2 d\mu(X)$$

is the  $L_2$ -risk (w.r.t. the Lebesgue measure  $\mu(\cdot)$ ) of an estimate  $\hat{f}$  constructed from the observed samples. It is well known that the best convergence rate for the minmax risk (8.3) is  $p \cdot n^{-2\alpha/(2\alpha+d)}$ . For fixed  $\alpha$ , such rate can be extremely slow when  $d$  becomes large. For example, if  $f_1, \dots, f_p \in W(\alpha, C)$ , then

$$\liminf_{n \rightarrow \infty} \frac{1}{p} \cdot n^{2\alpha/(2\alpha+d)} \inf_{\hat{f}} \sup_{f_1, \dots, f_p \in W(\alpha, C)} R(\hat{f}, f) > 0, \quad (8.3)$$

In this chapter, wherever possible, we suppress the dependence of  $\hat{f}$  on  $n$ . Thus the best rate is of convergence is  $p \cdot n^{-2\alpha/(2\alpha+d)}$ . For fixed  $\alpha$ , such a rate is not practical when  $d$  is large.

However, in many real world applications, the true regression function  $f$  may depend only on a small set of variables. In other words, the problem is *jointly sparse*:

$$f(X) = f(X_S) = (f_1(X_S), \dots, f_p(X_S)),$$

where  $X_S = (X_k : k \in S)$ ,  $S \subset \{1, \dots, d\}$  is a subset of covariates with size  $r = |S| \ll d$ . If  $S$  has been given, the minimax lower bound can be improved to be  $p \cdot n^{-2\alpha/(2\alpha+r)}$ , which is the best possible rate can be expected. For sparse learning problems, our task is to develop an estimator, which adaptively achieves this faster rate of convergence without knowing  $S$  in advance.

Previous research on these problems can be roughly divided into three categories: (i) parametric linear models, (ii) nonparametric additive models, and (iii) nonparametric tree models. The methods in the first category assume that the true models are linear and use some block-norm regularization to induce jointly sparse solutions [Turlach et al., 2005, Liu and Zhang, 2009, Obozinski et al., 2009, Chen et al., 2009]. If the linear model assumptions are correct, accurate estimates can be obtained. However, given the increasing complexity of modern applications, conclusions inferred under these restrictive linear model assumptions can be misleading. As has been discussed in the previous chapters, significant progress has been made on inferring nonparametric

additive models with joint sparsity constraints [Friedman, 1991, Liu et al., 2008]. For additive models, each  $f_j(X)$  is assumed to have an additive form:  $f_j(X) = \sum_{k=1}^d f_{jk}(X_k)$ . Although they are more flexible than linear models, the additivity assumptions might still be too stringent for real world applications.

A family of more flexible nonparametric methods are based on tree models. One of the most popular tree methods is the classification and regression tree (CART) [Breiman et al., 1984]. It first grows a full tree by orthogonally splitting the axes at locally optimal splitting points, then prunes back the full tree to form a subtree. Theoretically, CART is hard to analyze unless strong assumptions have been enforced [Gey and Nadelec, 2005]. In contrast to CART, dyadic decision trees (DDTs) are restricted to only axis-orthogonal dyadic splits, i.e. each dimension can only be split at its midpoint. For a broad range of classification problems, Scott and Nowak [2006b] showed that DDTs using a special penalty can attain nearly optimal rate of convergence in a minimax sense. Blanchard et al. [2007b] proposed a dynamic programming algorithm for constructing DDTs when the penalty term has an additive form, i.e. the penalty of the tree can be written as the sum of penalties on all terminal nodes. Though intensively studied for classification problems, the dyadic decision tree idea has not drawn as much attention in the regression settings. One of the closest results we are aware of is Castro et al. [2005], in which a single response dyadic regression procedure is considered for non-sparse learning problems. Another interesting tree model, “Bayesian Additive Regression Trees (BART)”, is proposed under Bayesian framework [Chipman et al., 2006], which is essentially a “sum-of-trees” model. Most of the existing work adopt the number of terminal nodes as the penalty. Such penalty cannot lead to sparse models since a tree with a small number of terminal nodes might still involve too many variables.

To obtain sparse models, we propose a new nonparametric method based on multivariate dyadic regression trees (MDRTs). Similar to DDTs, MDRTs are also constructed using penalized empirical risk minimization. The novelty of MDRT is to introduce a sparsity-inducing term in the penalty, which explicitly induces very sparse solutions. Our contributions are two-fold: (i) Theoretically, we show that MDRTs can simultaneously adapt to the unknown sparsity and smoothness of the true regression functions, and achieve the nearly optimal rate of convergence for the class of  $(\alpha, C)$ -smooth functions. (ii) Empirically, to avoid computationally prohibitive exhaustive search in high dimensions, we propose a two-stage greedy algorithm and its randomized version that achieve good performance in both function estimation and variable selection. Note that our theory and algorithm can also be adapted to univariate sparse regression problem, which is a special case of the multivariate one. The reason why we propose MDRT is due to its generality. To the best of our knowledge, this is the first time such a sparsity-inducing penalty is equipped to tree models for solving sparse regression problems.

The rest of this chapter is organized as follows. Section 8.2 presents MDRTs in detail. Section 8.3 studies the statistical properties of MDRTs. Section 8.4 presents the algorithms which approximately compute the MDRT solutions. Section 8.5 reports empirical results of MDRTs and their comparison with CARTs. Conclusions are made in the last section.

## 8.2 MULTIVARIATE DYADIC REGRESSION TREES

We adopt the notation in Scott and Nowak [2006b]. A MDRT  $T$  is a multivariate regression tree that recursively divides the input space  $\mathcal{X}$  by means of axis-orthogonal dyadic splits. The nodes of  $T$  are associated with hyperrectangles (cells) in  $\mathcal{X} = [0, 1]^d$ . The root node corresponds to  $\mathcal{X}$  itself. If a node is associated to the cell  $B = \prod_{j=1}^d [a_j, b_j]$ , after being dyadically split on the dimension  $k$ , the two children are associated to the subcells  $B^{k,1}$  and  $B^{k,2}$ :

$$B^{k,1} = \left\{ X^{(i)} \in B \mid X_k^{(i)} \leq \frac{a_k + b_k}{2} \right\} \text{ and } B^{k,2} = B \setminus B^{k,1}.$$

The set of terminal nodes of a MDRT  $T$  is denoted as  $\text{term}(T)$ . Let  $B_t$  be the cell in  $\mathcal{X}$  induced by a terminal node  $t$ , the partition induced by  $\text{term}(T)$  can be denoted as  $\pi(T) = \{B_t \mid t \in \text{term}(T)\}$ .

For each terminal node  $t$ , we can fit a multivariate  $m$ -th order polynomial regression on data points falling in  $B_t$ . Instead of using all covariates, such a polynomial regression is only fitted on a set of active variables, which is denoted as  $\mathcal{A}(t)$ . For each node  $b \in T$  (not necessarily a terminal node),  $\mathcal{A}(b)$  can be an arbitrary subset of  $\{1, \dots, d\}$  satisfying two rules:

1. If a node is dyadically split perpendicular to the axis  $k$ ,  $k$  must belong to the active sets of its two children.
2. For any node  $b$ , let  $\text{par}(b)$  be its parent node, then  $\mathcal{A}(\text{par}(b)) \subset \mathcal{A}(b)$ .

For a MDRT  $T$ , we define  $\mathcal{F}_T^m$  to be the class of  $p$ -valued measurable  $m$ -th order polynomials corresponding to  $\pi(T)$ . Furthermore, for a dyadic integer  $N = 2^L$ , let  $\mathcal{T}_N$  be the collection of all MDRTs such that no terminal cell has a side length smaller than  $2^{-L}$ .

Given integers  $M$  and  $N$ , let  $\mathcal{F}^{M,N}$  be defined as

$$\mathcal{F}^{M,N} = \cup_{0 \leq m \leq M} \cup_{T \in \mathcal{T}_N} \mathcal{F}_T^m.$$

The final MDRT estimator with respect to  $\mathcal{F}^{M,N}$ , denoted as  $\hat{f}^{M,N}$ , can then be defined as

$$\hat{f}^{M,N} = \arg \min_{f \in \mathcal{F}^{M,N}} \frac{1}{n} \sum_{i=1}^n \|Y^{(i)} - f(X^{(i)})\|_2^2 + \text{pen}(f). \quad (8.4)$$

To define in detail  $\text{pen}(f)$  for  $f \in \mathcal{F}^{M,N}$ , let  $T$  and  $m$  be the MDRT and the order of polynomials corresponding to  $f$ ,  $\text{pen}(f)$  then takes the following form:

$$\text{pen}(f) = \lambda \cdot \frac{p}{n} (\log n(r_T + 1)^m (N_T + 1)^{r_T} + |\pi(T)| \log d), \quad (8.5)$$

where  $\lambda > 0$  is a regularization parameter,  $r_T = |\cup_{t \in \text{term}(T)} \mathcal{A}(t)|$  corresponds to the number of relevant dimensions and

$$N_T = \min\{s \in \{1, 2, \dots, N\} \mid T \in \mathcal{T}_s\}.$$

There are two terms in (8.5) within the parenthesis. The latter one penalizing the number of terminal nodes  $|\pi(T)|$  has been commonly adopted in the existing tree literature. The former one is novel. Intuitively, it penalizes non-sparse models since the number of relevant dimensions  $r_T$  appears in the exponent term. In the next section, we will show that this sparsity-inducing term is derived by bounding the VC-dimension of the underlying subgraph of regression functions. Thus it has a very intuitive interpretation.

### 8.3 STATISTICAL PROPERTIES

In this section, we present theoretical properties of the MDRT estimator. Our main technical result is Theorem 8.1, which provides the nearly optimal rate of the MDRT estimator.

To evaluate the algorithm performance, we use the  $L_2$ -risk with respect to the Lebesgue measure  $\mu(\cdot)$ , which is defined as

$$R(\hat{f}, f) = \mathbb{E} \sum_{j=1}^p \int_{\mathcal{X}} |\hat{f}_j(X) - f_j(X)|^2 d\mu(X),$$

where  $\hat{f}$  is the function estimate constructed from  $n$  observed samples. Note that all the constants appear in this section are generic constants, i.e. their values can change from one line to another in the analysis.

Let  $\mathbb{N}_0 = \{0, 1, \dots\}$  be the set of natural number, we first define the class of  $(\alpha, C)$ -smooth functions.

**Definition 8.1.**  *$(\alpha, C)$ -smoothness* Let  $\alpha = q + \beta$  for some  $q \in \mathbb{N}_0$ ,  $0 < \beta \leq 1$ , and let  $C > 0$ . A function  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  is called  $(\alpha, C)$ -smooth if for every  $\alpha = (\alpha_1, \dots, \alpha_d)$ ,  $\alpha_i \in \mathbb{N}_0$ ,  $\sum_{j=1}^d \alpha_j = q$ , the partial derivative  $\frac{\partial^q g}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}}$  exists and satisfies, for all  $X, Z \in \mathbb{R}^d$ ,

$$\left| \frac{\partial^q g(X)}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} - \frac{\partial^q g(Z)}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} \right| \leq C \cdot \|X - Z\|_2^\beta.$$

In the following, we denote the class of  $(\alpha, C)$ -smooth functions by  $\mathcal{D}(\alpha, C)$ .

**Assumption 8.1.** We assume  $f_1, \dots, f_p \in \mathcal{D}(\alpha, C)$  for some  $\alpha, C > 0$  and for all  $j \in \{1, \dots, p\}$ ,  $f_j(X) = f_j(X_S)$  with  $r = |S| \ll d$ .

Theorem 3.2 of Györfi et al. [2002] shows that the minimax rate of convergence for class  $\mathcal{D}(\alpha, C)$  is exactly the same as that for class of  $d$ -dimensional Sobolev ball with order  $\alpha$  and radius  $C$ .

**Proposition 8.1.** (Györfi et al. [2002])

$$\liminf_{n \rightarrow \infty} \frac{1}{p} \cdot n^{2\alpha/(2\alpha+d)} \inf_{\hat{f}} \sup_{f_1, \dots, f_p \in \mathcal{D}(\alpha, C)} R(\hat{f}, f) > 0.$$

Therefore, the minimax rate of convergence is  $p \cdot n^{-2\alpha/(2\alpha+d)}$ . Similarly, if the problem is jointly sparse with the index set  $S$  and  $r = |S| \ll d$ , the best rate of convergence can be improved to  $p \cdot n^{-2\alpha/(2\alpha+r)}$  when  $S$  is given.

The following is another technical assumption needed for the main theorem.

**Assumption 8.2.** Let  $1 \leq \gamma < \infty$ . We assume that

$$\max_{1 \leq j \leq p} \sup_X |f_j(X)| \leq \gamma \text{ and } \max_{1 \leq i \leq n} \|Y^{(i)}\|_\infty \leq \gamma \text{ a.s.}$$

This condition is mild. Indeed, we can even allow  $\gamma$  to increase with the sample size  $n$  at a certain rate. This will not affect the final result. For example, when  $\{\epsilon_j^{(i)}\}_{i,j}$  are i.i.d. Gaussian random variables, this assumption easily holds with  $\gamma_n = O(\sqrt{\log n})$ , which only contributes a logarithmic term to the final rate of convergence.

The next assumption specifies the scaling of the relevant dimension  $r$  and ambient dimension  $d$  with respect to the sample size  $n$ .

**Assumption 8.3.**  $r = O(1)$  and  $d = O(\exp(n^\xi))$  for some  $0 < \xi < 1$ .

Here,  $r = O(1)$  is crucial, since even if  $r$  increases at a logarithmic rate with respect to  $n$ , i.e.  $r = O(\log n)$ , it is hopeless to get any consistent estimator for the class  $\mathcal{D}(\alpha, C)$  since  $n^{-(1/\log n)} = 1/e$ . On the other hand, the ambient dimension  $d$  can increase exponentially fast with the sample size, which is a realistic scaling for high dimensional settings.

The following is the main theorem.

**Theorem 8.1.** Under Assumptions 9.1 to 9.3, there exist a positive number  $\lambda$  that only depends on  $\alpha, \gamma$  and  $r$ , such that

$$\text{pen}(f) = \lambda \cdot \frac{p}{n} \left( (\log n)(r_T + 1)^m (N_T + 1)^{r_T} + |\pi(T)| \log d \right), \quad (8.6)$$

For large enough  $M, N$ , the solution  $\hat{f}^{M,N}$  obtained from (8.4) satisfies

$$R(\hat{f}^{M,N}, f) \leq c \cdot p \cdot \left( \frac{\log n + \log d}{n} \right)^{2\alpha/(2\alpha+r)}, \quad (8.7)$$

where  $c$  is some generic constant.



**Remark 8.1.** As discussed in Proposition 8.1, the obtained rate of convergence in (8.7) is nearly optimal up to a logarithmic term.

**Remark 8.2.** Since the estimator defined in (8.4) does not need to know the smoothness  $\alpha$  and the sparsity level  $r$  in advance, MDRTs are simultaneously adaptive to the unknown smoothness and sparsity level.

**Proof of Theorem 1:** To find an upper bound of  $R(\hat{f}^{M,N}, f)$ , we need to analyze and control the approximation and estimation errors separately. Our analysis closely follows the least squares regression analysis in Györfi et al. [2002] and some specific coding scheme of trees in Scott and Nowak [2006b].

Without loss of generality, we always assume  $\hat{f}^{M,N}$  obtained from (8.4) satisfies the condition that  $\max_{1 \leq j \leq p} \sup_X |f_j^{M,N}(X)| \leq \gamma$ . if this is not true, we can always truncate  $\hat{f}^{M,N}$  at the rate  $\gamma$  and obtain the desired result in Theorem 8.1.

Let  $\mathcal{S}_T^m$  be the class of scalar-valued measurable  $m$ -th order polynomials corresponding to  $\pi(T)$ , and let  $\mathcal{G}_T^m$  be the class of all subgraphs of functions of  $\mathcal{S}_T^m$ , i.e.

$$\mathcal{G}_T^m = \left\{ (Z, t) \in \mathbb{R}^d \times \mathbb{R}; t \leq g(Z); g \in \mathcal{S}_T^m \right\}.$$

Let  $V_{\mathcal{G}_T^m}$  be the VC-dimension of  $\mathcal{G}_T^m$ , we have the following lemma:

**Lemma 8.1.** Let  $r_T$  and  $N_T$  be defined as in (8.5), we know that

$$V_{\mathcal{G}_T^m} \leq (r_T + 1)^m \cdot (N_T + 1)^{r_T}. \quad (8.8)$$

**Sketch of Proof:** From Theorem 9.5 of Györfi et al. [2002], we only need to show the dimension of  $\mathcal{G}_T^m$  is upper bounded by the R.H.S. of (8.8). By the definition of  $r_T$  and  $N_T$ , the result follows from a straightforward combinatorial analysis.  $\square$

The next lemma provides an upper bound of the approximation error for the class  $\mathcal{D}(\alpha, C)$ .

**Lemma 8.2.** Let  $f = (f_1, \dots, f_p)$  be the true regression function, there exists a set of piecewise polynomials  $h_1, \dots, h_p \in \cup_{T \in \mathcal{T}_K} \mathcal{S}_T^m$

$$\forall j \in \{1, \dots, p\}, \sup_{X \in \mathcal{X}} |f_j(X) - h_j(X)| \leq cK^{-\alpha}$$

where  $K \leq N$ ,  $c$  is a generic constant depends on  $r$ .

**Sketch of Proof:** This is a standard approximation result using multivariate piecewise polynomials. The main idea is based on a multivariate Taylor expansion of the function  $f_j$  at a given point  $X_0$ . Then try to utilize Definition 8.1 to bound the remainder terms. For the sake of brevity, we omit the technical details.  $\square$

The next lemma is crucial, it provides an oracle inequality to bound the risk using an approximation term and an estimation term. Its analysis follows from a simple adaptation of Theorem 12.1 on page 227 of Györfi et al. [2002].

First, we define

$$\tilde{R}(g, f) = \sum_{j=1}^p \int_{\mathcal{X}} |g_j(X) - f_j(X)|^2 d\mu(X).$$

**Lemma 8.3.** Györfi et al. [2002] Choose

$$\text{pen}(f) \geq 5136 \cdot p \frac{\gamma^4}{n} \left( \log(120e\gamma^4 n) V_{\mathcal{G}_T^m} + \frac{[[T]] \log 2}{2} \right) \quad (8.9)$$

for some prefix code  $[[T]] > 0$  satisfying  $\sum_{T \in \mathcal{T}_N} 2^{-[[T]]} \leq 1$ . Then, we have

$$R(\hat{f}^{M,N}, f) \leq 12840 \cdot p \cdot \frac{\gamma^4}{n} + 2 \inf_{T \in \mathcal{T}_N} \inf_{g \in \mathcal{F}^{M,N}} \left\{ p \cdot \text{pen}(g) + \tilde{R}(g, f) \right\} \quad (8.10)$$

One appropriate prefix code  $[[T]]$  for each MDRT  $T$  is proposed in Scott and Nowak [2006b], which specifies that  $[[T]] = 3|\pi(T)| - 1 + (|\pi(T)| - 1) \log d / \log 2$ . A simpler upper bound for  $[[T]]$  is

$$[[T]] \leq (3 + \log d / \log 2) |\pi(T)|. \quad (8.11)$$

**Remark 8.3.** The derived constants in the Lemma 8.3 will be pessimistic due to the very large numerical values. This may result in selecting oversimplified tree structures. In practice, we always use cross-validation to choose the tuning parameters.

To prove Theorem 8.1, first, using Assumption 9.1 and Lemma 8.2, we know that for any  $K \leq N$ , there must exists generic constants  $c_1, c_2, c_3$  and a function  $f'$  that is conformal with a MDRT  $T' \in \mathcal{T}_K$ , satisfying  $f'(X) = f'(X_S)$  and  $|\pi(T')| \leq (K+1)^r$  such that

$$\tilde{R}(f', f) \leq c_1 \cdot p \cdot K^{-2\alpha}, \quad (8.12)$$

and

$$\text{pen}(f') \leq c_2 \frac{(\log n)(r+1)^M (K+1)^r}{n} + c_3 \frac{\log d (K+1)^r}{n}. \quad (8.13)$$

The desired result then follows by plugging (8.12) and (8.13) into (8.10) and balancing these three terms.

#### 8.4 COMPUTATIONAL ALGORITHM

Exhaustive search of  $\hat{f}^{M,N}$  in the MDRT space has similar complexity as that of DDTs and could be computationally very expansive. To make MDRTs scalable for high dimensional massive datasets, using similar ideas as CARTs,

we propose a two-stage procedure: (1) we grow a full tree in a greedy manner; (2) we prune back the full tree to form the final tree. Before going to the detail of the algorithm, we firstly introduce some necessary notations.

Given a MDRT  $T$ , denote the corresponding multivariate  $m$ -th order polynomial fit on  $\pi(T)$  by  $\hat{f}_T^m = \{\hat{f}_t^m\}_{t \in \pi(T)}$ , where  $\hat{f}_t^m$  is the  $m$ -th order polynomial regression fit on the partition  $B_t$ . For each  $X^{(i)}$  falling in  $B_t$ , let  $\hat{f}_t^m(X^{(i)}, \mathcal{A}(t))$  be the predicted function value for  $X^{(i)}$ . We denote the local squared error (LSE) on node  $t$  by  $\hat{R}^m(t, \mathcal{A}(t))$ :

$$\hat{R}^m(t, \mathcal{A}(t)) = \frac{1}{n} \sum_{X^{(i)} \in B_t} \|Y^{(i)} - \hat{f}_t^m(X^{(i)}, \mathcal{A}(t))\|_2^2.$$

It is worthwhile noting that  $\hat{R}^m(t, \mathcal{A}(t))$  is calculated as the average with respect to the total sample size  $n$ , instead of the number of data points contained in  $B_t$ . The total MSE of the tree  $\hat{R}(T)$  can then be computed by the following equation:

$$\hat{R}(T) = \sum_{t \in \text{term}(T)} \hat{R}^m(t, \mathcal{A}(t)).$$

The total cost of  $T$ , which is defined as the right hand side of (8.4), then can be written as:

$$\hat{C}(T) = \hat{R}(T) + \text{pen}(\hat{f}_T^m). \quad (8.14)$$

Our goal is to find the tree structure with the polynomial regression on each terminal node that can minimize the total cost.

The first stage is *tree growing*, in which a terminal node  $t$  is first selected in each step. We then perform one of two actions a1 and a2:

- a1: adding another dimension  $k \notin \mathcal{A}(t)$  to  $\mathcal{A}(t)$ , and refit the regression model on all data points falling in  $B_t$ ;
- a2: dyadically splitting  $t$  perpendicular to the dimension  $k \in \mathcal{A}(t)$ .

In each tree growing step, we need to decide which action to perform. For action a1, we denote the drop in LSE as:

$$\Delta \hat{R}_1^m(t, k) = \hat{R}^m(t, \mathcal{A}(t)) - \hat{R}^m(t, \mathcal{A}(t) \cup \{k\}). \quad (8.15)$$

For action a2, let  $\text{sl}(t^{(k)})$  be the side length of  $B_t$  on dimension  $k \in \mathcal{A}(t)$ . If  $\text{sl}(t^{(k)}) > 2^{-L}$ , the dimension  $k$  of  $B_t$  can then be dyadically split. In this case, let  $t_L^{(k)}$  and  $t_R^{(k)}$  be the left and right child of node  $t$ . The drop in LSE takes the following form:

$$\Delta \hat{R}_2^m(t, k) = \hat{R}^m(t, \mathcal{A}(t)) - \hat{R}^m(t_L^{(k)}, \mathcal{A}(t)) - \hat{R}^m(t_R^{(k)}, \mathcal{A}(t)). \quad (8.16)$$

For each terminal node  $t$ , we greedily perform the action  $a^*$  on the dimension  $k^*$ , which are determined by

$$(a^*, k^*) = \underset{a \in \{1, 2\}, k \in \{1 \dots d\}}{\text{argmax}} \quad \Delta \hat{R}_a^m(t, k). \quad (8.17)$$

In high dimensional setting, the above greedy procedure may not lead to the optimal tree since successively locally optimal splits cannot guarantee the global optimum. Once an irrelevant dimension has been added in or split, the greedy procedure can never fix the mistake. To make the algorithm more robust, we propose a randomized scheme. Instead of greedily performing the action on the dimension that leads the maximum drop in LSE, we randomly choose which action to perform according to a multinomial distribution. In particular, we normalize  $\Delta\hat{R}$  such that:

$$\sum_{a=1}^2 \sum_k \Delta\hat{R}_a^m(t, k) = 1. \quad (8.18)$$

And a sample  $(a^*, k^*)$  is drawn from  $\text{multinomial}(1, \Delta\hat{R})$ . The action  $a^*$  is then performed on the dimension  $k^*$ . In general, when the randomized scheme is adopted, we need to repeat our algorithm many times to pick the best tree.

The second stage is *cost complexity pruning*. For each step, we either merge a pair of terminal nodes or remove a variable from the active set of a terminal node such that the resulted tree has the smaller cost. We repeat this process until the tree becomes a single root node with an empty active set. The tree with the minimum cost in this process is returned as the final tree.

The pseudocode for the growing stage and cost complexity pruning stage are presented in Appendix. Moreover, to avoid a cell with too few data points, we pre-define a quantity  $n_{\max}$ . Let  $n(t)$  be the the number of data points fall into  $B_t$ , if  $n(t) \leq n_{\max}$ ,  $B_t$  will no longer be split. It is worthwhile noting that we ignore those actions that lead to  $\Delta R = 0$ . In addition, whenever we perform the  $m$ th order polynomial regression on the active set of a node, we need to make sure it is not rank deficient.

## 8.5 EXPERIMENTAL RESULTS

In this section, we present numerical results for MDRTs applied to both synthetic and real datasets. We compare five methods:

- 1 Greedy MDRT with  $M = 1$  (MDRT(G, M=1));
- 2 Randomized MDRT with  $M = 1$  (MDRT(R, M=1));
- 3 Greedy MDRT with  $M = 0$  (MDRT(G, M=0));
- 4 Randomized MDRT with  $M = 0$  (MDRT(R, M=0));
- 5 CART: Classification and Regression Trees

For randomized scheme, we run 50 random trials and pick the minimum cost tree.

As for CART, we adopt the MATLAB package from [Martinez and Martinez \[2008\]](#), which fits piecewise constant on each terminal node with the cost complexity criterion:  $\hat{C}(T) = \hat{R}(T) + \rho \frac{p}{n} |\pi(T)|$ , where  $\rho$  is the tuning parameter playing the same role as  $\lambda$  in (8.5).

### 8.5.1 Synthetic Data

For the synthetic data experiment, we consider the high dimensional *compound symmetry* covariance structure of the design matrix with  $n = 200$  and  $d = 100$ . Each dimension  $X_j$  is generated according to

$$X_j = \frac{W_j + tU}{1 + t}, \quad j = 1, \dots, d,$$

where  $W_1, \dots, W_d$  and  $U$  are i.i.d. sampled from  $\text{Uniform}(0,1)$ . Therefore the correlation between  $X_j$  and  $X_k$  is  $t^2 / (1 + t^2)$  for  $j \neq k$ .

We study three models as shown below: the first one is linear; the second one is nonlinear but additive; the third one is nonlinear with three-way interactions. All these models only involve four relevant variables. The noise terms, denoted as  $\epsilon$ , are independently drawn from a standard normal distribution.

$$\begin{aligned} \text{Model 1: } Y_1^{(i)} &= 2X_1^{(i)} + 3X_2^{(i)} + 4X_3^{(i)} + 5X_4^{(i)} + \epsilon_1^{(i)} \\ Y_2^{(i)} &= 5X_1^{(i)} + 4X_2^{(i)} + 3X_3^{(i)} + 2X_4^{(i)} + \epsilon_2^{(i)} \\ \text{Model 2: } Y_1^{(i)} &= \exp(X_1^{(i)}) + (X_2^{(i)})^2 + 3X_3^{(i)} + 2X_4^{(i)} + \epsilon_1^{(i)} \\ Y_2^{(i)} &= (X_1^{(i)})^2 + 2X_2^{(i)} + \exp(X_3^{(i)}) + 3X_4^{(i)} + \epsilon_2^{(i)} \\ \text{Model 3: } Y_1^{(i)} &= \exp(2X_1^{(i)}X_2^{(i)} + X_3^{(i)}) + X_4^{(i)} + \epsilon_1^{(i)} \\ Y_2^{(i)} &= \sin(X_1^{(i)}X_2^{(i)}) + (X_3^{(i)})^2 + 2X_4^{(i)} + \epsilon_2^{(i)} \end{aligned}$$

We compare the performances of different methods using two criteria: (i) variable selection and (ii) function estimation. For each model, we generate 100 designs and an equal-sized validation set per design. For more detailed experiment protocols, we set  $n_{\max} = 5$  and  $L = 6$ . By varying the values of  $\lambda$  or  $\rho$  from large to small, we obtain a full regularization path. The tree with the minimum MSE on the validation set is then picked as the best tree. For criterion (i), if the variables involved in the best tree are exactly the first four variables, the variable selection task for this design is deemed as successful. The numerical results are presented in Table 9. For each method, the three quantities reported in order are the number of success out of 100 designs, the mean and standard deviation of the MSE on the validation set. Note that we omit “MDRT” in Table 9 due to space limitations.

From Table 9, the performance of MDRT with  $M = 1$  is dominantly better in both variable selection and estimation than those of the others. For linear

Table 9.: Comparison of Variable Selection and Function Estimation on Synthetic Datasets

| Model 1   | R, M=1          | G, M=1          | R, M=0          | G, M=0         | CART           |
|-----------|-----------------|-----------------|-----------------|----------------|----------------|
| $t = 0$   | 100 2.03 (0.14) | 100 2.08 (0.15) | 100 5.84 (0.51) | 97 5.74 (0.54) | 52 6.17 (0.55) |
| $t = 0.5$ | 100 2.05 (0.14) | 100 2.06 (0.15) | 76 5.42 (0.53)  | 68 5.36 (0.60) | 29 5.48 (0.51) |
| $t = 1$   | 100 2.05 (0.13) | 100 2.05 (0.16) | 19 5.40 (0.60)  | 20 5.56 (0.69) | 3 5.30 (0.58)  |

| Model 2   | R, M=1          | G, M=1          | R, M=0         | G, M=0         | CART           |
|-----------|-----------------|-----------------|----------------|----------------|----------------|
| $t = 0$   | 100 2.07 (0.13) | 100 2.06 (0.15) | 39 3.21 (0.26) | 31 3.22 (0.28) | 25 3.52 (0.31) |
| $t = 0.5$ | 96 2.05 (0.15)  | 93 2.09 (0.17)  | 17 3.10 (0.25) | 11 3.15 (0.26) | 5 3.20 (0.27)  |
| $t = 1$   | 76 2.09 (0.14)  | 68 2.21 (0.19)  | 2 3.17 (0.30)  | 2 3.16 (0.26)  | 1 3.16 (0.27)  |

| Model 3   | R, M=1         | G, M=1         | R, M=0         | G, M=0         | CART           |
|-----------|----------------|----------------|----------------|----------------|----------------|
| $t = 0$   | 98 2.68 (0.31) | 95 2.67 (0.47) | 75 3.90 (0.47) | 63 4.03 (0.54) | 29 4.35 (0.73) |
| $t = 0.5$ | 84 2.56 (0.21) | 86 2.52 (0.25) | 32 3.63 (0.47) | 32 3.60 (0.40) | 15 3.69 (0.38) |
| $t = 1$   | 65 2.51 (0.26) | 50 2.62 (0.23) | 3 3.75 (0.45)  | 4 3.88 (0.51)  | 2 3.66 (0.38)  |

models, MDRT with  $M = 1$  always select the correct variables even for large  $t$ s. For variable selection, MDRT with  $M = 0$  has a better performance compared with CART due to its sparsity-inducing penalty. In contrast, CART is more flexible in the sense that its splits are not necessarily dyadic. As a consequence, they are comparable in function estimation. Moreover, the performance of randomized scheme is slightly better than its deterministic version in variable selection. Another observation is that, when  $t$  becomes larger, although the performance of variable selection decreases on all methods, the estimation performance becomes slightly better. This might be counter-intuitive at the first sight. In fact, with the increase of  $t$ , all methods tend to select more variables. Due to the high correlations, even the irrelevant variables are also helpful in predicting the responses. This is a common effect due to “collinearity” or “concurvity”.

### 8.5.2 Real Data

In this subsection, we compare these methods on three real datasets. The first dataset is the *Chemometrics* data (Chem for short), which has been extensively studied in Breiman and Friedman [1997]. The data are from a simulation of a low density tubular polyethylene reactor with  $n = 56$ ,  $d = 22$  and  $p = 6$ . Following the same procedures in Breiman and Friedman [1997],

Table 10.: Testing MSE on Real Datasets

|          | R, M=1         | G, M=1         | R, M=0         | G, M=0         | CART           |
|----------|----------------|----------------|----------------|----------------|----------------|
| Chem     | 0.15 (0.09)    | 0.18 (0.12)    | 0.38 (0.18)    | 0.52 (0.06)    | 0.40 (0.09)    |
| Housing  | 20.18 (2.94)   | 21.60 (2.83)   | 24.67 (2.05)   | 29.46 (1.95)   | 25.91 (3.05)   |
| Space_ga | 0.054 (7.8e-4) | 0.055 (8.0e-4) | 0.068 (7.2e-4) | 0.068 (9.2e-4) | 0.064 (8.3e-4) |

we log-transformed the responses because they are skewed. The second dataset is Boston *Housing*<sup>1</sup> with  $n = 506$ ,  $d = 10$  and  $p = 1$ . We add 10 irrelevant variables randomly drawn from  $\text{Uniform}(0,1)$  to evaluate the variable selection performance. The third one, *Space\_ga*<sup>2</sup>, is an election data with spatial coordinates on 3107 US counties. Our task is to predict the  $x, y$  coordinates of each county given 5 variables regarding voting information. For *Space\_ga*, we normalize the responses to  $[0, 1]$ . Similarly, we add other 15 irrelevant variables randomly drawn from  $\text{Uniform}(0,1)$ . For all these datasets, we scale the input variables into a unit cube.

For evaluation purpose, each dataset is randomly split such that half data are used for training and the other half for testing. We run a 5-fold cross-validation on the training set to pick the best tuning parameter  $\lambda^*$  and  $\rho^*$ . We then train MDRTs and CART on the entire training data using  $\lambda^*$  and  $\rho^*$ . We repeat this process 20 times and report the mean and standard deviation of the testing MSE in Table 10.  $n_{\max}$  is set to be 5 for the first dataset and 20 for the latter two. For all datasets, we set  $L = 6$ . Moreover, for randomized scheme, we run 50 random trials and pick the minimum cost tree.

From Table 10, we see that MDRT with  $M = 1$  has the best estimation performance. Moreover, randomized scheme does improve the performance compared to the deterministic counterpart. In particular, such an improvement is quite significant when  $M = 0$ . The performance of MDRT(G, M=0) is always worse than CART since CART can have more flexible splits. However, using randomized scheme, the performance of MDRT(R, M=0) achieves a comparable performance as CART.

As for variable selection of Housing data, in all the 20 runs, MDRT(G, M=1) and MDRT(R, M=1) never select the artificially added variables. However, for the other three methods, nearly 10 out of 20 runs involve at least one extraneous variable. In particular, we compare our results with those reported in Ravikumar et al. [2007]. They find that there are 4 (indus, age, dis, tax) irrelevant variables in the Housing data. Our experiments confirm this result since in 15 out of the 20 trials, MDRT(G, M=1) and MDRT(R, M=1) never select these four variables. Similarly, for *Space\_ga* data, there are only 2 and 1

<sup>1</sup> Available from *UCI Machine Learning Database Repository*: <http://archive.ics.uci.edu/ml>

<sup>2</sup> Available from *StatLib*: <http://lib.stat.cmu.edu/datasets/>

times that MDRT(G, M=1) and MDRT(R, M=1) involve the artificially added variables.

## 8.6 CONCLUSIONS

We propose a novel sparse learning method based on multivariate dyadic regression trees (MDRTs). Our approach adopts a new sparsity-inducing penalty that simultaneously conduct function estimation and variable selection. Some theoretical analysis and practical algorithms have been developed. To the best of our knowledge, it is the first time that such a penalty is introduced in the tree literature for high dimensional sparse learning problems.

## 8.7 APPENDIX: PSEUDO-CODE FOR GREEDY TREE LEARNING ALGORITHMS

---

### Algorithm 8.7.1 Tree Growing

---

**Input:**  $\{X^{(i)}, Y^{(i)}\}_{i=1}^n, m, n_{\max}, L, \text{deterministic}$

Build the initial tree  $T$  with a single root node  $r$  containing all the data points and set  $\mathcal{A}(r) \leftarrow \emptyset$

**while**  $\exists t \in \text{term}(T)$  such that  $n(t) > n_{\max}$  or  $|\mathcal{A}(t)| < d$  **do**

**if**  $|\mathcal{A}(t)| < d$  **then**

**for all** dimension  $k \notin \mathcal{A}(t)$  **do**

            calculate  $\Delta \hat{R}_1^m(t, k)$  according to (8.15)

**if**  $n(t) > n_{\max}$  **then**

**for all** dimension  $k \in \mathcal{A}(t)$  **do**

**if**  $\text{sl}(t^{(k)}) \geq 2^{-L+1}$  **then**

                calculate  $\Delta \hat{R}_2^m(t, k)$  according to (8.16)

**if deterministic then**

$(a^*, k^*) = \underset{a \in \{1,2\}, k \in \{1 \dots d\}}{\operatorname{argmax}} \Delta \hat{R}_a^m(t, k)$

**else**

        Normalize  $\Delta \hat{R}$  according to (8.18). Draw the sample  $(a^*, k^*)$  from the multinomial(1,  $\Delta \hat{R}$ )

**if**  $a^* = 1$  **then**

        Dyadically split the cell represented by node  $t$  perpendicular to dimension  $k^*$  and update  $T$

**else**

$\mathcal{A}(t) \leftarrow \mathcal{A}(t) \cup \{k\}$

**Output:** Tree  $T$

---

Note the boolean variable *deterministic* indicates whether the procedure is purely greedy or randomized.



---

**Algorithm 8.7.2** Cost Complexity Pruning
 

---

**Input:** Tree  $T$ , parameter  $\lambda$  for calculating  $\widehat{C}(T)$ 
 $i \leftarrow 1, T_1 \leftarrow T$ 
**while**  $T_i$  has more than one node **OR**  $T_i$  only has the root node  $r$  with  $\mathcal{A}(r) \neq \emptyset$  **do**

$$T^{(1)} \leftarrow \underset{t_L, t_R \in \text{term}(T_i)}{\text{argmin}} \widehat{C}(\text{Tree obtained by merging } t_L, t_R \text{ in } T_i)$$

$$T^{(2)} \leftarrow \underset{t \in \text{term}(T_i), k \in \mathcal{A}(t) \setminus \mathcal{A}(\text{par}(t))}{\text{argmin}} \widehat{C}(\text{Tree obtained by removing the dimension } k \text{ from } \mathcal{A}(t))$$

$$T_{i+1} \leftarrow \underset{T^{(l)} \in \{1,2\}}{\text{argmin}} \widehat{C}(T^{(l)})$$
 $i \leftarrow i + 1$ 

$$i^* \leftarrow \underset{i}{\text{argmin}} \widehat{C}(T_i)$$
**Output:** Optimal Tree  $T_{i^*}$ 


---



Undirected graphical models encode in a graph  $G$  the dependence structure of a random vector  $Y$ . In many applications, it is of interest to model  $Y$  given another random vector  $X$  as input. We refer to the problem of estimating the graph  $G(x)$  of  $Y$  conditioned on  $X = x$  as “graph-valued regression.” In this chapter, we propose a semiparametric method for estimating  $G(x)$  that builds a tree on the  $X$  space just as in CART (classification and regression trees), but at each leaf of the tree estimates a graph. We call the method “Graph-optimized CART,” or Go-CART. We study the theoretical properties of Go-CART using dyadic partitioning trees, establishing oracle inequalities on risk minimization and tree partition consistency. We also demonstrate the application of Go-CART to a meteorological dataset, showing how graph-valued regression can provide a useful tool for analyzing complex data.

## 9.1 INTRODUCTION

Let  $Y$  be a  $p$ -dimensional random vector with distribution  $P$ . A common way to study the structure of  $P$  is to construct the undirected graph  $G = (V, E)$ , where the vertex set  $V$  corresponds to the  $p$  components of the vector  $Y$ . The edge set  $E$  is a subset of the pairs of vertices, where an edge between  $Y_j$  and  $Y_k$  is absent if and only if  $Y_j$  is conditionally independent of  $Y_k$  given all the other variables. Suppose now that  $Y$  and  $X$  are both random vectors, and let  $P(\cdot | X)$  denote the conditional distribution of  $Y$  given  $X$ . In a typical regression problem, we are interested in the conditional mean  $\mu(x) = \mathbb{E}(Y | X = x)$ . But if  $Y$  is multivariate, we may also be interested in how the structure of  $P(\cdot | X)$  varies as a function of  $X$ . In particular, let  $G(x)$  be the undirected graph corresponding to  $P(\cdot | X = x)$ . We refer to the problem of estimating  $G(x)$  as *graph-valued regression*.

Let  $\mathcal{G} = \{G(x) : x \in \mathcal{X}\}$  be a set of graphs indexed by  $x \in \mathcal{X}$ , where  $\mathcal{X}$  is the domain of  $X$ . Then  $\mathcal{G}$  induces a partition of  $\mathcal{X}$ , denoted as  $\mathcal{X}_1, \dots, \mathcal{X}_m$ , where  $X^{(1)}$  and  $x_2$  lie in the same partition element if and only if  $G(X^{(1)}) = G(x_2)$ . Graph-valued regression is thus the problem of estimating the partition and estimating the graph within each partition element.

We present three different partition-based graph estimators; two that use global optimization, and one based on a greedy splitting procedure. One of

the optimization based schemes uses penalized empirical risk minimization; the other uses held-out risk minimization. As we show, both methods enjoy strong theoretical properties under relatively weak assumptions; in particular, we establish oracle inequalities on the excess risk of the estimators, and tree partition consistency (under stronger assumptions) in Section 10.4. While the optimization based estimates are attractive, they do not scale well computationally when the input dimension is large. An alternative is to adapt the greedy algorithms of classical CART, as we describe in Section 9.3.3. In Section 10.5 we present experimental results on both synthetic data and a meteorological dataset, demonstrating how graph-valued regression can be an effective tool for analyzing high dimensional data with covariates.

## 9.2 GRAPH-VALUED REGRESSION

Let  $Y^{(1)}, \dots, Y^{(n)}$  be a random sample of vectors from  $P$ , where each  $Y^{(i)} \in \mathbb{R}^p$ . We are interested in the case where  $p$  is large and, in fact, may diverge with  $n$  asymptotically. One way to estimate  $G$  from the sample is the *graphical lasso* or *glasso* [Yuan and Lin, 2007, Friedman et al., 2007, Banerjee et al., 2008], where one assumes that  $P$  is Gaussian with mean  $\mu$  and covariance matrix  $\Sigma$ . Missing edges in the graph correspond to zero elements in the precision matrix  $\Omega = \Sigma^{-1}$  [Whittaker, 1990, Edwards, 1995, Lauritzen, 1996]. A sparse estimate of  $\Omega$  is obtained by solving

$$\hat{\Omega} = \arg \min_{\Omega \succ 0} \{ \text{tr}(S\Omega) - \log |\Omega| + \lambda \|\Omega\|_1 \} \quad (9.1)$$

where  $\Omega$  is positive definite,  $S$  is the sample covariance matrix, and  $\|\Omega\|_1 = \sum_{j,k} |\Omega_{jk}|$  is the elementwise  $\ell_1$ -norm of  $\Omega$ . Friedman et al. [2007] develop a efficient algorithm for finding  $\hat{\Omega}$  that involves estimating a single row (and column) of  $\Omega$  in each iteration by solving a lasso regression. The theoretical properties of  $\hat{\Omega}$  have been studied by Rothman et al. [2008] and Ravikumar et al. [2009b]. In practice, it seems that the glasso yields reasonable graph estimators even if  $Y$  is not Gaussian; however, proving conditions under which this happens is an open problem.

We briefly mention three different strategies for estimating  $G(x)$ , the graph of  $Y$  conditioned on  $X = x$ , each of which builds upon the glasso.

**Parametric Estimators.** Assume that  $Z = (X, Y)$  is jointly multivariate Gaussian with covariance matrix

$$\Sigma = \begin{pmatrix} \Sigma_X & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_Y \end{pmatrix}.$$

We can estimate  $\Sigma_X$ ,  $\Sigma_Y$ , and  $\Sigma_{XY}$  by their corresponding sample quantities  $\hat{\Sigma}_X$ ,  $\hat{\Sigma}_Y$ , and  $\hat{\Sigma}_{XY}$ , and the marginal precision matrix of  $X$ , denoted as  $\Omega_X$ , can

be estimated using the glasso. The conditional distribution of  $Y$  given  $X = x$  is obtained by standard Gaussian formulas. In particular, the conditional covariance matrix of  $Y | X$  is  $\hat{\Sigma}_{Y|X} = \hat{\Sigma}_Y - \hat{\Sigma}_{YX} \hat{\Omega}_X \hat{\Sigma}_{XY}$  and a sparse estimate of  $\hat{\Omega}_{Y|X}$  can be obtained by directly plugging  $\hat{\Sigma}_{Y|X}$  into glasso. However, the estimated graph does not vary with different values of  $X$ .

**Kernel Smoothing Estimators.** We assume that  $Y$  given  $X$  is Gaussian, but without making any assumption about the marginal distribution of  $X$ . Thus  $Y | X = x \sim N(\mu(x), \Sigma(x))$ . Under the assumption that both  $\mu(x)$  and  $\Sigma(x)$  are smooth functions of  $x$ , we estimate  $\Sigma(x)$  via kernel smoothing:

$$\hat{\Sigma}(x) = \frac{\sum_{i=1}^n K\left(\frac{\|x - X^{(i)}\|}{h}\right) (Y^{(i)} - \hat{\mu}(x)) (Y^{(i)} - \hat{\mu}(x))^T}{\sum_{i=1}^n K\left(\frac{\|x - X^{(i)}\|}{h}\right)}$$

where  $K$  is a kernel (e.g. the probability density function of the standard Gaussian distribution),  $\|\cdot\|$  is the Euclidean norm,  $h > 0$  is a bandwidth and

$$\hat{\mu}(x) = \frac{\sum_{i=1}^n K\left(\frac{\|x - X^{(i)}\|}{h}\right) Y^{(i)}}{\sum_{i=1}^n K\left(\frac{\|x - X^{(i)}\|}{h}\right)}.$$

Now we apply glasso in (9.1) with  $S = \hat{\Sigma}(x)$  to obtain an estimate of  $G(x)$ . This method is appealing because it is simple and very similar to nonparametric regression smoothing; the method was analyzed for one-dimensional  $X$  by Zhou et al. [2010]. However, while it is easy to estimate  $G(x)$  at any given  $x$ , it requires global smoothness of the mean and covariance functions. It is also computationally challenging to reconstruct the partition  $\mathcal{X}_1, \dots, \mathcal{X}_m$ .

**Partition Estimators.** In this approach, we partition  $\mathcal{X}$  into finitely many connected regions  $\mathcal{X}_1, \dots, \mathcal{X}_m$ . Within each  $\mathcal{X}_j$ , we apply the glasso to get an estimated graph  $\hat{G}_j$ . We then take  $\hat{G}(x) = \hat{G}_j$  for all  $x \in \mathcal{X}_j$ . To find the partition, we appeal to the idea used in CART (classification and regression trees) [Breiman et al., 1984]. We take the partition elements to be recursively defined hyperrectangles. As is well-known, we can then represent the partition by a tree, where each leaf node corresponds to a single partition element. In CART, the leaves are associated with the means within each partition element; while in our case, there will be an estimated undirected graph for each leaf node. We refer to this method as Graph-optimized CART, or Go-CART. The remainder of this paper is devoted to the details of this method.

### 9.3 GRAPH-OPTIMIZED CART

Let  $X \in \mathbb{R}^d$  and  $Y \in \mathbb{R}^p$  be two random vectors, and let

$$\mathcal{D}_n = \left\{ (X^{(1)}, Y^{(1)}), \dots, (X^{(n)}, Y^{(n)}) \right\}$$

be  $n$  i.i.d. samples from the joint distribution of  $(X, Y)$ . The domains of  $X$  and  $Y$  are denoted by  $\mathcal{X}$  and  $\mathcal{Y}$  respectively; and for simplicity we take  $\mathcal{X} = [0, 1]^d$ . We assume that

$$Y | X = x \sim N_p(\mu(x), \Sigma(x)) \quad (9.2)$$

where  $\mu : \mathbb{R}^d \rightarrow \mathbb{R}^p$  is a vector-valued mean function and  $\Sigma : \mathbb{R}^d \rightarrow \mathbb{R}^{p \times p}$  is a matrix-valued covariance function. We also assume that for each  $x$ ,  $\Omega(x) = \Sigma(x)^{-1}$  is a sparse matrix, i.e., many elements of  $\Omega(x)$  are zero. In addition,  $\Omega(x)$  may also be a sparse function of  $x$ , i.e.,  $\Omega(x) = \Omega(x_R)$  for some  $R \subset \{1, \dots, d\}$  with cardinality  $|R| \ll d$ . The task of graph-valued regression is to find a sparse inverse covariance  $\hat{\Omega}(x)$  to estimate  $\Omega(x)$  for any  $x \in \mathcal{X}$ ; in some situations the graph of  $\Omega(x)$  is of greater interest than the entries of  $\Omega(x)$  themselves.

Go-CART is a partition-based conditional graph estimator. We partition  $\mathcal{X}$  into finitely many connected regions  $\mathcal{X}_1, \dots, \mathcal{X}_m$ , and within each  $\mathcal{X}_j$  we apply the glasso to estimate a graph  $\hat{G}_j$ . We then take  $\hat{G}(x) = \hat{G}_j$  for all  $x \in \mathcal{X}_j$ . To find the partition, we restrict ourselves to dyadic splits, as studied by [Scott and Nowak \[2006a\]](#), [Blanchard et al. \[2007a\]](#). The primary reason for such a choice is the computational and theoretical tractability of dyadic partition-based estimators.

### 9.3.1 Dyadic Partitioning Tree

Let  $\mathcal{T}$  denote the set of dyadic partitioning trees (DPTs) defined over  $\mathcal{X} = [0, 1]^d$ , where each DPT  $T \in \mathcal{T}$  is constructed by recursively dividing  $\mathcal{X}$  by means of axis-orthogonal dyadic splits. Each node of a DPT corresponds to a hyperrectangle in  $[0, 1]^d$ . If a node is associated to the hyperrectangle  $\mathcal{A} = \prod_{l=1}^d [a_l, b_l]$ , then after being dyadically split along dimension  $k$ , the two children are associated with the sub-hyperrectangles

$$\mathcal{A}_L^{(k)} = \prod_{l < k} [a_l, b_l] \times [a_k, \frac{a_k + b_k}{2}] \times \prod_{l > k} [a_l, b_l] \quad \text{and} \quad \mathcal{A}_R^{(k)} = \mathcal{A} \setminus \mathcal{A}_L^{(k)}.$$

Given a DPT  $T$ , we denote by  $\Pi(T) = \{\mathcal{X}_1, \dots, \mathcal{X}_{m_T}\}$  the partition of  $\mathcal{X}$  induced by the leaf nodes of  $T$ . For a dyadic integer  $N = 2^K$  where  $K \in \{0, 1, 2, \dots\}$ , we define  $\mathcal{T}_N$  to be the collection of all DPTs such that no partition has a side length smaller than  $2^{-K}$ . Let  $I(\cdot)$  denote the indicator function. We denote  $\mu_T(x)$  and  $\Omega_T(x)$  as the piecewise constant mean and precision functions associated with  $T$ :

$$\mu_T(x) = \sum_{j=1}^{m_T} \mu_{\mathcal{X}_j} \cdot I(x \in \mathcal{X}_j) \quad \text{and} \quad \Omega_T(x) = \sum_{j=1}^{m_T} \Omega_{\mathcal{X}_j} \cdot I(x \in \mathcal{X}_j), \quad (9.3)$$

where  $\mu_{\mathcal{X}_j} \in \mathbb{R}^p$  and  $\Omega_{\mathcal{X}_j} \in \mathbb{R}^{p \times p}$  are the mean vector and precision matrix for  $\mathcal{X}_j$ .

## 9.3.2 Go-CART: Risk Minimization Estimator

Before formally defining our graph-valued regression estimators, we require some further definitions. Given a DPT  $T$  with an induced partition  $\Pi(T) = \{\mathcal{X}_j\}_{j=1}^{m_T}$  and corresponding mean and precision functions  $\mu_T(x)$  and  $\Omega_T(x)$ , the negative conditional log-likelihood risk  $R(T, \mu_T, \Omega_T)$  and its sample version  $\hat{R}(T, \mu_T, \Omega_T)$  are defined as follows:

$$R(T, \mu_T, \Omega_T) = \sum_{j=1}^{m_T} \mathbb{E} \left[ \left( \text{tr} \left[ \Omega_{\mathcal{X}_j} \left( (Y - \mu_{\mathcal{X}_j})(Y - \mu_{\mathcal{X}_j})^T \right) \right] - \log |\Omega_{\mathcal{X}_j}| \right) \cdot I(X \in \mathcal{X}_j) \right], \quad (9.4)$$

$$\hat{R}(T, \mu_T, \Omega_T) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{m_T} \left[ \left( \text{tr} \left[ \Omega_{\mathcal{X}_j} \left( (Y^{(i)} - \mu_{\mathcal{X}_j})(Y^{(i)} - \mu_{\mathcal{X}_j})^T \right) \right] - \log |\Omega_{\mathcal{X}_j}| \right) \cdot I(X^{(i)} \in \mathcal{X}_j) \right]. \quad (9.5)$$

Let  $[[T]] > 0$  denote a prefix code over all DPTs  $T \in \mathcal{T}_N$  satisfying

$$\sum_{T \in \mathcal{T}_N} 2^{-[[T]]} \leq 1.$$

One such prefix code  $[[T]]$  is proposed in [Scott and Nowak \[2006a\]](#), and takes the form

$$[[T]] = 3|\Pi(T)| - 1 + (|\Pi(T)| - 1) \log d / \log 2.$$

A simple upper bound for  $[[T]]$  is

$$[[T]] \leq (3 + \log d / \log 2) |\Pi(T)|. \quad (9.6)$$

Our analysis will assume that the conditional means and precision matrices are bounded in the  $\|\cdot\|_\infty$  and  $\|\cdot\|_1$  norms; specifically we suppose there is a positive constant  $B$  and a sequence  $L_{1,n}, \dots, L_{m_T,n}$ , where each  $L_{j,n} \in \mathbb{R}^+$  is a function of the sample size  $n$ , and we define the domains of each  $\mu_{\mathcal{X}_j}$  and  $\Omega_{\mathcal{X}_j}$  as

$$\begin{aligned} M_j &= \{\mu \in \mathbb{R}^p : \|\mu\|_\infty \leq B\}, \\ \Lambda_j &= \{\Omega \in \mathbb{R}^{p \times p} : \Omega \text{ is P.D., symmetric, and } \|\Omega\|_1 \leq L_{j,n}\}. \end{aligned} \quad (9.7)$$

With this notation in place, we can now define two estimators.

**Definition 9.1.** *The penalized empirical risk minimization Go-CART estimator is defined as*

$$\hat{T}, \left\{ \hat{\mu}_{\hat{\mathcal{X}}_j}, \hat{\Omega}_{\hat{\mathcal{X}}_j} \right\}_{j=1}^{m_{\hat{T}}} = \operatorname{argmin}_{T \in \mathcal{T}_N, \mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} \left\{ \hat{R}(T, \mu_T, \Omega_T) + \operatorname{pen}(T) \right\} \quad (9.8)$$

where  $\hat{R}$  is defined in (9.5) and

$$\operatorname{pen}(T) = \gamma_n \cdot m_T \sqrt{\frac{[[T]] \log 2 + 2 \log(np)}{n}}.$$

Empirically, we may always set the dyadic integer  $N$  to be a reasonably large value; the regularization parameter  $\gamma_n$  is responsible for selecting a suitable DPT  $T \in \mathcal{T}_N$ . Once  $T$  is chosen, the tuning parameters  $L_{1,n}, \dots, L_{m_T,n}$  corresponding each partition element of  $T$  need to be determined in a data-dependent way. We will discuss further details about this in the next section.

We can also formulate an estimator by minimizing held-out risk. Practically, we split the data into two partitions; we use

$$\mathcal{D}_1 = \{(X^{(1)}, Y^{(1)}), \dots, (X^{(n_1)}, Y^{(n_1)})\}$$

for training and

$$\mathcal{D}_2 = \{(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), \dots, (\tilde{X}^{(n_2)}, \tilde{Y}^{(n_2)})\}$$

for validation with  $n_1 + n_2 = n$ . The held-out negative log-likelihood risk is then given by

$$\begin{aligned} \hat{R}_{\text{out}}(T, \mu_T, \Omega_T) \\ = \frac{1}{n_2} \sum_{i=1}^{n_2} \sum_{j=1}^{m_T} \left\{ \left( \text{tr} \left[ \Omega_{\mathcal{X}_j} \left( (\tilde{Y}^{(i)} - \mu_{\mathcal{X}_j}) (\tilde{Y}^{(i)} - \mu_{\mathcal{X}_j})^T \right) \right] - \log |\Omega_{\mathcal{X}_j}| \right) \cdot I \left( \tilde{X}^{(i)} \in \mathcal{X}_j \right) \right\}. \end{aligned} \quad (9.9)$$

**Definition 9.2.** For each DPT  $T$  define

$$\hat{\mu}_T, \hat{\Omega}_T = \underset{\mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j}{\text{argmin}} \hat{R}(T, \mu_T, \Omega_T) \quad (9.10)$$

where  $\hat{R}$  is defined in (9.5) but only evaluated on  $\{(X^{(1)}, Y^{(1)}), \dots, (X^{(n_1)}, Y^{(n_1)})\}$ . The held-out risk minimization Go-CART estimator is

$$\hat{T} = \underset{T \in \mathcal{T}_N}{\text{argmin}} \hat{R}_{\text{out}}(T, \hat{\mu}_T, \hat{\Omega}_T). \quad (9.11)$$

where  $\hat{R}_{\text{out}}$  is defined in (9.9) but only evaluated on  $\mathcal{D}_2$ .

### 9.3.3 Go-CART: Greedy Partitioning

The above procedures require us to find an optimal dyadic partitioning tree within  $\mathcal{T}_N$ . Although dynamic programming can be applied, as in [Blanchard et al., 2007a], the computation does not scale to large input dimensions  $d$ . We now propose a simple yet effective greedy algorithm to find an approximate solution  $(\hat{T}, \hat{\mu}_T, \hat{\Omega}_T)$ . We focus on the held-out risk minimization form as in Definition 9.2, due to its superior empirical performance. But note that our greedy approach is generic and can easily be adapted to the penalized empirical risk minimization form.

First, consider the simple case that we are given a dyadic tree structure  $T$  which induces a partition  $\Pi(T) = \{\mathcal{X}_1, \dots, \mathcal{X}_{m_T}\}$  on  $\mathcal{X}$ . For any partition element  $\mathcal{X}_j$ , we estimate the sample mean using  $\mathcal{D}_1$ :

$$\hat{\mu}_{\mathcal{X}_j} = \frac{1}{\sum_{i=1}^{n_1} I(X^{(i)} \in \mathcal{X}_j)} \sum_{i=1}^{n_1} Y^{(i)} \cdot I(X^{(i)} \in \mathcal{X}_j).$$



The glasso is then used to estimate a sparse precision matrix  $\hat{\Omega}_{\mathcal{X}_j}$ . More precisely, let  $\hat{\Sigma}_{\mathcal{X}_j}$  be the sample covariance matrix for the partition element  $\mathcal{X}_j$ , given by

$$\hat{\Sigma}_{\mathcal{X}_j} = \frac{1}{\sum_{i=1}^{n_1} I(X^{(i)} \in \mathcal{X}_j)} \sum_{i=1}^{n_1} (Y^{(i)} - \hat{\mu}_{\mathcal{X}_j}) (Y^{(i)} - \hat{\mu}_{\mathcal{X}_j})^T \cdot I(X^{(i)} \in \mathcal{X}_j).$$

The estimator  $\hat{\Omega}_{\mathcal{X}_j}$  is obtained by optimizing

$$\hat{\Omega}_{\mathcal{X}_j} = \arg \min_{\Omega \succ 0} \{ \text{tr}(\hat{\Sigma}_{\mathcal{X}_j} \Omega) - \log |\Omega| + \lambda_j \|\Omega\|_1 \},$$

where  $\lambda_j$  is in one-to-one correspondence with  $L_{j,n}$  in (9.7). In practice, we run the full regularization path of the glasso, from large  $\lambda_j$ , which yields very sparse graph, to small  $\lambda_j$ , and select the graph that minimizes the held-out negative log-likelihood risk. To further improve the model selection performance, we refit the parameters of the precision matrix after the graph has been selected. That is, to reduce the bias of the glasso, we first estimate the sparse precision matrix using  $\ell_1$ -regularization, and then we refit the Gaussian model without  $\ell_1$ -regularization, but enforcing the sparsity pattern obtained in the first step. Liu et al. [2010c] demonstrate that such a refitting step will yield a significantly better model selection performance when estimating graphs.

The natural, standard greedy procedure starts from the coarsest partition  $\mathcal{X} = [0, 1]^d$  and then computes the decrease in the held-out risk by dyadically splitting each hyperrectangle  $\mathcal{A}$  along dimension  $k \in \{1, \dots, d\}$ . The dimension  $k^*$  that results in the largest decrease in held-out risk is selected. More precisely, let  $\text{sl}_k(\mathcal{A})$  be the side length of  $\mathcal{A}$  on the dimension  $k$ . If  $\text{sl}_k(\mathcal{A}) > 2^{-K}$ , where  $K = \log_2 N$ , we dyadically split  $\mathcal{A}$  along the dimension  $k$ . In this case, let  $\mathcal{A}_L^{(k)}$  and  $\mathcal{A}_R^{(k)}$  be the two resulting sub-hyperrectangles. The decrease in held-out risk takes the form

$$\begin{aligned} \Delta \hat{R}_{\text{out}}^{(k)}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}}) \\ = \hat{R}_{\text{out}}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}}) - \hat{R}_{\text{out}}(\mathcal{A}_L^{(k)}, \hat{\mu}_{\mathcal{A}_L^{(k)}}, \hat{\Omega}_{\mathcal{A}_L^{(k)}}) - \hat{R}_{\text{out}}(\mathcal{A}_R^{(k)}, \hat{\mu}_{\mathcal{A}_R^{(k)}}, \hat{\Omega}_{\mathcal{A}_R^{(k)}}). \end{aligned} \quad (9.12)$$

Note that if splitting any dimension  $k$  of  $\mathcal{A}$  leads to an increase in the risk, we set a Boolean variable  $S(\mathcal{A}) = \text{False}$  which indicates that the partition element  $\mathcal{A}$  should no longer be split and hence  $\mathcal{A}$  should be a partition element of  $\Pi(T)$ . The greedy Go-CART, as presented in Algorithm 9.3.1, recursively applies the previous procedure to split each partition element until all the partition elements cannot be further split. Note that we also record the dyadic partition tree structure in the implementation.

This greedy partitioning method parallels the classical algorithms for classification and regression trees that have been used in statistical learning for decades. However, the strength of the procedures given in Definitions 9.1

**Algorithm 9.3.1** Greedy Go-CART using Dyadic Partitioning

**Input:** training data  $\{X^{(i)}, Y^{(i)}\}_{i=1}^{n_1}$ , held-out validation data  $\{\tilde{X}^{(i)}, \tilde{Y}^{(i)}\}_{i=1}^{n_2}$ , and an integer  $K$ .

Start from  $\mathcal{X} = [0, 1]^d$ . Set the Boolean variable  $S(\mathcal{X}) = \text{True}$  and estimate  $\hat{\mu}_{\mathcal{X}}, \hat{\Omega}_{\mathcal{X}}$

**while** exists a hyperrectangle  $\mathcal{A}$  such that  $S(\mathcal{A}) = \text{True}$  **do**

**for all** dimensions  $k \in \{1, \dots, d\}$  **do**

**if**  $\text{sl}_k(\mathcal{A}) \geq 2^{-K+1}$  **then**

      Calculate  $\Delta \hat{R}_{\text{out}}^{(k)}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}})$  according to (9.12)

**else**

      Set  $\Delta \hat{R}_{\text{out}}^{(k)}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}}) = -\infty$

  Determine the best splitting dimension  $k^* = \arg \max_{k \in \{1, \dots, d\}} \Delta \hat{R}_{\text{out}}^{(k)}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}})$

**if**  $\Delta \hat{R}_{\text{out}}^{(k^*)}(\mathcal{A}, \hat{\mu}_{\mathcal{A}}, \hat{\Omega}_{\mathcal{A}}) > 0$  **then**

    Dyadically split  $\mathcal{A}$  along dimension  $k^*$ , yielding two hyperrectangles  $\mathcal{A}_L^{(k^*)}$  and  $\mathcal{A}_R^{(k^*)}$ .

    Estimate  $\hat{\mu}_{\mathcal{A}_L^{(k^*)}}, \hat{\Omega}_{\mathcal{A}_L^{(k^*)}}, \hat{\mu}_{\mathcal{A}_R^{(k^*)}}, \hat{\Omega}_{\mathcal{A}_R^{(k^*)}}$  and set  $S(\mathcal{A}_L^{(k^*)}) = S(\mathcal{A}_R^{(k^*)}) = \text{True}$ .

**else**

    Set  $S(\mathcal{A}) = \text{False}$  and put  $\mathcal{A}$  into the final partition set.

**Output:** Partition  $\Pi(\hat{T}) = \{\mathcal{X}_j\}_{j=1}^{m_{\hat{T}}}$  and the corresponding DPT  $\hat{T}$  with the estimated  $\hat{\mu}_{\mathcal{X}_j}$ .

and 9.2 is that they lend themselves to a theoretical analysis under relatively weak assumptions, as we show in the following section. The theoretical properties of greedy Go-CART are left to future work.

## 9.4 THEORETICAL PROPERTIES

We define the oracle risk  $R^*$  over  $\mathcal{T}_N$  as

$$R^* = R(T^*, \mu_{T^*}^*, \Omega_{T^*}^*) = \inf_{T \in \mathcal{T}_N, \mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} R(T, \mu_T, \Omega_T).$$

Note that  $T^*$ ,  $\mu_{T^*}^*$ , and  $\Omega_{T^*}^*$  might not be unique, since the finest partition always achieves the oracle risk. To obtain oracle inequalities, we make the following two technical assumptions.

**Assumption 9.1.** Let  $T \in \mathcal{T}_N$  be an arbitrary DPT which induces a partition  $\Pi(T) = \{\mathcal{X}_1, \dots, \mathcal{X}_{m_T}\}$  on  $\mathcal{X}$ , we assume that there exists a constant  $B$ , such that

$$\max_{1 \leq j \leq m_T} \|\mu_{\mathcal{X}_j}\|_{\infty} \leq B \quad \text{and} \quad \max_{1 \leq j \leq m_T} \sup_{\Omega \in \Lambda_j} \log |\Omega| \leq L_n$$

where  $\Lambda_j$  is defined in (9.7) and  $L_n = \max_{1 \leq j \leq m_T} L_{j,n}$ , where  $L_{j,n}$  is the same as in (9.7). We also assume that

$$L_n = o(\sqrt{n}).$$

**Assumption 9.2.** Let  $Y = (Y_1, \dots, Y_p)^T \in \mathbb{R}^p$ . For any  $\mathcal{A} \subset \mathcal{X}$ , we define

$$Z_{k\ell}(\mathcal{A}) = Y_k Y_\ell \cdot I(X \in \mathcal{A}) - \mathbb{E}(Y_k Y_\ell \cdot I(X \in \mathcal{A})) \quad (9.13)$$

$$Z_j(\mathcal{A}) = Y_j \cdot I(X \in \mathcal{A}) - \mathbb{E}(Y_j \cdot I(X \in \mathcal{A})). \quad (9.14)$$

We assume there exist constants  $M_1, M_2, v_1$ , and  $v_2$ , such that

$$\sup_{k, \ell, \mathcal{A}} \mathbb{E}|Z_{k\ell}(\mathcal{A})|^m \leq \frac{m! M_2^{m-2} v_2}{2} \quad \text{and} \quad \sup_{j, \mathcal{A}} \mathbb{E}|Z_j(\mathcal{A})|^m \leq \frac{m! M_1^{m-2} v_1}{2}$$

for all  $m \geq 2$ .

**Theorem 9.1.** Let  $T \in \mathcal{T}_N$  be a DPT that induces a partition  $\Pi(T) = \{\mathcal{X}_1, \dots, \mathcal{X}_{m_T}\}$  on  $\mathcal{X}$ . For any  $\delta \in (0, 1)$ , let  $\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}$  be the estimator obtained using the penalized empirical risk minimization Go-CART in Definition 9.1, with a penalty term  $\text{pen}(T)$  of the form

$$\text{pen}(T) = (C_1 + 1) L_n m_T \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(48/\delta)}{n}} \quad (9.15)$$

where  $C_1 = 8\sqrt{v_2} + 8B\sqrt{v_1} + B^2$ . Then for sufficiently large  $n$ , the excess risk inequality

$$R(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) - R^* \quad (9.16)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ 2\text{pen}(T) + \inf_{\mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} (R(T, \mu_T, \Omega_T) - R^*) \right\} \quad (9.17)$$

holds with probability at least  $1 - \delta$ .

A similar oracle inequality holds when using the held-out risk minimization Go-CART.

**Theorem 9.2.** Let  $T \in \mathcal{T}_N$  be a DPT which induces a partition  $\Pi(T) = \{\mathcal{X}_1, \dots, \mathcal{X}_{m_T}\}$  on  $\mathcal{X}$ . For any  $\delta \in (0, 1)$ , we define  $\phi_n(T)$  to be a function of  $n$  and  $T$ :

$$\phi_n(T) = (C_2 + \sqrt{2}) L_n m_T \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(384/\delta)}{n}}$$

where  $C_2 = 8\sqrt{2v_2} + 8B\sqrt{2v_1} + \sqrt{2}B^2$  and  $L_n = \max_{1 \leq j \leq m_T} L_{j,n}$ . Partition the data into

$$\mathcal{D}_1 = \{(X^{(1)}, Y^{(1)}), \dots, (X^{(n_1)}, Y^{(n_1)})\}$$

and

$$\mathcal{D}_2 = \{(\tilde{X}^{(1)}, \tilde{Y}^{(1)}), \dots, (\tilde{X}^{(n_2)}, \tilde{Y}^{(n_2)})\}$$

with sizes  $n_1 = n_2 = n/2$ . Let  $\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}$  be the estimator constructed using the held-out risk minimization criterion of Definition 9.2. Then, for sufficiently large  $n$ , the excess risk inequality

$$R(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) - R^* \quad (9.18)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ 3\phi_n(T) + \inf_{\mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} (R(T, \mu_T, \Omega_T) - R^*) \right\} + \phi_n(\hat{T}) \quad (9.19)$$

holds with probability at least  $1 - \delta$ .

Note that in contrast to the statement in Theorem 9.1, Theorem 9.2 results in a stochastic upper bound due to the extra  $\phi_n(\hat{T})$  term, which depends on the complexity of the final estimate  $\hat{T}$ . The proofs of both theorems are given in the appendix.

We now temporarily make the strong assumption that the model is correct, so that  $Y$  given  $X$  is conditionally Gaussian, with a partition structure that is given by a dyadic tree. We show that with high probability, the true dyadic partition structure can be correctly recovered.

**Assumption 9.3.** *The true model is*

$$Y | X = x \sim N_p(\mu_{T^*}^*(x), \Omega_{T^*}^*(x)) \quad (9.20)$$

where  $T^* \in \mathcal{T}_N$  is a DPT with induced partition  $\Pi(T^*) = \{\mathcal{X}_j^0\}_{j=1}^{m_{T^*}}$  and

$$\mu_{T^*}^*(x) = \sum_{j=1}^{m_{T^*}} \mu_j^* I(x \in \mathcal{X}_j^0), \quad \Omega_{T^*}^*(x) = \sum_{j=1}^{m_{T^*}} \Omega_j^* I(x \in \mathcal{X}_j^0).$$

Under this assumption, clearly

$$R(T^*, \mu_{T^*}^*, \Omega_{T^*}^*) = \inf_{T \in \mathcal{T}_N, \mu_T, \Omega_T \in \mathcal{M}_T} R(T, \mu_T, \Omega_T), \quad (9.21)$$

where  $\mathcal{M}_T$  is given by

$$\mathcal{M}_T = \left\{ \mu(x) = \sum_{j=1}^{m_T} \mu_{\mathcal{X}_j} I(x \in \mathcal{X}_j), \Omega(x) = \sum_{j=1}^{m_T} \Omega_{\mathcal{X}_j} I(x \in \mathcal{X}_j) : \right. \\ \left. \text{where } \mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j, \Pi(T) = \{\mathcal{X}_j\}_{j=1}^{m_T} \right\}.$$

Let  $T_1$  and  $T_2$  be two DPTs, if  $\Pi(T_1)$  can be obtained by further split the hyperrectangles within  $\Pi(T_2)$ , we say  $\Pi(T_2) \subset \Pi(T_1)$ . We then have the following definitions:

**Definition 9.3.** *A tree estimation procedure  $\hat{T}$  is tree partition consistent in case*

$$\mathbb{P}(\Pi(T^*) \subset \Pi(\hat{T})) \rightarrow 1 \text{ as } n \rightarrow \infty. \quad (9.22)$$

Note that the estimated partition may be finer than the true partition. Establishing a tree partition consistency result requires further technical assumptions. The following assumption specifies that for arbitrary adjacent subregions of the true dyadic partition, either the means or the variances should be sufficiently different. Without such an assumption, of course, it is impossible to detect the boundaries of the true partition.

**Assumption 9.4.** *Let  $\mathcal{X}_i^0$  and  $\mathcal{X}_j^0$  be adjacent partition elements of  $T^*$ , so that they have a common parent node within  $T^*$ . Let  $\Sigma_{\mathcal{X}_i^0}^* = (\Omega_{\mathcal{X}_i^0}^*)^{-1}$ . We assume there exist positive constants  $c_1, c_2, c_3, c_4$ , such that either*

$$2 \log \left| \frac{\Sigma_{\mathcal{X}_i^0}^* + \Sigma_{\mathcal{X}_j^0}^*}{2} \right| - \log |\Sigma_{\mathcal{X}_i^0}^*| - \log |\Sigma_{\mathcal{X}_j^0}^*| \geq c_4 \quad (9.23)$$

or  $\|\mu_{\mathcal{X}_i^0}^* - \mu_{\mathcal{X}_j^0}^*\|_2^2 \geq c_3$ . We also assume

$$\rho_{\min}(\Omega_{\mathcal{X}_j^0}^*) \geq c_1, \quad \forall j = 1, \dots, m_{T^*}, \quad (9.24)$$

where  $\rho_{\min}(\cdot)$  denotes the smallest eigenvalue. Furthermore, for any  $T \in \mathcal{T}_N$  and any  $\mathcal{A} \in \Pi(T)$ , we have  $\mathbb{P}(X \in \mathcal{A}) \geq c_2$ .

**Theorem 9.3.** *Under the above assumptions, we have*

$$\inf_{T \in \mathcal{T}_N, \Pi(T^*) \not\subseteq \Pi(T)} \inf_{\mu_T, \Omega_T \in \mathcal{M}_T} R(T, \mu_T, \Omega_T) - R(T^*, \mu_{T^*}^*, \Omega_{T^*}^*) > \min\left\{\frac{c_1 c_2 c_3}{2}, c_2 c_4\right\}$$

where  $c_1, c_2, c_3, c_4$  are defined in Assumption 9.4. Moreover, the Go-CART estimator in both the penalized risk minimization and held-out risk minimization form is tree partition consistent.

This result shows that, with high probability, we obtain a finer partition than  $T^*$ ; the assumptions do not, however, control the size of the resulting partition. The proof of this result appears in the appendix.

## 9.5 EXPERIMENTAL RESULTS

We evaluate the performance of the greedy Go-CART learning algorithm in Section 9.3.3 on both synthetic datasets and a meteorological dataset. In each experiment, we set the dyadic integer to  $N = 2^{10}$  to ensure that we can obtain fine-tuned partitions of the input space  $\mathcal{X}$ . Furthermore, we always ensure that the region (hyperrectangle) represented by each leaf node contains at least 10 data points to guarantee reasonable estimates of the sample means and sparse inverse covariance matrices.

### 9.5.1 Synthetic Data

We generate  $n$  data points  $X^{(1)}, \dots, X^{(n)} \in \mathbb{R}^d$  with  $n = 10,000$  and  $d = 10$  uniformly distributed on the unit hypercube  $[0, 1]^d$ . We split the square  $[0, 1]^2$  defined by the first two dimensions into 22 subregions, as shown in Figure 53(a). For the  $t$ -th subregion where  $1 \leq t \leq 22$ , we generate an Erdős-Rényi random graph  $G^t = (V^t, E^t)$  with  $p = 20$  vertices and  $|E| = 10$  edges, with maximum node degree four. As an illustration, the random graphs for subregion 4 (the smallest region), 17 (middle region) and 22 (large region) are presented in Figures 53(b), (c) and (d), respectively. For each graph  $G^t$ , we generate an inverse covariance matrix  $\Omega^t$  according to:

$$\Omega_{i,j}^t = \begin{cases} 1 & \text{if } i = j, \\ 0.245 & \text{if } (i, j) \in E^t, \\ 0 & \text{otherwise,} \end{cases}$$

where 0.245 guarantees positive-definiteness of  $\Omega^t$  when the maximum node degree is four.

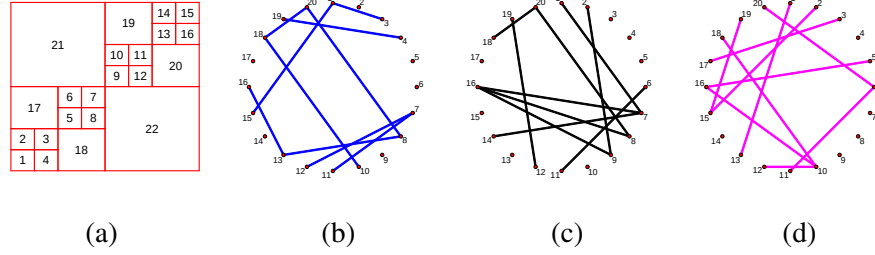


Figure 53.: (a) The 22 subregions defined on  $[0, 1]^2$ . The horizontal axis corresponds to the first dimension denoted as  $X_1$  while the vertical axis corresponds to the second dimension denoted as  $X_2$ . The bottom left point corresponds to  $[0, 0]$  and the upper right point corresponds to  $[1, 1]$ . (b) The true graph for subregion 4. (c) The true graph for subregion 17. (d) The true graph for subregion 22.

To each data point  $X^{(i)}$  in the  $t$ -th subregion we associate a 20-dimensional response vector  $Y^{(i)}$  generated from a multivariate Gaussian distribution  $N_{20}(0, (\Omega^t)^{-1})$ . We also create an equally-sized held-out dataset in the same manner based on  $\{\Omega^t\}_{t=1}^{22}$ .

We apply Algorithm 9.3.1 to this synthetic dataset. The estimated dyadic tree structure and its induced partitions are presented in Figure 54. Estimated graphs for some nodes are also illustrated. Note that the label for each subregion in subplot (c) is the leaf node ID of the tree in subplot (a). We conduct 100 Monte-Carlo simulations and find that in 82 out of 100 runs our algorithm perfectly recovers the ground truth partition of the  $X_1$ - $X_2$  plane, and never wrongly splits on any of the irrelevant dimensions, ranging from  $X_3$  to  $X_{10}$ . Moreover, the estimated graphs have interesting patterns. Even though the graphs within each subregion are sparse, the estimated graph obtained by pooling all the data together is highly dense. As the algorithm progresses, the estimated graphs become more sparse. However, for the immediate parent nodes of the true subregions, the graphs become denser again.

Out of the 82 simulations where we correctly identify the tree structure, we list the graph estimation performance for subregions 1, 4, 17, 18, 21, 22 in terms of precision, recall, and  $F_1$ -score. Let  $\hat{E}$  be the estimated edge set while  $E$  be the true edge set. These criteria are defined as:

$$\text{precision} = \frac{|\hat{E} \cap E|}{|\hat{E}|}, \quad \text{recall} = \frac{|\hat{E} \cap E|}{|E|}, \quad F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

We see that for a larger subregion, it is easier to obtain better recovery performance, while good recovery for a very small region is more challenging. Of course, in the smaller regions there is less data; In Figure 53(a), there are

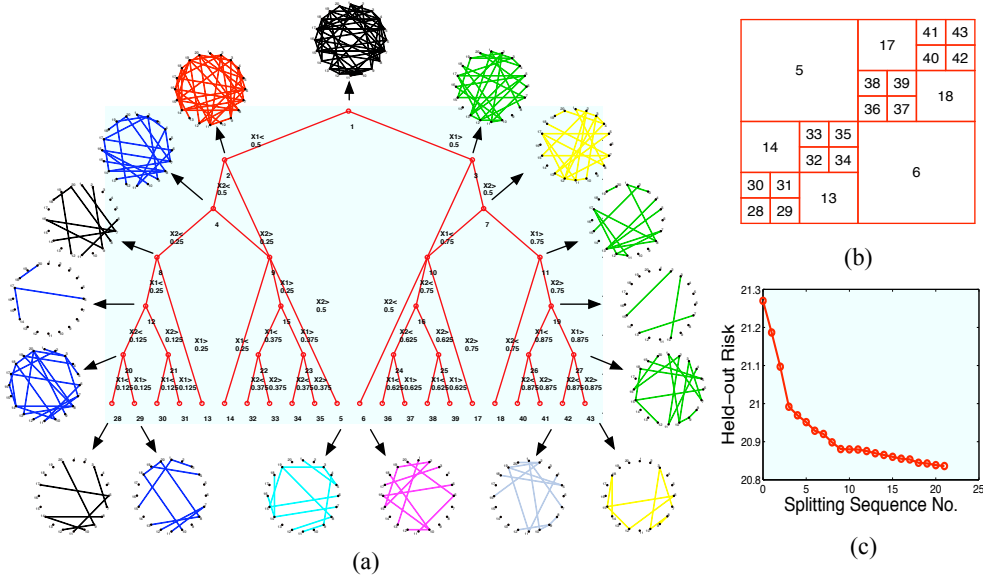


Figure 54.: (a) The estimated dyadic tree structure; (b) the induced partition on  $[0, 1]^2$  and the number labeled on each subregion corresponds to each leaf node ID of the tree in (a); (c) the held-out negative log-likelihood risk for each split. The order of the splits corresponds the ID of the tree node (from small to large)

only  $10000/64 \approx 156$  data points appear in subregion 1 (the smallest one). In contrast, approximately  $10000/16 = 625$  data points fall inside subregion 18, so that the graph corresponding to this region can be better estimated.

We also plot the held-out risk in the subplot (c). As can be seen, the first few splits lead to the most significant decrease in the held-out risk.

Table 11.: The graph estimation performance over different subregions

| subregion     | Mean values over 100 runs (Standard deviation) |               |               |               |               |               |
|---------------|------------------------------------------------|---------------|---------------|---------------|---------------|---------------|
|               | region 1                                       | region 4      | region 17     | region 18     | region 21     | region 22     |
| Precision     | 0.8327 (0.15)                                  | 0.8429 (0.15) | 0.9821 (0.05) | 0.9853 (0.04) | 0.9906 (0.04) | 0.9899 (0.05) |
| Recall        | 0.7890 (0.16)                                  | 0.7990 (0.18) | 1.0000 (0.00) | 1.0000 (0.00) | 1.0000 (0.00) | 1.0000 (0.00) |
| $F_1$ - score | 0.7880 (0.11)                                  | 0.7923 (0.12) | 0.9904 (0.03) | 0.9921 (0.02) | 0.9949 (0.02) | 0.9913 (0.02) |

#### 9.5.1.1 Further Simulations

To further demonstrate the performance of the method, this section presents simulations where the true conditional covariance matrix is continuous in  $X$ . We compare the graphs estimated by our method to the single graph obtained by applying the glasso directly to the entire dataset.

In this subsection, we consider the case where  $X$  lies on a one dimensional chain. More precisely, we generate  $n$  equally spaced points  $X^{(1)}, \dots, X^{(n)} \in \mathbb{R}$  with  $n = 10,000$  on  $[0, 1]$ . We generate an Erdős-Rényi random graph  $G^1 = (V^1, E^1)$  with  $p = 20$  vertices,  $|E| = 10$  edges, and maximum node degree four. Then, we simulate the output  $Y^{(1)}, \dots, Y^{(n)} \in \mathbb{R}^p$  as follows:

1. For  $t = 2$  to  $T$ , we construct the graph  $G^t = (V^t, E^t)$  as follows: (a) with probability 0.05, remove one edge from  $G^{t-1}$  and (b) with probability 0.05, add one edge to the graph generated in (a). We make sure that the total number of edges is between 5 and 15, and that the maximum node degree four.
2. For each graph  $G^t$ , generate the inverse covariance matrix  $\Omega^t$ :

$$\Omega^t(i, j) = \begin{cases} 1 & \text{if } i = j, \\ 0.245 & \text{if } (i, j) \in E^t, \\ 0 & \text{otherwise,} \end{cases}$$

where 0.245 guarantees positive-definiteness of  $\Omega^t$  under the degree constraint.

3. For each  $t$ , we sample  $y_t$  from a multivariate Gaussian distribution with mean  $\mu = (0, \dots, 0) \in \mathbb{R}^p$  and covariance matrix  $\Sigma^t = (\Omega^t)^{-1}$ .

We generate an equal-sized held-out dataset in the same manner, using the same  $\mu$  and  $\Sigma^t$ . Greedy Go-CART is used to estimate the dyadic tree structure and corresponding inverse covariance matrices; these are displayed in Figure 55.

### 9.5.2 Chain Structure

To examine the recovery quality of the underlying graph structure, we compare our estimated graphs to the graph estimated by directly applying the glasso to the entire dataset. Comparisons in terms of precision, recall and  $F_1$ -score are given in Figure 56 (a), (b) and (c) respectively. As we can see, the partition-based method achieves much higher precision and  $F_1$ -Score. As for recall, glasso is slightly better, due to the fact that the glasso graphs estimated on the entire data are very dense, as shown in 56 (d). The dense graphs lead to fewer false negatives (thus large recall) but many false positives (thus small precision).

### 9.5.3 Two-way Grid Structure

In this section, we apply Go-CART to a two dimensional design  $X$ . The underlying graph structures and  $Y$  are generated in manner similar to that



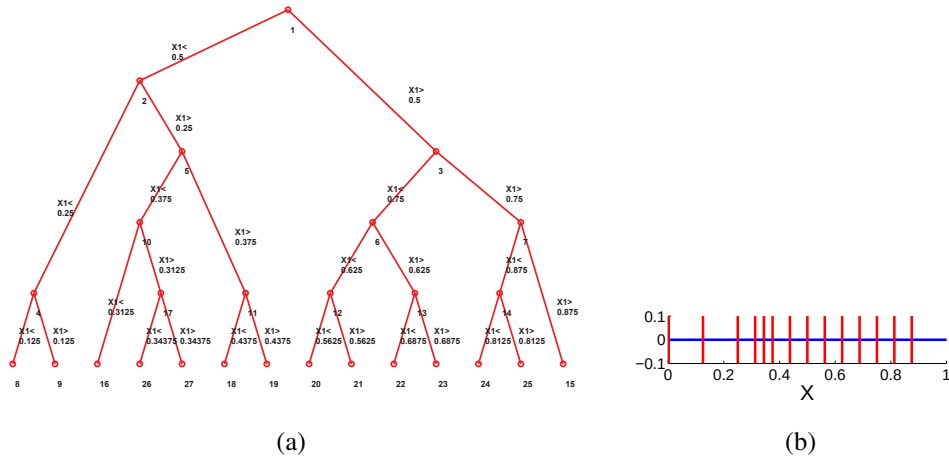
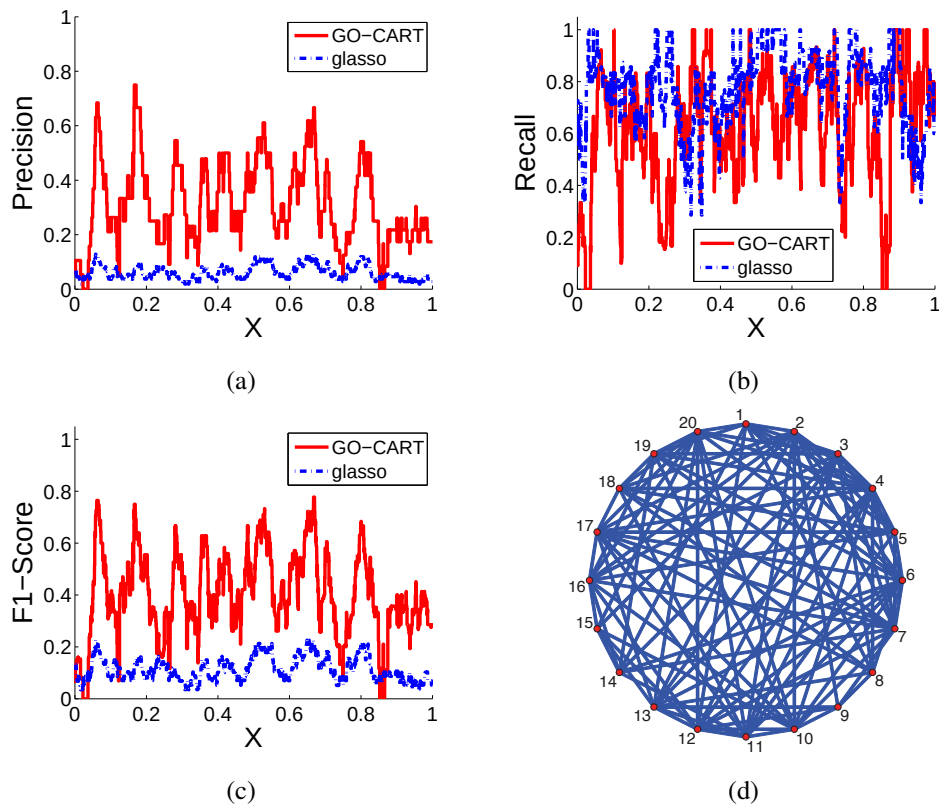


Figure 55.: (a) Estimated tree structure; (b) corresponding partitions

Figure 56.: Comparison of our algorithm with glasso (a) Precision; (b) Recall; (c)  $F_1$ -score; (d) Estimated graph by applying glasso on the entire dataset

used in the previous section. In particular, we generate equally spaced  $X^{(1)}, \dots, X^{(n)} \in \mathbb{R}^2$  with  $n = 10,000$  on the unit two-dimensional grid  $[0, 1]^2$ . We generate an Erdős-Rényi random graph  $G^{1,1} = (V^{1,1}, E^{1,1})$  with  $p = 20$  vertices,  $|E| = 10$  edges, and maximum node degree four, then construct the graphs for each  $x$  along diagonals. More precisely, for each pair of  $i, j$ , where  $1 \leq i \leq 100$  and  $1 \leq j \leq 100$ , we randomly select either  $G^{i-1,j}$  (if it exists) or  $G^{i,j-1}$  (if it exists) with equal probability as the basis graph. Then, we construct the graph  $G^{i,j} = (V^{i,j}, E^{i,j})$  by removing one edge and adding one edge with probability 0.05 based on the selected basis graph, taking care that the number of edges is between 5 and 15 and the maximum degree is still four. Given the underlying graphs, we generate the covariance matrix and output  $Y$  in the same way as in the last section.

We apply the greedy algorithm to learn the dyadic tree structure and corresponding inverse covariance matrices, shown in Figure 57. We plot the  $F_1$ -score obtained by glasso on the entire data compared against the our method in Figure 58. It is seen that for most  $x$ , the partitioning method achieves significantly higher  $F_1$ -score than directly applying the glasso. Note that since the graphs near the middle part of the diagonal (the line connecting  $[0, 1]$  and  $[1, 0]$ ) have the greatest variability, the  $F_1$ -scores for both methods are low in this region.

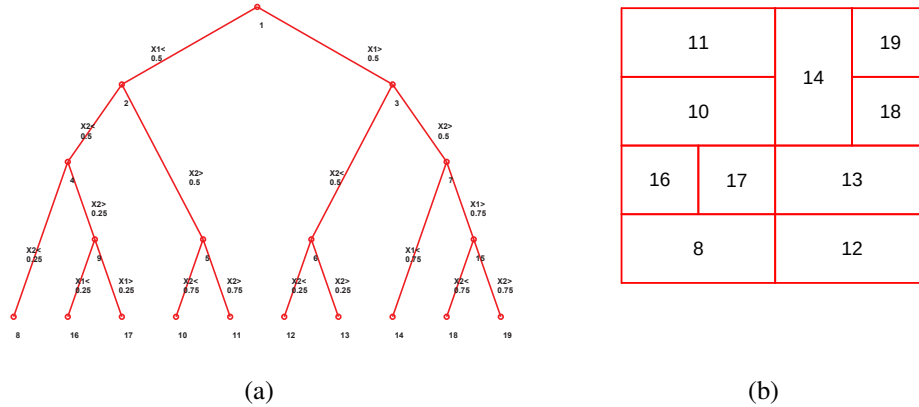


Figure 57.: (a) Estimated tree structure; (b) estimated partitions where the labels correspond to the index of the leaf node in (a)

#### 9.5.4 Climate Data Analysis

In this section, we use graph-valued regression to analyze a meteorology dataset [Lozano et al., 2009] that contains monthly data of 18 different meteorological factors from 1990 to 2002. We use the data from 1990 to 1995 as the training data and the data from 1996 to 2002 as the held-out validation

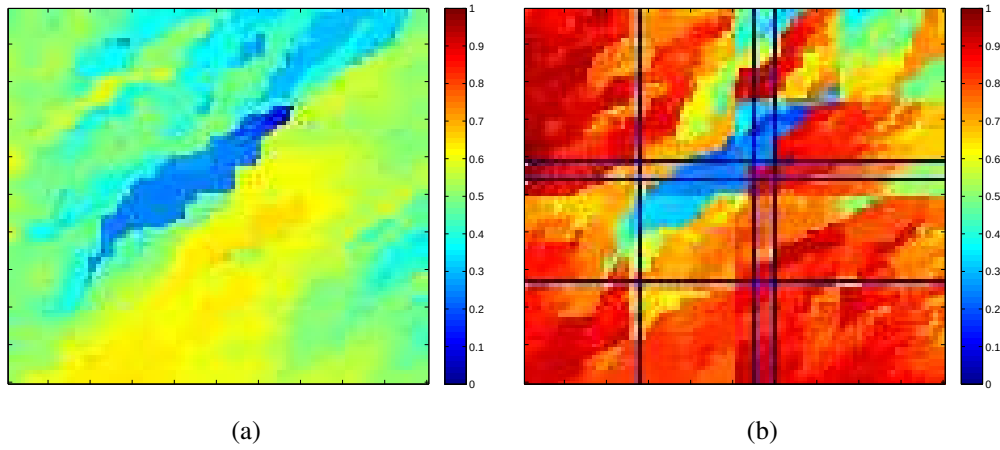


Figure 58.: (a) Color map of  $F_1$ -score for glasso run on the entire dataset; (b) color map of  $F_1$ -score for Go-CART. Red indicates large values (approaching 1) and blue indicates small values (approaching 0), as shown in the color bar.

data. The observations span 125 locations in the US on an equally spaced grid between latitude 30.475 and 47.975 and longitude -119.75 to -82.25. The 18 meteorological factors measured for each month include levels of  $\text{CO}_2$ ,  $\text{CH}_4$ ,  $\text{H}_2$ ,  $\text{CO}$ , average temperature (TMP) and diurnal temperature range (DTR), minimum temperature (TMN), maximum temperature (TMX), precipitation (PRE), vapor (VAP), cloud cover (CLD), wet days (WET), frost days (FRS), global solar radiation (GLO), direct solar radiation (DIR), extraterrestrial radiation (ETR), extraterrestrial normal radiation (ETRN) and UV aerosol index (UV). For further detail, see [Lozano et al. \[2009\]](#).

As a baseline, we estimate a sparse graph on the data from all 125 locations, using the glasso algorithm; the estimated graph is shown in Figure 59 (b). It is seen that there is no edge connecting to any of the greenhouse gas factors  $\text{CO}_2$ ,  $\text{CH}_4$ ,  $\text{H}_2$  or  $\text{CO}$ . This apparently contradicts basic domain knowledge that these four factors should correlate with the solar radiation factors (including GLO, DIR, ETR, ETRN, and UV), according to the IPCC report [IPCC \[2007\]](#), one of the most authoritative reports in the field of meteorology. The reason for the missing edges in the pooled data may be that positive correlations at one location are canceled by negative correlations at other locations.

Treating the longitude and latitude of each site as two-dimensional covariate  $X$ , and the meteorology data of the  $p = 18$  factors as the response  $Y$ , we estimate a dyadic tree structure using the greedy algorithm. The result is a partition with 87 subregions, shown in Figure 59, with the corresponding dyadic partition tree is shown in Figure 60. The graphs for subregion 2 (corresponding to the strip of land from Los Angeles, California to Phoenix, Arizona) and subregion 3 (Bakersfield, California to Flagstaff, Arizona) are shown in subplot (a) of Figure 59. The graphs for these two adjacent subregions are

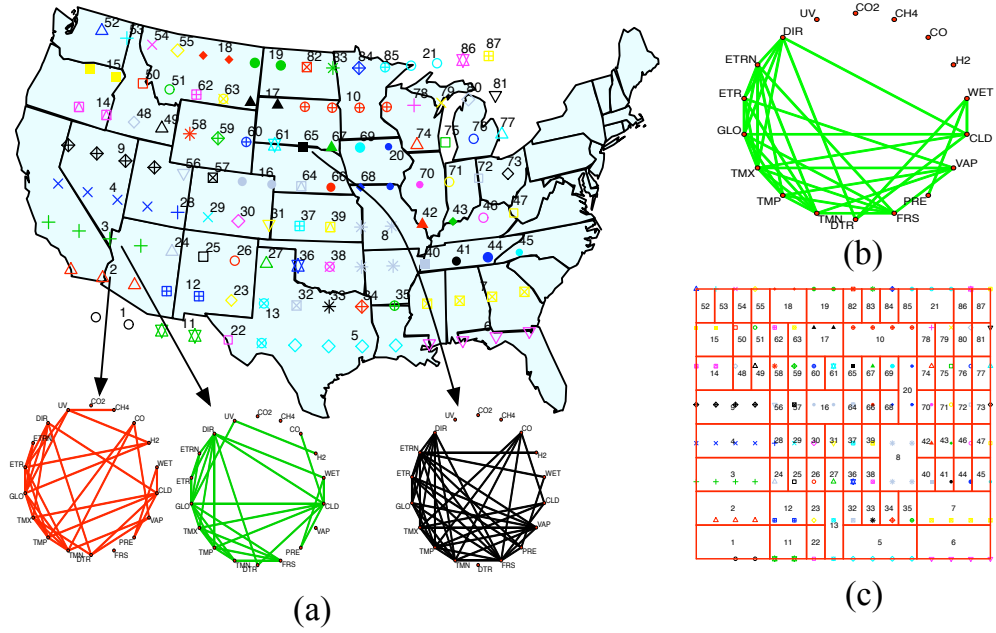


Figure 59.: Analysis of the climate data. (a) Estimated partitions for 125 locations projected to the US map, with the estimated graphs for subregions 2, 3, and 65; (b) estimated graph with data pooled from all 125 locations; (c) the re-scaled partition pattern induced by the dyadic tree structure.

quite similar, suggesting spatial smoothness of the learned graphs. Moreover, for both graphs, CO is connected to solar radiation factors in either a direct or indirect way, and H<sub>2</sub> is connected to UV, which is accordance with Chapter 7 of the IPCC report [IPCC \[2007\]](#). In contrast, for subregion 65, which corresponds to the border of South Dakota and Nebraska; here the graph is quite different. In general, it is found that the graphs corresponding to the locations along the coasts are sparser than those corresponding to the locations in the mainland.

Such observations, which require validation and interpretation by domain experts, are examples of the capability of graph-valued regression to provide a useful tool for high dimensional data analysis.

## 9.6 CONCLUSIONS

In this chapter, we present Go-CART, a partition-based estimator of the family of undirected graphs associated with a high dimensional conditional distribution. Dyadic partitioning estimators, either using penalized empirical risk minimization or data splitting, are attractive due to their simplicity and theoretical guarantees. We derive finite sample oracle inequalities on excess risk, together with a tree partition consistency result. Our theory allows the scale of the graphs to increase with the sample size, which is relevant since the methods are targeted at high dimensional data analysis applications. Greedy

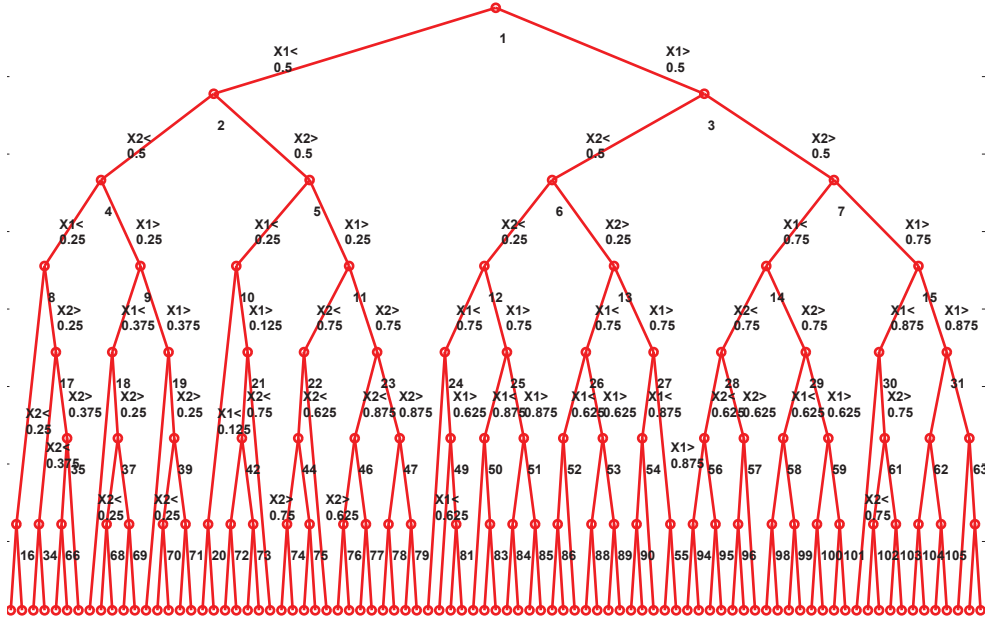


Figure 60.: The estimated dyadic tree structure on the climate data.

partitioning estimators are proposed that are computationally attractive, combining classical greedy algorithms for decision trees with recent advances in  $\ell_1$ -regularization techniques for graph selection. The practical potential of Go-CART is indicated by experiments on a meteorology dataset. A theoretical analysis of greedy Go-CART is one of several interesting directions for future work.

## 9.7 APPENDIX: TECHNICAL PROOFS

### 9.7.0.1 Proof of Theorem 9.1

For any  $T \in \mathcal{T}_N$ , we denote

$$S_{j,n} = \frac{1}{n} \sum_{i=1}^n (Y^{(i)} - \mu_{\mathcal{X}_j})(Y^{(i)} - \mu_{\mathcal{X}_j})^T \cdot I(X^{(i)} \in \mathcal{X}_j) \quad (9.25)$$

$$\bar{S}_j = \mathbb{E}(Y - \mu_{\mathcal{X}_j})(Y - \mu_{\mathcal{X}_j})^T \cdot I(X \in \mathcal{X}_j). \quad (9.26)$$

We then have

$$\begin{aligned} & \left| R(T, \mu_T, \Omega_T) - \widehat{R}(T, \mu_T, \Omega_T) \right| \\ & \leq \left| \sum_{j=1}^m \text{tr} \left[ \Omega_{\mathcal{X}_j} (S_{j,n} - \bar{S}_j) \right] \right| \end{aligned} \quad (9.27)$$

$$+ \left| \sum_{j=1}^m \log |\Omega_{\mathcal{X}_j}| \cdot \left[ \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}I(X \in \mathcal{X}_j) \right] \right| \quad (9.28)$$

$$\leq \underbrace{\sum_{j=1}^m \|\Omega_{\mathcal{X}_j}\|_1 \cdot \|S_{j,n} - \bar{S}_j\|_\infty}_{A_1} \quad (9.29)$$

$$+ \underbrace{\sum_{j=1}^m \left| \log |\Omega_{\mathcal{X}_j}| \right| \cdot \left| \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}I(X \in \mathcal{X}_j) \right|}_{A_2}. \quad (9.30)$$

We now analyze the terms  $A_1$  and  $A_2$  separately.

For  $A_2$ , using the Hoeffding's inequality, for  $\epsilon > 0$ , we get

$$\mathbb{P} \left( \left| \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}I(X \in \mathcal{X}_j) \right| > \epsilon \right) \leq 2 \exp(-2n\epsilon^2), \quad (9.31)$$

which implies that,

$$\mathbb{P} \left( \sup_{T \in \mathcal{T}_N} \left| \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}I(X \in \mathcal{X}_j) \right| / \epsilon_T > 1 \right) \quad (9.32)$$

$$\leq 2 \sum_{T \in \mathcal{T}_N} \exp(-2n\epsilon_T^2), \quad (9.33)$$

where  $\epsilon_T$  means  $\epsilon$  is a function of  $T$ . For any  $\delta \in (0, 1)$ , we have, with probability at least  $1 - \delta/4$ ,

$$\forall T \in \mathcal{T}_N, \left| \frac{1}{n} \sum_{i=1}^n I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}I(X \in \mathcal{X}_j) \right| \leq \sqrt{\frac{[[T]] \log 2 + \log(8/\delta)}{2n}}$$

where  $[[T]] > 0$  is the prefix code of  $T$  given in (9.6).

From Assumption 9.1, since  $\Omega_{\mathcal{X}_j} \in \Lambda_j$ , we have that

$$\max_{1 \leq j \leq m_T} \log |\Omega_{\mathcal{X}_j}| \leq L_n \quad (9.34)$$

Therefore, with probability at least  $1 - \delta/4$ ,

$$A_2 \leq L_n m_T \sqrt{\frac{[[T]] \log 2 + \log(8/\delta)}{2n}}. \quad (9.35)$$

Next, we analyze the term  $A_1$ . Since

$$\max_{1 \leq j \leq m_T} \|\Omega_{\mathcal{X}_j}\|_1 \leq L_n. \quad (9.36)$$

we only need to bound the term  $\|S_{j,n} - \bar{S}_j\|_\infty$ . By Assumption 9.2 and the union bound, we have, for any  $\epsilon > 0$ ,

$$\begin{aligned} & \mathbb{P}(\|S_{j,n} - \bar{S}_j\|_\infty > \epsilon) \\ & \leq \mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n Y^{(i)} y_i^T I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}[Y Y^T I(X \in \mathcal{X}_j)]\right\|_\infty > \frac{\epsilon}{4}\right) \end{aligned} \quad (9.37)$$

$$+ \mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n Y^{(i)} \mu_{\mathcal{X}_j}^T I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}[Y \mu_{\mathcal{X}_j}^T I(X \in \mathcal{X}_j)]\right\|_\infty > \frac{\epsilon}{4}\right) \quad (9.38)$$

$$+ \mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n \mu_{\mathcal{X}_j} y_i^T I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}[\mu_{\mathcal{X}_j} Y^T I(X \in \mathcal{X}_j)]\right\|_\infty > \frac{\epsilon}{4}\right) \quad (9.39)$$

$$+ \mathbb{P}\left(\left\|\frac{1}{n} \sum_{i=1}^n \mu_{\mathcal{X}_j} \mu_{\mathcal{X}_j}^T I(X^{(i)} \in \mathcal{X}_j) - \mathbb{E}[\mu_{\mathcal{X}_j} \mu_{\mathcal{X}_j}^T I(X \in \mathcal{X}_j)]\right\|_\infty > \frac{\epsilon}{4}\right) \quad (9.40)$$

Using the fact that  $\|\mu\|_\infty \leq B$  and the Assumption 9.2, we can apply Bernstein's exponential inequality on (9.37), (9.38), and (9.39). Also, since the indicator function is bounded, we can apply Hoeffding's inequality on (9.40). In this way we obtain

$$\mathbb{P}(\|S_{j,n} - \bar{S}_j\|_\infty > \epsilon) \quad (9.41)$$

$$\leq 2p^2 \exp\left(-\frac{1}{32} \left(\frac{n\epsilon^2}{v_2 + M_2\epsilon}\right)\right) \quad (9.42)$$

$$+ 4p^2 \exp\left(-\frac{1}{32B^2} \left(\frac{n\epsilon^2}{v_1 + M_1\epsilon}\right)\right) + 2p^2 \exp\left(-\frac{2n\epsilon^2}{B^4}\right). \quad (9.43)$$

Therefore, for any  $\delta \in (0, 1)$ , we have, for any  $\epsilon \rightarrow 0$  as  $n$  goes to infinity, with probability at least  $1 - \delta/4$

$$\forall T \in \mathcal{T}_N, \|S_{j,n} - \bar{S}_j\|_\infty \quad (9.44)$$

$$\leq (8\sqrt{v_2}) \cdot \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(24/\delta)}{n}} \quad (9.45)$$

$$+ (8B\sqrt{v_1}) \cdot \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(48/\delta)}{n}} \quad (9.46)$$

$$+ B^2 \cdot \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(24/\delta)}{2n}}. \quad (9.47)$$

Combined with (9.36), we get that

$$A_1 \leq C_1 L_n m_T \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(48/\delta)}{n}} \quad (9.48)$$

where  $C_1 = 8\sqrt{v_2} + 8B\sqrt{v_1} + B^2$ .

Since the above analysis holds uniformly over  $\mathcal{T}_N$ , when choosing

$$\text{pen}(T) = (C_1 + 1)L_n m_T \sqrt{\frac{[[T]] \log 2 + 2 \log p + \log(48/\delta)}{n}}, \quad (9.49)$$

we then get, with probability at least  $1 - \delta/2$ ,

$$\sup_{T \in \mathcal{T}_N, \mu_j \in M_j, \Omega_j \in \Lambda_j} \left| R(T, \mu_T, \Omega_T) - \hat{R}(T, \mu_T, \Omega_T) \right| \leq \text{pen}(T) \quad (9.50)$$

for large enough  $n$ .

Given a DPT  $T$ , we define

$$\mu_T^0, \Omega_T^0 = \arg \min_{\mu_T \in M_j, \Omega_T \in \Lambda_j} R(T, \mu_T, \Omega_T). \quad (9.51)$$

From the uniform deviation inequality in (9.50), we have, for large enough  $n$ : for any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ ,

$$R(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) \leq \hat{R}(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) + \text{pen}(\hat{T}) \quad (9.52)$$

$$= \inf_{T \in \mathcal{T}_N, \mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} \left\{ \hat{R}(T, \mu_T, \Omega_T) + \text{pen}(T) \right\} \quad (9.53)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ \hat{R}(T, \mu_T^0, \Omega_T^0) + \text{pen}(T) \right\} \quad (9.54)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ R(T, \mu_T^0, \Omega_T^0) + 2\text{pen}(T) \right\} \quad (9.55)$$

$$= \inf_{T \in \mathcal{T}_N} \left\{ \inf_{\mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} (R(T, \mu_T, \Omega_T) + 2\text{pen}(T)) \right\}. \quad (9.56)$$

The desired result of the theorem follows by subtracting  $R^*$  from both sides.

#### 9.7.0.2 Proof of Theorem 9.2

From (9.50) we have, for large enough  $n$ , on the dataset  $\mathcal{D}_1$ , with probability at least  $1 - \delta/4$

$$\sup_{T \in \mathcal{T}_N, \mu_j \in M_j, \Omega_j \in \Lambda_j} \left| R(T, \mu_T, \Omega_T) - \hat{R}(T, \mu_T, \Omega_T) \right| \leq \phi_n(T). \quad (9.57)$$

Following the same line of analysis, we can also get that on the validation dataset  $\mathcal{D}_2$ , with probability at least  $1 - \delta/4$ ,

$$\sup_{T \in \mathcal{T}_N} \left| R(T, \hat{\mu}_T, \hat{\Omega}_T) - \hat{R}_{\text{out}}(T, \hat{\mu}_T, \hat{\Omega}_T) \right| \leq \phi_n(T) \quad (9.58)$$

for large enough  $n$ . Here  $\hat{\mu}_T, \hat{\Omega}_T$  are as defined in (9.10).

Given a DPT  $T$ , we define

$$\mu_T^0, \Omega_T^0 = \arg \min_{\mu_T \in M_j, \Omega_T \in \Lambda_j} R(T, \mu_T, \Omega_T). \quad (9.59)$$



Using the fact that

$$\hat{T} = \operatorname{argmin}_{T \in \mathcal{T}_N} \hat{R}_{\text{out}}(T, \hat{\mu}_T, \hat{\Omega}_T), \quad (9.60)$$

we have, for large enough  $n$  and any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ ,

$$R(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) \leq \hat{R}_{\text{out}}(\hat{T}, \hat{\mu}_{\hat{T}}, \hat{\Omega}_{\hat{T}}) + \phi_n(\hat{T}) \quad (9.61)$$

$$= \inf_{T \in \mathcal{T}_N} \hat{R}_{\text{out}}(T, \hat{\mu}_T, \hat{\Omega}_T) + \phi_n(\hat{T}) \quad (9.62)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ R(T, \hat{\mu}_T, \hat{\Omega}_T) + \phi_n(T) \right\} + \phi_n(\hat{T}) \quad (9.63)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ \hat{R}(T, \hat{\mu}_T, \hat{\Omega}_T) + \phi_n(T) + \phi_n(T) \right\} + \phi_n(\hat{T}) \quad (9.64)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ \hat{R}(T, \mu_T^0, \Omega_T^0) + \phi_n(T) + \phi_n(T) \right\} + \phi_n(\hat{T}) \quad (9.65)$$

$$\leq \inf_{T \in \mathcal{T}_N} \left\{ 3\phi_n(T) + \inf_{\mu_{\mathcal{X}_j} \in M_j, \Omega_{\mathcal{X}_j} \in \Lambda_j} R(T, \mu_T, \Omega_T) \right\} + \phi_n(\hat{T}).$$

The result follows by subtracting  $R^*$  from both sides.

### 9.7.0.3 Proof of Theorem 9.3

For any  $T \in \mathcal{T}_N$ ,  $\Pi(T^*) \not\subseteq \Pi(T)$ , there must exist a subregion  $\mathcal{X}' \in \Pi(T)$  such that no  $\mathcal{A} \in \Pi(T^*)$  satisfies  $\mathcal{X}' \subset \mathcal{A}$ . We can thus find a minimal class of disjoint subregions  $\{\mathcal{X}_1^0, \dots, \mathcal{X}_{k'}^0\} \in \Pi(T^*)$ , such that

$$\mathcal{X}' \subset \cup_{i=1}^{k'} \mathcal{X}_i^0, \quad (9.66)$$

where  $k' \geq 2$ . We define  $\mathcal{X}_i^* = \mathcal{X}_i^0 \cap \mathcal{X}'$  for  $i = 1, \dots, k'$ . Then we have

$$\mathcal{X}' = \cup_{i=1}^{k'} \mathcal{X}_i^*. \quad (9.67)$$

Let  $\{\mu_{\mathcal{X}_j}^*, \Omega_{\mathcal{X}_j}^*\}_{j=1}^{k'}$  be the true parameters on  $\mathcal{X}_1^0, \dots, \mathcal{X}_{k'}^0$ . We denote by  $R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*)$  the risk of  $\mu_{T^*}^*$  and  $\Omega_{T^*}^*$  on the subregion  $\mathcal{X}'$ , so that

$$\begin{aligned} R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) &= \sum_{j=1}^{k'} \mathbb{E} \left[ \left( \operatorname{tr} \left[ \Omega_{\mathcal{X}_j}^* ((Y - \mu_{\mathcal{X}_j}^*)(Y - \mu_{\mathcal{X}_j}^*)^T) \right] - \log |\Omega_{\mathcal{X}_j}^*| \right) \cdot I(X \in \mathcal{X}_j^*) \right] \\ &= p \mathbb{P}(X \in \mathcal{X}') - \sum_{j=1}^{k'} \mathbb{P}(X \in \mathcal{X}_j^*) \log |\Omega_{\mathcal{X}_j}^*|. \end{aligned} \quad (9.68)$$

Since the DPT  $T$  does not further partition  $\mathcal{X}'$ , we have, for any  $\mu_T, \Omega_T \in \mathcal{M}_T$

$$\begin{aligned} R(\mathcal{X}', \mu_T, \Omega_T) &= \sum_{j=1}^{k'} \mathbb{E} \left[ \left( \operatorname{tr} \left[ \Omega_T ((Y - \mu_T)(Y - \mu_T)^T) \right] - \log |\Omega_T| \right) \cdot I(X \in \mathcal{X}_j^*) \right] \\ &= \sum_{j=1}^{k'} \mathbb{E} \left[ \left( \operatorname{tr} \left[ \Omega_T ((Y - \mu_T)(Y - \mu_T)^T) \right] \right) \cdot I(X \in \mathcal{X}_j^*) \right] - \mathbb{P}(X \in \mathcal{X}') \log |\Omega_T|. \end{aligned}$$

Using the decomposition

$$\begin{aligned} & (Y - \mu_T)(Y - \mu_T)^T \\ &= (Y - \mu_{\mathcal{X}_j^*}^*)(Y - \mu_{\mathcal{X}_j^*}^*)^T + (Y - \mu_{\mathcal{X}_j^*}^*)(\mu_{\mathcal{X}_j^*}^* - \mu_T)^T \end{aligned} \quad (9.69)$$

$$+ (\mu_{\mathcal{X}_j^*}^* - \mu_T)(Y - \mu_{\mathcal{X}_j^*}^*)^T + (\mu_{\mathcal{X}_j^*}^* - \mu_T)(\mu_{\mathcal{X}_j^*}^* - \mu_T)^T, \quad (9.70)$$

we obtain

$$\begin{aligned} & \sum_{j=1}^{k'} \mathbb{E} \left[ \left( \text{tr} \left[ \Omega_T \left( (Y - \mu_T)(Y - \mu_T)^T \right) \right] \right) \cdot I(X \in \mathcal{X}_j^*) \right] \\ &= \sum_{j=1}^{k'} \mathbb{P} \left( X \in \mathcal{X}_j^* \right) \left[ \text{tr}(\Omega_T(\Omega_j^*)^{-1}) + \text{tr}(\Omega_T(\mu_{\mathcal{X}_j^*}^* - \mu_T)(\mu_{\mathcal{X}_j^*}^* - \mu_T)^T) \right]. \end{aligned}$$

Using the bound

$$R(\mathcal{X}', \mu_T, \Omega_T) \geq \max\{R(\mathcal{X}', \mu_{T^*}^*, \Omega_T), R(\mathcal{X}', \mu_T, \Omega_{T^*}^*)\}, \quad (9.71)$$

we proceed by cases.

**Case 1:** The  $\mu$ 's are different. We know that

$$\begin{aligned} & \inf_{\mu_T, \Omega_T \in \mathcal{M}_T} R(\mathcal{X}', \mu_T, \Omega_T) - R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) \\ & \geq \inf_{\mu_T} R(\mathcal{X}', \mu_T, \Omega_{T^*}^*) - R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) \\ &= \inf_{\mu_T} \sum_{j=1}^{k'} \mathbb{P} \left( X \in \mathcal{X}_j^* \right) (\mu_{\mathcal{X}_j^*}^* - \mu_T)^T \Omega_{\mathcal{X}_j^*}^* (\mu_{\mathcal{X}_j^*}^* - \mu_T) \\ & \geq c_1 c_2 \inf_{\mu_T} \sum_{j=1}^{k'} \|\mu_{\mathcal{X}_j^*}^* - \mu_T\|_2^2 \end{aligned} \quad (9.72)$$

where the last inequality follows from that fact that

$$\rho_{\min}(\Omega_{\mathcal{X}_j^*}^*) \geq c_1, \mathbb{P} \left( X \in \mathcal{X}_j^* \right) \geq c_2.$$

It's easy to see that a lower bound of the last term is achieved at  $\bar{\mu}_T$ ,

$$\bar{\mu}_T = \frac{1}{k'} \sum_{j=1}^{k'} \mu_{\mathcal{X}_j^*}^*. \quad (9.73)$$

Furthermore, for any two DPTs  $T$  and  $T'$ , if  $\Pi(T) \subset \Pi(T')$  it's clear that

$$\inf_{\mu_T, \Omega_T \in \mathcal{M}_T} R(T, \mu_T, \Omega_T) \geq \inf_{\mu_{T'}, \Omega_{T'} \in \mathcal{M}_{T'}} R(T', \mu_{T'}, \Omega_{T'}). \quad (9.74)$$

Therefore, in the sequel, without loss of generality we only need to consider the case  $k' = 2$ .

The result in this case then follows from the fact that

$$\sum_{j=1}^2 \|\mu_{\mathcal{X}_j}^* - \bar{\mu}_T\|_2^2 = \frac{1}{2} \|\mu_{\mathcal{X}_1}^* - \mu_{\mathcal{X}_2}^*\|_2^2 \geq \frac{c_3}{2}. \quad (9.75)$$

**Case 2:** The  $\Omega$ 's are different. In this case, we have

$$\inf_{\mu_T, \Omega_T \in \mathcal{M}_T} R(\mathcal{X}', \mu_T, \Omega_T) - R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) \quad (9.76)$$

$$\geq \inf_{\Omega_T} R(\mathcal{X}', \mu_{T^*}^*, \Omega_T) - R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) \quad (9.77)$$

$$\begin{aligned} &= \inf_{\Omega_T} \sum_{j=1}^{k'} \mathbb{P}(X \in \mathcal{X}_j^*) \left( \text{tr} \left[ \Omega_{\mathcal{X}_j^*}^{-1} (\Omega_T - \Omega_{\mathcal{X}_j^*}^*) \right] - \left( \log |\Omega_T| - \log |\Omega_{\mathcal{X}_j^*}^*| \right) \right) \\ &\geq c_2 \inf_{\Omega_T} \sum_{j=1}^{k'} \left( \text{tr} \left[ \Omega_{\mathcal{X}_j^*}^{-1} (\Omega_T - \Omega_{\mathcal{X}_j^*}^*) \right] - \left( \log |\Omega_T| - \log |\Omega_{\mathcal{X}_j^*}^*| \right) \right) \end{aligned} \quad (9.78)$$

$$\geq c_2 \inf_{\Sigma_T} \sum_{j=1}^{k'} \left( \text{tr} \left[ \Sigma_{\mathcal{X}_j^*}^* (\Sigma_T^{-1} - \Omega_{\mathcal{X}_j^*}^*) \right] + \log \frac{|\Sigma_T|}{|\Sigma_{\mathcal{X}_j^*}^*|} \right) \quad (9.79)$$

$$= c_2 \inf_{\Sigma_T} \sum_{j=1}^{k'} \left( \text{tr} \left( \Sigma_{\mathcal{X}_j^*}^* \Sigma_T^{-1} \right) + \log \frac{|\Sigma_T|}{|\Sigma_{\mathcal{X}_j^*}^*|} - p \right) \quad (9.80)$$

where  $\Sigma_T = \Omega_T^{-1}$ .

As before, we only need to consider the case  $k' = 2$ . A lower bound of the last term is achieved at

$$\bar{\Sigma}_T = \frac{\Sigma_{\mathcal{X}_1^*} + \Sigma_{\mathcal{X}_2^*}}{2}. \quad (9.81)$$

Plugging in  $\bar{\Sigma}_T$ , we get

$$\begin{aligned} &\inf_{\Sigma_T} \sum_{j=1}^2 \left( \text{tr} \left( \Sigma_{\mathcal{X}_j^*}^* \Sigma_T^{-1} \right) + \log \frac{|\Sigma_T|}{|\Sigma_{\mathcal{X}_j^*}^*|} - p \right) \\ &\geq \sum_{j=1}^2 \left( \text{tr} \left( \Sigma_{\mathcal{X}_j^*}^* \bar{\Sigma}_T^{-1} \right) + \log \frac{|\bar{\Sigma}_T|}{|\Sigma_{\mathcal{X}_j^*}^*|} - p \right) \end{aligned} \quad (9.82)$$

$$\begin{aligned} &= \text{tr} \left( (2\bar{\Sigma}_T - \Sigma_{\mathcal{X}_2^*}^*) \bar{\Sigma}_T^{-1} \right) + \log \frac{|\bar{\Sigma}_T|}{|\Sigma_{\mathcal{X}_1^*}^*|} - p + \text{tr} \left( \Sigma_{\mathcal{X}_2^*}^* \bar{\Sigma}_T^{-1} \right) + \log \frac{|\bar{\Sigma}_T|}{|\Sigma_{\mathcal{X}_2^*}^*|} - p \\ &= \log \frac{|\bar{\Sigma}_T|}{|\Sigma_{\mathcal{X}_1^*}^*|} + \log \frac{|\bar{\Sigma}_T|}{|\Sigma_{\mathcal{X}_2^*}^*|} \end{aligned} \quad (9.83)$$

$$= 2 \log \left| \frac{\Sigma_{\mathcal{X}_1^*} + \Sigma_{\mathcal{X}_2^*}}{2} \right| - \log |\Sigma_{\mathcal{X}_1^*}^*| - \log |\Sigma_{\mathcal{X}_2^*}^*| \quad (9.84)$$

$$\geq c_4 \quad (9.85)$$

where the last inequality follows from the given assumption.

Therefore, we have

$$\inf_{\mu_T, \Omega_T \in \mathcal{M}_T} R(\mathcal{X}', \mu_T, \Omega_T) - R(\mathcal{X}', \mu_{T^*}^*, \Omega_{T^*}^*) \geq c_2 c_4. \quad (9.86)$$

The theorem is obtained by combining the two cases.

## Part V

### REGULARIZATION PARAMETER SELECTION



## STARS: STABILITY APPROACH FOR REGULARIZATION SELECTION

---

All the methods discussed in this thesis have at least one tuning parameter. A challenging problem is to choose the regularization parameter in a data-dependent way. The standard techniques include  $K$ -fold cross-validation ( $K$ -CV), Akaike information criterion (AIC), and Bayesian information criterion (BIC). Though cross-validation works fine for high dimensional supervised methods, these methods are not suitable in high dimensional unsupervised settings. In this chapter, we present StARS: a new stability-based method for choosing the regularization parameter in high dimensional inference. The method is quite general and can be applied to different kinds of parametric and nonparametric models. In this chapter, we only consider the problem of estimating high dimensional undirected graphs as a pilot study. The method has a clear interpretation: we use the least amount of regularization that simultaneously makes a graph sparse and replicable under random sampling. This interpretation requires essentially no conditions. Under mild conditions, we show that StARS is partially sparsistent in terms of graph estimation: i.e. with high probability, all the true edges will be included in the selected model even when the graph size diverges with the sample size. Empirically, the performance of StARS is compared with the state-of-the-art model selection procedures, including  $K$ -CV, AIC, and BIC, on both synthetic data and a real microarray dataset. StARS outperforms all these competing procedures.

### 10.1 INTRODUCTION

Undirected graphical models have emerged as a useful tool because they allow for a stochastic description of complex associations in high-dimensional data. For example, biological processes in a cell lead to complex interactions among gene products. It is of interest to determine which features of the system are conditionally independent. Such problems require us to infer an undirected graph from i.i.d. observations. Each node in this graph corresponds to a random variable and the existence of an edge between a pair of nodes represent their conditional independence relationship.

Gaussian graphical models [Dempster, 1972, Whittaker, 1990, Edwards, 1995, Lauritzen, 1996] are by far the most popular approach for learning high dimensional undirected graph structures. Under the Gaussian assumption, the graph can be estimated using the sparsity pattern of the inverse covariance matrix. If two variables are conditionally independent, the corresponding element of the inverse covariance matrix is zero. In many applications, estimating the the inverse covariance matrix is statistically challenging because the number of features measured may be much larger than the number of collected samples. To handle this challenge, the *graphical lasso* or *glasso* [Friedman et al., 2007, Yuan and Lin, 2007, Banerjee et al., 2008] is rapidly becoming a popular method for estimating sparse undirected graphs. To use this method, however, the user must specify a regularization parameter  $\lambda$  that controls the sparsity of the graph. The choice of  $\lambda$  is critical since different  $\lambda$ 's may lead to different scientific conclusions of the statistical inference. Other methods for estimating high dimensional graphs include [Meinshausen and Bühlmann, 2006, Peng et al., 2009, Liu et al., 2009a]. They also require the user to specify a regularization parameter.

The standard methods for choosing the regularization parameter are AIC [Akaike, 1973], BIC [Schwarz, 1978] and cross validation [Efron, 1982]. Though these methods have good theoretical properties in low dimensions, they are not suitable for high dimensional problems. In regression, cross-validation has been shown to overfit the data [Wasserman and Roeder, 2009]. Likewise, AIC and BIC tend to perform poorly when the dimension is large relative to the sample size. Our simulations confirm that these methods perform poorly when used with glasso.

A new approach to model selection, based on model stability, has recently generated some interest in the literature [Lange et al., 2004]. The idea, as we develop it, is based on subsampling [Politis et al., 1999] and builds on the approach of Meinshausen and Bühlmann [2010]. We draw many random subsamples and construct a graph from each subsample (unlike  $K$ -fold cross-validation, these subsamples are overlapping). We choose the regularization parameter so that the obtained graph is sparse and there is not too much variability across subsamples. More precisely, we start with a large regularization which corresponds to an empty, and hence highly stable, graph. We gradually reduce the amount of regularization until there is a small but acceptable amount of variability of the graph across subsamples. In other words, we regularize to the point that we control the dissonance between graphs. The procedure is named StARS: Stability Approach to Regularization Selection. We study the performance of StARS by simulations and theoretical analysis in Sections 4 and 5. Although we focus here on graphical models, StARS is quite general and can be adapted to other settings including regression, classification, clustering, and dimensionality reduction.



In the context of clustering, results of stability methods have been mixed. Weaknesses of stability have been shown in [Ben-david et al., 2006]. However, the approach was successful for density-based clustering [Rinaldo and Wasserman, 2009a]. For graph selection, Meinshausen and Bühlmann [2010] also used a stability criterion; however, their approach differs from StARS in its fundamental conception. They use subsampling to produce a new and more stable regularization path then select a regularization parameter from this newly created path, whereas we propose to use subsampling to directly select one regularization parameter from the original path. Our aim is to ensure that the selected graph is sparse, but inclusive, while they aim to control the familywise type I errors. As a consequence, their goal is contrary to ours: instead of selecting a larger graph that contains the true graph, they try to select a smaller graph that is contained in the true graph. As we will discuss in Section 3, in specific application domains like gene regulatory network analysis, our goal for graph selection is more natural.

In Section 10.2 we review the basic notion of estimating high dimensional undirected graphs; in Section 10.3 we develop StARS; in Section 10.4 we present a theoretical analysis of the proposed method; and in Section 10.5 we report experimental results on both simulated data and a gene microarray dataset, where the problem is to construct gene regulatory network based on natural variation of the expression levels of human genes.

## 10.2 ESTIMATING A HIGH-DIMENSIONAL UNDIRECTED GRAPH

Let  $X = (X_1, \dots, X_d)^T$  be a random vector with distribution  $P$ . The undirected graph  $G = (V, E)$  associated with  $P$  has vertices  $V = \{X_1, \dots, X_d\}$  and a set of edges  $E$  corresponding to pairs of vertices. In this paper, we also interchangeably use  $E$  to denote the adjacency matrix of the graph  $G$ . The edge corresponding to  $X_j$  and  $X_k$  is absent if  $X_j$  and  $X_k$  are conditionally independent given the other coordinates of  $X$ . The graph estimation problem is to infer  $E$  from i.i.d. observed data  $X^{(1)}, \dots, X^{(n)}$  where  $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})^T$ .

Suppose now that  $P$  is Gaussian with mean vector  $\mu$  and covariance matrix  $\Sigma$ . Then the edge corresponding to  $X_j$  and  $X_k$  is absent if and only if  $\Omega_{jk} = 0$  where  $\Omega = \Sigma^{-1}$ . Hence, to estimate the graph we only need to estimate the sparsity pattern of  $\Omega$ . When  $d$  could diverge with  $n$ , estimating  $\Omega$  is difficult. A popular approach is the *graphical lasso* or *glasso* [Friedman et al., 2007, Yuan and Lin, 2007, Banerjee et al., 2008]. Using glasso, we estimate  $\Omega$  as follows: Ignoring constants, the log-likelihood (after maximizing over  $\mu$ ) can be written as

$$\ell(\Omega) = \log |\Omega| - \text{trace}(\hat{\Sigma}\Omega)$$

where  $\widehat{\Sigma}$  is the sample covariance matrix. With a positive regularization parameter  $\lambda$ , the glasso estimator  $\widehat{\Omega}(\lambda)$  is obtained by minimizing the regularized negative log-likelihood

$$\widehat{\Omega}(\lambda) = \arg \min_{\Omega \succ 0} \left\{ -\ell(\Omega) + \lambda \|\Omega\|_1 \right\} \quad (10.1)$$

where  $\|\Omega\|_1 = \sum_{j,k} |\Omega_{jk}|$  is the elementwise  $\ell_1$ -norm of  $\Omega$ . The estimated graph  $\widehat{G}(\lambda) = (V, \widehat{E}(\lambda))$  is then easily obtained from  $\widehat{\Omega}(\lambda)$ : for  $i \neq j$ , an edge  $(i, j) \in \widehat{E}(\lambda)$  if and only if the corresponding entry in  $\widehat{\Omega}(\lambda)$  is nonzero. [Friedman et al. \[2007\]](#) give a fast algorithm for calculating  $\widehat{\Omega}(\lambda)$  over a grid of  $\lambda$ s ranging from small to large. By taking advantage of the fact that the objective function in (10.1) is convex, their algorithm iteratively estimates a single row (and column) of  $\Omega$  in each iteration by solving a lasso regression [[Tibshirani, 1996](#)]. The resulting regularization path  $\widehat{\Omega}(\lambda)$  for all  $\lambda$ s has been shown to have excellent theoretical properties [[Rothman et al., 2008](#), [Ravikumar et al., 2009b](#)]. For example, [Ravikumar et al. \[2009b\]](#) show that, if the regularization parameter  $\lambda$  satisfies a certain rate, the corresponding estimator  $\widehat{\Omega}(\lambda)$  could recover the true graph with high probability. However, these types of results are either asymptotic or non-asymptotic but with very large constants. They are not practical enough to guide the choice of the regularization parameter  $\lambda$  in finite-sample settings.

### 10.3 REGULARIZATION SELECTION

In Equation (10.1), the choice of  $\lambda$  is critical because  $\lambda$  controls the sparsity level of  $\widehat{G}(\lambda)$ . Larger values of  $\lambda$  tend to yield sparser graphs and smaller values of  $\lambda$  yield denser graphs. It is convenient to define  $\Lambda = 1/\lambda$  so that small  $\Lambda$  corresponds to a more sparse graph. In particular,  $\Lambda = 0$  corresponds to the empty graph with no edges. Given a grid of regularization parameters  $\mathcal{G}_n = \{\Lambda_1, \dots, \Lambda_K\}$ , our goal of graph regularization parameter selection is to choose one  $\widehat{\Lambda} \in \mathcal{G}_n$ , such that the true graph  $E$  is contained in  $\widehat{E}(\widehat{\Lambda})$  with high probability. In other words, we want to “overselect” instead of “underselect”. Such a choice is motivated by application problems like gene regulatory networks reconstruction, in which we aim to study the interactions of many genes. For these types of studies, we tolerate some false positives but not false negatives. Specifically, it is acceptable that an edge presents but the two genes corresponding to this edge do not really interact with each other. Such false positives can generally be screened out by more fine-tuned downstream biological experiments. However, if one important interaction edge is omitted at the beginning, it's very difficult for us to re-discover it by follow-up analysis. There is also a tradeoff: we want to select a denser graph which contains the true graph with high probability. At the same time, we want the graph to be as sparse as possible so that important information will not be

buried by massive false positives. Based on this rationale, an “underselect” method, like the approach of [Meinshausen and Bühlmann \[2010\]](#), does not really fit our goal. In the following, we start with an overview of several state-of-the-art regularization parameter selection methods for graphs. We then introduce our new StARS approach.

### 10.3.1 Existing Methods

The regularization parameter is often chosen using AIC or BIC. Let  $\hat{\Omega}(\Lambda)$  denote the estimator corresponding to  $\Lambda$ . Let  $\text{df}(\Lambda)$  denote the degree of freedom (or the effective number of free parameters) of the corresponding Gaussian model. AIC chooses  $\Lambda$  as

$$\text{(AIC)} \quad \hat{\Lambda} = \arg \min_{\Lambda \in \mathcal{G}_n} \{-2\ell(\hat{\Omega}(\Lambda)) + 2\text{df}(\Lambda)\}, \quad (10.2)$$

and BIC chooses  $\Lambda$  as

$$\text{(BIC)} \quad \hat{\Lambda} = \arg \min_{\Lambda \in \mathcal{G}_n} \{-2\ell(\hat{\Omega}(\Lambda)) + \text{df}(\Lambda) \cdot \log n\}. \quad (10.3)$$

The usual theoretical justification for these methods assumes that the dimension  $d$  is fixed as  $n$  increases; however, in the case where  $d > n$  this justification is not applicable. In fact, it’s even not straightforward how to estimate the degree of freedom  $\text{df}(\Lambda)$  when  $d$  is larger than  $n$ . A common practice is to calculate  $\text{df}(\Lambda)$  as  $\text{df}(\Lambda) = m(\Lambda)(m(\Lambda) - 1)/2 + p$  where  $m(\Lambda)$  denotes the number of nonzero elements of  $\hat{\Omega}(\Lambda)$ . As we will see in our experiments, AIC and BIC tend to select overly dense graphs in high dimensions.

Another popular method is  $K$ -fold cross-validation ( $K$ -CV). For this procedure the data is partitioned into  $K$  subsets. Of the  $K$  subsets one is retained as the validation data, and the remaining  $K - 1$  ones are used as training data. For each  $\Lambda \in \mathcal{G}_n$ , we estimate a graph on the  $K - 1$  training sets and evaluate the negative log-likelihood on the retained validation set. The results are averaged over all  $K$  folds to obtain a single CV score. We then choose  $\Lambda$  to minimize the CV score over the whole grid  $\mathcal{G}_n$ . In regression, cross-validation has been shown to overfit [[Wasserman and Roeder, 2009](#)]. Our experiments will confirm this is true for graph estimation as well.

### 10.3.2 StARS: Stability Approach to Regularization Selection

The StARS approach is to choose  $\Lambda$  based on stability. When  $\Lambda$  is 0, the graph is empty and two datasets from  $P$  would both yield the same graph. As we increase  $\Lambda$ , the variability of the graph increases and hence the stability decreases. We increase  $\Lambda$  just until the point where the graph becomes variable as measured by the stability. StARS leads to a concrete rule for choosing  $\Lambda$ .

Let  $b = b(n)$  be such that  $1 < b(n) < n$ . We draw  $N$  random subsamples  $S_1, \dots, S_N$  from  $X^{(1)}, \dots, X^{(n)}$ , each of size  $b$ . There are  $\binom{n}{b}$  such subsamples. Theoretically one uses all  $\binom{n}{b}$  subsamples. However, [Politis et al. \[1999\]](#) show that it suffices in practice to choose a large number  $N$  of subsamples at random. Note that, unlike bootstrapping [\[Efron, 1982\]](#), each subsample is drawn without replacement. For each  $\Lambda \in \mathcal{G}_n$ , we construct a graph using the glasso for each subsample. This results in  $N$  estimated edge matrices  $\hat{E}_1^b(\Lambda), \dots, \hat{E}_N^b(\Lambda)$ . Focus for now on one edge  $(s, t)$  and one value of  $\Lambda$ . Let  $\psi^\Lambda(\cdot)$  denote the glasso algorithm with the regularization parameter  $\Lambda$ . For any subsample  $S_j$  let  $\psi_{st}^\Lambda(S_j) = 1$  if the algorithm puts an edge and  $\psi_{st}^\Lambda(S_j) = 0$  if the algorithm does not put an edge between  $(s, t)$ . Define

$$\theta_{st}^b(\Lambda) = \mathbb{P}(\psi_{st}^\Lambda(X^{(1)}, \dots, X^{(b)}) = 1).$$

To estimate  $\theta_{st}^b(\Lambda)$ , we use a U-statistic of order  $b$ , namely,

$$\hat{\theta}_{st}^b(\Lambda) = \frac{1}{N} \sum_{j=1}^N \psi_{st}^\Lambda(S_j).$$

Now define the parameter  $\zeta_{st}^b(\Lambda) = 2\theta_{st}^b(\Lambda)(1 - \theta_{st}^b(\Lambda))$  and let  $\hat{\zeta}_{st}^b(\Lambda) = 2\hat{\theta}_{st}^b(\Lambda)(1 - \hat{\theta}_{st}^b(\Lambda))$  be its estimate. Then  $\zeta_{st}^b(\Lambda)$ , in addition to being twice the variance of the Bernoulli indicator of the edge  $(s, t)$ , has the following nice interpretation: For each pair of graphs, we can ask how often they disagree on the presence of the edge:  $\zeta_{st}^b(\Lambda)$  is the fraction of times they disagree. For  $\Lambda \in \mathcal{G}_n$ , we regard  $\zeta_{st}^b(\Lambda)$  as a measure of instability of the edge across subsamples, with  $0 \leq \zeta_{st}^b(\Lambda) \leq 1/2$ .

Define the total instability by averaging over all edges:

$$\hat{D}_b(\Lambda) = \frac{\sum_{s < t} \hat{\zeta}_{st}^b}{\binom{d}{2}}.$$

Clearly on the boundary  $\hat{D}_b(0) = 0$ , and  $\hat{D}_b(\Lambda)$  generally will increase as  $\Lambda$  increases. However, when  $\Lambda$  gets very large, all the graphs will become dense and  $\hat{D}_b(\Lambda)$  will begin to decrease. Subsample stability for large  $\Lambda$  is essentially an artifact. We are interested in stability for sparse graphs not dense graphs. For this reason we monotinize  $\hat{D}_b(\Lambda)$  by defining

$$\overline{D}_b(\Lambda) = \sup_{0 \leq t \leq \Lambda} \hat{D}_b(t).$$

Finally, our StARS approach chooses  $\Lambda$  by defining

$$\hat{\Lambda}_s = \sup \left\{ \Lambda : \overline{D}_b(\Lambda) \leq \beta \right\}$$

for a specified cut point value  $\beta$ .

It may seem that we have merely replaced the problem of choosing  $\Lambda$  with the problem of choosing  $\beta$ , but  $\beta$  is an interpretable quantity and we always set a default value  $\beta = 0.05$ . One thing to note is that all quantities  $\hat{E}, \hat{\theta}, \hat{\xi}, \hat{D}$  depend on the subsampling block size  $b$ . Since StARS is based on subsampling, the effective sample size for estimating the selected graph is  $b$  instead of  $n$ . Compared with methods like BIC and AIC which fully utilize all  $n$  data points. StARS has some efficiency loss in low dimensions. However, in high dimensional settings, the gain of StARS on better graph selection significantly dominate this efficiency loss. This fact is confirmed by our experiments.

#### 10.4 THEORETICAL PROPERTIES

The StARS procedure is quite general and can be applied with any graph estimation algorithms. Here, we provide its theoretical properties. We start with a key theorem which establishes the rates of convergence of the estimated stability quantities to their population means. We then discuss the implication of this theorem on general graph regularization selection problems.

Let  $\Lambda$  be an element in the grid  $\mathcal{G}_n = \{\Lambda_1, \dots, \Lambda_K\}$  where  $K$  is a polynomial of  $n$ . We denote  $D_b(\Lambda) = \mathbb{E}(\hat{D}_b(\Lambda))$ . The quantity  $\hat{\xi}_{st}^b(\Lambda)$  is an estimate of  $\xi_{st}^b(\Lambda)$  and  $\hat{D}_b(\Lambda)$  is an estimate of  $D_b(\Lambda)$ . Standard  $U$ -statistic theory guarantees that these estimates have good uniform convergence properties to their population quantities:

**Theorem 10.1.** (Uniform Concentration) *The following statements hold with no assumptions on  $P$ . For any  $\delta \in (0, 1)$ , with probability at least  $1 - \delta$ , we have*

$$\forall \Lambda \in \mathcal{G}_n, \max_{s < t} |\hat{\xi}_{st}^b(\Lambda) - \xi_{st}^b(\Lambda)| \leq \sqrt{\frac{18b(2 \log d + \log(2/\delta))}{n}}, \quad (10.4)$$

$$\max_{\Lambda \in \mathcal{G}_n} |\hat{D}_b(\Lambda) - D_b(\Lambda)| \leq \sqrt{\frac{18b(\log K + 4 \log d + \log(1/\delta))}{n}}. \quad (10.5)$$

*Proof.* Note that  $\hat{\theta}_{st}^b(\Lambda)$  is a  $U$ -statistic of order  $b$ . Hence, by Hoeffding's inequality for  $U$ -statistics [Serfling, 1980], we have, for any  $\epsilon > 0$ ,

$$\mathbb{P}(|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)| > \epsilon) \leq 2 \exp(-2n\epsilon^2/b). \quad (10.6)$$

Now  $\hat{\xi}_{st}^b(\Lambda)$  is just a function of the  $U$ -statistic  $\hat{\theta}_{st}^b(\Lambda)$ . Note that

$$|\hat{\xi}_{st}^b(\Lambda) - \xi_{st}^b(\Lambda)| \quad (10.7)$$

$$= 2|\hat{\theta}_{st}^b(\Lambda)(1 - \hat{\theta}_{st}^b(\Lambda)) - \theta_{st}^b(\Lambda)(1 - \theta_{st}^b(\Lambda))| \quad (10.8)$$

$$= 2|\hat{\theta}_{st}^b(\Lambda) - (\hat{\theta}_{st}^b(\Lambda))^2 - \theta_{st}^b(\Lambda) + (\theta_{st}^b(\Lambda))^2| \quad (10.9)$$

$$\leq 2|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)| + 2|(\hat{\theta}_{st}^b(\Lambda))^2 - (\theta_{st}^b(\Lambda))^2| \quad (10.10)$$

$$\leq 2|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)| + 2|(\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda))(\hat{\theta}_{st}^b(\Lambda) + \theta_{st}^b(\Lambda))| \quad (10.11)$$

$$\leq 2|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)| + 4|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)| \quad (10.12)$$

$$= 6|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)|, \quad (10.13)$$

we have  $|\hat{\zeta}_{st}^b(\Lambda) - \zeta_{st}^b(\Lambda)| \leq 6|\hat{\theta}_{st}^b(\Lambda) - \theta_{st}^b(\Lambda)|$ . Using (10.6) and the union bound over all the edges, we obtain: for each  $\Lambda \in \mathcal{G}_n$ ,

$$\mathbb{P}(\max_{s < t} |\hat{\zeta}_{st}^b(\Lambda) - \zeta_{st}^b(\Lambda)| > 6\epsilon) \leq 2d^2 \exp(-2n\epsilon^2/b). \quad (10.14)$$

Using two union bound arguments over the  $K$  values of  $\Lambda$  and all the  $d(d-1)/2$  edges, we have:

$$\mathbb{P}\left(\max_{\Lambda \in \mathcal{G}_n} |\hat{D}_b(\Lambda) - D_b(\Lambda)| \geq \epsilon\right) \quad (10.15)$$

$$\leq |\mathcal{G}_n| \cdot \frac{d(d-1)}{2} \cdot \mathbb{P}(\max_{s < t} |\hat{\zeta}_{st}^b(\Lambda) - \zeta_{st}^b(\Lambda)| > \epsilon) \quad (10.16)$$

$$\leq K \cdot d^4 \cdot \exp(-n\epsilon^2/(18b)). \quad (10.17)$$

Equations (10.4) and (10.5) follow directly from (10.14) and the above exponential probability inequality.  $\square$

Theorem 10.1 allows us to explicitly characterize the high-dimensional scaling of the sample size  $n$ , dimensionality  $d$ , subsampling block size  $b$ , and the grid size  $K$ . More specifically, we get

$$\frac{n}{b \log(nd^4K)} \rightarrow \infty \implies \max_{\Lambda \in \mathcal{G}_n} |\hat{D}_b(\Lambda) - D_b(\Lambda)| \xrightarrow{P} 0 \quad (10.18)$$

by setting  $\delta = 1/n$  in Equation (10.5). From (10.18), let  $c_1, c_2$  be arbitrary positive constants, if  $b = c_1 \sqrt{n}$ ,  $K = n^{c_2}$ , and  $d \leq \exp(n^\gamma)$  for some  $\gamma < 1/2$ , the estimated total stability  $\hat{D}_b(\Lambda)$  still converges to its mean  $D_b(\Lambda)$  uniformly over the whole grid  $\mathcal{G}_n$ .

We now discuss the implication of Theorem 10.1 to graph regularization selection problems. Due to the generality of StARS, we provide theoretical justifications for a whole family of graph estimation procedures satisfying certain conditions. Let  $\psi$  be a graph estimation procedure. We denote  $\hat{E}^b(\Lambda)$  as the estimated edge set using the regularization parameter  $\Lambda$  by applying  $\psi$  on a subsampled dataset with block size  $b$ . To establish graph selection result, we start with two technical assumptions:

(A1)  $\exists \Lambda_o \in \mathcal{G}_n$ , such that  $\max_{\Lambda \leq \Lambda_o \wedge \Lambda \in \mathcal{G}_n} D_b(\Lambda) \leq \beta/2$  for large enough  $n$ .

(A2) For any  $\Lambda \in \mathcal{G}_n$  and  $\Lambda \geq \Lambda_o$ ,  $\mathbb{P}(E \subset \hat{E}^b(\Lambda)) \rightarrow 1$  as  $n \rightarrow \infty$ .

Note that  $\Lambda_o$  here depends on the sample size  $n$  and does not have to be unique. To understand the above conditions, (A1) assumes that there exists a threshold  $\Lambda_o \in \mathcal{G}_n$ , such that the population quantity  $D_b(\Lambda)$  is small for all  $\Lambda \leq \Lambda_o$ . (A2) requires that all estimated graphs using regularization parameters  $\Lambda \geq \Lambda_o$  contain the true graph with high probability. Both assumptions are mild and should be satisfied by most graph estimation algorithm with reasonable behaviors. There is a tradeoff on the design of the subsampling block

size  $b$ . To make (A2) hold, we require  $b$  to be large. However, to make  $\widehat{D}_b(\Lambda)$  concentrate to  $D_b(\Lambda)$  fast, we require  $b$  to be small. Our suggested value is  $b = \lfloor 10\sqrt{n} \rfloor$ , which balances both the theoretical and empirical performance well. The next theorem provides the graph selection performance of StARS:

**Theorem 10.2.** (Partial Sparsistency): *Let  $\psi$  to be a graph estimation algorithm. We assume (A1) and (A2) hold for  $\psi$  using  $b = \lfloor 10\sqrt{n} \rfloor$  and  $|\mathcal{G}_n| = K = n^{c_1}$  for some constant  $c_1 > 0$ . Let  $\widehat{\Lambda}_s \in \mathcal{G}_n$  be the selected regularization parameter using the StARS procedure with a constant cutting point  $\beta$ . Then, if  $d \leq \exp(n^\gamma)$  for some  $\gamma < 1/2$ , we have*

$$\mathbb{P}(E \subset \widehat{E}^b(\widehat{\Lambda}_s)) \rightarrow 1 \text{ as } n \rightarrow \infty. \quad (10.19)$$

*Proof.* We define  $\mathcal{A}_n$  to be the event that  $\max_{\Lambda \in \mathcal{G}_n} |\widehat{D}_b(\Lambda) - D_b(\Lambda)| \leq \beta/2$ . The scaling of  $n, K, b, p$  in the theorem satisfies the L.H.S. of (10.18), which implies that  $\mathbb{P}(\mathcal{A}_n) \rightarrow 1$  as  $n \rightarrow \infty$ .

Using (A1), we know that, on  $\mathcal{A}_n$ ,

$$\max_{\Lambda \leq \Lambda_o \wedge \Lambda \in \mathcal{G}_n} \widehat{D}_b(\Lambda) \leq \max_{\Lambda \in \mathcal{G}_n} |\widehat{D}_b(\Lambda) - D_b(\Lambda)| + \max_{\Lambda \leq \Lambda_o \wedge \Lambda \in \mathcal{G}_n} D_b(\Lambda) \leq \beta. \quad (10.20)$$

This implies that, on  $\mathcal{A}_n$ ,  $\widehat{\Lambda}_s \geq \Lambda_o$ . The result follows by applying (A2) and a union bound.  $\square$

## 10.5 EXPERIMENTAL RESULTS

We now provide empirical evidence to illustrate the usefulness of StARS and compare it with several state-of-the-art competitors, including 10-fold cross-validation (K-CV), BIC, and AIC. For StARS we always use subsampling block size  $b(n) = \lfloor 10 \cdot \sqrt{n} \rfloor$  and set the cut point  $\beta = 0.05$ . We first quantitatively evaluate these methods on two types of synthetic datasets, where the true graphs are known. We then illustrate StARS on a microarray dataset that records the gene expression levels from immortalized B cells of human subjects. On all high dimensional synthetic datasets, StARS significantly outperforms its competitors. On the microarray dataset, StARS obtains a remarkably simple graph while all competing methods select what appear to be overly dense graphs.

### 10.5.1 Synthetic Data

To quantitatively evaluate the graph estimation performance, we adapt the criteria including precision, recall, and  $F_1$ -score from the information retrieval literature. Let  $G = (V, E)$  be a  $d$ -dimensional graph and let  $\widehat{G} = (V, \widehat{E})$  be an estimated graph. We define

$$\text{precision} = \frac{|\widehat{E} \cap E|}{|\widehat{E}|}, \quad \text{recall} = \frac{|\widehat{E} \cap E|}{|E|}, \quad F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$



In other words, Precision is the number of correctly estimated edges divided by the total number of edges in the estimated graph; recall is the number of correctly estimated edges divided by the total number of edges in the true graph; the  $F_1$ -score can be viewed as a weighted average of the precision and recall, where an  $F_1$ -score reaches its best value at 1 and worst score at 0. On the synthetic data where we know the true graphs, we also compare the previous methods with an oracle procedure which selects the optimal regularization parameter by minimizing the total number of different edges between the estimated and true graphs along the full regularization path. Since this oracle procedure requires the knowledge of the truth graph, it is not a practical method. We only present it here to calibrate the inherent challenge of each simulated scenario. To make the comparison fair, once the regularization parameters are selected, we estimate the oracle and StARS graphs only based on a subsampled dataset with size

$$b(n) = \lfloor 10\sqrt{n} \rfloor.$$

In contrast, the  $K$ -CV, BIC, and AIC graphs are estimated using the full dataset. More details about this issue were discussed in Section 10.3.

We generate data from sparse Gaussian graphs, *neighborhood graphs* and *hub graphs*, which mimic characteristics of real-world biological networks. The mean is set to be zero and the covariance matrix  $\Sigma = \Omega^{-1}$ . For both graphs, the diagonal elements of  $\Omega$  are set to be one. More specifically:

1. *Neighborhood graph*: We first uniformly sample  $y_1, \dots, y_n$  from a unit square. We then set  $\Omega_{ij} = \Omega_{ji} = \rho$  with probability

$$\left(\sqrt{2\pi}\right)^{-1} \exp(-4\|y_i - y_j\|^2).$$

All the rest  $\Omega_{ij}$  are set to be zero. The number of nonzero off-diagonal elements of each row or column is restricted to be smaller than  $\lfloor 1/\rho \rfloor$ . In this paper,  $\rho$  is set to be 0.245.

2. *Hub graph*: The rows/columns are partitioned into  $J$  equally-sized disjoint groups:  $V_1 \cup V_2 \dots \cup V_J = \{1, \dots, d\}$ , each group is associated with a “pivotal” row  $k$ . Let  $|V_1| = s$ . We set  $\Omega_{ik} = \Omega_{ki} = \rho$  for  $i \in V_k$  and  $\Omega_{ik} = \Omega_{ki} = 0$  otherwise. In our experiment,  $J = \lfloor d/s \rfloor$ ,  $k = 1, s+1, 2s+1, \dots$ , and we always set  $\rho = 1/(s+1)$  with  $s = 20$ .

We generate synthetic datasets in both low-dimensional ( $n = 800, d = 40$ ) and high-dimensional ( $n = 400, d = 100$ ) settings. Table 12 provides comparisons of all methods, where we repeat the experiments 100 times and report the averaged precision, recall,  $F_1$ -score with their standard errors.

For low-dimensional settings where  $n \gg d$ , the BIC criterion is very competitive and performs the best among all the methods. In high dimensional settings, however, StARS clearly outperforms all the competing methods for



Table 12.: Quantitative comparison of different methods on the datasets from the neighborhood and hub graphs.

| Neighborhood graph: $n=800, d=40$ |               |               |               | Neighborhood graph: $n=400, d=100$ |               |               |
|-----------------------------------|---------------|---------------|---------------|------------------------------------|---------------|---------------|
| Methods                           | Precision     | Recall        | $F_1$ -score  | Precision                          | Recall        | $F_1$ -score  |
| Oracle                            | 0.9222 (0.05) | 0.9070 (0.07) | 0.9119 (0.04) | 0.7473 (0.09)                      | 0.8001 (0.06) | 0.7672 (0.07) |
| StARS                             | 0.7204 (0.08) | 0.9530 (0.05) | 0.8171 (0.05) | 0.6366 (0.07)                      | 0.8718 (0.06) | 0.7352 (0.07) |
| K-CV                              | 0.1394 (0.02) | 1.0000 (0.00) | 0.2440 (0.04) | 0.1383 (0.01)                      | 1.0000 (0.00) | 0.2428 (0.01) |
| BIC                               | 0.9738 (0.03) | 0.9948 (0.02) | 0.9839 (0.01) | 0.1796 (0.11)                      | 1.0000 (0.00) | 0.2933 (0.13) |
| AIC                               | 0.8696 (0.11) | 0.9996 (0.01) | 0.9236 (0.07) | 0.1279 (0.00)                      | 1.0000 (0.00) | 0.2268 (0.01) |

| Hub graph: $n=800, d=40$ |               |               |               | Hub graph: $n=400, d=100$ |               |               |
|--------------------------|---------------|---------------|---------------|---------------------------|---------------|---------------|
| Methods                  | Precision     | Recall        | $F_1$ -score  | Precision                 | Recall        | $F_1$ -score  |
| Oracle                   | 0.9793 (0.01) | 1.0000 (0.00) | 0.9895 (0.01) | 0.8976 (0.02)             | 1.0000 (0.00) | 0.9459 (0.01) |
| StARS                    | 0.4377 (0.02) | 1.0000 (0.00) | 0.6086 (0.02) | 0.4572 (0.01)             | 1.0000 (0.00) | 0.6274 (0.01) |
| K-CV                     | 0.2383 (0.09) | 1.0000 (0.00) | 0.3769 (0.01) | 0.1574 (0.01)             | 1.0000 (0.00) | 0.2719 (0.00) |
| BIC                      | 0.4879 (0.05) | 1.0000 (0.00) | 0.6542 (0.05) | 0.2155 (0.00)             | 1.0000 (0.00) | 0.3545 (0.01) |
| AIC                      | 0.2522 (0.09) | 1.0000 (0.00) | 0.3951 (0.00) | 0.1676 (0.00)             | 1.0000 (0.00) | 0.2871 (0.00) |

both neighborhood and hub graphs. This is consistent with our theory. At first sight, it might be surprising that for data from low-dimensional neighborhood graphs, BIC and AIC even outperform the oracle procedure! This is due to the fact that both BIC and AIC graphs are estimated using all the  $n = 800$  data points, while the oracle graph is estimated using only the subsampled dataset with size  $b(n) = \lfloor 10 \cdot \sqrt{n} \rfloor = 282$ . Direct usage of the full sample is an advantage of model selection methods that take the general form of BIC and AIC. In high dimensions, however, we see that even with this advantage, StARS clearly outperforms BIC and AIC. The estimated graphs for different methods in the setting  $n = 400, d = 100$  are provided in Figures 61 and 62, from which we see that the StARS graph is almost as good as the oracle, while the K-CV, BIC, and AIC graphs are overly too dense.

### 10.5.2 Microarray Data

We apply StARS to a dataset based on Affymetrix GeneChip microarrays for the gene expression levels from immortalized B cells of human subjects. The sample size is  $n = 294$ . The expression levels for each array are pre-processed by log-transformation and standardization as in [Nayak et al., 2009]. Using

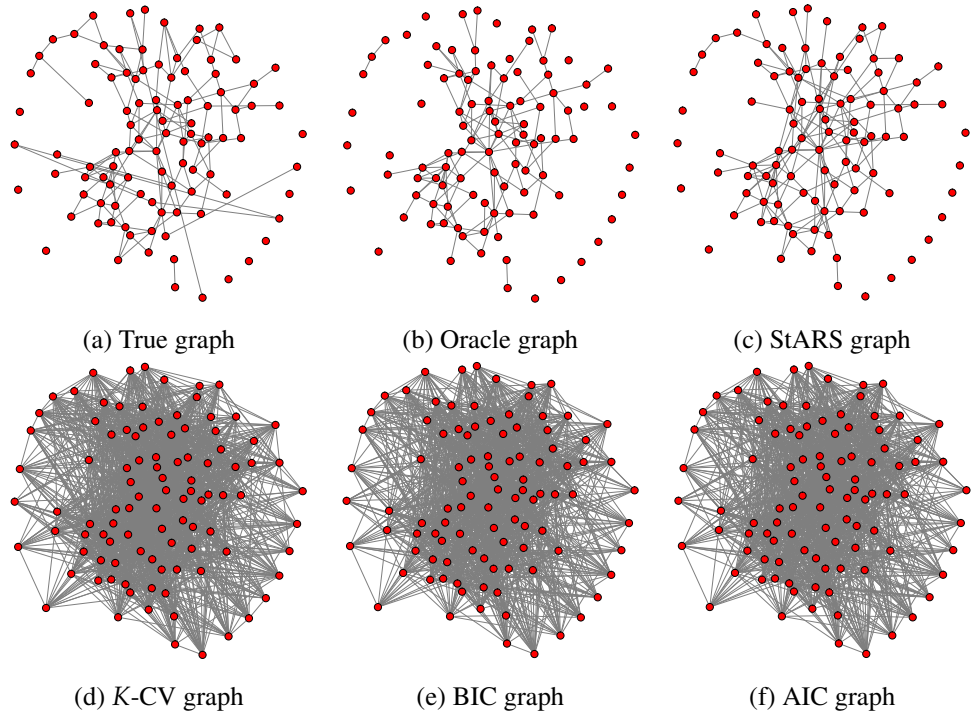


Figure 61.: Comparison of different methods on the data from the neighborhood graphs ( $n = 400, d = 100$ ).

a previously estimated sub-pathway subset containing 324 genes [Liu et al., 2010c], we study the estimated graphs obtained from each method under investigation. The StARS and BIC graphs are provided in Figure 63. We see that the StARS graph is remarkably simple and informative, exhibiting some cliques and hub genes. In contrast, the BIC graph is very dense and possible useful association information is buried in the large number of estimated edges. The selected graphs using AIC and K-CV are even more dense than the BIC graph and is omitted here. A full treatment of the biological implication of these two graphs validated by enrichment analysis will be left as a future study.

## 10.6 CONCLUSIONS

The problem of estimating structure in high dimensions is very challenging. Casting the problem in the context of a regularized optimization has led to some success, but the choice of the regularization parameter is critical. We present a new method, StARS, for choosing this parameter in high dimensional inference for undirected graphs. Like Meinshausen and Bühlmann's stability selection approach [Meinshausen and Bühlmann, 2010], our method makes use of subsampling, but it differs substantially from their approach in

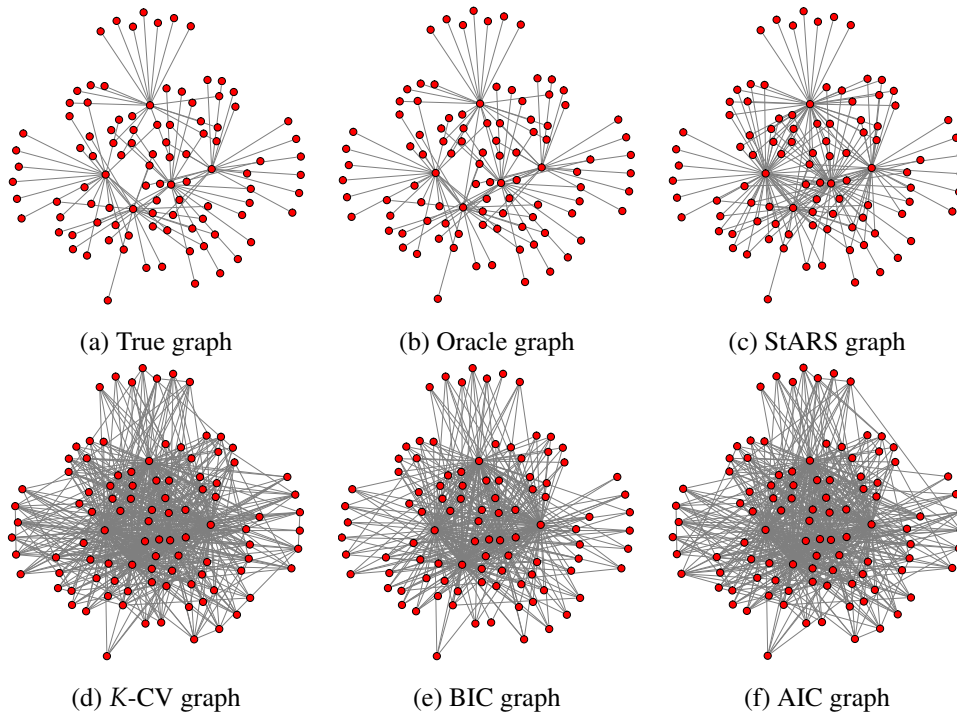


Figure 62.: Comparison of different methods on the data from the hub graphs ( $n = 400, d = 100$ ).

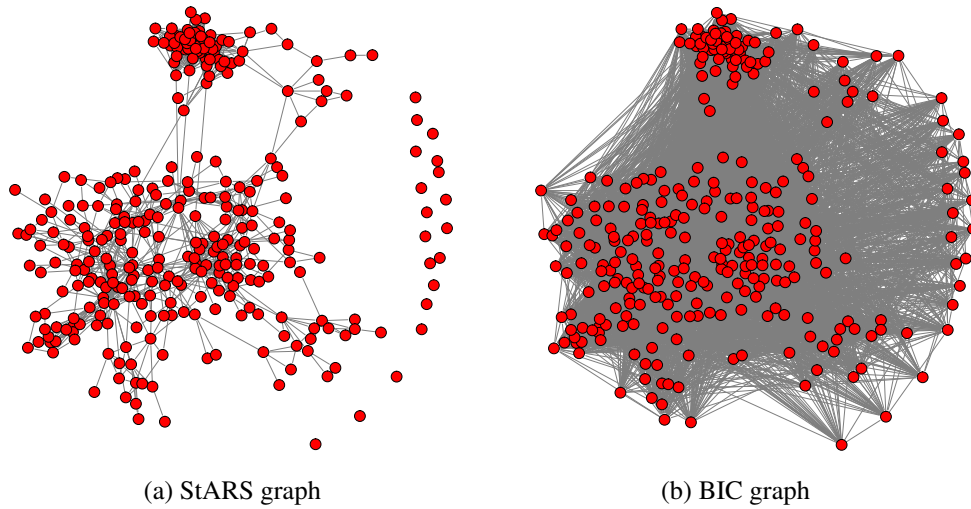


Figure 63.: Microarray data example. The StARS graph is more informative graph than the BIC graph.

both implementation and goals. For graphical models, we choose the regularization parameter directly based on the edge stability. Under mild conditions, StARS is partially sparsistent. However, even without these conditions, StARS

has a simple interpretation: we use the least amount of regularization that simultaneously makes a graph sparse and replicable under random sampling.

Empirically, we show that StARS works significantly better than existing techniques on both synthetic and microarray datasets. Although we focus here on graphical models, our new method is generally applicable to many problems that involve estimating structure, including regression, classification, density estimation, clustering, and dimensionality reduction.

Part VI

CONCLUSION



## CONCLUSION AND FUTURE DIRECTIONS

---

### 11.1 SUMMARY AND DISCUSSIONS

In this thesis, we developed principled nonparametric methods to explore and predict high dimensional complex datasets. The results of this thesis are applicable in many modern scientific fields, including genomics, proteomics, cognitive neuroscience, and computational meteorology. The data in these fields are usually very high dimensional and are generated by some unknown complex processes, which makes nonparametric methods especially suitable for building accurate predictive models or discovering new scientific facts. In this chapter, we first summarize several applications of this thesis. We then conclude this thesis with some future directions.

#### 11.1.1 *Building Computational Models to Predict Brain Activities*

In Chapter 6, we applied the multi-task sparse additive models to build a computational model that predicts human brain activities represented by fMRI images. The basic approach is to predict the neural activities that would be recorded using fMRI images when a person thinks about an arbitrary word in English. Creating such a predictive model not only enables us to explore new analytical tools for the fMRI data, but also helps us to gain a deeper understanding of how human brains represent knowledge. Existing solutions to this problem either resort to human experts [Mitchell et al., 2008] or sparse linear models [Liu et al., 2009b]. Compared to the previous work, our nonparametric solution achieves a significantly higher prediction accuracy and good interpretability. This finding is important for building more realistic models of the fMRI data. In the near future, we will further investigate the obtained model from a cognitive neuroscience perspective.

#### 11.1.2 *Inferring Gene Regulatory Networks*

One of the most important and challenging knowledge discovery tasks in genomics is the reverse engineering of gene regulatory networks from DNA microarray data. Such networks can be inferred from data by estimating the undirected graphical models. However, this problem is extremely challenging since the number of genes being studied is generally much larger than the number of measurements. To avoid the curse of dimensionality, most existing

methods assume that the gene expression data are Gaussian distributed and infer the undirected graphical models by estimating the inverse covariance matrix. In Chapter 5, we applied the forest density estimator to estimate gene regulatory graphs for the isoprenoid biosynthetic pathway and humans using microarray data. Evaluated by the held-out likelihood, our method significantly outperforms sparse Gaussian models. This is not to say that the estimated networks from the existing methods are wrong, but it does reveal the fact that the normality assumption is not appropriate in this dataset. Such a result is quite interesting since it may lead to dramatically different scientific conclusions from the previous parametric analysis.

### 11.1.3 *Tumor Classification using Microarray Data*

Another successful application of our method is to classify small round blue cell tumors (SRBCT) using high dimensional microarray data. This dataset contains 2,308 genes and 4 tumor categories. Compared to previous analyses on the same data, our sparse multi-category additive logistic regression model achieves the best predictive accuracy on the test set (100% accuracy) using the most compact set of predictors (20 genes). The fitted marginal effects of these selected genes are highly nonlinear, which confirms that high-dimensional nonparametric inference is quite suitable for this dataset.

### 11.1.4 *Climate Data Analysis*

Global warming has become one of the most critical socio-technological issues we are facing in the 21st century. Among the various ways in which computer scientists can play a role, we are particularly interested in providing better understanding and quantifying the causal effects of climate and climate-forcing factors. In Chapter 9, we applied the Go-CART to estimate the graphical models in different locations of the United States. Our results by estimating the conditional independence graphs (conditional on the geographic locations) are more interpretable than those obtained by estimating an unconditional universal graph.

## 11.2 FUTURE DIRECTIONS

The results of this thesis lead to many future directions. Here we summarize some in terms of theory, methods, and applications.



### 11.2.1 *Theory*

Almost all current analyses for high dimensional nonparametric methods are based on empirical process theory, where success crucially depends on controlling the complexity of the hypothesis space. However, many widely used nonparametric methods, such as kernel or local polynomial smoothers, are left outside of this framework. The main reason is that their hypothesis spaces are too rich to be easily controlled. In Chapter 6, we proposed the smooth sparse backfitting framework to address this problem. The key is to construct an extended product Hilbert space that enables us to formulate a nonparametric estimator as the solution to an infinite-dimensional convex optimization problem. Explorations in this realm could significantly push the frontier of modern statistics and learning theory.

### 11.2.2 *Methods*

The field of high dimensional nonparametric inference has mainly focused on sequential algorithms, which are suitable when there is one powerful CPU and sufficient memory. However, over the last five years, the computing paradigms have changed significantly: serial speedups of single processors are relatively stalled and the new trend is to make processors multicore. Such an evolution poses both challenges and opportunities for high dimensional nonparametric learning. On the one hand, most existing sequential algorithms may no longer be compatible with the new parallel paradigm. On the other hand, many current computationally intractable methods may become feasible in the future. We believe it would be fruitful to explore the interaction of parallel computing with high dimensional nonparametric learning.

### 11.2.3 *Applications*

We expect that the results of this thesis could have more applications in cognitive neuroscience and bioinformatics. One interesting problem is to develop a unified nonparametric framework that enables transfer learning from multiple fMRI datasets provided by different labs. Another potential application problem is *peptide identification using data-independent tandem mass spectrometry*. This is a new technique recently invented in proteomics, and its main goal is to use shotgun methods to simultaneously identify many peptides in a tissue by searching a large sequence database. To our knowledge, there are not yet any effective sequencing algorithms. Indeed, this problem can be formulated into a multi-task regression with joint sparsity constraints and our MT-SpAM could be appropriate for it.



Part VII

APPENDIX



## MORE TECHNICAL DETAILS OF THE COSSO

The COSSO is developed by [Lin and Zhang \[2006\]](#), it can be viewed as a generalization of the lasso estimator to the nonparametric functional ANOVA model, which has the form

$$Y^{(i)} = \sum_{j=1}^n m_j(X_j^{(i)}) + \sum_{j < k} m_{jk}(X_j^{(i)}, X_k^{(i)}) + \sum_{j < k < \ell} m(X_j^{(i)}, X_k^{(i)}, X_\ell^{(i)}) + \cdots + \epsilon^{(i)}.$$

COSSO formulates the estimation as an optimization problem:

$$\hat{m}(x) = \arg \min_m \left\{ \frac{1}{n} \sum_{i=1}^n \left( Y_i - m(X^{(i)}) \right)^2 + \lambda \sum_{\alpha=1}^d \|P^\alpha m\| \right\} \quad (\text{A.1})$$

where  $P^\alpha$  is the projection onto the subspace of the  $\alpha$ -th component. An equivalent form of the COSSO estimator is

$$\begin{aligned} \hat{m}(x) &= \arg \min_m \left\{ \frac{1}{n} \sum_{i=1}^n \left( Y^{(i)} - m(X^{(i)}) \right)^2 + \lambda_0 \sum_{\alpha=1}^d \theta_\alpha^{-1} \|P^\alpha m\|^2 + \lambda \sum_{\alpha=1}^d \theta_\alpha \mid \theta_\alpha \geq 0 \right\}. \end{aligned} \quad (\text{A.2})$$

It is very similar to the nonnegative garrote estimator. This explains why COSSO can induce sparsity. For the fixed data dimension  $d$ , [Lin and Zhang \[2006\]](#) proved the nearly optimal rate of convergence for the COSSO estimator when using additive models.

A recent work of [Jeon and Lin \[2006\]](#) extended the idea of COSSO to density estimation setting. Let  $X^{(1)}, X^{(2)}, \dots, X^{(n)}$  be a  $d$ -dimensional sample from a distribution  $F$  with density  $p(x)$ , assuming  $\rho$  be a fixed positive density function over the support  $\mathcal{X}$ , they first find a function  $g(x) = \sum_A g(x_A)$ , where  $A$  varies over the subspaces corresponding to the all-two-way-interaction terms, in the reproducing kernel Hilbert space(RKHS)  $\mathcal{H}$  and satisfies

$$\begin{aligned} \hat{g} = \arg \min_g \left\{ 1 + \log \left( \frac{1}{n} \sum_{i=1}^n \exp(-g(X^{(i)})) \right) \right. \\ \left. + \int_{\mathcal{X}} g(x) \rho(x) dx + \lambda \sum_A \theta_A^{-1} \|P_A g\|^2 \right\} \end{aligned} \quad (\text{A.3})$$

where  $P_A$  is the orthogonal projector. Once the optimal estimate  $\hat{g}(x)$  is obtained, the final estimator for the density is of the form

$$\hat{f}(x) = \text{constant} \cdot \rho(x) \cdot \exp(\hat{g}(x)) \quad (\text{A.4})$$

By taking  $\rho(x)$  with a multiplicative form of the marginal baseline densities  $\rho(x) = \prod_{j=1}^d \rho^{(j)}(x_j)$ , a Newton-Raphson procedure is developed for model fitting. Even though no theoretical analysis is provided, the resulting algorithm can be used to perform density estimation in very high dimensions and the resulting sparse all-two-way-interaction log-density ANOVA model is natural to build nonparametric graphical models.

## BIBLIOGRAPHY

---

- A. Azzalini and A. W. Bowman. A look at some data on the old faithful geyser. *Applied Statistics*, 39:357–365, 1990.
- Felix Abramovich, Yoav Benjamini, David L. Donoho, and Iain M. Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *The Annals of Statistics*, 34(2):584–653, 2006.
- Martin Aigner and Günter Ziegler. *Proofs from THE BOOK*. Springer-Verlag, 1998.
- H. Akaike. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, pages 267–281, 1973.
- Francis Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 8:1179–1225, 2008a.
- Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in Neural Information Processing Systems 21*. MIT Press, 2008b.
- Francis R. Bach and Michael I. Jordan. Beyond independent components: Trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.
- O. Banerjee, L. E. Ghaoui, and A. d’Aspremont. Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9:485–516, March 2008.
- Andrew R. Barron, Albert Cohen, Wolfgang Dahmen, and Ronald A. DeVore. Approximation and learning by greedy algorithms. *The Annals of Statistics*, 36:64–94, 2008.
- Shai Ben-david, Ulrike Von Luxburg, and David Pal. A sober look at clustering stability. In *Proceedings of the Conference of Learning Theory*, pages 5–19. Springer, 2006.
- Peter J. Bickel and Bo Li. Local polynomial regression on unknown manifolds. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2101–2117, 2006.
- G. Blanchard, C. Schäfer, Y. Rozenholc, and K.-R. Müller. Optimal dyadic decision trees. *Mach. Learn.*, 66(2-3):209–241, 2007a. ISSN 0885-6125.

- G. Blanchard, C. Schäfer, Y. Rozenholc, and K.-R. Müller. Optimal dyadic decision trees. *Machine Learning Journal*, 66(2-3):209–241, 2007b.
- Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37:373–384, 1995.
- Leo Breiman and Jerome H. Friedman. Predicting multivariate responses in multiple linear regression. *J. Roy. Statist. Soc. B*, 59:3, 1997.
- Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. *Classification and regression trees*. Wadsworth Publishing Co Inc, 1984.
- Peter Bühlmann and Bin Yu. Sparse boosting. *Journal of Machine Learning Research*, 7:1001–1024, 2006.
- Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510, June 1989.
- Florentina Bunea, Alexandre B. Tsybakov, and Marten H. Wegkamp. Aggregation for Gaussian regression. *The Annals of Statistics*, 35(4):1674–1697, 2007.
- Tony Cai, Cun-Hui Zhang, and Harrison H. Zhou. Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, 38(4):2118–2144, 2010.
- Emmanuel Candes and Terence Tao. The dantzig selector: statistical estimation when  $p$  is much larger than  $n$ . *The Annals of Statistics*, 35:2313–2351, 2007.
- Rui Castro, Rebecca Willett, and Robert Nowak. Faster rates in regression via active learning. In *Advances in Neural Information Processing Systems 18*, 2005.
- Arthur Cayley. A theorem on trees. *Quart. J. Math.*, 23:376–378, 1889.
- Anton Chechotka and Carlos Guestrin. Efficient principled learning of thin junction trees. In *In Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2007.
- Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific and Statistical Computing*, 20:33–61, 1998.
- Xi Chen, Weike Pan, James T. Kwok, and Jamie G. Carbonell. Accelerated gradient method for multi-task sparse learning problem. In *ICDM*, 2009.
- Xi Chen, Yan Liu, Han Liu, and Jaime G. Carbonell. Learning spatial-temporal varying graphs with applications to climate data analysis. In *AAAI*, 2010.



- Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bart: Bayesian additive regression trees. Technical report, Department of Mathematics and Statistics, Acadia University, Canada, 2006.
- C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467, 1968.
- I. Daubechies, M. Defrise, and C. DeMol. An iterative thresholding algorithm for linear inverse problems. *Comm. Pure Appl. Math*, 57(11), 2004.
- I. Daubechies, M. Fornasier, and I. Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. Technical report, Princeton University, 2007. arXiv:0706.4297.
- A.P. Dempster, N.M. Laord, and D.B. Rubin. Maximum likelihood estimation from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.
- Arthur P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.
- Mathias Drton and Michael D. Perlman. Multiple testing and error control in Gaussian graphical model selection. *Statistical Science*, 22(3):430–449, 2007.
- Mathias Drton and Michael D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4): 1179–1200, 2008.
- David Edwards. *Introduction to graphical modelling*. Springer-Verlag Inc, 1995.
- Bradley Efron. *The jackknife, the bootstrap and other resampling plans*. SIAM [Society for Industrial and Applied Mathematics], 1982.
- Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32:407–499, 2004.
- M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1994.
- J. Fan and J. Jiang. Nonparametric inference for additive models. *Journal of the American Statistical Association*, 100:890–907, 2005.
- J. Fan and R. Z. Li. Variable selection via penalized likelihood. *Journal of American Statistical Association*, 96:1348–1360, 2001.
- Jianqing Fan and Irène Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall. New York, NY, 1996.

- Massimo Fornasier and Holger Rauhut. Recovery algorithms for vector valued data with joint sparsity constraints. *SIAM J. Numer. Anal.*, 46(2):577–613, 2008.
- David Friedenbergh and Christopher Genovese. Straight to the source: Detecting aggregate objects in astronomical images. *arXiv:0910.5449*, 2009.
- J. Friedman, W. Stuetzle, and A. Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.
- Jerome H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19:1–67, 1991.
- Jerome H. Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2007.
- C.R. Genovese and L.A. Wasserman. Rates of convergence for the gaussian mixture sieve. *The Annals of Statistics*, 28(4):1105–1127, 2000.
- S. Gey and E. Nedelec. Model selection for cart regression trees. *IEEE Tran. on Info. Theory*, 51(2):658–670, 2005.
- S. Ghosal, J.K. Ghosh, and A.W. van der Vaart. Convergence rates of posterior distributions. *The Annals of Statistics*, 28(2):500–531, 2000.
- E. Giné and A. Guillaou. Rates of strong uniform consistency for multivariate kernel density estimators. *Annales de l’institut Henri Poincaré (B), Probabilités et Statistiques*, 38:907–921, 2002.
- E. Greenshtein and Y. Ritov. Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Journal of Bernoulli*, 10: 971–988, 2004.
- László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, 2002.
- Trevor Hastie and Robert Tibshirani. *Generalized Additive Models*. Chapman and Hall. New York, NY, 1999.
- N.L. Hjort and I.K. Glad. Nonparametric density estimation with a parametric start. *The Annals of Statistics*, 23(3):882–904, 1995.
- N.L. Hjort and M.C.Jones. Locally parametric nonparametric density estimation. *The Annals of Statistics*, 24:1619–1647, 1996.
- IPCC. Climate change 2007—the physical science basis. *IPCC Fourth Assessment Report*, 2007.

- Y. Jeon and Y. Lin. An effective method for high dimensional log-density anova estimation, with application to nonparametric graphical model building. *Statistical Sinica*, 16:353–374, 2006.
- A. Juditsky and A. Nemirovski. Functional aggregation for nonparametric regression. *The Annals of Statistics*, 28:681–712, 2000.
- Javed Khan, Jun S. Wei, Markus Ringner, Lao H. Saa, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R. Antonescu, Carsten Peterson, and Paul S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679, 2001.
- Chris A. J. Klaassen and Jon A. Wellner. Efficient estimation in the bivariate normal copula model: Normal margins are least-favorable. *Bernoulli*, 3(1): 55–77, 1997.
- Donald E. Knuth. Computer Programming as an Art. *Communications of the ACM*, 17(12):667–673, December 1974.
- Balaji Krishnapuram, Lawrence Carin, Mário A. T. Figueiredo, and Alexander J. Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 27(6): 957–968, June 2005.
- Joseph B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.
- John Lafferty and Larry Wasserman. Rodeo: Sparse, greedy nonparametric regression. *The Annals of Statistics*, 36(1):28–63, 2008.
- N.M. Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73:805–811, 1978.
- Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6): 1299–1323, 2004.
- Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.
- Han Liu and Xi Chen. Nonparametric greedy algorithm for the sparse learning problems. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 1141–1149, 2009.

- Han Liu and Xi Chen. Multivariate dyadic regression trees for sparse learning problems. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, 2010.
- Han Liu and Jian Zhang. On the estimation consistency of the group lasso and its applications. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.
- Han Liu and Jian Zhang. On the  $\ell_{1,q}$ -regularized regression. *Technical Report, Department of Statistics, Carnegie Mellon University*, 2008.
- Han Liu, John Lafferty, and Larry Wasserman. Sparse nonparametric density estimation using the rodeo. In *Proceedings of the Eleventh Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 283–290, 2007.
- Han Liu, John Lafferty, and Larry Wasserman. Nonparametric regression and classification with joint sparsity constraints. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 969–976, 2008.
- Han Liu, John Lafferty, and Larry Wasserman. The nonparanormal: Semi-parametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:2295–2328, 2009a.
- Han Liu, Mark Palatucci, and Jian Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *The Twenty-sixth International Conference on Machine Learning (ICML)*, 2009b.
- Han Liu, Xi Chen, John Lafferty, and Larry Wasserman. Graph-valued regression. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, 2010a.
- Han Liu, Kathryn Roeder, and Larry Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, 2010b.
- Han Liu, Min Xu, Haijie Gu, Anupam Gupta, John Lafferty, and Larry Wasserman. Forest density estimation. arXiv:1001.1557, 2010c.
- C.R. Loader. Local likelihood density estimation. *The Annals of Statistics*, 24: 1602–1618, 1996.
- Aurelie C. Lozano, Hongfei Li, Alexandru Niculescu-Mizil, Yan Liu, Claudia Perlich, Jonathan Hosking, and Naoki Abe. Spatial-temporal causal modeling for climate change attribution. In *ACM SIGKDD*, 2009.

- S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415, 1993.
- Colin L. Mallows, editor. *The collected works of John W. Tukey. Volume VI: More mathematical, 1938–1984*. Wadsworth & Brooks/Cole, 1990.
- Enno Mammen, O. Linton, and J. Nielsen. The existence and asymptotic properties of a backfitting projection algorithm under weak conditions. *The Annals of Statistics*, 27:1443–1490, 1999.
- J.S. Marron and M.P. Wand. Exact mean integrated squared error. *The Annals of Statistics*, 20:712–736, 1992.
- Wendy L. Martinez and Angel R. Martinez. *Computational Statistics Handbook with MATLAB*. Chapman & Hall CRC, 2 edition, 2008.
- Lukas Meier, Sara van de Geer, and Peter Bühlmann. High-dimensional additive modelling. *Annals of Statistics (to appear)*, 37(6B):3779–3821, 2009.
- Meinshausen and Bin Yu. Lasso-type recovery of sparse representations from highdimensional data. *Annals of Statistics*, pages 246–270, 2009.
- N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3), 2006.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society, Series B, Methodological*, 72:417–473, 2010.
- Tom Mitchell et al. Predicting human brain activity associated with the meanings of nouns. *Science*, 320:1191–1195, 2008.
- Renuka R. Nayak, Michael Kearns, Richard S. Spielman, and Vivian G. Cheung. Coexpression network based on natural variation in human gene expression reveals gene interactions and functions. *Genome Research*, 19(11):1953–1962, November 2009.
- Sahand Negahban and Martin J. Wainwright. Phase transitions for high-dimensional joint support recovery. In *Annual Conference on Neural Information Processing Systems*, pages 1161–1168, 2008.
- Deborah Nolan and David Pollard. U-processes: Rates of convergence. *The Annals of Statistics*, 15(2):780 – 799, 1987.
- G. Obozinski, M. J. Wainwright, and M. I. Jordan. High-dimensional union support recovery in multivariate regression. In *Advances in Neural Information Processing Systems*. MIT Press, 2009.
- Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 1999.

- Emanuel Parzen. On the estimation of a probability density function and the mode. *The Annals of Mathematical Statistics*, 33:1065–1076, 1962.
- Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.
- Dimitris N. Politis, Joseph P. Romano, and Michael Wolf. *Subsampling (Springer Series in Statistics)*. Springer, 1 edition, August 1999. ISBN 0387988548.
- Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming, Aug 2010.
- P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional ising model selection using  $\ell_1$ -regularized logistic regression. *Annals of Statistics*, 38: 1287–1319, 2010.
- Pradeep Ravikumar, Han Liu, John Lafferty, and Larry Wasserman. Spam: Sparse additive models. In *Advances in Neural Information Processing Systems 20*, 2007.
- Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, Series B, Methodological*, 71(5):1009–1030, 2009a.
- Pradeep Ravikumar, Martin Wainwright, Garvesh Raskutti, and Bin Yu. Model selection in Gaussian graphical models: High-dimensional consistency of  $\ell_1$ -regularized MLE. In *Advances in Neural Information Processing Systems 22*, Cambridge, MA, 2009b. MIT Press.
- S. Richardson and P.J. Green. On bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B*, 59(4):731–792, 1997.
- P. Rigollet and R. Vert. Fast rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.
- Alessandro Rinaldo and Larry Wasserman. Generalized density clustering. *arXiv/0907.3454*, 2009a.
- Alessandro Rinaldo and Larry Wasserman. Low-noise density clustering. *Technical report, Carnegie Mellon University*, 2009b.
- R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer-Verlag Inc, 1998.
- M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27:642–669, 1956.

- Adam J. Rothman, Peter J. Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.
- Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464, 1978.
- C. Scott and R. Nowak. Learning minimum volume sets. *Journal of Machine Learning Research*, 7:665–704, April 2006a.
- C. Scott and R. Nowak. Minimax-optimal classification with dyadic decision trees. *IEEE Transactions on Information Theory*, 52(4):1335–1353, April 2006b.
- David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley. New York, NY, 1992.
- Robert J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley. New York, NY, 1980.
- B. W. Silverman. Spline smoothing: The equivalent variable kernel method. *The Annals of Statistics*, 12:898–916, 1984.
- B.W. Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, 10:795–810, 1982.
- Abe Sklar. Fonctions de répartition à  $n$  dimensions et leurs marges. *Publications de l'Institut de Statistique de L'Université de Paris* 8, pages 229–231, 1959.
- C.J. Stone. Large sample inference for log-spline models. *The Annals of Statistics*, 18:717–741, 1990.
- V. Tan, A. Anandkumar, L. Tong, and A. Willsky. A large-deviation analysis for the maximum likelihood learning of tree structures. arXiv:0905.0940, 2009a.
- V. Tan, A. Anandkumar, and A. Willsky. Learning Gaussian tree models: Analysis of error exponents and extremal structures. arXiv:0909.5216, 2009b.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, Methodological*, 58:267–288, 1996.
- Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, , and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci U.S.A.*, 99:6567–6572, 2002.
- J. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. Part II: Convex relaxation. *Signal Processing*, 86:572–588, 2006.

- Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50(10):2231–2241, October 2004.
- Hideatsu Tsukahara. Semiparametric estimation in copula models. *Canadian Journal of Statistics*, 33:357–375, 2005.
- Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 2008.
- B.A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 27:349–363, 2005.
- A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.
- Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer-Verlag, 1996.
- Grace Wahba. *Spline models for observational data*. SIAM. New York, NY, 1990.
- M. Wainwright. Sharp thresholds for high-dimensional and noisy recovery of sparsity. Technical Report 709, Department of Statistics, UC Berkeley, May 2006.
- Martin Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, 55(12):5728–5741, December 2009.
- Martin J. Wainwright, Pradeep Ravikumar, and John D. Lafferty. High-dimensional graphical model selection using  $\ell_1$ -regularized logistic regression. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1465–1472. MIT Press, Cambridge, MA, 2007.
- Larry Wasserman and Kathryn Roeder. High dimensional variable selection. *Annals of statistics*, 37(5A):2178–2201, January 2009.
- J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley, 1990.
- Anja Wille, Philip Zimmermann, Eva Vranová, Andreas Fürholz, Oliver Laule, Stefan Bleuler, Lars Hennig, Amela Prelić, Peter von Rohr, Lothar Thiele, Eckart Zitzler, Wilhelm Gruissem, and Peter Bühlmann. Sparse Gaussian graphical modelling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biology*, 5:R92, 2004.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.
- Ming Yuan. Nonnegative garrote component selection in functional ANOVA models. *Proceedings of AI and Statistics, AISTATS*, 2007.



- Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- Hao Helen Zhang, Yufeng Liu, Yichao Wu, and Ji Zhu. Variable selection for the multcategory SVM via adaptive sup-norm regularization. *Electronic Journal of Statistics*, 2:149–1167, 2008.
- Jian Zhang. A probabilistic framework for multitask learning. Technical Report CMU-LTI-06-006, Ph.D. thesis, Carnegie Mellon University, 2006.
- Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.
- Tong Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928, 2008.
- P. Zhao and B. Yu. On model selection consistency of lasso. *J. of Mach. Learn. Res.*, 7:2541–2567, 2007.
- Shuheng Zhou, John Lafferty, and Larry Wasserman. Time varying undirected graphs. *Machine Learning*, 78(4), 2010.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2005.
- Hui Zou. The adaptive lasso and its oracle properties. *Journal of American Statistical Association*, 101(476):1418–1429, 2006.



## COLOPHON

This thesis was typeset with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

*Final Version* as of December 8, 2010 at 14:24.



## DECLARATION

---

The work was done under the supervision of Professor John Lafferty and Professor Larry Wasserman, at Carnegie Mellon University, Pittsburgh.

*Pittsburgh, December 2010*

---

Han Liu