Correlated Data Analysis with Copula Models or Bayesian Nonparametric Methods

by

Haoxin Zhuang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. Radu V. Craiu Professor, University of Toronto
Supervisor(s):	Dr. Liqun Diao Research Assistant Professor, University of Waterloo
	Dr. Grace Y. Yi Professor, Western University (2019-Present) University of Waterloo (before 2019, now Adjunct Professor)
Internal Member:	Dr. Cecilia Cotton Associate Professor, University of Waterloo
Internal Member:	Dr. Leilei Zeng Associate Professor, University of Waterloo
Internal-External Member:	Dr. Yaoliang Yu Assistant Professor, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Different types of correlated data arise commonly in many studies and present considerable challenges in modeling and characterizing complex dependence structures. This thesis considers statistical issues in analyzing such kinds of data. Chapters 2-4 of the thesis aim to develop models to account for complex dependence structures and propose new statistical inference methods. In particular, our attention focuses on using copula models and their variants to delineate association structures for dependent data. As "big data" has increasingly versatile applications in many fields, more and more data with irregular distributions emerge, which calls for more flexible and robust nonparametric statistical methods. Chapters 5 and 6 of the thesis develop novel Bayesian nonparametric methods on sampling algorithms and regression models.

More specifically, in Chapter 2, we consider longitudinal data with a time-span, of which common examples include temperature and precipitation data. We utilize a vine copula model to account for the dependence among longitudinal responses; the joint distribution of responses is factorized as a product of marginal distributions and bivariate conditional copulas. To release the computational burden and concentrate on the structure of interest, we propose composite likelihood methods which divide the responses into time blocks and leave the connecting structure between time blocks unspecified. We explore the efficiency, robustness, model selection and prediction of our proposed methods by simulation studies. The proposed model is applied to analyze an Ontario temperature dataset.

In Chapter 3, we consider dependent data with a hierarchical structure. Analysis of such data is often challenging due to the complexity in modeling different dependence structures as well as the demand of intensive computation sources. To alleviate these issues, we propose a Bayesian hierarchical copula model (BHCM) to accommodate the hierarchical structures of the dependent data, where the subject-level dependence is facilitated by the copula-based model and the hierarchical structure is described using random dependence parameters. We introduce a layer-by-layer sampling scheme for conducting inferences. Our proposed BHCM enjoys the flexibility of modeling various complex association structures, while retaining manageable computation. Extensive simulation studies show that our proposed estimators outperform conventional likelihood-based estimators in finite sample settings. We apply the BHCM to analyze the Vertebral Column dataset from the UCI Machine Learning Repository.

In Chapter 4, we consider dependent data coming from multiple sources where we aim to group similar dependence structures together and then conduct model selection and parameter estimation based on copula models. We propose a mixture of Dirichlet process

mixture copula model (M-DPM-CM) to identify similar dependence structures and select copula models, in which the model selection parameters and copula parameters are assigned a Dirichlet process prior. Simulation studies and data analysis are conducted to compare the M-DPM-CM to the conventional copula selection method using the AIC criterion. The results show that the M-DPM-CM can accurately recover the true grouping structure with a moderate sample size, and achieve a more accurate model selection results than the conventional AIC method. The M-DPM-CM is also applied to analyze the Vertebral Column dataset used in Chapter 3 to obtain more insights into the dependence structures.

In Chapter 5, we focus on developing sampling algorithms from a complex distribution. To remedy the limitations of Markov Chain Monte Carlo (MCMC) algorithms, we propose a novel sampling method, called Polya tree Monte Carlo (PTMC). Our proposed PTMC method can feasibly approximate the posterior Polya tree by the Monte Carlo method, which is justified theoretically that the approximated Polya tree posterior converges to the target distribution under regularity conditions. We further propose a series of simple and efficient sampling algorithms which are useful for different scenarios. Extensive numerical studies are conducted to demonstrate the appealing performance of the proposed method, including its superiority to the usual MCMC algorithms, under various settings. The evaluation and comparison are carried out in terms of sampling efficiency, computational speed and the capacity of identifying distribution modes.

In Chapter 6, we consider the topic of nonparametric regression models. The Polya tree (PT) based nearest neighbor regression model is introduced as a fully nonparametric regression method. To approximate the true conditional probability measure of the response given the covariate value, we construct a PT-distributed probability measure of the response in the nearest neighborhood of the covariate value of interest. Our proposed method gives consistent and robust estimators, and has a faster convergence rate than the kernel density estimation. We conduct extensive simulation studies and analyze the Combined Cycle Power Plant dataset to compare the performance of our method to other nonparametric or semi-parametric methods.

Summary remarks and discussion of future research topics are presented in Chapter 7.

Acknowledgements

I would like to give my most sincere thanks to everyone who helps and supports me during my Ph.D. period. Firstly, I want to express my deepest gratitude to my supervisors, Dr. Liqun Diao and Dr. Grace Y. Yi for their patience, encouragement, and support during my Ph.D. studies. I learned not only the profound knowledge, but also the rigorous academic attitude and the enthusiasm for academic research from them. It is a real honor to be their student.

Besides my supervisors, I wish to thank my committee members Drs. Radu Craiu, Cecilia Cotton, Leilei Zeng and Yaoliang Yu for reviewing my thesis and providing their constructive suggestions. I would also like to thank Dr. Wenqing He for sharing a good time in the GW-DSRG, and Dr. Pengfei Li for teaching my first Ph.D. courses and giving me a lot of help. I want to deliver my gratitude to Mrs. Mary Lou Dufton and Mr. Greg Preston, who provide helpful administrative and technical support during my Ph.D. period.

Thirdly, I would like to give my thanks to my friends in Waterloo, including my academic brothers and sisters: Di Shu, Li-Pang Chen, Junhan Fang, Qihuang Zhang, Ce Yang and Yechao Meng, and other good friends: Hongcan Lin, Meng Yuan, Menglu Che, Yilin Chen, Chi-Kuang Yeh, Cong Jiang, Tom Chen, Changbao Zhang and Paul Xu.

Finally, I would like to give my gratitude to my family, especially my parents, Xiaofeng and Yao, who give their love and unconditional support to me. Special thanks give to my wife, Rongrong, for her love, understanding and sacrifices all these years.

Dedication

This is dedicated to my grandparents Xinming Zhuang and Yuyin Chen, who passed away in 2014 and 2019. I have my happy childhood with them. I will love and miss them forever.

Table of Contents

List of Tables **xiv**

List of Figures **xix**

1 Introduction **1**

1.1 Background 1

1.2 Copula 2

1.2.1 Definition 2

1.2.2 Model Selection and Parameter Estimation 4

1.3 Vine Copula 5

1.3.1 Definition 5

1.3.2 Canonical Vine and D-Vine 6

1.3.3 Model Selection and Parameter Estimation 8

1.4 Longitudinal Data Analysis 9

1.5 Composite Likelihood 10

1.6 Hierarchical Models 11

1.7 Bayesian Nonparametric Methods 12

1.7.1 Dirichlet Process 12

1.7.2 Polya Tree 14

1.8 Nonparametric Regression 18

1.9 Outline of the Thesis 20

2 Composite Likelihood Methods for Analyzing Longitudinal Data with a Time-Span under Vine Copula Models **24**

2.1 Introduction 24

2.2 Model Formulation 25

2.2.1 Joint Distribution of ε_i 26

2.2.2 Joint Model of the Responses Y_i 29

2.3 Estimation Methods 30

2.3.1 Simultaneous Estimation with Composite Likelihood 30

2.3.2 Two-Stage Estimation with Composite Likelihood 32

2.4 Copula Selection and Prediction 33

2.5 Simulation Studies 35

2.5.1 Validity and Efficiency 35

2.5.2 Robustness 39

2.5.3 Copula Selection 40

2.5.4 Prediction 41

2.6 Data Analysis 50

2.6.1 Dataset 50

2.6.2 Statistical Models 50

2.7 General Remarks 59

3 A Bayesian Hierarchical Copula Model **61**

3.1 Introduction 61

3.2 Model Formulation 62

3.2.1 Copula-based Dependence Models 63

3.2.2 Bayesian Hierarchical Models 63

3.3 Bayesian Inference 65

3.3.1 Posterior Distributions 66

3.3.2 Sampling Scheme 66

3.3.3 Asymptotic Properties 68

3.4 Transformation of the Dependence Parameters 69

3.4.1 Transformation Function 69

3.4.2 Choice of Scaling Parameter 70

3.5 Simulation Studies 72

3.5.1 Simulation Settings 73

3.5.2 Evaluation Metrics 74

3.5.3 Simulation Results 75

3.6 Data Analysis 78

3.6.1 Marginal Model 79

3.6.2 Dependence Model 79

3.6.3 Results 80

3.7 General Remarks 81

4 Grouping Dependence Structure and Selection of Copula-Based Models Using Bayesian Nonparametric Methods 84

4.1 Introduction 84

4.2 Model Formulation 85

4.2.1 Bayesian Hierarchical Model with Dirichlet Process Prior 87

4.2.2 Model Selection and Grouping under Dirichlet Process Prior 88

4.3 Bayesian Inference Process 89

4.3.1 Posterior and Hyper-Posterior Distribution 89

4.3.2 Sampling Scheme 91

4.4 Simulation Studies 93

4.4.1 Simulation Settings 93

4.4.2 Evaluation Metrics 95

4.4.3 Simulation Results 97

4.5 Data Analysis 100

4.5.1 Marginal Model 100
4.5.2 Dependence Model 100
4.6 General Remarks 102

5 Polya Tree Monte Carlo Method 103

5.1 Introduction 103
5.2 Polya Tree Monte Carlo Method 104
5.2.1 Polya Tree Monte Carlo Method 105
5.2.2 Sampling Algorithms 109
5.3 Simulation Studies 115
5.3.1 Setting 5.1 115
5.3.2 Setting 5.2 119
5.4 Data Analysis 124
5.4.1 Models 125
5.4.2 Sampling Results 125
5.5 General Remarks 127

6 Polya Tree Based Nearest Neighbor Regression 129

6.1 Introduction 129
6.2 Model Formulation 130
6.3 Asymptotic Properties 132
6.4 Inference Procedures 134
6.4.1 Selection of Tuning Parameter h 134
6.4.2 Sampling Algorithm 135
6.5 Simulation Studies 137
6.5.1 Simulation Settings 137
6.5.2 Evaluation Metrics 139
6.5.3 Simulation Results 139

6.6 Data Analysis 146
6.6.1 Dataset Description 146
6.6.2 Selection of Tuning Parameter η 147
6.6.3 Models to Compare 148
6.6.4 Results 149
6.7 General Remarks 150

7 Discussion and Future Work **151**

References **154**

APPENDICES **172**

A Appendix for Chapter 2 **173**
A.1 Additional Simulation Results 173
A.1.1 Efficiency 174
A.1.2 Robustness 177
A.1.3 Prediction 181
A.2 Data Analysis 193
A.2.1 Dataset Description 193
A.2.2 Model Fitting Results 194

B Appendix for Chapter 3 **195**
B.1 Variability of the Transformed Dependence Parameters 195
B.2 Additional Simulation Results 197
B.3 Additional Results for Data Analysis 199
B.3.1 Marginal Distribution of Six Features in Three Health Groups . . . 199
B.3.2 Dependence Model 203

C Appendix for Chapter 4 **205**

C.1 Additional Simulation Results 205

D Appendix for Chapter 5 **209**

D.1 Proofs of Theorems 209

D.1.1 Proof of Theorem 5.1 209

D.1.2 Proof of Theorem 5.2 214

D.2 Additional Simulation Results 220

D.2.1 Simulation Results of Setting 5.1 220

D.2.2 Simulation Results of Setting 5.2 222

D.3 Additional Results of Data Analysis 227

E Appendix for Chapter 6 **230**

E.1 Proof of Theorems 230

E.1.1 Proof of Theorem 6.1 230

E.1.2 Proof of Theorem 6.2 235

E.2 Additional Simulation Results 245

E.2.1 Setting 6.1: Monte Carlo-Based Results 246

E.2.2 Setting 6.1: Grid-Based Results 248

E.2.3 Setting 6.2: Monte Carlo-Based Results 250

E.2.4 Setting 6.3: Monte Carlo-Based Results 252

E.2.5 Setting 6.3: Grid-Based Results 254

E.2.6 Setting 6.4: Monte Carlo-Based Results 256

List of Tables

2.1 Copula functions and the values of the dependence parameters in the dependence structure within each time block 36

2.2 Simulation results using the four estimation methods: strong dependence and $n = 500$ 38

2.3 Mis-selected rates for copula functions within each block 41

2.4 Copula functions and the values of dependence parameters in dependence structure within time blocks for strong and moderate dependence settings . 43

2.5 MAEs of different models for subject extrapolation under the proposed scenarios 46

2.6 MAEs of different models for time extrapolation under the proposed scenarios 47

2.7 Summary of the selected bivariate copula functions for the C-Vine structure within each year 52

2.8 The estimates of marginal parameters for each month under simultaneous estimation and two-stage estimation of composite likelihood method (standard error in the bracket) 53

2.9 Prediction results for subject extrapolation (prediction standard error in the brackets) 56

2.10 Prediction result for time extrapolation in year 2018 (prediction standard error in the brackets) 57

2.11 Prediction results for subject extrapolation of month 4-12, given the first 3 months (prediction standard error in the brackets) 58

2.12 Prediction results for time extrapolation of month 4-12, given the first 3 months (prediction standard error in the brackets) 59

3.1 Transformation functions for copula parameters 70
3.2 Simulation settings: copula forms and parameters 73
3.3 Simulation results for Setting 3.5 75
3.4 Log-likelihood and DIC of three models for each cluster 80
3.5 Copula functions and estimates for six interested dependence of 3 health groups 82

4.1 Transformation functions and distributions for $G_{\eta_{\gamma r}}$ 88
4.2 Sampling algorithm for M-DPM-CM 92
4.3 Copula Forms and Parameter Values in Each Cluster in the Simulation Set-ups 94
4.4 Simulation results for grouping effects 97
4.5 Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Common Copula Form Setting 99
4.6 Selected copula functions and estimated parameters for the dependence of six pairs of interest in three health groups 101

5.1 Algorithm 5.1: Polya Tree Monte Carlo algorithm 110
5.2 Algorithm 5.2: Polya-Tree Monte Carlo algorithm for $k \geq 2$ 111
5.3 Algorithm 5.3: PTMC Gibbs sampler for a high-dimensional distribution . 112
5.4 Algorithm 5.4: PTMC-MH algorithm for a high-dimensional distribution . 114
5.5 One- and two-dimensional distributions $f(x|\beta)$ with parameter values . . . 120
5.6 D-Vine copulas and the corresponding parameters 121

6.1 Sampling algorithm of PTNN Model 136
6.2 The distributions of the covariate(s), the regression function and the distributions of random errors of six scenarios in each of the four simulation settings 138
6.3 Prediction performance of PTNN versus the kernel density estimation (KDE), kernel regression (KR), linear dependent tail free process (LDTFP1), linear model I (LM1) (6.10), linear model II (LM2) (6.11) and Polya tree density estimation (PT) methods 149

A.1 Simulation results using the four estimation methods: strong dependence and $n = 1000$ . . . 174
A.2 Simulation results using the four estimation methods: moderate dependence and $n = 500$ . . . 175
A.3 Simulation results using the four estimation methods: moderate dependence and $n = 1000$ . . . 176
A.4 Simulation results using the four estimation methods when block-connecting structure is misspecified: strong dependence and $n = 500$ . . . 177
A.5 Simulation results using the four estimation methods when block-connecting structure is misspecified: strong dependence and $n = 1000$ . . . 178
A.6 Simulation results using the four estimation methods when block-connecting structure is misspecified: moderate dependence and $n = 500$ . . . 179
A.7 Simulation results using the four estimation methods when block-connecting structure is misspecified: moderate dependence and $n = 1000$ . . . 180
A.8 Simulation results for subject extrapolation and time extrapolation in terms of percentage outperformance VINE4 versus the other models . . . 181
A.9 Location information of 47 observation stations . . . 193
A.10 Estimates of first parameters of the copula functions in the C-Vine structure obtained by the two-stage estimation procedure (standard error in the bracket) 194
A.11 Estimates of second parameters of the copula functions in the C-Vine structure obtained by the two-stage estimation procedure (standard error in the bracket) . . . 194

B.1 Empirical standard error of the MLE of transformed dependence parameter under various copula functions . . . 196
B.2 Simulation results for Setting 3.1 . . . 197
B.3 Simulation results for Setting 3.2 . . . 197
B.4 Simulation results for Setting 3.3 . . . 198
B.5 Simulation results for Setting 3.4 . . . 198
B.6 MLE of marginal parameters in the generalized skewed-t distributions . . . 200

C.1 Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for High Signal Setting . . . 206

C.2 Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Low Signal Setting 207

C.3 Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Nearly Independent Setting 208

D.1 Simulation results for the Dog bowl distribution 220

D.2 Simulation results for 25-normal mixture distribution 221

D.3 Simulation results for 5-normal mixture distribution 221

D.4 Simulation results for one-dimensional distribution 223

D.5 Simulation results for two-dimensional distribution 224

D.6 Simulation results for Gamma-normal mixture distribution 225

D.7 Simulation results for D-Vine 226

D.8 Data analysis results for the Fishery Data 228

D.9 Data analysis results for the Hidalgo Stamp Data 229

E.1 Setting 6.1: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 246

E.2 Setting 6.1: Square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 247

E.3 Setting 6.1: Grid-based K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500 248

E.4 Setting 6.1: Grid-based square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500 249

E.5 Setting 6.2: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 250

E.6 Setting 6.2: Square root of MISE (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 251

E.7 Setting 6.3: K-L divergence (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 252

E.8 Setting 6.3: Square root of MISE (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 253

E.9 Setting 6.3: Grid-based K-L divergence (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500 254

E.10 Setting 6.3: Grid-based square root of MISE (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500 255

E.11 Setting 6.4: K-L divergence (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 256

E.12 Setting 6.4: Square root of MISE (standard error$\times$10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500 257

List of Figures

1.1 An illustration of a C-Vine with 4 random variables 7

1.2 An illustration of a D-Vine with 4 random variables 8

1.3 The first two levels of the sequence of the nested partitions of an interval space $\mathcal{S} = [a, b]$ 15

1.4 Quaternary partition of $\mathcal{S} = [a_1, b_1] \times [a_2, b_2]$ 18

2.1 A R-Vine structure for 4 time blocks and 4 time points within each block . 28

2.2 Boxplots of MAEs of different models for subject extrapolation 48

2.3 Boxplots of MAEs of different models for time extrapolation 49

2.4 The monthly temperature of all 47 stations from Jan. 1978 to Dec. 2018 . 50

2.5 Boxplot of MAEs for the short-term (on the left) and mid-term (on the right) time extrapolation 55

3.1 A three-level hierarchical structure 62

3.2 The top level of a D-Vine structure 73

3.3 Sample trace plots and sample density plots of mean parameters φ_l and μ_{jl} for $j = 1, 2, 3, 4$ and $l = 1, 2$ of the BHCM with $L = 4$ in Setting 3.5 77

3.4 Sample trace plots and sample density plots of copula parameters θ_{jl} for $j = 1, 2, 3, 4$ and $l = 1, 2$ of the BHCM with $L = 4$ in Setting 3.5 78

4.1 Scatter plots for the three groups identified by the M-DPM-CM 102

5.1 An example of a multimodal distribution 113

5.2 The 3-D density plots of target distributions 116

5.3 Plots of samples from the dog bowl distribution using various algorithms . 117

5.4 Plots of samples from the 25-normal mixture using various algorithms . . . 118

5.5 Plots of samples from the 5-normal mixture using various algorithms . . . 119

5.6 D-Vine structure and copula functions 121

5.7 RCTs for different distributions in Setting 5.2.1 123

5.8 Histograms and 3-component Gaussian mixture density of two datasets . . 124

5.9 The Fishery data: Sample plots of the means and standard deviations of the Gaussian mixture model 126

5.10 The Hidalgo Stamp data: Sample plots of of the means and standard deviations of the Gaussian mixture model 128

6.1 A scatterplot of the response versus the covariate in Settings 6.1-6.3 ($n = 500$) 137

6.2 K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.1 141

6.3 K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.2 142

6.4 K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.3 142

6.5 K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.4 143

6.6 Grid-based K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method, Polya tree density estimation, LDTFP1 with linear predictor and LDTFP2 with quadratic predictor for Setting 6.1 145

6.7 Grid-based K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method, Polya tree density estimation, LDTFP1 with linear predictor and LDTFP2 with quadratic predictor for Setting 6.3 145

6.8 Scatter plots of the Net Hourly Electrical Energy Output (y) versus Temperature (x_1), Ambient Pressure (x_2), Relative Humidity (x_3) and Exhaust Vacuum (x_4), respectively, and the plot of Exhaust Vacuum (x_4) versus Ambient Pressure (x_2) 146

6.9 The cross-validated mean absolute error changes with respect to the value of η 148

6.10 Histograms with superimposed curves of the estimated conditional densities by PTNN model at five different covariate values 149

B.2 Histograms of six biomedical features on three groups 202

B.4 Scatter plots of six pairs of bivariate dependence in 3 health groups 204

Chapter 1

Introduction

1.1 Background

In this thesis, we concentrate on exploring topics regarding correlated data modeling with copula-based models and Bayesian nonparametric methods on sampling and regression problems. More specifically, the thesis is divided into two parts. The first part investigates copula-based models on correlated data, including parameter estimation, model selection and similar dependence structure grouping. The second part is devoted to the Bayesian nonparametric methods on sampling and nonparametric regression.

Correlated data of multiple types are more than common in real life and the analyses of such data are also thriving topics in statistical science. Many researchers proposed different modeling methods to account for dependence in different areas, such as survival analysis (e.g., Andersen, 2005; Braekers and Veraverbeke, 2005; Bogaerts and Lesaffre, 2008; Geerdens et al., 2016) and longitudinal data analysis (e.g., Diggle et al., 2002; Hedeker and Gibbons, 2006; Fitzmaurice et al., 2009; Verbeke and Molenberghs, 2009; Verbeke et al., 2014). A large part of the literature utilized dependence modeling as an extension of the original marginal models, yet still considered the marginal models of prime interest. Focusing on multivariate dependence modeling, we use copula models and vine copula models as fundamental building blocks to investigate new estimation procedures and propose valid model selection and grouping methods for dependent data analysis.

Since the concept of "big data" becomes popular, more and more data with irregular distributions, often featured with strong skewness and/or multiple modes, emerge. For example, many machine learning models, especially neural networks, create irregular and multi-modal parameter spaces, which are difficult to be characterized by a parametric form. In comparison, nonparametric methods, requiring less model assumptions, provide more robust results for statistical modeling and inference on such kinds of data. In this thesis, we propose Bayesian nonparametric sampling algorithms and regression models to explore and describe data with irregular distributions.

For the rest of the chapter, we review the topics related to the thesis. In Section 1.2, we introduce the basic formulation of copula models and relevant topics, including estimation and model selection. In Section 1.3, we introduce the vine copula models and relevant topics. In Sections 1.4 and 1.5, we summarize the basic theory for longitudinal studies and composite likelihood, respectively, which will be explored in depth in Chapter 2. In Section 1.6, we discuss hierarchical models from the frequentist's viewpoint and as well as the Bayesian perspective. In Section 1.7, we introduce two Bayesian nonparametric models, Dirichlet Process and Polya Tree. In Section 1.8, nonparametric regression models are introduced and briefly reviewed. In Section 1.9, we provide the outline of the thesis.

1.2 Copula

1.2.1 Definition

For n random variables $X_1, \ldots, X_n$, the dependence between them can be described by their joint distribution function, denoted $F(x_1, \ldots, x_n)$. Copula models were proposed by Sklar (1959) to separate the joint distribution into a part that represents the dependence structure and other parts that describe the marginal distributions of the random variables. The formal definition of a copula (Kolev et al., 2006) is given as follows.

Definition 1.1 (Copula)**.** *An n-dimensional copula is a function C: $[0,1]^n \to [0,1]$ with the properties:*

1. *For every $u = (u_1, \ldots, u_n)^T \in [0,1]^n$, $C(u) = 0$ if there is at least one index $i = 1, \ldots, n$ such that $u_i = 0$;*
2. *For every $u \in [0,1]^n$ and $v \in [0,1]^n$ with $u_j \leq v_j$ for $j = 1, \ldots, n$, the C-volume $V_C([u, v])$ is non-negative (see Nelsen (2007), Definition 2.10.1 for the definition of C-volume);*
3. *$C(1, \ldots, 1, u_j, 1, \ldots, 1) = u_i$ for all $u_j \in [0,1]$ with $j = 1, \ldots, n$.*

The copula function can be equivalently defined as a multivariate distribution with uniform margins, with the copula density calculated by

$$c(u_1, \ldots, u_n) = \frac{\partial^n C(u_1, \ldots, u_n)}{\partial u_1 \ldots \partial u_n}.$$

Theorem 1.1 (Sklar's Theorem (Sklar, 1959))**.** *Let F be an n-dimensional distribution function with margins $F_{X_1}, \ldots, F_{X_n}$. Then there exists an n-dimension copula C such that for all $(x_1, \ldots, x_n)^T \in (-\infty, \infty)^n$,*

$$F(x_1, \ldots, x_n) = C(F_{X_1}(x_1), \ldots, F_{X_n}(x_n)). \tag{1.1}$$

Conversely, if C is an n-dimension copula and $F_{X_1}, \ldots, F_{X_n}$ are distribution functions, the function F in (1.1) is an n-dimension distribution function with margins $F_{X_1}, \ldots, F_{X_n}$. Furthermore, if the marginals are all continuous, C is unique. Otherwise, C is uniquely determined on Ran(F_{X_1})$\times \cdots \times$Ran(F_{X_n}), where Ran(F_{X_j}) represents the range of F_{X_j} for $j = 1, \ldots, n$.

Using the Sklar's Theorem, the density of an n-dimensional distribution function can be expressed as

$$f(x_1, \ldots, x_n) = \left[\prod_{j=1}^{n} f_{X_j}(x_j)\right] c(F_{X_1}(x_1), \ldots, F_{X_n}(x_n)).$$

where $f_{X_j}(\cdot)$ is the marginal density function of X_j and $c(\cdot)$ is the copula density.

Sklar's Theorem ensures the existence of copula functions, serving as the core of copula theory. Many useful parametric forms for copula functions are available, especially for describing bivariate data. Commonly used parametric forms include Gaussian copula, t copula, which are derived from the multivariate Gaussian distribution and the multivariate t distribution, and the Archimedean family, in which the copulas assume the form

$$C(u_1, \ldots, u_n) = \psi^{[-1]}(\psi(u_1) + \cdots + \psi(u_n)),$$

where $\psi : [0, 1] \to [0, \infty)$, called the generator function, is a continuous and strictly decreasing function such that $\psi(1) = 0$, and the $pseudo{-}inverse$ of ψ, $\psi^{[-1]}$, is a continuous and non-increasing function defined on $[0, \infty)$ such that

$$\psi^{[-1]}(t) = \begin{cases} \psi^{-1}(t), & 0 \le t \le \psi(0), \\ 0, & \psi(0) < t \le \infty. \end{cases}$$

where $\psi^{-1}(\cdot)$ is the inverse function of $\psi(\cdot)$. Commonly used copulas in the Archimedean family include the Clayton, Gumbel, Frank and Joe copulas. More details can be found in Nelsen (2007).

Copula has been widely applied in finance and econometrics (Cherubini et al., 2004; Van Den Goorbergh et al., 2005; Chen and Fan, 2006b; Hu, 2006; Jondeau and Rockinger, 2006; Aas et al., 2009; Chollete et al., 2009; Patton, 2012), survival analysis, especially in semi-competing risk modeling (Andersen, 2005; Braekers and Veraverbeke, 2005; Jiang et al., 2005; Romeo et al., 2006; Huang and Zhang, 2008; Bogaerts and Lesaffre, 2008; Geerdens et al., 2016), spatial analysis (Staicu et al., 2012; Boehm et al., 2013; Erhardt et al., 2015; Krupskii and Genton, 2017), and genetic data analysis (He et al., 2012). In most of these papers, bivariate or tri-variate copulas were employed to model the dependence, mainly because multivariate copulas are not flexible enough to describe complex dependence structures and the interpretation of the model parameters becomes difficult. In the case of three dimensions or higher, vine copula models were introduced to circumvent those issues, which will be elaborated in Section 1.3.

1.2.2 Model Selection and Parameter Estimation

The selection of copula functions and the estimation of copula parameters are thriving research topics. Fermanian (2005) and Genest et al. (2006) proposed two different goodness-of-fit tests for copula models, mainly focusing on the bivariate case. Genest et al. (2009) provided a comprehensive review and studied the possible types of goodness-of-fit tests on copula. Besides the goodness-of-fit tests, Chen and Fan (2005) proposed a pseudo-likelihood ratio test for copula model selection, and Chen and Fan (2006a) introduced a model selection and estimation method for copula models with misspecification. A method of combining the traditional model selection criterion, such as AIC and BIC, with the copula model selection was used by Hans (2007), which has remained to be the most prevailing method in copula model selection.

Several methods of estimating the copula parameters are available in the literature. The maximum likelihood (ML) method (Joe, 1997; Dissmann et al., 2013; Stober and Schepsemeier, 2013) is the most commonly-used. However, it requires a lot of computational resources when a large number of parameters appear. A computationally friendly but less efficient alternative is the inference functions for margin (IFM) method, which was proposed by Joe and Xu (1996) and whose asymptotic properties was studied by Joe (2005). Another estimation method, the ranked-based method, estimates copula parameters by utilizing its relationship with Kendall's τ. The method is restrictive to single-parameter copula functions for the problems with an explicit form linking the dependence parameter and Kendall's τ. Despite the popularity of copula in dependency modeling, applications of the Bayesian theory to the copula field are relatively limited, briefly summarized by Smith (2011).

1.3 Vine Copula

1.3.1 Definition

Compared to the bivariate case, using multivariate copulas to describe multivariate distributions seems relatively limited. Multivariate versions of Gaussian and t copulas usually fail to model the possible tail dependence in real life. Multivariate copulas in the Archimedean family are difficult to interpret as they use only one association parameter to describe complex multivariate dependence. In order to flexibly model dependence structures in high dimensional settings using copula models, Bedford and Cooke (2002) proposed the concept of vine copula. With the pair-copula construction proposed by (Aas et al., 2009), regular vine (R-Vine) copulas can be used to decompose an n-dimensional multivariate distribution into $n(n-1)/2$ bivariate distributions, where n is a positive integer greater than 2. This kind of decomposition enjoys the convenience of parameter estimation and the flexibility of modeling the dependence structure among random variables. Bedford and Cooke (2002) gave the definition of vine and R-Vine.

Definition 1.2 (Vine Copula)**.** *$\mathcal{V} = (T_1, \ldots, T_m)$ is a* vine *on n elements with m vine trees if:*

1. *T_1 is a tree with nodes $N_1 \in \{1, \ldots, n\}$ and a set of edges, denoted E_1, containing edges that connect two nodes;*

2. *For $i = 2, \ldots, m$, T_i is a tree with nodes $N_i \subset N_1 \cup E_1 \cup E_2 \cup \cdots \cup E_{i-1}$ and the edge set E_i containing edges that connect two nodes.*

A vine $\mathcal{V}$ is a regular vine *on n elements if:*

1. *$m = n - 1$;*

2. *T_i is a connected tree with the edge set E_i and the node set $N_i = E_{i-1}$, with the cardinality of N_i equal to $n - (i - 1)$ for $i = 1, \ldots, n$, where E_0 is the null set;*

3. *The proximity condition holds: for $i = 2, \dots, n-1$, if $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$ are two nodes in N_i connected by an edge, with $a_1, a_2, b_1, b_2 \in N_{i-1}$, then the cardinality of $a \cap b$ equals 1.*

In this thesis, we only consider the regular vine (R-Vine), in which an n-dimensional copula is decomposed into $n(n-1)/2$ bivariate (conditional) copulas. With the decomposition of the R-Vine model, the joint density is divided into $(n-1)$ vine trees and in tree T_k for $k = 1, \dots, n-1$, there are $(n-k)$ edges, representing the bivariate (conditional) dependence of any two random variables in $(u_1, \dots, u_n)^{\mathrm{T}}$. The joint density is given as

$$c(u_1, \dots, u_n) = \prod_{k=1}^{n-1} \prod_{(e_1,e_2) \in E_k} c_{e_1 e_2}(u_{e_1|D_e}, u_{e_2|D_e}; \theta_{e_1 e_2}),$$

where e_1, e_2 can be any two random variables in $(u_1, \dots, u_n)^{\mathrm{T}}$, E_k is the set of edges in vine tree T_k, D_e is the conditioning set for the edge (e_1, e_2) including all random variables connected with e_1, e_2 in the previous vine trees, and $(e_1|D_e, e_2|D_e)$ is an edge in the edge set E_k with the copula density $c_{e_1 e_2}(\cdot, \cdot)$ and parameters $\theta_{e_1 e_2}$. The conditional terms $u_{e_1|D_e}$ and $u_{e_2|D_e}$ are calculated by applying the following formulas iteratively,

$$u_{p|q} = \frac{\partial C_{pq}(u_p, u_q)}{\partial u_q} \quad \text{and} \quad u_{q|p} = \frac{\partial C_{pq}(u_p, u_q)}{\partial u_p}. \tag{1.2}$$

1.3.2 Canonical Vine and D-Vine

We now introduce two most commonly used vine copula forms, Canonical vine (C-Vine) and D-Vine. C-Vine is a special case of R-Vine such that each vine tree has a dominating variable connected with the remaining variables. A 4-variable C-Vine is illustrated in Figure 1.1. For $i = 1, \dots, n-1$, T_i has $(n+1-i)$ nodes and $(n-i)$ edges. In T_1, the $(n-1)$ edges represent the dependence between random variable u_1 and other $(n-1)$ random variables. In T_2, the $(n-2)$ edges represent the conditional dependence relation between random variable u_2 and other remaining $(n-2)$ random variables, given random variable u_1. Similarly, for T_i, the $(n-i)$ edges represent the conditional dependence relation between random variable u_i and other remaining $(n-i)$ random variables, given random variables $u_1, \dots, u_{i-1}$. As a result, the n-dimensional joint density function $c(u_1, \dots, u_n)$ of $U_1, \dots, U_n \in [0,1]^n$ can be decomposed as

$$c(u_1, \dots, u_n; \theta) = \prod_{i=1}^{n-1} \prod_{k=i+1}^{n} c_{ik}(u_{i|\mathcal{D}_{ik}}, u_{k|\mathcal{D}_{ik}}; \theta_{ik}),$$

where $c_{ik}(\cdot,\cdot)$ denotes the bivariate copula density of random variables u_i and u_k conditional on the conditioning set $\mathcal{D}_{ik} = \{u_1, \ldots, u_{i-1}\}$, and $\theta = (\theta_{ik} : i = 1, \ldots, n-1; k = i+1, \ldots, n)^{\mathrm{T}}$ denotes the parameter vector. Note that when $i = 1$, the conditioning set is empty.

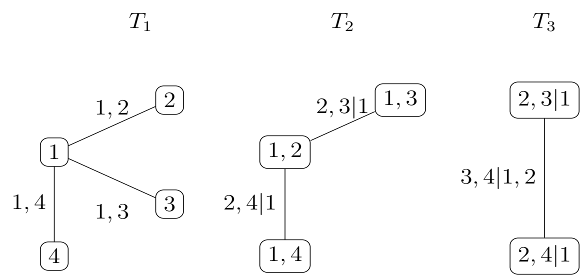


Figure 1.1: An illustration of a C-Vine with 4 random variables

D-Vine is another special case of R-Vine that in each vine tree, a variable is connected with the two closest variables. A 4-variable D-Vine is illustrated in Figure 1.2. In T_1, for $k = 1, \ldots, n-1$, random variable u_k is connected with variable u_{k+1}. In T_2, for $k = 1, \ldots, n-2$, random variable u_k is connected with variable u_{k+2} conditional on random variable $\mathcal{D}_{k,k+2} = u_{k+1}$. Similarly, for T_i, random variable u_k is connected with variable u_{k+i} conditional on random variables $\mathcal{D}_{k,k+i} = \{u_{k+1}, \ldots, u_{k+i-1}\}$. As a result, the n-dimensional joint density function $c(u_1, \ldots, u_n)$ of $U_1, \ldots, U_n$ can be decomposed as

$$c(u_1, \ldots, u_n; \theta) = \prod_{i=1}^{n-1} \prod_{k=1}^{n-i} c_{k,k+i}(u_{k|\mathcal{D}_{k,k+i}}, u_{k+i|\mathcal{D}_{k,k+i}}; \theta_{k,k+i}),$$

where $c_{k,k+i}(\cdot,\cdot)$ denotes the bivariate copula density of random variables u_k and u_{k+i} conditional on the conditioning set $\mathcal{D}_{k,k+i}$, and $\theta = (\theta_{ik} : i = 1, \ldots, n-1; k = i+1, \ldots, n)^{\mathrm{T}}$ denotes the parameter vector. Note that when $i = 1$, the conditioning set is a null set.

It is worth mentioning that the order of listing the random variables matters in determining the dependence structure. In practice, the orders of the random variables are usually set based on the context of the problem. For example, for longitudinal data, we can label the variables based on the temporal order, and for spatial data, we can label the variables based on the spatial distance.

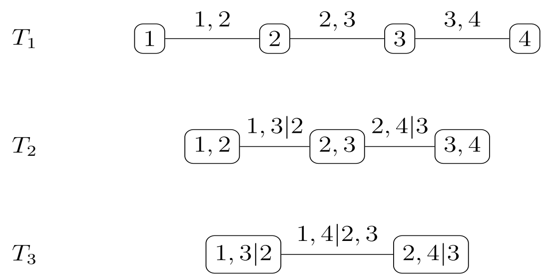


Figure 1.2: An illustration of a D-Vine with 4 random variables

1.3.3 Model Selection and Parameter Estimation

The selection of bivariate copula functions for each (conditional) bivariate dependence in the R-Vine structure is commonly done through a sequential way (Dissmann et al., 2013). The copula forms are first selected for each bivariate dependence relation in tree T_1 separately in the same way as discussed in Section 1.2.2. After obtaining the selected copula forms and the corresponding estimates, we calculate the transformed copula inputs for T_2, $(u_{e_1|D_e}, u_{e_2|D_e})$, through (1.2). We proceed to select the copula forms for bivariate dependence in tree T_2. The selection of the remaining bivariate dependence is done through a similar tree-by-tree way. Dissmann et al. (2013) also proposed the sequential estimation method for vine copula models, which is a fast estimation method for high dimensional data compared to the ML method at the price of losing efficiency. For more details, refer to Dissmann et al. (2013).

Likelihood methods can also be applied to estimating parameters in vine copula models. But as the dimension increases, the number of parameters in an R-Vine increases quadratically, which makes the implementation of likelihood based methods prohibitive in high dimensional settings. Brechmann et al. (2012) proposed the truncated and simplified vine copula based on the Vuong test to reduce computation burdens. From the Bayesian viewpoint, Smith et al. (2010) used the Bayesian model selection methods to identify possible independent (conditional) bivariate copulas in the vine copula model. Min and Czado (2010) suggested using Bayesian inference on the pair-copula constructions (PCC). Gruber and Czado (2015) and Gruber and Czado (2018) discussed both sequential and simultaneous methods for selecting copula forms in a regular vine structure using reversible jump Markov Chain Monte Carlo (MCMC). Generally speaking, the fast and robust inference on vine copula model is a key concern in the literature of vine copulas.

1.4 Longitudinal Data Analysis

Longitudinal data analysis, which studies the change of repeated observations of the same subjects over time, has long been a thriving topic in statistical research. There have been a large body of books, papers and reviews in this field, such as Diggle et al. (2002), Hedeker and Gibbons (2006), Fitzmaurice et al. (2009) and Verbeke et al. (2014). Linear mixed effects (LME) models (Verbeke and Lefaffre, 1996, 1997; Muthén and Shedden, 1999; Heagerty and Zeger, 2000; Zhang and Davidian, 2001; Ghidey et al., 2004; Litière et al., 2008; Verbeke and Molenberghs, 2009) are one of the commonly-used models for continuous repeated observations:

$$Y = X\beta + Zu + \varepsilon,$$

where X and Z are design matrices featuring fixed effects and random effects, respectively, and u is the vector of subject-wise random effects which is often assumed to have mean 0, and ε is the vector of random error terms assumed to have mean 0. In LME models, subject-specific random effects are mixed with the fixed effects to account for the within-subject variability and the between-subject variability. LME models usually lead to analytically intractable likelihood functions, except for the case where both u and $varepsilon$ follow a normal distribution. .

Generalized linear mixed effects (GLME) models can be seen as a combination of LME models and generalized linear models. GLME models are constructed by introducing random effects in the linear predictor of a generalized linear model. GLME models and their extensions are widely discussed in the literature, including Breslow and Clayton (1993), Breslow and Lin (1995), McCulloch (1997), Natarajan and Kass (2000), Raudenbush et al. (2000), Duchateau and Janssen (2005), Lee et al. (2006), Ng et al. (2006), Craiu et al. (2011), Goldstein (2011), and Yi et al. (2017).

Introduced by Liang and Zeger (1986), generalized estimating equations (GEE) have become one of the most popular methods to estimate marginal model parameters. Lipsitz et al. (1991), Becker and Balagtas (1993), Miller et al. (1993), Kenward et al. (1994), Lipsitz et al. (1994), Molenbergh and Lesaffre (1994), Heagerty and Zeger (2000), Qu et al. (2000), Wang and Carey (2004) and Ye and Pan (2006) are also important references on this topic. Estimating equations are constructed without specifying the complete joint distribution of the repeated responses, but they provide consistent estimators of the marginal parameters, provided the correct specification of the mean structure, together with other regularity conditions.

To model longitudinal discrete data, transitional models are proposed to characterize the alteration of responses over time and the influence of the covariates on the transitional probabilities under the usual Markov assumption. Transitional models concentrate on the conditional expectation of current observation Y_{ij}, given the past observations $Y_{i,j-1}, \ldots, Y_{i1}$. From Diggle et al. (2002), a simple example of logistic regression on longitudinal binary data is

$$\log\left\{\frac{\mathrm{P}(Y_{ij}=1|Y_{i,j-1},\ldots,Y_{i1},X_{ij})}{1-\mathrm{P}(Y_{ij}=1|Y_{i,j-1},\ldots,Y_{i1},X_{ij})}\right\} = X_{ij}^{\mathrm{T}}\beta + \alpha Y_{i,j-1},$$

where β and α are model parameters associated with covariates and past observations, respectively. Conventionally, transitional models are employed to study the covariate effects on transition probabilities for univariate longitudinal data, in which a single longitudinal sequence of responses is analyzed. Related works include Muenz and Rubinstein (1985), Lee and Kim (1998), Cook (1999), Albert (2000), Heagerty (2002), Koru-Sengul et al. (2007), Zeng and Cook (2007) and Cheon et al. (2014).

1.5 Composite Likelihood

First coined by Lindsay (1988), composite likelihood is a pseudo-likelihood which is defined by multiplying a collection of component likelihoods. The set of variables included in the composite likelihood is often determined by the particular context of the problems. Although composite likelihood may not be as efficient as the ordinary likelihood, it provides estimators that are consistent and have asymptotic normal distributions under regularity conditions. Moreover, composite likelihood is generally perceived as a more robust estimation method than the ordinary likelihood (though there are exceptions), because it reduces the risk of model misspecification by leaving the part of less interest in full likelihood unspecified. In addition, composite likelihood also reduces the computational burden significantly when we leave high-order association structures unspecified. Generally speaking, composite likelihood is a trade-off between estimation efficiency and robustness. For a comprehensive review of composite likelihood, see Varin (2008), Varin et al. (2011), Lindsay et al. (2011) and Yi (2017a).

For a given $k = 1, \ldots, d$, let S_k be the collection of subsets of k elements of a set of random variables $\{Y_1, \ldots, Y_d\}$ with $\{y_{i1}, \ldots, y_{id}\}$ for $i = 1, \ldots, n$ to be the realization. For $S \in S_k$, let $f(s; \theta_S)$ be the corresponding k-dimensional probability density function of S, where θ_S is the vector of parameters of interest. Let s_i denote a realization of S for $i = 1, \ldots, n$. Then for a given subset $\mathcal{K} \in \{1, \ldots, d\}$, a composite likelihood is defined as

$$CL_i(\theta) = \prod_{k\in\mathcal{K}} \prod_{s_i \in S_k} f(s_i; \theta_S),$$

where $\theta = (\theta_S : S \in S_k; k \in \mathcal{K})^{\mathrm{T}}$, and the parameters θ can be estimated as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} CL_i(\theta).$$

Following Varin et al. (2011), under regularity conditions, the following asymptotic results of $\hat{\theta}$ hold:

(1) $\hat{\theta} \xrightarrow{p} \theta$ as $n \to \infty$,

(2) $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} \mathrm{MVN}(0, G^{-1}(\theta))$ as $n \to \infty$,

where $G(\theta) = H(\theta)J^{-1}(\theta)H^{\mathrm{T}}(\theta)$ is the Godambe information matrix, and the sensitivity matrix $H(\theta)$ and the variability matrix $J(\theta)$ are of the forms

$$\begin{aligned} H(\theta) =& E\left[\frac{\partial^2}{\partial\theta\partial\theta^{\mathrm{T}}} CL_i(\theta)\right]; \\ J(\theta) =& E\left\{\left[\frac{\partial}{\partial\theta} CL_i(\theta)\right]\left[\frac{\partial}{\partial\theta} CL_i(\theta)\right]^{\mathrm{T}}\right\}. \end{aligned}$$

1.6 Hierarchical Models

Multilevel models, also known as hierarchical linear models, are multi-stage statistical models used for modeling nested data with hierarchical structures. A commonly used example for nested data or hierarchical data in literature is school data. Students in a school belong to different classes in different grades. As a result, the individual measure of a certain student has a hierarchical structure, which can be described by a multilevel model. Multilevel models extends linear regression models with the hierarchical structures incorporated; they can also be generalized to nonlinear problems. In multilevel models, the regression models in a certain stage are constructed based on the regression parameters in the previous stage, which allows the parameters to vary at multiple levels. A simple form of multilevel model is as follows:

$$\begin{aligned} \text{Level 1:} \quad & Y_{ij} = \beta_{i0} + \beta_{i1}X_{ij} + \varepsilon_{ij}; \\ \text{Level 2:} \quad & \beta_{i0} = \alpha_{00} + \alpha_{01}Z_i + e_{i0}; \\ & \beta_{i1} = \alpha_{10} + \alpha_{11}Z_i + e_{i1}; \end{aligned} \tag{1.3}$$

where X_{ij} and Z_i are covariates for level 1 and level 2, respectively, and ε_{ij} and $\{e_{i0}, e_{i1}\}$ are random error terms of level 1 and level 2, respectively. Raudenbush and Bryk (2002), Tabachnick et al. (2007), Goldstein (2011) and Garson (2012) provided comprehensive discussions on the multilevel models.

Bayesian hierarchical models (Congdon, 2010, 2014) are used to analyze nested data with hierarchical structures in the Bayesian framework. By assuming exchangeability between parameters, Bayesian hierarchical models place prior distributions on the parameters and hyperprior distribution on parameters in the prior distribution, which is a sensible way to model the hierarchical structure of the data. A simple 3-stage Bayesian hierarchical model is

$$\begin{aligned} &\text{Stage 1:} \quad y_{ij}|\theta_j \sim F(y_{ij}|\theta_j); \\ &\text{Stage 2:} \quad \theta_j|\phi \sim F(\theta_j|\phi); \\ &\text{Stage 3:} \quad \phi \sim F(\phi); \end{aligned} \tag{1.4}$$

where θ_j and ϕ are parameters in the prior and hyperprior distribution representing different levels of hierarchies in the data. Bayesian hierarchical models are widely applied in genetic studies (Broët et al., 2002), spatial temporal analysis in epidemiology and ecology (Wikle et al., 1998; Borsuk et al., 2001; Wikle et al., 2001; Lawson, 2013), longitudinal and survival analysis (Brown and Ibrahim, 2003), and machine learning (Fei-Fei and Perona, 2005; George and Hawkins, 2005).

1.7 Bayesian Nonparametric Methods

1.7.1 Dirichlet Process

The Dirichlet process (DP) prior is a Bayesian nonparametric model introduced by Ferguson (1973), which was originally used to approximate certain probability density functions. From Müller et al. (2015), the formal definition of a Dirichlet process is given as follows.

Definition 1.3 (Dirichlet Process)**.** *Let $\alpha > 0$ and G_0 be a probability measure defined on probability space S. A DP with parameters (α, G_0) is a random probability measure G defined on S which assigns probability $G(B)$ to every set B such that for each finite partition $(B_1, \dots, B_k)$ of S, the joint distribution of the vector $(G(B_1), \dots, G(B_k))$ is the Dirichlet distribution with parameters $(\alpha G_0(B_1), \dots, \alpha G_0(B_k))$.*

A Dirichlet process $\mathrm{DP}(\alpha, G_0)$ consists of two components: a positive tuning parameter α and a base distribution G_0. The tuning parameter α controls the closeness of the generated distribution G to the base distribution G_0 in a way that as $\alpha \to \infty$, $G \to G_0$. The tuning parameter α also controls the degree of discreteness of the generated probability distribution G such that as $\alpha \to \infty$, the distribution G becomes continuous and approaches G_0.

Ferguson (1973) proved the existence of the process and the conjugacy of the DP prior on independent and identically distributed (i.i.d.) samples. Blackwell and MacQueen (1973) proposed the Pólya Urn sampling scheme for the DP prior to induce the marginal distribution for certain samples. Sethuraman (1994) provided the "Stick Breaking Construction" of the Dirichlet process, which offered more insights into the Dirichlet process.

Definition 1.4 (Stick Breaking Construction)**.** *For $h = 1, 2, \dots,$ let $w_h = v_h \prod_{l<h}(1 - v_l)$ with $v_h \sim Beta(1, \alpha)$, and $m_h \sim G_0$, where v_h and m_h are independent. Then*

$$G(\cdot) = \sum_{h=1}^{\infty} w_h \delta_{m_h}(\cdot) \tag{1.5}$$

defines a $DP(\alpha, G_0)$ random probability measure, where $\delta_{m_h}(\cdot)$ is a Dirac measure defined on m_h.

Since the DP generates discrete distributions, sometimes it is mixed with some simple continuous parametric functions to accommodate the continuous data, which is called the Dirichlet process mixture (DPM) model (Ferguson, 1983; Lo, 1984; Escobar, 1994; Escobar and West, 1995). A simple example of the DPM model is

$$\begin{aligned} y_{ij}|\theta_j &\sim F(y_{ij}|\theta_j); \\ \theta_j|G &\sim G; \end{aligned} \tag{1.6}$$

where $G \sim \mathrm{DP}(\alpha, G_0)$. Neal (2000) introduced several efficient sampling algorithms for inference with both the conjugate and non-conjugate DP or DPM models. If we further assume that the prior parameters in DP to be random parameters with a prior distribution, we can obtain the mixture of DPM as:

$$\begin{aligned} y_{ij}|\theta_j &\sim F(y_{ij}|\theta_j); \\ \theta_j|G &\sim G; \\ G|\alpha, \eta &\sim \mathrm{DP}(\alpha, G_\eta) \\ \alpha, \eta &\sim \pi(\alpha, \eta) \end{aligned} \tag{1.7}$$

Due to the discrete nature in (1.5), DP or DPM models are widely applied in many clustering research problems, such as Kim et al. (2006), Dahl (2006), Vlachos et al. (2009), and Yu et al. (2010). For details, refer to Müller et al. (2015).

1.7.2 Polya Tree

The discrete nature of the Dirichlet process refrains its performance on distributions with continuous densities. An attractive alternative of the Dirichlet process on low-dimensional sample space is the Polya tree (PT). The Polya tree includes the Dirichlet process as its special case, but with an appropriate setting of the PT parameters, the PT will generate continuous distribution with probability one (Müller et al., 2015).

Essentially defining a random histogram, the PT was first introduced and studied by Ferguson (1973) and Blackwell and MacQueen (1973). Mauldin et al. (1992) and Lavine (1992, 1994) provided more systematic research of the property of PT. Mauldin et al. (1992) proved that PT can be viewed as the De Finneti measure in a generalized Polya urn scheme. PT was connected with the Pólya Urn scheme (e.g., Monticino, 2001), which led to the proof of many properties. A complete introduction of the Polya tree can be found in Müller et al. (2015).

A Polya tree distribution, denoted $\mathrm{PT}(\Pi, \mathcal{A})$, is indexed by a sequence of partitions Π, which is of the form of nested binary trees, and a set of parameter $\mathcal{A}$. Before introducing the definition of the Polya tree, two important components of the Polya tree, Π and $\mathcal{A}$, are first discussed. In the following discussion, the univariate or one-dimensional sample space is first considered, and the extension of the Polya tree to a higher dimensional space will be discussed later.

Without loss of generality, we assume that Y is a random variable with domain $\mathcal{S}$. Let $\pi_0 = \mathcal{S}$, $\pi_1 = \{\mathcal{B}_0, \mathcal{B}_1\}$, $\pi_2 = \{\mathcal{B}_{00}, \mathcal{B}_{01}, \mathcal{B}_{10}, \mathcal{B}_{11}\}$, $\ldots$, $\pi_m = \{\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_m} : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$, $\ldots$, be a sequence of nested partitions of $\mathcal{S}$ such that $\cup_{\varepsilon_m=0,1} \mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m} = \mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}}$ and $\cap_{\varepsilon_m=0,1} \mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m} = \emptyset$ for every $\varepsilon_j \in \{0,1\}$ with $j = 1, \ldots, m$ and $m \in N^+$. In other words, the m-level partition π_m splits the domain space $\mathcal{S}$ into 2^m subsets $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$ with $\varepsilon_j \in \{0,1\}$, for $j = 1, \ldots, m$; and π_{m+1} is a refined partition of the domain by further splitting each subset $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$ into $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m 0}$ and $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m 1}$. Therefore, the sequence of partitions is formed by binary splitting of subsets from the previous level of partition and assumes a nested tree structure. The collection of the partitions forms $\Pi = \{\pi_m : m \in N^+\}$. An illustration of the first two levels of Π for an interval space $\mathcal{S} = [a, b]$ with equal-sized partitions is provided in Figure 1.3.

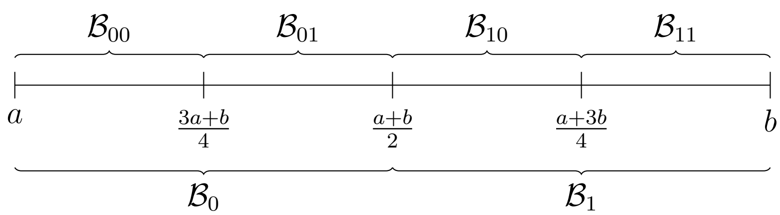


Figure 1.3: The first two levels of the sequence of the nested partitions of an interval space $\mathcal{S} = [a, b]$

Next, we assign a random probability to the subset $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$ through a sequence of conditional random probabilities. Let $G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$ denote the conditional probability that Y falls in the subset $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}\varepsilon_m}$, given that Y falls in the subset $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}}$:

$$G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}\varepsilon_m} = P\big(Y \in \mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}\varepsilon_m} | Y \in \mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}}\big).$$

In a Polya tree distribution, the conditional probabilities $G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}$ from different levels are commonly assumed to be mutually independent Beta random variables

$$G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0} \sim \text{Beta}(\alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}, \alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}1}), \tag{1.8}$$

where $\mathcal{A}_m := \{\alpha_{\varepsilon_1\ldots\varepsilon_m} : \varepsilon_j \in \{0, 1\}, j = 1, \ldots, m\}$ is a set of positive parameters for the m-level partition. $\mathcal{A} = \{\mathcal{A}_m : m \in N^+\}$ is the collection of the parameters for all levels of partitions. Kraft (1964) proved that $\alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m} = m^2$ is a sufficient condition to guarantee probability one assigned to the set of continuous distributions, and m^2 becomes the canonical choice for $\alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$. Schervish (1995) proved the more general conditions that $\alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m} = c\rho(m^2)$ with $\sum_{m=1}^{\infty} \rho(m) < \infty$ is sufficient to guarantee that the Polya tree generates continuous distributions. Following Walker and Mallick (1997), the default choice is $\alpha_{\varepsilon_1\ldots\varepsilon_m} = \phi m^2$, with $\phi > 0$.

Therefore, the probability that Y falls in the subset $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$, denoted $G(\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m})$, can be expressed as the product of the sequence of conditional probabilities

$$G(\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}) = \prod_{j=1}^{m} G_{\varepsilon_1\ldots\varepsilon_j},$$

which is a random probability.

Definition 1.5 (Polya Tree)**.** *Let Π be a sequence of nested binary partitions defined above and let $\mathcal{A} = \{\mathcal{A}_m : m \in N^+\}$ be a collection of non-negative numbers. A random probability* *measure G on $\mathcal{S}$ is said to be a Polya tree with parameters $(\Pi, \mathcal{A})$ if for every $m = 1, 2, \ldots$ and every $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m} \in \pi_m$*

$$G(\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}) = \prod_{j=1}^{m} G_{\varepsilon_1\ldots\varepsilon_j},$$

where the conditional probabilities $G_{\varepsilon_1\ldots\varepsilon_j}$ are mutually independent Beta random variables with

$$G_{\varepsilon_1\ldots\varepsilon_{j-1}0} \sim Beta(\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}0}, \alpha_{\varepsilon_1\ldots\varepsilon_{j-1}1})$$

and $G_{\varepsilon_1\ldots\varepsilon_{j-1}0} = 1 - G_{\varepsilon_1\ldots\varepsilon_{j-1}1}$. We write $G \sim PT(\Pi, \mathcal{A})$.

Polya trees are used as conjugate priors in Bayesian nonparametric statistics in the following sense. If the random variable Y follows a random probability measure G, of which the prior distribution is assumed to be a PT distribution, i.e.,

$$\begin{aligned} Y|G &\sim G \\ G &\sim PT(\Pi, \mathcal{A}), \end{aligned}$$

then the posterior distribution of probability measure G, given the data Y, still follows a PT distribution with

$$G|Y \sim PT(\Pi, \mathcal{A}(Y)), \tag{1.9}$$

where $\mathcal{A}(Y) = \{\mathcal{A}_m(Y) : m \in N^+\}$, $\mathcal{A}_m(Y) = \{\alpha_{\varepsilon_1\ldots\varepsilon_m}(Y) : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$, and

$$\alpha_{\varepsilon_1\ldots\varepsilon_m}(Y) = \begin{cases} \alpha_{\varepsilon_1\ldots\varepsilon_m} + 1 & \text{if } Y \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}, \\ \alpha_{\varepsilon_1\ldots\varepsilon_m} & \text{otherwise.} \end{cases}$$

In other words, if we have n i.i.d. random copies of Y, denoted $(Y_1, \ldots, Y_n)$, the random conditional probabilities, given the sample, are updated as

$$G_{\varepsilon_1\ldots\varepsilon_{m-1}0}|(Y_1, \ldots, Y_n) \sim \text{Beta}\left(\alpha_{\varepsilon_1\ldots\varepsilon_{m-1}0} + N_{\varepsilon_1\ldots\varepsilon_{m-1}0}, \alpha_{\varepsilon_1\ldots\varepsilon_{m-1}1} + N_{\varepsilon_1\ldots\varepsilon_{m-1}1}\right), \tag{1.10}$$

where $N_{\varepsilon_1\ldots\varepsilon_m}$ denotes the number of the sample points in $(Y_1, \ldots, Y_n)$ that fall in the subset $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}$.

Sampling from a PT distribution might be hindered by the need to update an infinite number of parameters which characterize the tree structure. We consider a finite PT (FPT), denoted by $\text{FPT}(\Pi, \mathcal{A}, M)$ (Lavine, 1994) by only updating parameters up to level

M. Under the FPT$(\Pi, \mathcal{A}, M)$, for partition level $m \leq M$, the expectation of the posterior probability of falling into a certain subspace is

$$\begin{aligned} E[G(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})|(Y_1,\ldots,Y_n)] &= E\bigg[\prod_{j=1}^{m} G_{\varepsilon_1\ldots\varepsilon_j}|(Y_1,\ldots,Y_n)\bigg] \\ &= \prod_{j=1}^{m} \frac{\alpha_{\varepsilon_1\ldots\varepsilon_j} + N_{\varepsilon_1\ldots\varepsilon_j}}{\sum_{l=0}^{1} \alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l}}. \end{aligned}$$

For partition level $m > M$, the expectation of the posterior probability of falling into a certain subspace becomes

$$E[G(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})|(Y_1,\ldots,Y_{n^*})] = \bigg(\prod_{j=1}^{M} \frac{\alpha_{\varepsilon_1\ldots\varepsilon_j} + N_{\varepsilon_1\ldots\varepsilon_j}}{\sum_{l=0}^{1} \alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l}}\bigg)\bigg(\prod_{j=M}^{m} \frac{\alpha_{\varepsilon_1\ldots\varepsilon_j}}{\sum_{l=0}^{1} \alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l}}\bigg).$$

Next, we discuss the extension of PT to a higher dimensional sample space. The formulation of the two-dimensional Polya tree is discussed, and the construction on higher dimensions can be derived in a similar way.

Under the two-dimension case, the quaternary partition, illustrated in Figure 1.4, is considered. In the quaternary partition, each subspace of $\mathcal{S}$ in the previous level will be partitioned into 4 parts. Let $\pi_0 = \{\mathcal{S}\}$, $\pi_1 = \{\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}, \ldots, \pi_m = \{\mathcal{B}_{\varepsilon_1,\ldots,\varepsilon_m} : \varepsilon_j = 0,1,2,3; j = 1,\ldots,m\}$, $\ldots$, be a sequence of nested partitions of $\mathcal{S}$ such that $\bigcup_{j=0}^{3} \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m j} = \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}$ and $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m i} \bigcap \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m j} = \emptyset$ for $i \neq j$. In other words, the m-level partition π_m splits the domain space $\mathcal{S}$ into 2^{2m} subsets $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}$ with $\varepsilon_j \in \{0,1,2,3\}$, for $j = 1,\ldots,m$. The nested partition set Π is defined as the collection of all levels of partition $\Pi = \{\pi_m : m \in n^+\}$. The prior parameter set $\mathcal{A} = \{\mathcal{A}_m : m \in N^+\}$ with $\mathcal{A}_m = \{\alpha_{\varepsilon_1\ldots\varepsilon_m} : \varepsilon_j = 0,1,2,3; j = 1,\ldots,m\}$ is defined similarly as the 1-dimension case.

Instead of the Beta distribution, the (conditional) probability of the subspace $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_m}$, given the $(m-1)$-level of partition $\mathcal{B}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}}$, is assumed to be,

$$\begin{aligned} &(G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}, G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}1}, G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}2}, G_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}3}) \\ &\qquad \sim \text{Dirichlet}(\alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}, \alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}1}, \alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}2}, \alpha_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}3}) \end{aligned} \tag{1.11}$$

As the multivariate counterpart of the Beta distribution, the conjugacy property of the Polya tree is retained. Moreover, Ning and Shephard (2018) proved the continuous condition and the consistency of the proposed Dirichlet-based Polya tree. It is noteworthy that the Dirichlet-based Polya tree can be naturally extended to dimensions greater than two.

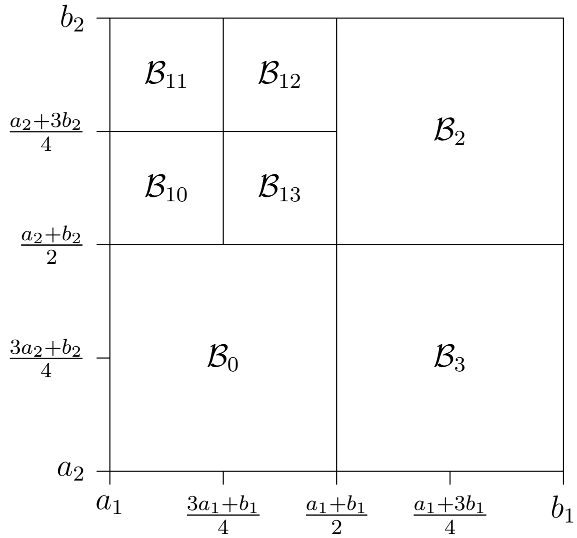


Figure 1.4: Quaternary partition of $\mathcal{S} = [a_1, b_1] \times [a_2, b_2]$

1.8 Nonparametric Regression

Regression analysis is a powerful statistical method for delineating the relationship between responses and covariates of interest. In parametric or semi-parametric regression models (e.g. Kleinbaum and Klein, 2002; Ruppert et al., 2003; Seber and Lee, 2012), specific parametric forms are given to the regression function and/or the error distribution, which are subject to the risks of model misspecification. To mitigate this potential issue, the nonparametric regression models are often considered, which make no assumptions on the form of the regression function and/or the error distribution. The nonparametric regression in the literature often focuses on either the regression function or the random error distribution. There are rich studies on methods of nonparametrically formulating the regression function in the frequentist's literature including kernel method (Gasser and Müller, 1979; Wand and Jones, 1994), spline method (Reinsch, 1967; Eubank, 1999; Wahba, 1990; Schumaker, 2007), and regression trees (Breiman et al., 1984). Under the Bayesian framework, one commonly used Bayesian nonparametric prior for a regression function is the Gaussian process prior (O'Hagan, 1978). It is also common to model the regression function using basis expansions, including the wavelets basis (Chui, 2016), neural network (Hansen and Salamon, 1990; Demuth et al., 2014) and B-splines (De Boor, 1978). Chipman et al. (2010) proposed Bayesian regression trees, which aggregate single regression trees to approximate the regression function. On the other hand, nonparametric modeling of the random error distribution essentially reduces to nonparametric density estimation. The kernel density estimation (Davis et al., 2011) is the most popular approach in the frequentist's framework. The Dirichlet process (DP) prior (Hanson and Johnson, 2004), or the Polya tree (PT) prior (Walker and Mallick, 1999; Müller et al., 2015) are attractive choices as Bayesian nonparametric priors for the random error distribution in the Bayesian framework.

As opposed to parametric regression or semi-parametric regression, fully nonparametric regression characterizes the conditional probability measures of the responses given covariates. A fully nonparametric regression is often formulated as

$$Y|x \sim G_x, \tag{1.12}$$

where G_x is a conditional probability measure of the response variable Y given the covariate value $X = x$. Fully nonparametric regression is usually studied in the framework of Bayesian nonparametric analysis, where Bayesian nonparametric priors serve as the building blocks in the models. For instance, MacEachern (1999) extended the Dirichlet Process (DP) prior to a regression setting and modeled $\{G_x : x \in \mathcal{S}_x\}$ jointly using a dependent DP (DDP) prior, where each G_x follows a DP marginally. Noting that a DP-distributed measure G, indexed by the concentration parameter $\alpha > 0$ and the base distribution H, can be expressed by a stick-breaking construction (Sethuraman, 1994) as $G = \sum_{h=1}^{\infty} w_h \delta_{\theta_h}$, where $w_h = \gamma_h \prod_{l=1}^{h-1}(1 - \gamma_l)$, $\gamma_h \overset{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha)$, δ_θ denotes the Dirac measure at θ, and $\theta_h \overset{\text{i.i.d.}}{\sim} H$. MacEachern (1999) replaced θ_h with stochastic processes $\{\theta_h(x) : x \in \mathcal{S}_x\}$, and De Iorio et al. (2004) introduced ANOVA DDP by assuming $\theta_h = x^{\mathrm{T}}\beta_h$. Griffin and Steel (2006) and Dunson et al. (2007) further considered models in which the weights w_h are dependent on the covariates. Chung and Dunson (2011) proposed the "local Dirichlet Process" to aggregate sample points that are close in the covariate space. The Polya tree (PT) prior, like the DP prior, was also extended to model G_x and the dependent PT (DPT) was proposed parallel to DDP by allowing the random splitting probabilities in a PT distribution to depend on x. Trippa et al. (2011) proposed one such construction by expressing the Beta prior of the random splitting probabilities as a ratio of Gamma random variables, which are modeled by Gamma processes for an area centered around x. Jara and Hanson (2011) proposed the linear dependent tail-free process (LDTFP) and modeled the logistic transformation of the random splitting probabilities by a regression function of covariates x. Generally speaking, the fully nonparametric regression methods are distinguished by the prior adopted for G_x and how G_x is connected with x, which usually demand tremendous computational resources for inferences. In our view, not all aforementioned methods are "fully" nonparametric. For instance, the ANOVA DDP and

LDTFP models assume a regression form of covariates as part of their formulations and thus their performance may be compromised if the assumption is violated.

1.9 Outline of the Thesis

In this section, we briefly introduce the backgrounds and topics for each chapter of the thesis. In Chapter 2, we consider longitudinal data, and incorporate vine copula models together with the regression model to account for the temporal dependence between observations. Chapter 3 studies the modeling of dependent data exhibiting hierarchical structures. Chapter 4 discusses the topics related to identifying similar dependence structures and performing model selection for dependent data from multiple sources using DP process. In Chapter 5, sampling algorithms based on PT are proposed to provide more efficient and powerful alternatives for MCMC in handling complex distributions. Chapter 6 discusses a fully nonparametric regression model based on the Polya tree and the nearest neighbor method. Finally, Chapter 7 presents the concluding remarks of the thesis and some possible working directions for future research. Summaries of Chapters 2-6 are provided below.

Chapter 2: Composite Likelihood Methods for Analyzing Longitudinal Data with a Time-Span under Vine Copula Models

In longitudinal data analysis, modeling temporal dependence among observations is an important topic as reviewed in Section 1.4. Before the development of copula and vine copula models, multivariate normal distributions were usually used to describe the temporal dependence of continuous longitudinal data, of which the dependence is completely determined by the covariance matrix. Using the multivariate normal distributions can greatly simplify the likelihood function in many circumstances. However, as a symmetric distribution, multivariate normal distributions fail to address the possible tail dependence of observations at different time points.

In this chapter, we model longitudinal responses with a time-span using vine copula models to address the temporal dependence between different observations. The temporal length of the longitudinal data determines the number of parameters in regular vine models, which increases quadratically as the time length increases. Thus, directly using vine copula model for longitudinal data with a large time-span will introduce a large number of parameters and hence create difficulties for parameter estimation. As a result, we use composite likelihood to simplify the inference procedures and concentrate on the parameters of primary interest. We also compare different estimation procedures, simultaneous estimation and two-stage estimation, in terms of efficiency and robustness. Moreover, we find in simulation studies that the composite likelihood is robust to misspecification of the structure linking between time blocks, achieves accurate selection of the (conditional) bivariate copulas, and provides a convenient structure for prediction. The material in this chapter has been wrapped up as a paper submitted for publication and will appear in the Journal of Data Science.

Chapter 3: A Bayesian Hierarchical Copula Model

Complex data structures arise commonly in modern scientific research. Examples include data with a hierarchical nesting structure, data collected from different research centers, studies or resources, and data configured at multiple locations or multiple time points, etc. In this chapter, we are interested in studying dependent data with hierarchical structures, and analysis of such data is often challenging due to the complexity in modeling different dependence structures and computation intensity.

To account for the dependence relations and the hierarchical structure, we propose a Bayesian hierarchical copula model (BHCM), which combines the ideas of the copula models and the Bayesian hierarchical models. Different copula models are used to describe the dependence structures of different clusters, and the Bayesian hierarchical model, which is built on the copula parameters, is used to describe between cluster relations of different dependence structures. The model can be applied to settings, such as the time varying dependence modeling and clustered dependence modeling. The material in this chapter has been wrapped up as a paper submitted for publication and will appear in the Electronic Journal of Statistics.

Chapter 4: Grouping Dependence Structure and Selection of Copula-Based Models Using Bayesian Nonparametric Methods

When analyzing dependent data, an insufficient sample size can lead to inaccurate model selection of dependence structures and estimation of the dependence parameters. Some bivariate pairs in multivariate data may share the same dependence structure, or dependent data that arises from multiple sources may share the same dependence structure, thus grouping the data according to the similarity in their dependence structure is a natural way to increase the sample size and carry out valid and efficient inferences.

In this chapter, we still consider data arising from multiple sources as we assume in Chapter 3, and we are interested in modeling subject-level dependence using copula-based models. Instead of focusing on parameter estimation with given copula forms as we do in Chapter 3, we mainly focus on selection of copula forms. We propose a copula-based model with copula selection indicators and dependence parameters following a DP prior, and we call this model the mixture of DPM copula model (M-DPM-CM). The M-DPM-CM is able to group the clusters with similar dependence structures together. The grouping of clusters sharing similar dependence relations can benefit the copula selection and parameter estimation by facilitating a larger sample size. The material in this chapter has been wrapped up as a paper submitted for publication.

Chapter 5: Polya Tree Monte Carlo Method

In this chapter, we investigate the problem of sampling from a distribution, which has been an important research topic in statistics and enjoys broad applications in different contexts, including the Bayesian framework and the machine learning paradigm(e.g., Goodfellow et al., 2014). Markov Chain Monte Carlo (MCMC) is the dominating algorithm for sampling from distributions, but it suffers from emerging difficulties, such as correlated samples and inefficiency in exploring multi-modal distributions. Motivated by this, we propose the Polya tree Monte Carlo (PTMC) method, which is based on the approximated Polya tree posterior using the Monte Carlo method.

In our proposed PTMC method, we first approximate the posterior Polya tree by the Monte Carlo method and prove theoretically that this approximated Polya tree posterior converges to the target distribution, provided regularity conditions. Based on this result, we further propose a series of simple, efficient and computational friendly sampling algorithms for sampling from the approximated posterior Polya tree. The proposed algorithms provide independent samples from the target distribution and exhibit superior performance in discovering modes for multi-modal distributions as we illustrate in the numerical studies. The material in this chapter has been wrapped up as a paper submitted for publication.

Chapter 6: Polya Tree Based Nearest Neighbor Regression

Regression analysis is a powerful statistical method for delineating the relationship between responses and covariates of interest and has become one of the most thriving topics in statistics. In this chapter, we propose a fully nonparametric regression model to provide robust description of highly irregular regression relations.

As opposed to parametric regression or semi-parametric regression, fully nonparametric regression characterizes the conditional probability measures of the responses given covariates. A fully nonparametric regression is often formulated as

$$Y \mid x \sim G_x,$$

where G_x is a conditional probability measure of the response variable Y, given the covariate value $X = x$. The key component in a fully nonparametric regression model is to build the connection between G_x and x.

In this chapter, we introduce a new fully nonparametric regression model, called the Polya tree based nearest neighbor (PTNN) regression, which constructs a PT-distributed probability measure of the responses in a "nearest" neighborhood of the covariates of interest. Here "a nearest neighbor" is loosely used in the same way as the nearest neighbor method (Cover and Hart, 1967; Beyer et al., 1999), though strictly speaking, there is no "nearest" neighborhood of a center in a continuous metric (unless the center itself is taken as its nearest neighborhood). The constructed probability measure well approximates the true probability measure of the response given covariates, and the resulting nonparametric estimates are easy to obtain based on a sample from the constructed PT distribution. The model enjoys several merits including simple formulation, consistent estimates of the conditional distribution G_x and computational efficiency. The proposed method does not require any parametric model assumption and thus possesses the robustness property. The material in this chapter has been wrapped up as a paper submitted for publication.

Chapter 2

Composite Likelihood Methods for Analyzing Longitudinal Data with a Time-Span under Vine Copula Models

2.1 Introduction

Longitudinal data analysis, which studies the change of repeated observations of the same subjects over time, has long been a thriving topic in statistical research. Longitudinal data with a long time span, such as temperature data or precipitation data, imposes challenges to conventional methods for longitudinal data analysis in modeling the temporal dependence. In this chapter, the objective of our research is to develop a new statistical model to better characterize and forecast such kinds of data.

With the increasing focus on dependence modeling with copula models, applications of copulas and vine copulas to longitudinal data are relatively limited. Lambert and Vandenhende (2002) introduced copula to model the multivariate non-normal longitudinal data. Smith et al. (2010) considered using D-Vine copula to model the serial dependence in time series, but they focused more on the estimation of the vine copulas and did not include covariates into the model. Killiches and Czado (2018) considered modeling the repeated measurements with a homogeneous vine copula model under a unbalanced design. Each bivariate copula in the vine structure is assumed to be Gaussian copula, so that the model can be used to make prediction easily. Other studies include Frees and Wang (2006), Domma et al. (2009), Madsen and Fang (2011), and Ruscone and Osmetti (2016). Most of the studies focused more on estimation instead of prediction, and the time ordering of the longitudinal data did not receive much attention.

Applying vine copula models to longitudinal data with a long time span can be prohibitive, since the number of parameters in the model will increase considerably as time length increases. In Chapter 2, we propose a special R-Vine structure to describe the temporal dependence of longitudinal data that exhibits periodic patterns. The R-Vine structure can be easily combined with composite likelihood ideas to simplify the estimation procedure and reduce the computation burden. It also provides a convenient structure for prediction of future observations. Furthermore, we consider using composite likelihood to simplify inference procedures and concentrate on the parameters of primary interest. We also compare different estimation procedures, simultaneous estimation and two-stage estimation, to further facilitate the fast inference of our proposed model.

The rest of the chapter is organized as follows. In Section 2.2, we discuss the model formulation, including marginal and association models. In Sections 2.3, we describe how to estimate the parameters, and in Section 2.4, we give the procedure for copula selection and prediction. In Sections 2.5 and 2.6, simulation studies and analysis of Ontario temperature data are provided, respectively.

2.2 Model Formulation

Suppose that we are interested in a particular type of longitudinal data which exhibits a periodic pattern, such as longitudinal data of temperature and precipitation. To feature periodic patterns, we examine the data by periods, called time blocks in what follows, and let b denote the number of time points in each time blocks. Suppose that we have a time blocks, let $m = ab$ denote the total number of observed occasions, and n subjects are observed at the m occasions. For longitudinal data with no periodic pattern, we set $a = 1$. Let Y_{ikl} be the continuous response for the ith subject at the lth time point in the kth time block, and let x_{ikl} be the associated covariate matrices. Let $Y_{ik} = (Y_{ik1}, \ldots, Y_{ikb})^{\mathrm{T}}$ be the vector of responses of the ith subject in the kth time block, and let $Y_i = (Y_{i1}^{\mathrm{T}}, \ldots, Y_{ia}^{\mathrm{T}})^{\mathrm{T}}$ be the full vector of responses of subject i for $i = 1, \ldots, n$ and $k = 1, \ldots, a$. Let lower case letters y_{ik} and y_i denote the realizations of of Y_{ik} and Y_i, respectively, and let x_{ik} and x_i denote the corresponding covariates.

We now introduce the joint model for Y_i which shows the dependence of Y_i on x_i. It is difficult to directly specify a meaningful joint distribution of Y_i, given x_i, to account for the dependence structure of the components of Y_i. To come up with an interpretable joint model for Y_i given x_i, we take two steps. In the first step, we characterize the dependence of Y_i on x_i via regression models, which contain random errors; in the second step, we further delineate the dependence structures of the components of Y_i by characterizing the dependence structures of the random errors resulted from the first step.

Specifically, for $i = 1, \ldots, n$, $k = 1, \ldots, a$, and $l = 1, \ldots, b$, we assume that

$$Y_{ikl} = \mu_{ikl} + \varepsilon_{ikl}, \tag{2.1}$$

where $\mu_{ikl} = E(Y_{ikl}|x_{ikl})$, and ε_{ikl} is the associated random error term. We further assume that

$$g_l(\mu_{ikl}) = x_{ikl}^{\mathrm{T}}\beta_l,$$

where $g_l(\cdot)$ is the link function and β_l is the parameter vector associated with time l. Let $\beta = (\beta_1^{\mathrm{T}}, \ldots, \beta_b^{\mathrm{T}})^{\mathrm{T}}$. For $i = 1, \ldots, n$ and $k = 1, \ldots, a$, we let $\varepsilon_{ik} = (\varepsilon_{ik1}, \ldots, \varepsilon_{ikb})^{\mathrm{T}}$ and $\varepsilon_i = (\varepsilon_{i1}^{\mathrm{T}}, \ldots, \varepsilon_{ia}^{\mathrm{T}})^{\mathrm{T}}$.

To reflect that responses from the same subject across time points are possibly associated, in the next step, we focus on characterizing the dependence structure among the components of ε_i using vine copula models.

2.2.1 Joint Distribution of ε_i

Marginal Distribution of ε_i

For $l = 1, \ldots, b$, we assume that marginally, the random errors $\{\varepsilon_{ikl} : i = 1, \ldots, n;\ k = 1, \ldots, a\}$ share the same distribution function and let $F_l(\cdot; \omega_l)$ and $f_l(\cdot; \omega_l)$, respectively, denote their cumulative distribution function (CDF) and the density function indexed by parameter vector ω_l, i.e.,

$$\varepsilon_{ikl} \sim F_l(\varepsilon_{ikl}; \omega_l), \tag{2.2}$$

for $i = 1, \ldots, n$; $k = 1, \ldots, a$. Let $\omega = (\omega_1^{\mathrm{T}}, \ldots, \omega_b^{\mathrm{T}})^{\mathrm{T}}$ and let $\eta = (\beta^{\mathrm{T}}, \omega^{\mathrm{T}})^{\mathrm{T}}$ denote the parameter vector associated with the marginal distribution of the Y_{ikl}.

Dependence Structure of ε_i

We employ vine copula models (Bedford and Cooke, 2002) to delineate the dependence structures of the random vector ε_i. In particular, D-Vine and Canonical vine (C-Vine) are two useful cases of regular vine copula models, which pertain to pair-copula constructions (Aas et al., 2009).

As longitudinal data has a natural temporal order, Smith et al. (2010) and Killiches and Czado (2018) both considered modeling longitudinal data using a D-Vine structure under different settings. However, in the second or higher levels of D-Vine trees, describing the stochastic behavior of the current responses needs to be conditional on future responses, which creates difficulties in interpreting the copula parameters. As a result, we adopt a C-Vine to model the dependence structure between different time points within a block to avoid this problem and yield an interpretable model.

Specifically, we propose to use an R-Vine structure to model the dependence structures within ε_i. Within each time block, the dependence structure between time points is assumed to be identical and modeled with a C-Vine structure; and different time blocks are connected by a D-Vine structure. To illustrate this idea, in Figure 2.1 we present an example with 4 time blocks and 4 time points within each block, where T_1, T_2 and T_3 represent the first three levels of trees in the vine copula model, and the nodes in the (blue) boxes represent the error terms of time points within time blocks, which have a C-Vine model structure.

We first introduce necessary notation before we give the mathematical form of the R-Vine structure. For $c = 1, \ldots, a$ and $d = 2, \ldots, b + 1$, let $\mathcal{G}_{icd} = \{\varepsilon_{icl} : l = 1, \ldots, d - 1\}$. For $s, g \in \{1, \ldots, a\}$ and $h, r \in \{1, \ldots, b\}$, let

$$\mathcal{D}_{ish,igr} = \begin{cases} \left\{ \bigcup_{c=s+1}^{g-1} \mathcal{G}_{ic(b+1)} \right\} \cup \mathcal{G}_{ish} \cup \mathcal{G}_{igr}, & \text{if } s < g - 1; \\ \mathcal{G}_{ish} \cup \mathcal{G}_{igr}, & \text{if } s = g - 1; \\ \mathcal{G}_{ish}, & \text{if } s = g \text{ and } h < r. \end{cases}$$

Furthermore, for a random variable Z_1, a random vector $Z_2 = (Z_{21}, \ldots, Z_{2d_1})^{\mathrm{T}}$ and a random vector $Z_3 = (Z_{31}, \ldots, Z_{3d_2})^{\mathrm{T}}$ with $1 + d_1 + d_2 = m$, let $F_{\mathrm{Z}_1\mathrm{Z}_2\mathrm{Z}_3}(z_1, z_2, z_3)$ denote the joint CDF of Z_1,Z_2 and Z_3, with $f_{\mathrm{Z}_1\mathrm{Z}_2\mathrm{Z}_3}$ as the corresponding density function. As a result, the joint density of the random vector Z_2 is derived as $f_{\mathrm{Z}_2}(z_2) = \int\int f_{\mathrm{Z}_1\mathrm{Z}_2\mathrm{Z}_3}(z_1, z_2, z_3)dz_1dz_3$, and the conditional CDF of Z_1, given Z_2 is

$$F_{\mathrm{Z}_1|\mathrm{Z}_2}(z_1|z_2) = \frac{\partial^{d_1} F_{\mathrm{Z}_1\mathrm{Z}_2}(z_1, z_2)}{\partial z_{21} \ldots \partial z_{2d_1}} \times \frac{1}{f_{\mathrm{Z}_2}(z_2)}, \tag{2.3}$$

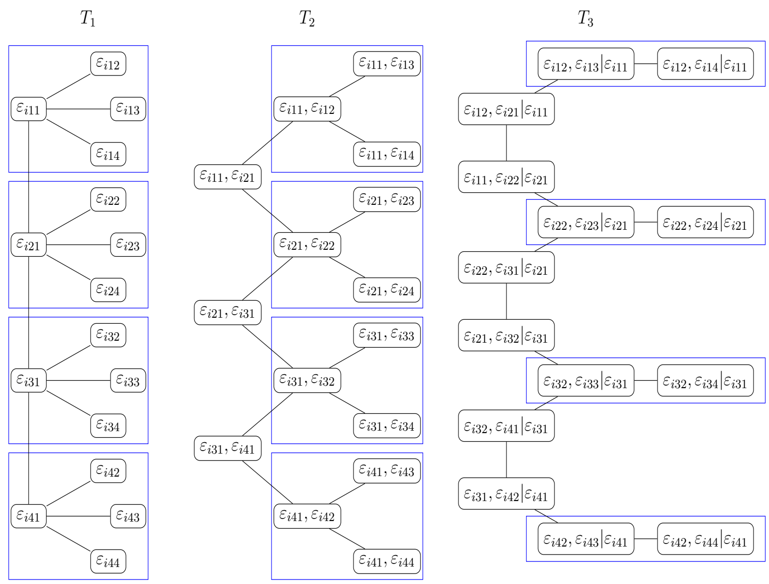


Figure 2.1: A R-Vine structure for 4 time blocks and 4 time points within each block

where $F_{Z_1Z_2}(z_1, z_2) = \lim\limits_{z_3 \to \infty} F_{Z_1Z_2Z_3}(z_1, z_2, z_3)$ is the joint CDF of Z_1 and Z_2.

For ε_{ikh} and ε_{ikr} with $h < r$ in the same time block k, let $c_{kh,kr}(\cdot, \cdot)$ denote the conditional copula density function between ε_{ikh} and ε_{ikr}, given the conditioning set $\mathcal{D}_{ikh,ikr}$, where the first and second arguments in the copula density are given by $u_{ikh|\mathcal{D}_{ikh,ikr}} = F_{\varepsilon_{ikh}|\mathcal{D}_{ikh,ikr}}(\varepsilon_{ikh}|\mathcal{D}_{ikh,ikr})$ and $u_{ikr|\mathcal{D}_{ikh,ikr}} = F_{\varepsilon_{ikr}|\mathcal{D}_{ikh,ikr}}(\varepsilon_{ikr}|\mathcal{D}_{ikh,ikr})$ respectively, and $F_{\varepsilon_{ikh}|\mathcal{D}_{ikh,ikr}}$ and $F_{\varepsilon_{ikr}|\mathcal{D}_{ikh,ikr}}$ are the conditional CDFs of ε_{ikh} and ε_{ikr}, given the conditioning set $\mathcal{D}_{ikh,ikr}$ respectively, which are obtained from (2.3) by letting $Z_1 = \varepsilon_{ikh}$ or ε_{ikr}, $Z_2 = \mathcal{D}_{ikh,ikr}$ and $Z_3 = \varepsilon_i \backslash \{\varepsilon_{ikh} \cup \mathcal{D}_{ikh,ikr}\}$ or $\varepsilon_i \backslash \{\varepsilon_{ikr} \cup \mathcal{D}_{ikh,ikr}\}$.

For ε_{ish} and ε_{igr} in different time block with $s < g$, let $c_{sh,gr}(\cdot, \cdot)$ denotes the conditional copula density function between ε_{ish} and ε_{igr}, given the conditioning set $\mathcal{D}_{ish,igr}$, where the first and second arguments in the copula density are given by $u_{ish|\mathcal{D}_{ish,igr}} = F_{\varepsilon_{ish}|\mathcal{D}_{ish,igr}}(\varepsilon_{ish}|\mathcal{D}_{ish,igr})$ and $u_{igr|\mathcal{D}_{ish,igr}} = F_{\varepsilon_{igr}|\mathcal{D}_{ish,igr}}(\varepsilon_{igr}|\mathcal{D}_{ish,igr})$ respectively, and $F_{\varepsilon_{ish}|\mathcal{D}_{ish,igr}}$ and $F_{\varepsilon_{igr}|\mathcal{D}_{ish,igr}}$ are the conditional CDFs of ε_{ish} and ε_{igr}, given the conditioning set $\mathcal{D}_{ish,igr}$ respectively, which are obtained from (2.3) by letting $Z_1 = \varepsilon_{ish}$ or ε_{igr}, $Z_2 = \mathcal{D}_{ish,igr}$ and $Z_3 = \varepsilon_i \backslash \{\varepsilon_{ish} \cup \mathcal{D}_{ish,igr}\}$ or $\varepsilon_i \backslash \{\varepsilon_{igr} \cup \mathcal{D}_{ish,igr}\}$.

Combining the marginal model and the dependence structures specified, we write the joint density function of ε_i as

$$f(\varepsilon_i;\omega,\theta,\psi) = \left\{\prod_{k=1}^{a}\prod_{l=1}^{b} f_l(\varepsilon_{ikl};\omega_l)\right\}$$
$$\times\left\{\prod_{k=1}^{a}\prod_{h=1}^{b-1}\prod_{r=h+1}^{b} c_{kh,kr}\big(u_{ikh|\mathcal{D}_{ikh,ikr}}, u_{ikr|\mathcal{D}_{ikh,ikr}};\theta_{kh,kr}\big)\right\}$$
$$\times\left\{\prod_{s=1}^{a-1}\prod_{g=s+1}^{a}\prod_{h=1}^{b}\prod_{r=1}^{b} c_{sh,gr}\big(u_{ish|\mathcal{D}_{ish,igr}}, u_{igr|\mathcal{D}_{ish,igr}};\psi_{sh,gr}\big)\right\}, \tag{2.4}$$

where the product in the first set of brackets corresponds to the marginal densities of the ε_{ikl}, the product in the second set of brackets corresponds to the C-Vine structure within time blocks indexed by the dependence parameter vector $\theta = \{\theta_{kh,kr} : k = 1,\ldots,a; h = 1,\ldots,b-1; r = (h+1),\ldots,b\}$, and the product in the third set of brackets corresponds to the D-Vine structure connecting the time blocks indexed by the dependence parameter vector $\psi = \{\psi_{sh,gr} : s = 1,\ldots,a-1; g = s+1,\ldots,a; h,r = 1,\ldots,b\}$. Let $\vartheta = (\theta^{\mathrm{T}},\psi^{\mathrm{T}})^{\mathrm{T}}$ denote the vector of dependence parameters.

We comment that although in formulating the sequence of bivariate conditional copula density functions for (2.4), we employ an m-dimensional joint CDF via (2.3), the determination of the joint density function ε_i is done through the density decomposition in combination with the componentwise specification via (2.4), which is different from directly specifying an m-dimensional joint distribution of ε_i.

2.2.2 Joint Model of the Responses Y_i

Applying the one-to-one transformation to the random variables defined by (2.1) in combination with the joint density function (2.4) for ε_i, we obtain the joint distribution of responses Y_i, given by

$$f(y_i;\eta,\vartheta) = \left\{\prod_{k=1}^{a}\prod_{l=1}^{b} f_l(y_{ikl} - g^{-1}(x_{ikl}^{\mathrm{T}}\beta_l);\omega_l)\right\}$$
$$\times\left\{\prod_{k=1}^{a}\prod_{h=1}^{b-1}\prod_{r=h+1}^{b} c_{kh,kr}\big(u_{ikh|\mathcal{D}_{ikh,ikr}}, u_{ikr|\mathcal{D}_{ikh,ikr}};\theta_{kh,kr}\big)\right\}$$
$$\times\left\{\prod_{s=1}^{a-1}\prod_{g=s+1}^{a}\prod_{h=1}^{b}\prod_{r=1}^{b} c_{sh,gr}\big(u_{ish|\mathcal{D}_{ish,igr}}, u_{igr|\mathcal{D}_{ish,igr}};\psi_{sh,gr}\big)\right\} \tag{2.5}$$

where $u_{ish|\mathcal{D}_{ish,igr}} = F_{\varepsilon_{ish}|\mathcal{D}_{ish,igr}}(\varepsilon_{ish}|\mathcal{D}_{ish,igr})$ in (2.4) is now expressed as $u_{ish|\mathcal{D}_{sh,gr}} = F_{\varepsilon_{ish}|\mathcal{D}_{ish,igr}}\left(y_{ish} - g^{-1}(x_{ish}^{\mathrm{T}}\beta_h)|\mathcal{D}_{ish,igr}\right)$ by using (2.1). Here $g^{-1}(\cdot)$ represents the inverse function of $g(\cdot)$.

2.3 Estimation Methods

Given the availability of the joint distribution of Y_i, it is natural to use the likelihood method to estimate the marginal parameters η and dependence parameters ϑ simultaneously. Let

$$L_i(\eta, \vartheta) = f(y_{i11}, \ldots, y_{iab}; \eta, \vartheta)$$

be the likelihood contributed from subject i. Then the full likelihood is

$$L(\eta, \vartheta) = \prod_{i=1}^{n} L_i(\eta, \vartheta). \tag{2.6}$$

Maximizing the likelihood function (2.6) with respect to η and ϑ gives the maximum likelihood estimator of $(\eta^{\mathrm{T}}, \vartheta^{\mathrm{T}})^{\mathrm{T}}$, denoted by $(\hat{\eta}^{\mathrm{T}}, \hat{\vartheta}^{\mathrm{T}})^{\mathrm{T}}$.

The likelihood method is conceptually easy to implement, and it yields consistent and efficient estimators if the associated models are correctly specified. However, this method has two major limitations. Computationally, when the dimension of Y_i increases, the number of parameters in the likelihood function will increase dramatically, and thus, using the likelihood for estimation can be computationally prohibitive. Theoretically, the validity of the maximum likelihood estimator hinges on the correctness of all the assumed models. Any model misspecification may result in biased results.

To overcome the weakness of the likelihood method, we explore the alternative estimation methods using the composite likelihood framework (Lindsay, 1988; Varin, 2008; Varin et al., 2011; Lindsay et al., 2011; Yi, 2017b), of which the details are elaborated in following sections.

2.3.1 Simultaneous Estimation with Composite Likelihood

Rather than working with the joint distribution of Y_i in (2.5), we ignore the dependence structure between time blocks. This ignorance is driven by the fact that the parameters ψ, which consists mostly of the parameters in high levels of R-Vine tree, are not of primary interest (Brechmann et al., 2012).

Let $\phi = (\eta^{\mathrm{T}}, \theta^{\mathrm{T}})^{\mathrm{T}}$, we consider the joint distribution of Y_{ik} for subject i within the kth time block

$$\begin{aligned} f(y_{ik1}, \ldots, y_{ikb}; \phi) = & \prod_{l=1}^{b} f_l(y_{ikl} - g_l^{-1}(x_{ikl}^{\mathrm{T}}\beta_l); \omega_l) \\ & \times \left\{ \prod_{h=1}^{b-1} \prod_{r=h+1}^{b} c_{kh,kr}(u_{ikh|\mathcal{D}_{kh,kr}}, u_{ikr|\mathcal{D}_{ikh,ikr}}; \theta_{kh,kr}) \right\} \end{aligned} \tag{2.7}$$

for $i = 1, \ldots, n$ and $k = 1, \ldots, a$. This distribution form is simpler than (2.5).

Next, we formulate a composite likelihood for the parameters ϕ using (2.7) and ignoring the dependence among different time blocks:

$$\begin{aligned} L_{ci}(\phi) &= \prod_{k=1}^{a} f(y_{ik1}, \ldots, y_{ikb}; \phi); \\ L_c(\phi) &= \prod_{i=1}^{n} L_{ci}(\phi). \end{aligned} \tag{2.8}$$

Maximizing (2.8) with respect to ϕ yields a composite maximum likelihood estimator of ϕ, denoted by $\hat{\phi}_{\mathrm{CS}}$.

The asymptotic results of composite likelihood have been discussed by Varin (2008), Varin et al. (2011), and Yi (2017a) among others. Under regularity conditions, the estimator $\hat{\phi}_{\mathrm{CS}}$ has the following asymptotic properties:

(1) $\hat{\phi}_{\mathrm{CS}} \xrightarrow{p} \phi$ as $n \to \infty$;

(2) $\sqrt{n}(\hat{\phi}_{\mathrm{CS}} - \phi) \xrightarrow{d} \mathrm{MVN}(0, G^{-1}(\phi))$ as $n \to \infty$,

where $G(\phi) = H(\phi)J^{-1}(\phi)H(\phi)$ is the Godambe information matrix, $H(\phi)$ is the sensitivity matrix, and $J(\phi)$ is the variability matrix, defined, respectively, as

$$\begin{aligned} H(\phi) &= E\left(\frac{\partial^2 L_{ci}(\phi)}{\partial\phi\partial\phi^{\mathrm{T}}}\right) \\ \text{and} \quad J(\phi) &= E\left\{\left(\frac{\partial L_{ci}(\phi)}{\partial\phi}\right)\left(\frac{\partial L_{ci}(\phi)}{\partial\phi}\right)^{\mathrm{T}}\right\}. \end{aligned}$$

Inference about ϕ can be carried out by using the asymptotic distribution of $\hat{\phi}$. When doing so, it is necessary to estimate $G(\phi)$ consistently, which is available from consistent estimators of $H(\phi)$ and $J(\phi)$, given by

$$\hat{H}(\hat{\phi}_{\text{CS}}) = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial^2 L_{ci}(\phi)}{\partial\phi\partial\phi^{\text{T}}}\Big|_{\phi=\hat{\phi}_{\text{CS}}},$$
$$\text{and} \quad \hat{J}(\hat{\phi}_{\text{CS}}) = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\partial L_{ci}(\phi)}{\partial\phi}\right)\left(\frac{\partial L_{ci}(\phi)}{\partial\phi}\right)^{\text{T}}\Big|_{\phi=\hat{\phi}_{\text{CS}}},$$

respectively.

2.3.2 Two-Stage Estimation with Composite Likelihood

To further ease computation burdens, we treat η and θ differently when employing (2.8) for estimation. Specifically, we estimate η using a simpler formulation than (2.8) and then use (2.8) to estimate θ only. We now describe a two-stage estimation procedure. In the first stage, for $l = 1, \ldots, b$ we construct the marginal likelihood functions for marginal parameters $\eta_l = (\beta_l^{\text{T}}, \omega_l^{\text{T}})^{\text{T}}$,

$$L_{il}(\eta_l) = \prod_{k=1}^{a} f_l\left(y_{ikl} - g_l^{-1}(x_{ikl}^{\text{T}}\beta_l); \omega_l\right),$$

and

$$L_l(\eta_l) = \prod_{i=1}^{n} L_{il}(\eta_l). \tag{2.9}$$

Maximizing (2.9) with respect to η_l yields an estimator of η_l, denoted by $\hat{\eta}_l$, for $l = 1, \ldots, b$. Let $\hat{\eta}_{\text{CT}} = (\hat{\eta}_1^{\text{T}}, \ldots, \hat{\eta}_b^{\text{T}})^{\text{T}}$.

In the second stage, we plug $\hat{\eta}_{\text{CT}}$ into (2.8) and obtain $L_c(\hat{\eta}_{\text{CT}}, \theta)$. Then maximizing $L_c(\hat{\eta}_{\text{CT}}, \theta)$ with respect to θ provides an estimator of θ, denoted by $\hat{\theta}_{\text{CT}}$. Let $\hat{\phi}_{\text{CT}} = (\hat{\eta}_{\text{CT}}^{\text{T}}, \hat{\theta}_{\text{CT}}^{\text{T}})^{\text{T}}$.

Let $Q_i(\eta) = \frac{\partial}{\partial\eta}\sum_{l=1}^{b}\log[L_{il}(\eta_l)]$ and $U_i(\eta,\theta) = \frac{\partial}{\partial\theta}\log[L_{ci}(\eta,\theta)]$. Define

$$H(\phi) = E\begin{pmatrix} \frac{\partial}{\partial\eta^{\text{T}}}Q_i(\eta) & 0 \\ \frac{\partial}{\partial\eta^{\text{T}}}U_i(\eta,\theta) & \frac{\partial}{\partial\theta^{\text{T}}}U_i(\eta,\theta) \end{pmatrix} \quad \text{and}$$

$$J(\phi) = E\{W_i(\eta,\theta)W_i(\eta,\theta)^{\mathrm{T}}\},$$

where $W_i(\eta,\theta) = (Q_i(\eta)^{\mathrm{T}}, U_i(\theta,\eta)^{\mathrm{T}})^{\mathrm{T}}$. Similarly, based on the results of Varin (2008), Varin et al. (2011) and Yi (2017a), under regularity conditions, the estimator $\hat{\phi}_{\mathrm{CT}}$ has the asymptotic results

(1) $\hat{\phi}_{\mathrm{CT}} \xrightarrow{p} \phi$ as $n \to \infty$;

(2) $\sqrt{n}(\hat{\phi}_{\mathrm{CT}} - \phi) \xrightarrow{d} \mathrm{MVN}(0, G^{-1}(\phi))$ as $n \to \infty$,

where $G(\phi) = H(\phi)J^{-1}(\phi)H(\phi)$ is the Godambe information matrix. $H(\phi)$ and $J(\phi)$ can be consistently estimated by

$$\hat{H}(\hat{\phi}_{\mathrm{CT}}) = \frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} \frac{\partial}{\partial \eta^{\mathrm{T}}}Q_i(\eta) & 0 \\ \frac{\partial}{\partial \eta^{\mathrm{T}}}U_i(\eta,\theta) & \frac{\partial}{\partial \theta^{\mathrm{T}}}U_i(\eta,\theta) \end{pmatrix}\Bigg|_{\phi=\hat{\phi}_{\mathrm{CT}}} \quad \text{and}$$

$$\hat{J}(\hat{\phi}_{\mathrm{CT}}) = \frac{1}{n}\sum_{i=1}^{n}\{W_i(\eta,\theta)W_i(\eta,\theta)^{\mathrm{T}}\}\Big|_{\phi=\hat{\phi}_{\mathrm{CT}}},$$

respectively.

2.4 Copula Selection and Prediction

Dissmann et al. (2013) proposed a sequential procedure which selects copula forms for each of the (conditional) bivariate copulas level by level, where the selection is carried out with a prespecified vine structure from a set of candidate copula functions. The sequential procedure facilitates a fast model selection process by considering each (conditional) pair separately. In the same spirit of the composite likelihood formulation (2.8), we assume the same dependence structures within time blocks and ignore the dependence between blocks. Pretending to have $n \times a$ independent time blocks, we apply sequential selection procedure of Dissmann et al. (2013) to select copula functions in the C-Vine structure within blocks.

We are interested in predicting the observations for a subject in the study for a future time point (i.e., time extrapolation) or for some new subjects at a given time point (i.e., subject extrapolation). Please see supplementary materials for our discussion on subject extrapolation through simulation studies and data analysis. We focus on the time extrapolation in this subsection.

Suppose that for subject i, at time block k, the observations for all time points $j \leq h$ have been observed, and we would like to predict the observation at time $(h+1)$, where h is a given time point. First, the estimate of the mean for the marginal model is calculated as

$$\hat{\mu}_{ikl} = g_l^{-1}(x_{ikl}^{\mathrm{T}}\hat{\beta}_l)$$

for $l = 1, \ldots, (h+1)$. Then, the error terms of the h observed time points can be calculated and transformed as "pseudo-observations", i.e., for $l = 1, \ldots, h$,

$$\hat{\varepsilon}_{ikl} = y_{ikl} - \hat{\mu}_{ikl} \quad \text{and} \quad \hat{u}_{ikl} = F_l(\hat{\varepsilon}_{ikl}; \hat{\omega}_l).$$

Next, the conditional distribution of the error term at time $(h+1)$ can be approximated as

$$f(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}) = \frac{f(\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}, \varepsilon_{ik(h+1)})}{f(\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh})},$$

which by (2.4), is equal to

$$\frac{f(\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}) f_{h+1}(\varepsilon_{ik(h+1)}) \prod_{r=1}^{h} c_{kr,k(h+1)}(\hat{u}_{ikr|\mathcal{D}_{ikr,ik(h+1)}}, u_{ik(h+1)|\mathcal{D}_{ikr,ik(h+1)}})}{f(\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh})}$$
$$= f_{h+1}(\varepsilon_{ik(h+1)}; \hat{\omega}_{h+1}) \times \prod_{r=1}^{h} c_{kr,k(h+1)}(\hat{u}_{ikr|\mathcal{D}_{ikr,ik(h+1)}}, u_{ik(h+1)|\mathcal{D}_{ikr,ik(h+1)}}), \tag{2.10}$$

where the conditional terms $\hat{u}_{ikr|\mathcal{D}_{ikr,ik(h+1)}}$ and $u_{ik(h+1)|\mathcal{D}_{ikr,ik(h+1)}}$ are calculated by applying the formulas $u_{p|q} = \frac{\partial c_{pq}(u_p,u_q)}{\partial u_q}$ and $u_{q|p} = \frac{\partial c_{pq}(u_p,u_q)}{\partial u_p}$ iteratively, in which p and q can be any unconditional label, such as ikr, or conditional label, such as $ikr|\mathcal{D}_{ikr,ik(h+1)}$. As a result, the predicted outcome $\hat{y}_{ik(h+1)}$ for subject i at time point $(h+1)$ in time block k is given by

$$\begin{aligned}\hat{y}_{ik(h+1)} &= E(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}) + \hat{\mu}_{ik(h+1)} \\ &= \int_{-\infty}^{\infty} \varepsilon_{ik(h+1)} f(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}) d\varepsilon_{ik(h+1)} + \hat{\mu}_{ik(h+1)}\end{aligned}$$

with $f(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh})$ determined by (2.10). The prediction variance of $\hat{y}_{ik(h+1)}$ is calculated as

$$\mathrm{Var}(\hat{y}_{ik(h+1)}) = \mathrm{Var}(\varepsilon_{ik(h+1)})/(k-1) + \mathrm{Var}(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh}),$$

where the first component is related to the marginal model at time $h+1$, and the second component can be calculated from the conditional density $f(\varepsilon_{ik(h+1)}|\hat{\varepsilon}_{ik1}, \ldots, \hat{\varepsilon}_{ikh})$.

2.5 Simulation Studies

In this section, we conduct simulation studies to examine the finite sample performance of the proposed composite likelihood under simultaneous and two-stage estimation procedures in terms of efficiency, robustness, mis-selection rate and prediction accuracy, which will be elaborated in Sections 2.5.1, 2.5.2, 2.5.3 and 2.5.4, respectively.

2.5.1 Validity and Efficiency

In this subsection, we explore the validity and efficiency loss of the proposed composite likelihood method relative to the likelihood-based methods. We first introduce various simulation settings, describe evaluation metrics, and finally report the simulation results.

Simulation Settings

We consider scenarios where the sample size $n = 500$ or 1000, the number of time blocks is $a = 4$ and the number of time points in each time block is $b = 4$. The covariates x_{ikl} are independently generated from a uniformly distribution on $[0, 5]$ for $i = 1, \ldots, n$, $k = 1, \ldots, a$ and $l = 1, \ldots, b$. Suppose that the marginal model is

$$Y_{ikl} = \beta_{0l} + \beta_{1l} x_{ikl} + \beta_{2l} k + \varepsilon_{ikl}, \tag{2.11}$$

where $\varepsilon_{ikl} \sim N(0, \sigma_l^2)$, for $i = 1, \ldots, n$, $k = 1, 2, 3, 4$ and $l = 1, 2, 3, 4$. We set the values of the marginal parameters as $\eta_l = (\beta_{0l}, \beta_{1l}, \beta_{2l}, \sigma_l)^{\mathrm{T}} = (l, l+1, l+2, 2)^{\mathrm{T}}$ for $l = 1, 2, 3, 4$.

In this subsection, we assume the error terms bear the R-Vine structure as demonstrated in Figure 2.1 and we further assume the conditional independence in tree structure T_4 and beyond for simplicity. We consider two scenarios where the dependence is either strong or moderate. For the scenario of strong or moderate dependence, the (conditional) bivariate copulas connecting the time blocks in T_1, T_2 and T_3 are all Gaussian(0.8) or Gaussian(0.5). More specifically, the bivariate copula functions and their corresponding parameter values for the C-Vine structure within each time block are given in Table 2.1. In the scenario of strong dependence, the Kendall's Taus of the bivariate copulas in T_1, T_2 and T_3 are set to be 0.7, 0.6 and 0.5, respectively; in that of moderate dependence, they are set to be 0.4, 0.3 and 0.2, respectively. The values of the dependence parameters are set to reach the desired degree of dependence. We generate the error terms ε_i from joint density (2.4), in which the marginal distribution is normal and the dependence structure is the previously specified R-Vine; the values of Y_{ikl} are determined by (2.11).

Table 2.1: Copula functions and the values of the dependence parameters in the dependence structure within each time block

Bivariate Variable	$\varepsilon_{ik1}, \varepsilon_{ik2}$	$\varepsilon_{ik1}, \varepsilon_{ik3}$	$\varepsilon_{ik1}, \varepsilon_{ik4}$	$\varepsilon_{ik2}, \varepsilon_{ik3} \vert \varepsilon_{ik1}$	$\varepsilon_{ik2}, \varepsilon_{ik4} \vert \varepsilon_{ik1}$	$\varepsilon_{ik3}, \varepsilon_{ik4} \vert \varepsilon_{ik1}, \varepsilon_{ik2}$
Copula Function	Clayton	Gumbel	Gaussian	Frank	Gaussian	Frank
Strong Dependence						
Kendall's Tau	0.7	0.7	0.7	0.6	0.6	0.5
Dependence Parameter	$\theta_{k1,k2} = 4.67$	$\theta_{k1,k3} = 3.33$	$\theta_{k1,k4} = 0.89$	$\theta_{k2,k3} = 7.93$	$\theta_{k2,k4} = 0.81$	$\theta_{k3,k4} = 5.74$
Moderate Dependence						
Kendall's Tau	0.4	0.4	0.4	0.3	0.3	0.2
Dependence Parameter	$\theta_{k1,k2} = 1.33$	$\theta_{k1,k3} = 1.67$	$\theta_{k1,k4} = 0.59$	$\theta_{k2,k3} = 2.92$	$\theta_{k2,k4} = 0.45$	$\theta_{k3,k4} = 1.86$

The simulation is repeated 500 times. We compare the performance of the following four estimation methods:

(1) Method 1: full likelihood using simultaneous estimation procedure,

(2) Method 2: full likelihood using two-stage estimation procedure,

(3) Method 3: composite likelihood using simultaneous estimation procedure described in Section 3.1,

(4) Method 4: composite likelihood using two-stage estimation procedure described in Section 3.2.

Note that the first stage of Method 2 and 4 are both using the marginal likelihood (2.9) and essentially provide the same estimates for marginal parameters.

Evaluation Metrics

The following five evaluation metrics are used to evaluate different aspects of the estimators obtained by using the four estimation methods.

- *Empirical Bias (EBias)*: The difference between the average of the estimated values from 500 simulations and the true value of the parameters;
- *Empirical Standard Error (ESE)*: The sample standard deviation of the 500 estimates;
- *Asymptotic Standard Error (ASE)*: The average of 500 estimated asymptotic standard deviation of the estimators;

- *Empirical Coverage Probability (ECP)*: The proportion of the 500 confidence intervals that contain the true parameter value;
- *Asymptotic Efficiency (Efficiency)*: The ratio of the asymptotic variance of an estimator obtained from Methods 2, 3 or 4 relative to those of Method 1.

Simulation Results

We report the simulation results which include *EBias*, *ESE*, *ASE*, *ECP* and *Efficiency* for the four estimation methods. For the setting of strong dependence and $n = 500$, Table 2.2 summarizes the results for marginal parameters and dependence parameters. The results for strong dependence and $n = 1000$ are reported in Table A.1 and those for moderate dependence and $n = 500, 1000$ are summarized in Tables A.2 and A.3 in Appendix A.1.1.

The results in Table 2.2 show that when dependence is strong and sample size is 500, the finite sample biases for the estimates of the marginal parameters η obtained from all four estimation methods are fairly small, ASEs and ESEs are close to each other, and ECP is close to the 95% nominal level. These results suggest that the proposed composite likelihood methods (i.e. Method 3 and 4) yield consistent estimates. However, these methods may incur noticeable efficiency loss; Method 3 is more efficient than Method 4, as expected. Similar patterns are observed for the estimates of the dependence parameters within blocks θ, as shown in Table 2.2.

As expected, the performance of the four methods becomes better as the sample size increases, as displayed in Tables A.1 and A.3. The efficiency loss incurred by the composite likelihood methods becomes less severe when the dependence among the response components is weaker, as illustrated in Tables A.2 and A.3. We notice that the efficiency loss remains stable as the sample size increases by exploring the performance under the settings with $n = 1000$. Generally speaking, the efficiency loss of using the simultaneous composite likelihood (i.e., Method 3) is mild to moderate for within-block parameters θ, while the computational time is significantly reduced compared to using the simultaneous full likelihood (i.e., Method 1). The efficiency loss of coefficient estimators β using the two-stage estimation procedure is obviously more severe by further ignoring dependence structure within blocks. In Table 2.2, the two-stage estimation procedures based on full likelihood and composite likelihood (Method 2 and 4) suffer from a similar amount of efficiency loss when estimating the dependence parameters θ, suggesting that the efficiency loss is mainly due to the variation introduced from the first stage when estimating marginal parameters, and is not aggravated much by making working conditional independence assumptions. Under the moderate dependence setting, the two-stage procedure still leads to

Table 2.2: Simulation results using the four estimation methods: strong dependence and $n = 500$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	0.317	0.424	0.179	0.141	-0.008	-0.017	0.021	0.010	-0.032	-0.015	-0.012	0.035	-0.364	-0.332	-0.422	-0.354	-0.118	-0.072	-0.017	1.963	-0.012	0.744
	ESE[2]	0.064	0.065	0.067	0.068	0.004	0.003	0.004	0.006	0.007	0.006	0.007	0.010	0.031	0.032	0.032	0.034	0.180	0.076	0.004	0.219	0.008	0.184
	ASE[3]	0.065	0.066	0.066	0.070	0.003	0.003	0.003	0.006	0.007	0.006	0.007	0.010	0.031	0.032	0.032	0.034	0.177	0.071	0.004	0.220	0.007	0.182
	ECP[4]	0.954	0.951	0.951	0.954	0.954	0.936	0.946	0.956	0.954	0.949	0.949	0.951	0.949	0.949	0.949	0.944	0.951	0.951	0.951	0.949	0.951	0.954
Method 2: Full likelihood Two-stage Estimation	EBias*	0.745	0.316	-0.107	0.389	0.158	0.010	0.096	-0.130	-0.081	-0.011	0.039	0.060	-0.910	-0.799	-1.057	-0.920	-10.607	-3.791	-0.233	-21.440	-0.364	-6.405
	ESE	0.119	0.116	0.121	0.123	0.031	0.030	0.032	0.033	0.013	0.012	0.011	0.010	0.053	0.054	0.057	0.058	0.221	0.110	0.007	0.266	0.009	0.218
	ASE	0.119	0.119	0.119	0.119	0.031	0.031	0.031	0.031	0.013	0.012	0.011	0.011	0.056	0.055	0.058	0.060	0.228	0.107	0.006	0.302	0.009	0.246
	ECP	0.945	0.955	0.945	0.953	0.948	0.938	0.960	0.958	0.953	0.948	0.950	0.955	0.945	0.953	0.953	0.950	0.925	0.940	0.933	0.880	0.933	0.953
	Efficiency	0.299	0.309	0.308	0.343	0.012	0.008	0.012	0.036	0.278	0.264	0.423	0.827	0.305	0.336	0.298	0.319	0.601	0.441	0.460	0.533	0.613	0.550
Method 3: Composite likelihood Simultaneous Estimation	EBias*	0.256	0.378	0.166	0.047	0.038	-0.022	-0.004	0.011	-0.041	0.005	0.029	0.076	-0.709	-0.733	-0.838	-0.773	-0.256	-0.472	-0.052	1.418	-0.051	0.310
	ESE	0.086	0.088	0.090	0.090	0.006	0.005	0.005	0.006	0.016	0.017	0.018	0.019	0.039	0.042	0.044	0.046	0.241	0.111	0.006	0.241	0.009	0.185
	ASE	0.089	0.091	0.091	0.094	0.006	0.005	0.005	0.006	0.017	0.018	0.019	0.020	0.041	0.044	0.045	0.048	0.235	0.105	0.006	0.252	0.008	0.185
	ECP	0.948	0.955	0.955	0.958	0.950	0.958	0.958	0.958	0.955	0.958	0.950	0.948	0.953	0.943	0.953	0.955	0.945	0.950	0.935	0.948	0.955	0.953
	Efficiency	0.535	0.530	0.526	0.554	0.345	0.330	0.455	0.980	0.166	0.118	0.145	0.241	0.556	0.516	0.493	0.494	0.567	0.458	0.501	0.766	0.760	0.974
Method 4: Composite likelihood Two-stage Estimation	EBias*	0.745	0.316	-0.107	0.389	0.158	0.010	0.096	-0.130	-0.081	-0.011	0.039	0.060	-0.910	-0.799	-1.057	-0.920	-6.816	-2.362	-0.157	-19.722	-0.461	-7.777
	ESE	0.119	0.116	0.121	0.123	0.031	0.030	0.032	0.033	0.013	0.012	0.011	0.010	0.053	0.054	0.057	0.058	0.253	0.129	0.007	0.274	0.010	0.222
	ASE	0.119	0.119	0.119	0.119	0.031	0.031	0.031	0.031	0.013	0.012	0.011	0.011	0.056	0.055	0.058	0.060	0.249	0.121	0.007	0.306	0.010	0.247
	ECP	0.945	0.955	0.945	0.953	0.948	0.938	0.960	0.958	0.953	0.948	0.950	0.955	0.945	0.953	0.953	0.950	0.935	0.935	0.935	0.885	0.918	0.945
	Efficiency	0.299	0.309	0.308	0.343	0.012	0.008	0.012	0.036	0.278	0.264	0.423	0.827	0.305	0.336	0.298	0.319	0.505	0.347	0.393	0.517	0.545	0.547

[1] EBias*=EBias$\times 10^2$
[2] ESE: Empirical Standard Error
[3] ASE: Asymptotic Standard Error
[4] ECP: Empirical Coverage Probability

significant efficiency loss on marginal parameters, but comparable and mild efficiency loss on dependence parameters using both full likelihood and composite likelihood as shown in Tables A.2 and A.3.

In summary, all four methods are valid and provide consistent results for the estimation of parameters of the models. Simultaneous composite likelihood provides consistent estimates for all within-block parameters with moderate efficiency loss, even when the sample size is small and dependence is strong, while the two-stage estimation procedure of full likelihood and composite likelihood could introduce biases and significant efficiency loss under the strong dependence structure, although it can greatly speed up the estimation process.

2.5.2 Robustness

In this section, we examine the robustness of the simultaneous and two-stage composite likelihood estimation procedures (i.e., Method 3 and Method 4 in Section 2.5.1) in contrast to the counterparts based on full likelihood formulation (i.e., Method 1 and Method 2 in Section 2.5.1).

Simulation Settings

The simulation studies have the same settings as those in Section 2.5.1. To examine how the four methods behave when the dependence structure connecting different time blocks is misspecified, we simulate data from settings where all (conditional) bivariate copulas connecting the time blocks are all specified as Frank(7.93) for strong dependence and Frank(2.92) for moderate dependence setting, respectively, but we assume them to be Gaussian copula functions for model fitting.

Simulation Results

We report the performance of the four estimation methods in terms of the same evaluation metrics as described in Section 2.5.1. The results for the strong dependence or moderate dependence and $n = 500$ or $n = 1000$ are summarized in Tables A.4, A.5, A.6 and A.7 in Appendix A.1.2.

Under simultaneous estimation procedure, the full likelihood fails to provide consistent estimators for both marginal and dependence parameters, with non-ignorable empirical biases, gaps between ASEs and ESEs, and discrepancies between the ECPs and the 95% nominal level. These patterns are not improved by increasing the sample size, while they are less severe for a weaker dependence. The full likelihood based two-stage estimation provides inefficient yet valid estimators for marginal parameters but invalid results for the dependence parameters. Both simultaneous and two-stage estimation procedures based on the proposed composite likelihood function (Methods 3 and 4) provide valid results for both marginal parameters $(\beta^{\mathrm{T}}, \omega^{\mathrm{T}})^{\mathrm{T}}$ and dependence parameters θ within time blocks. Estimators using Method 3 incur less finite sample biases and are more efficient than Method 4. The proposed composite likelihood provide robustness with respect to misspecification of dependence structure linking time blocks.

2.5.3 Copula Selection

In this subsection, we aim to explore the capacity of the proposed copula selection procedure in Section 2.4 and examine how frequently we can select the correct copula forms for C-Vine structure within the time blocks.

Simulation Setting

We simulate data from the same setting as that in Section 2.5.1. We evaluate the performance for copula selection under both the strong and moderate dependence settings, and $n = 500$ or 1000. The simulation is repeated 500 times.

Copula Set and Evaluation Metrics

For simplicity, we construct a set of candidate copula functions including the commonly-used copulas in the Archimedean family (Clayton, Gumbel, Frank and Joe copula), Gaussian copula and t copula. The *mis-selected rate* of a copula function is used to evaluate the copula selection performance, which is computed as the number of times for which the copula function is incorrectly selected divided by the number of simulations.

Simulation Results

We report the mis-selected rates for the six (conditional) bivariate copulas in Table 2.3, where the correct forms are specified in Table 2.1.

Table 2.3: Mis-selected rates for copula functions within each block

Degree of Dependence	Sample Size	$\varepsilon_{ik1}, \varepsilon_{ik2}$	$\varepsilon_{ik1}, \varepsilon_{ik3}$	$\varepsilon_{ik1}, \varepsilon_{ik4}$	$\varepsilon_{ik2}, \varepsilon_{ik3} \mid \varepsilon_{ik1}$	$\varepsilon_{ik2}, \varepsilon_{ik4} \mid \varepsilon_{ik1}$	$\varepsilon_{ik3}, \varepsilon_{ik4} \mid \varepsilon_{ik1}, \varepsilon_{ik2}$
Strong Dependence	500	0.264	0	0.008	0	0	0
	1000	0.192	0	0.006	0	0	0
Moderate Dependence	500	0.182	0	0	0.002	0.002	0.024
	1000	0.074	0	0	0	0.002	0.006

The mis-selected rates for all the (conditional) bivariate copulas are close to 0, except for the bivariate copula between ε_{ik1} and ε_{ik2}, for which the true form is a Clayton copula. The mis-selected rates of all (conditional) bivariate copulas drop, as the sample size increases or the dependence becomes weaker. The mis-selected rate for the Clayton copula drops from 26.4% to 19.2% by increasing the sample size from 500 to 1000 in the scenario of a strong dependence and drops even more dramatically from 18.2% to 7.4% for the scenario of a moderate dependence. Generally speaking, we are confident with the proposed copula selection method with fairly low mis-selected rates.

2.5.4 Prediction

In this subsection, we evaluate the prediction performance of the proposed R-Vine model and compare it to that of the conventional regression models and time-series models. We consider various settings and evaluation metrics first. We described the two kinds of prediction of interest: subject extrapolation and time extrapolation, respectively, and finally report our findings.

Simulation Settings

We consider the following scenarios. For all the scenarios, we simulate 200 datasets of the sample size $n = 500$. The covariates x_{ikl} are generated independently from the uniform distribution on $[0, 5]$ for $i = 1, \ldots, n$; $k = 1, \ldots, a$; and $l = 1, \ldots, b$.

- *Scenario 1*: The first simulation setting is the same as that in Section 2.5.1 with $a = 5$ and $b = 4$. We consider both the strong dependence (S) and the moderate dependence (M) settings.
- *Scenario 2*: We consider the same settings as those in Scenarios 1, except that we restrict the parameters in the marginal model across different time points to be the same. Specifically, we set $\eta_l = (\beta_{0l}, \beta_{1l}, \beta_{2l}, \sigma_l)^{\mathrm{T}} = (2.5, 3.5, 4.5, 2)^{\mathrm{T}}$ for $l = 1, 2, 3, 4$.

- *Scenario 3*: We consider the same setting as those of Scenarios 1, except that the dependence structures within each time block previously assumed to be the same are allowed to be different from block to block. More specifically, the bivariate copulas and the value of dependence parameters for the strong and the moderate dependence settings are given in Table 2.4, in which the kth row corresponds to the set-up for the kth time block for $k = 1, 2, 3, 4, 5$.

- *Scenario 4*: We consider the same settings as those of Scenarios 3, except that we further restrict the parameters in the marginal model across different time points to be the same parameters and their values are set to be $\eta_l = (\beta_{0l}, \beta_{1l}, \beta_{2l}, \sigma_l)^{\mathrm{T}} = (2.5, 3.5, 4.5, 2)^{\mathrm{T}}$ for $l = 1, 2, 3, 4$.

- *Scenario 5*: The error terms ε_i are simulated from an AR(1) structure instead of a R-Vine. We set $\rho = 0.5$ for $m = ab = 20$ time points. The marginal model is assumed to be

$$Y_{ij} = 2.5 + 3.5x_{ij} - 50sin\left(\frac{\pi j}{2}\right) + 50cos\left(\frac{\pi j}{2}\right) + \varepsilon_{ij},$$

  where ε_{ij} are independently generated from $N(0, 1)$ for $i = 1, \dots, n$ and $j = 1, \dots, m$. The sine and cosine functions are used to model the periodic trend.

- *Scenario 6*: We consider the same setting as that of Scenario 5, except that the marginal model does not contain the periodic sine and cosine functions but is of the form

$$Y_{ij} = 2.5 + 3.5x_{ij} + 4.5j + \varepsilon_{ij},$$

  where ε_{ij} are independently generated from $N(0, 1)$ for $i = 1, \dots, n$ and $j = 1, \dots, m$.

Table 2.4: Copula functions and the values of dependence parameters in dependence structure within time blocks for strong and moderate dependence settings

Bivariate Variables	$\varepsilon_{i11}, \varepsilon_{i12}$	$\varepsilon_{i11}, \varepsilon_{i13}$	$\varepsilon_{i11}, \varepsilon_{i14}$	$\varepsilon_{i12}, \varepsilon_{i13} \vert \varepsilon_{i11}$	$\varepsilon_{i12}, \varepsilon_{i14} \vert \varepsilon_{i11}$	$\varepsilon_{i13}, \varepsilon_{i14} \vert \varepsilon_{i11}, \varepsilon_{i12}$
Copula Function	Clayton	Gumbel	Gaussian	Frank	Gaussian	Frank
Strong Dependence	4.67	3.33	0.89	7.93	0.81	5.74
Moderate Dependence	1.33	1.67	0.59	2.92	0.45	1.86

Bivariate Variables	$\varepsilon_{i21}, \varepsilon_{i22}$	$\varepsilon_{i21}, \varepsilon_{i23}$	$\varepsilon_{i21}, \varepsilon_{i24}$	$\varepsilon_{i22}, \varepsilon_{i23} \vert \varepsilon_{i21}$	$\varepsilon_{i22}, \varepsilon_{i24} \vert \varepsilon_{i21}$	$\varepsilon_{i23}, \varepsilon_{i24} \vert \varepsilon_{i21}, \varepsilon_{i22}$
Copula Function	Joe	Clayton	Gumbel	Joe	Clayton	Joe
Strong Dependence	5.46	4.67	3.33	3.83	3.00	2.86
Moderate Dependence	2.22	1.33	1.67	1.77	0.86	1.44

Bivariate Variables	$\varepsilon_{i31}, \varepsilon_{i32}$	$\varepsilon_{i31}, \varepsilon_{i33}$	$\varepsilon_{i31}, \varepsilon_{i34}$	$\varepsilon_{i32}, \varepsilon_{i33} \vert \varepsilon_{i31}$	$\varepsilon_{i32}, \varepsilon_{i34} \vert \varepsilon_{i31}$	$\varepsilon_{i33}, \varepsilon_{i34} \vert \varepsilon_{i31}, \varepsilon_{i32}$
Copula Function	Frank	Gumbel	Gaussian	Frank	Gumbel	Frank
Strong Dependence	11.41	3.33	0.89	7.93	2.50	5.74
Moderate Dependence	4.16	1.67	0.59	2.92	1.43	1.86

Bivariate Variables	$\varepsilon_{i41}, \varepsilon_{i42}$	$\varepsilon_{i41}, \varepsilon_{i43}$	$\varepsilon_{i41}, \varepsilon_{i44}$	$\varepsilon_{i42}, \varepsilon_{i43} \vert \varepsilon_{i41}$	$\varepsilon_{i42}, \varepsilon_{i44} \vert \varepsilon_{i41}$	$\varepsilon_{i43}, \varepsilon_{i44} \vert \varepsilon_{i41}, \varepsilon_{i42}$
Copula Function	Joe	Clayton	Gumbel	Joe	Clayton	Joe
Strong Dependence	5.46	4.67	3.33	7.93	2.50	5.74
Moderate Dependence	2.22	1.33	1.67	2.92	1.43	1.86

Bivariate Variables	$\varepsilon_{i51}, \varepsilon_{i52}$	$\varepsilon_{i51}, \varepsilon_{i53}$	$\varepsilon_{i51}, \varepsilon_{i54}$	$\varepsilon_{i52}, \varepsilon_{i53} \vert \varepsilon_{i51}$	$\varepsilon_{i52}, \varepsilon_{i54} \vert \varepsilon_{i51}$	$\varepsilon_{i53}, \varepsilon_{i54} \vert \varepsilon_{i51}, \varepsilon_{i52}$
Copula Function	Clayton	Joe	Frank	Joe	Clayton	Joe
Strong Dependence	11.41	3.33	0.89	3.83	3.00	2.86
Moderate Dependence	4.16	1.67	0.59	1.77	0.86	1.44

Scenarios 3 and 4 are designed to evaluate the prediction performance of the proposed R-Vine model where the dependence structures within each time block are not identical.

We fit the following models and compare their prediction performance.

- *VINE*: The proposed R-Vine copula model is fitted using the proposed composite likelihood method described in Section 2.3.1 and 2.3.2. For this model, we consider the following four estimation procedures:

  (1) *VINE1* : For Scenarios 1-4, the (conditional) bivariate copula functions are assumed to follow the forms in Table 2.1. For Scenarios 5 and 6, the (conditional) bivariate copula functions are all assumed to be the Gaussian copula. The parameters are estimated using simultaneous estimation.

(2) *VINE2* : For Scenarios 1-4, the (conditional) bivariate copula functions are assumed to follow the forms in Table 2.1. For Scenarios 5 and 6, the (conditional) bivariate copula functions are all assumed to be the Gaussian copula. The parameters are estimated using two-stage estimation procedure.

(3) *VINE3* : The (conditional) bivariate copulas are selected using the methods presented in Section 4 and the parameters are estimated under simultaneous estimation presented in Section 2.3.1.

(4) *VINE4* : The (conditional) bivariate copulas are selected using the methods presented in Section 4 and the parameters are estimated under two-stage estimation presented in Section 2.3.2.

- *MRM*: We assume that the marginal model for the lth time point is identical across time blocks. A marginal regression model of the form (2.11) is fitted. The dependence structure is completely ignored.

- *LRM*: A linear regression model is fitted, which takes both time block k and time point l as covariates and is of the form

$$Y_{ikl} = \beta_0 + \beta_1 x_{ikl} + \beta_2 k + \beta_3 l + \varepsilon_{ikl},$$

  where ε_{ikl} are assumed to follow $N(0, \sigma^2)$, for $i = 1, \ldots, n; k = 1, 2, 3, 4, 5; l = 1, 2, 3, 4$

- *AR*: An AR model in time series analysis is considered. The model form and the time lag are determined from the data.

Subject Extrapolation and Time Extrapolation

Two kinds of prediction are of our interest: *subject extrapolation* and *time extrapolation*. We explain the meaning of two kinds of predictions, how we create the training and test set and how we conduct prediction in both cases.

- *Subject Extrapolation*: We are interested in predicting the value of the response for a new subject at a past or current time point. We partition the data by subjects, use 90% of the subjects as the training set, denoted by $\{(y_i^{\mathrm{T}}, x_i^{\mathrm{T}})^{\mathrm{T}} : i = 1, \ldots, 450\}$, and reserve 10% of the subjects as the test set, denoted by $\{(y_i^{\mathrm{T}}, x_i^{\mathrm{T}})^{\mathrm{T}} : i = 451, \ldots, 500\}$. The training set is used to fit a model, which is utilized to predict y_{ikl} for a subject from the test set using its covariate information and responses from the first $l - 1$ time points in the kth time block.

- *Time Extrapolation*: We are interested in predicting the response value for a subject at a future time point. We partition the data by time points, use the time points from the first four blocks as the training set, denoted by $\{(y_{ikl}^{\mathrm{T}}, x_{ikl}^{\mathrm{T}})^{\mathrm{T}} : i = 1, \ldots, 500; k = 1, 2, 3, 4; l = 1, 2, 3, 4\}$, and reserve the time points in the fifth block as the test set, denoted by $\{(y_{ikl}^{\mathrm{T}}, x_{ikl}^{\mathrm{T}})^{\mathrm{T}} : i = 1, \ldots, 500; k = 5; l = 1, 2, 3, 4\}$. The training set is used to fit a model, which is utilized to predict y_{ikl} for a time point in time block $k = 5$, based on the covariate information and the first $l - 1$ time points in the 5th time block.

Evaluation Metrics

Let $y_{ikl}^{(r)}$ denote the response value of the ith subject at the lth time point in the kth time block from the rth independent dataset and let $\hat{y}_{ikl}^{(r)}$ be the corresponding predicted value. We consider the following two evaluation metrics:

- *Mean Absolute Error (MAE)*: the mean of the absolute difference between the predicted value and the true value over all time points in the test set across 200 simulations. To evaluate subject extrapolation, the MAE is computed as

$$\frac{1}{200 \times 50 \times 5 \times 4} \sum_{r=1}^{200} \sum_{i=451}^{500} \sum_{k=1}^{5} \sum_{l=1}^{4} |\hat{y}_{ikl}^{(r)} - y_{ikl}^{(r)}|;$$

  to evaluate time extrapolation, it is computed by

$$\frac{1}{200 \times 500 \times 4} \sum_{r=1}^{200} \sum_{i=1}^{500} \sum_{l=1}^{4} |\hat{y}_{i5l}^{(r)} - y_{i5l}^{(r)}|.$$

  The model that provides a smaller MAE is preferred.

- *Percentage Outperformance*: Percentage outperformance of Model 1 versus Model 2 is calculated as the number of times that Model 1 provides a smaller MAE than Model 2, divided by the number of time points in the test set and then averaged over 200 simulations. If percentage outperformance is over 50%, Model 1 provides better prediction accuracy than Model 2. Let $\hat{y}_{ikl}^{(1r)}$ and $\hat{y}_{ikl}^{(2r)}$ be the predicted values from Models 1 and 2, respectively. To evaluate subject extrapolation, percentage outperformance is computed as

$$\frac{1}{200 \times 50 \times 5 \times 4} \sum_{r=1}^{200} \sum_{i=451}^{500} \sum_{k=1}^{5} \sum_{l=1}^{4} I(|\hat{y}_{ikl}^{(1r)} - y_{ikl}^{(r)}| \leq |\hat{y}_{ikl}^{(2r)} - y_{ikl}^{(r)}|);$$

to evaluate time extrapolation, it is computed as

$$\frac{1}{200 \times 500 \times 4} \sum_{r=1}^{200} \sum_{i=1}^{500} \sum_{l=1}^{4} I(|\hat{y}_{i5l}^{(1r)} - y_{i5l}^{(r)}| \leq |\hat{y}_{i5l}^{(2r)} - y_{i5l}^{(r)}|).$$

Percentage outperformance is more robust than the MAE, which may be sensitive to extreme prediction values.

Prediction Results

We report simulation results for subject and time extrapolations using all candidate models. The boxplots of MAEs of 200 simulations for subject and time extrapolation are given in Figures 2.2 and 2.3, respectively. There are 10 sub-figures in both figures, corresponding to each considered simulation scenario. In each subfigure, there are 7 boxplots corresponding to the 7 models to be compared. From Figures 2.2 and 2.3, the boxplots of the four vine-based methods do not differ noticeably. The biases of estimators of VINE2 and VINE4 using the two-stage estimation are larger than those obtained from the simultaneous procedure, as we find in Section 2.5.1. The mis-selected rate of some copula functions can be as high as about 26% when using VINE3 and VINE4, as we find in Section 2.5.3. However, the prediction results are fairly robust with respect to estimation biases and model misspecification.

Table 2.5: MAEs of different models for subject extrapolation under the proposed scenarios

	VINE1	VINE2	VINE3	VINE4	MRM	LRM	AR
Scenario 1(S)	0.761 (1.158)	0.762 (1.159)	0.767 (1.159)	0.767 (1.160)	1.598 (1.977)	2.249 (2.841)	5.331 (6.550)
Scenario 1(M)	1.145 (1.486)	1.145 (1.487)	1.147 (1.487)	1.147 (1.488)	1.604 (1.980)	2.252 (2.843)	5.332 (6.549)
Scenario 2(S)	0.761 (1.158)	0.762 (1.159)	0.767 (1.159)	0.767 (1.160)	1.598 (1.977)	1.598 (1.977)	1.599 (1.975)
Scenario 2(M)	1.145 (1.486)	1.145 (1.487)	1.147 (1.487)	1.147 (1.488)	1.604 (1.980)	1.604 (1.980)	1.606 (1.978)
Scenario 3(S)	0.871 (1.216)	0.889 (1.270)	0.831 (1.217)	0.834 (1.270)	1.599 (1.986)	2.248 (2.856)	6.098 (7.336)
Scenario 3(M)	1.232 (1.555)	1.235 (1.572)	1.199 (1.555)	1.201 (1.572)	1.605 (1.986)	2.253 (2.858)	6.100 (7.337)
Scenario 4(S)	0.871 (1.216)	0.889 (1.270)	0.831 (1.217)	0.835 (1.271)	1.600 (1.986)	1.600 (1.985)	1.601 (1.985)
Scenario 4(M)	1.232 (1.555)	1.235 (1.572)	1.199 (1.555)	1.201 (1.573)	1.606 (1.986)	1.606 (1.986)	1.607 (1.985)
Scenario 5	0.830 (1.038)	0.830 (1.039)	0.830 (1.038)	0.830 (1.039)	0.923 (1.153)	0.923 (1.154)	0.922 (1.152)
Scenario 6	0.830 (1.038)	0.830 (1.039)	0.830 (1.038)	0.830 (1.039)	0.923 (1.153)	0.922 (1.154)	0.922 (1.152)

S: strong dependence setting; M: moderate dependence setting

Table 2.6: MAEs of different models for time extrapolation under the proposed scenarios

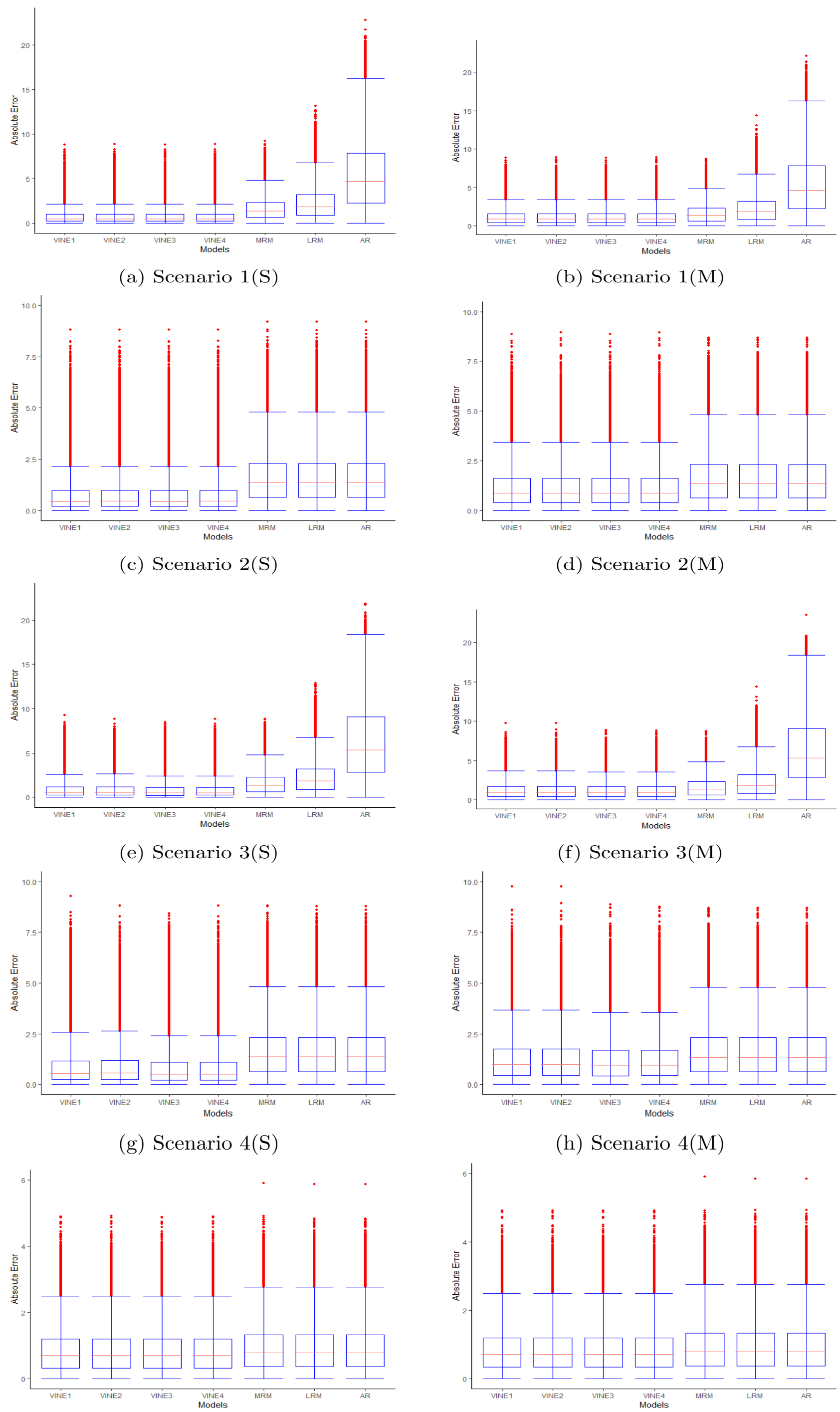
	VINE1	VINE2	VINE3	VINE4	MRM	LRM	AR
Scenario 1(S)	0.760 (1.076)	0.760 (1.082)	0.765 (1.076)	0.765 (1.083)	1.596 (1.954)	2.999 (2.823)	11.356 (7.681)
Scenario 1(M)	1.145 (1.344)	1.145 (1.352)	1.146 (1.345)	1.146 (1.352)	1.598 (1.963)	3.002 (2.832)	11.360 (7.682)
Scenario 2(S)	0.760 (1.076)	0.760 (1.083)	0.765 (1.076)	0.765 (1.083)	1.596 (1.953)	1.596 (1.953)	1.597 (1.951)
Scenario 2(M)	1.145 (1.344)	1.145 (1.352)	1.146 (1.344)	1.146 (1.353)	1.598 (1.963)	1.597 (1.963)	1.598 (1.962)
Scenario 3(S)	0.847 (0.663)	0.865 (0.675)	0.837 (0.664)	0.888 (0.675)	1.596 (1.942)	3.000 (2.818)	11.356 (7.665)
Scenario 3(M)	1.219 (1.168)	1.222 (1.190)	1.230 (1.169)	1.232 (1.190)	1.599 (1.951)	3.002 (2.827)	11.359 (7.781)
Scenario 4(S)	0.847 (0.663)	0.865 (0.675)	0.837 (0.664)	0.888 (0.675)	1.596 (1.942)	1.596 (1.942)	1.597 (2.976)
Scenario 4(M)	1.219 (1.168)	1.222 (1.190)	1.230 (1.169)	1.232 (1.190)	1.599 (1.951)	1.598 (1.951)	1.599 (2.981)
Scenario 5	0.830 (1.040)	0.830 (1.040)	0.830 (1.040)	0.830 (1.040)	0.922 (1.154)	0.922 (1.154)	0.920 (1.153)
Scenario 6	0.830 (1.040)	0.831 (1.040)	0.831 (1.040)	0.831 (1.040)	0.923 (1.156)	0.923 (1.156)	0.922 (1.154)

S: strong dependence setting; M: moderate dependence setting

The four vine-based methods provide smaller and less variant MAEs across all the considered scenarios and for both subject and time extrapolations, suggesting superiority in prediction performance compared to other models. In Scenarios 1-4, it is not surprising that the vine-based models outperform the other ones, since the true models hold a vine structure. But the vine-based models still slightly outperform the AR model when the true model holds an AR(1) structure in Scenarios 5-6. AR performs either comparably to MRM and LRM or a lot worse (e.g., in scenarios 1 and 3). The four vine-based models have smaller MAEs when the dependence is stronger while the MAEs are comparable in the strong and moderate settings when using MRM, LRM and AR models.

VINE1 and VINE3 yield smaller prediction standard errors than VINE2 and VINE4, because the simultaneous estimation tends to be more efficient than the two-stage estimation. However, factoring in the computation cost, the improvement of using the former method over the latter one seems marginal; in applications, it may not always be worthwhile to pursue the simultaneous estimation method due to its computation cost. Incorporating the observation history can greatly reduce the prediction standard errors. Moreover, prediction standard errors decrease as the strength of dependence increases.

We report the MAEs of different models for subject extrapolation in Table 2.5, for time extrapolation in Table 2.6, and percentage outperformance of VINE4 versus the other models in Table A.8, which further supports our comments above. In Appendix A.1.3, we report the boxplots of MAEs by time points for subject and time extrapolation, respectively. We find the MAEs for a later time point are always smaller and less variant when using the vine models, which is the benefit of taking into account the dependence structure within time blocks.



(a) Scenario 1(S) (b) Scenario 1(M)

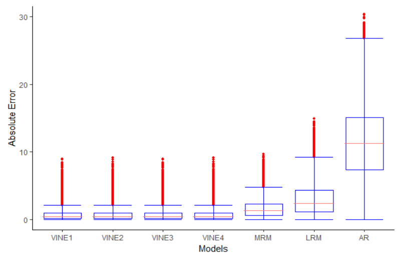
(c) Scenario 2(S) (d) Scenario 2(M)

(e) Scenario 3(S) (f) Scenario 3(M)

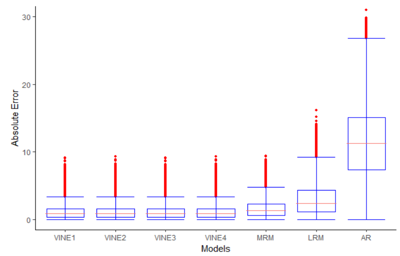
(g) Scenario 4(S) (h) Scenario 4(M)

(i) Scenario 5 (j) Scenario 6

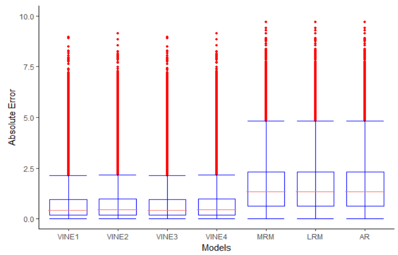
Figure 2.2: Boxplots of MAEs of different models for subject extrapolation



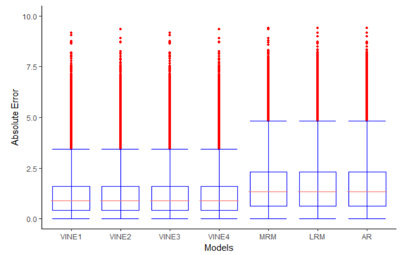
(a) Scenario 1(S)



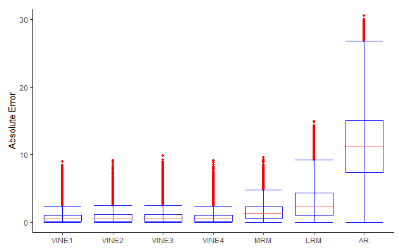
(b) Scenario 1(M)



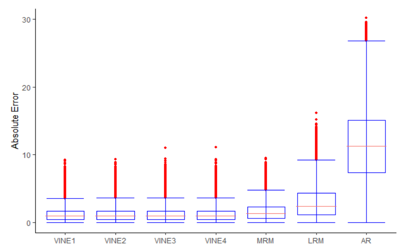
(c) Scenario 2(S)



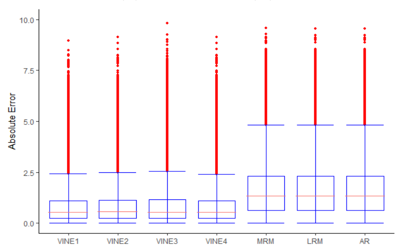
(d) Scenario 2(M)



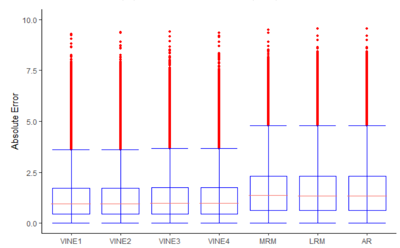
(e) Scenario 3(S)



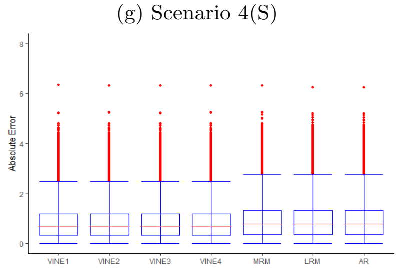
(f) Scenario 3(M)



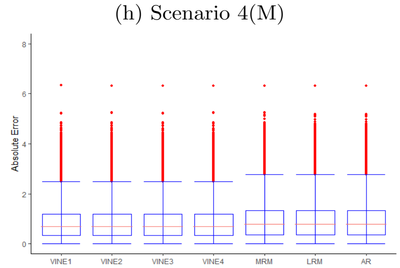
(g) Scenario 4(S)



(h) Scenario 4(M)



(i) Scenario 5



(j) Scenario 6

Figure 2.3: Boxplots of MAEs of different models for time extrapolation

2.6 Data Analysis

2.6.1 Dataset

We consider the climate data available publicly on the website of Government of Canada. It is homogenized Canadian surface air temperature data (Vincent et al., 2012). The data is available at https://www.canada.ca/en/environment-climate-change/services/climate-change/science-research-data/climate-trends-variability/adjusted-homogenized-canadian-data.html. The dataset we use contains monthly mean of daily mean temperature in Celsius degree at 47 Ontarian observation stations from January 1978 to December 2018. Figure 2.4 is a run chart of the monthly temperature of the 47 stations from January 1978 to December 2018, which obviously exhibits a yearly periodic pattern and a mild overall increasing trend.

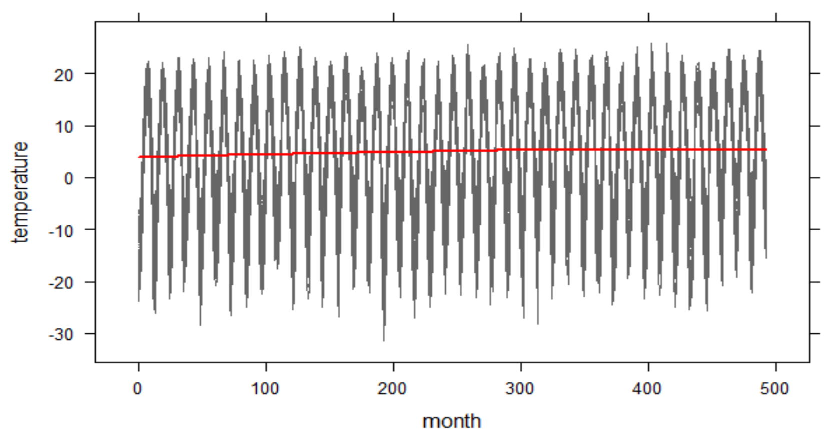


Figure 2.4: The monthly temperature of all 47 stations from Jan. 1978 to Dec. 2018

2.6.2 Statistical Models

In our analysis, the monthly `temperature` is used as the response variable, and the geographical information, `latitude`, `longitude` and `elevation`, and the time variables `year` are covariates. It is natural to select a year as a time block, yielding $a = 40$ time blocks (years) in total and $b = 12$ time points (months) in each block. We partition the 47 stations into a training group with 42 stations, and a test group with 5 stations, and we make a division in time by letting January 1978 to December 2008 be the training period and January 2009 to December 2018 as the testing period. The station information and the division of stations into training and test groups are given in Table A.9 in supplementary materials. We use the data of the 42 stations from January 1978 to December 2008 to fit a model.

Marginal Model

The `temperature` highly depends on the geographical information, i.e., `latitude`, `longitude` and `elevation`, and tends to have an increasing trend with respect to `year` in some months. Preliminary marginal regression analysis (not shown here) suggests that the four covariates all have linear or quadratic relation with the responses, and the identity link function seems to be adequate, and the error terms of each month are appropriate to be modeled by a normal distribution with mean 0.

We assume that the marginal model for the lth month is of the following form: for $l = 1, 2, 10, 11, 12$,

$$\begin{aligned} Y_{ikl} =& \beta_{0l} + \beta_{1l} \cdot \texttt{latitude} + \beta_{2l} \cdot \texttt{longitude} + \beta_{3l} \cdot \texttt{elevation} \\ &+ \beta_{4l} \cdot \texttt{year} + \varepsilon_{ikl}; \end{aligned} \tag{2.12}$$

and for $l = 3, 4, 5, 6, 7, 8, 9$,

$$\begin{aligned} Y_{ikl} =& \beta_{0l} + \beta_{1l} \cdot \texttt{latitude} + \beta_{2l} \cdot \texttt{longitude} + \beta_{2l2} \cdot \texttt{longitude}^2 \\ &+ \beta_{3l} \cdot \texttt{elevation} + \beta_{4l} \cdot \texttt{year} + \varepsilon_{ikl}, \end{aligned} \tag{2.13}$$

where the ε_{ikl} are marginally distributed as $N(0, \sigma_l^2)$, for $l = 1, \ldots, 12$.

Dependence Model

We ignore the dependence structure between years, model the dependence between months within each year through a C-Vine. We first select the copula functions for the C-Vine structure within each year by using the copula selection method we proposed in Section 2.4, which is implemented using the `VineCopula` package in `R` based on a dataset of 1260 years with each of the 42 stations in the training group contributing 30 years (the training period). All copula functions available in the `VineCopula` package are included in the candidate set for selection; the available copula functions, are described by Schepsmeier et al. (2018). Table 2.7 summarizes the selected bivariate copula functions, where the

lth row corresponds to the lth level of tree in the C-Vine structure and variable l is the dominating variable in this level of tree. The lth tree and the l'th month in Table 2.7 gives the selected (conditional) bivariate copula functions between variables ε_{ikl} and $\varepsilon_{ikl'}$. The minimum ($\min(\hat{\tau})$) and maximum ($\max(\hat{\tau})$) values of the corresponding Kendall's Tau for each level of the tree are also provided in the last two columns in Table 2.7. We can see that the dependence between time points are moderate, especially in higher level of trees.

Table 2.7: Summary of the selected bivariate copula functions for the C-Vine structure within each year

Tree \ Month	2	3	4	5	6	7	8	9	10	11	12	$\min(\hat{\tau})$	$\max(\hat{\tau})$
1	RT1(180)	T2	Ga	Cl	In	SCl	Cl	Fr	Ga	In	In	-0.151	0.186
2		JF	SCl	In	SCl	SCl	SCl	T1	Jo	T2	T2	0.000	0.215
3			T	Cl	In	Cl	RT1(90)	In	SJC	RT2(180)	T	-0.054	0.179
4				RT1(180)	T	Ga	In	In	RCl(90)	SJF	Ga	-0.089	0.165
5					In	SCl	RT2(180)	In	SCl	In	Jo	0.000	0.076
6						Ga	JC	Fr	Fr	RJo(90)	In	-0.048	0.371
7							SJC	SCl	RJo(90)	Gu	RGu(90)	-0.081	0.178
8								In	RT2(180)	In	T	-0.109	0.111
9									SGu	RT2(180)	In	0.000	0.180
10										RT2(180)	RCl(90)	-0.077	0.053
11											JF	0.208	0.208

Cl=Clayton, Fr=Frank, Ga=Gaussian, Gu=Gumbel, In=Independent, Jo=Joe, T=Student t, T1=Tawn type 1, T2=Tawn Type 2. CG=Clayton-Gumbel mixed, JC=Joe-Clayton mixed, JF=Joe-Frank mixed. R means rotated with rotated degree in the bracket and S means survival copula.

Model Fitting, Model Comparison and Prediction

Based on the selected copula functions, we perform composite likelihood estimation. The total number of parameters, which is around 150, is too large for common optimization algorithm to optimize simultaneously and obtain simultaneous estimators. The four vine-based methods provide comparable prediction results by simulations, thus we implement composite likelihood estimation under two-stage estimation procedures (*VINE4*) here. The estimation for marginal parameters are summarized in Table 2.8 and those for dependence parameters are summarized in Tables A.10 and A.11 in supplementary materials.

Table 2.8: The estimates of marginal parameters for each month under simultaneous estimation and two-stage estimation of composite likelihood method (standard error in the bracket)

	Two-Stage Estimation						
month l	β_{0l}	β_{1l}	β_{2l}	β_{2l2}	β_{3l}	β_{4l}	σ_l
1	-135.740(62.240)	-1.978(0.041)	-0.210(0.037)	-	-0.009(0.002)	0.101(0.030)	3.204(0.064)
2	-34.827(27.651)	-1.739(0.042)	-0.256(0.039)	-	-0.008(0.003)	0.043(0.014)	3.164(0.067)
3	22.785(89.626)	-1.429(0.119)	-44.929(19.069)	16.994(5.543)	-0.005(0.003)	0.021(0.044)	2.207(0.116)
4	2.431(26.704)	-1.012(0.099)	-28.691(12.179)	25.054(4.627)	-0.003(0.002)	0.025(0.133)	1.944(0.090)
5	44.601(6.172)	-0.708(0.100)	-14.893(26.378)	25.789(6.031)	-0.002(0.007)	0.007(0.102)	1.939(0.034)
6	-100.571(21.824)	0.681(0.046)	-17.278(12.666)	22.259(3.084)	-0.002(0.003)	0.075(0.010)	1.578(0.034)
7	30.540(45.497)	-0.626(0.036)	-21.546(5.831)	19.626(2.988)	-0.004(<0.001)	0.010(0.022)	1.417(0.034)
8	1.261(23.072)	-0.685(0.032)	-27.126(5.479)	15.480(3.112)	-0.006(0.001)	0.025(0.011)	1.482(0.032)
9	-90.891(13.597)	-0.877(0.073)	-27.676(10.608)	8.679(3.121)	-0.007(0.001)	0.074(0.006)	1.335(0.063)
10	-54.479(10.110)	-0.918(0.020)	-0.117(0.012)	-	-0.008(<0.001)	0.048(0.005)	1.518(0.029)
11	-28.415(11.045)	-1.275(0.028)	-0.061(0.018)	-	-0.009(<0.001)	0.042(0.006)	2.085(0.047)
12	-68.781(19.581)	-1.764(0.045)	-0.112(0.028)	-	-0.008(<0.001)	0.068(0.010)	3.324(0.067)

In the estimation results, β_{1l} is negative for all 12 months, which suggests high-latitude areas tend to have lower temperature year around and this trend is more obvious in winter months (i.e., $|\beta_{1l}|$ is larger in months 1, 2, 3, 11 and 12). For winter months, i.e., months 1-2 and 10-12, the mean temperature has a linear negative relation with the longitude. For months 3-9 in spring and summer, the mean temperature has a quadratic relation with the longitude. β_{3l} is negative but close to zero, suggesting that as the elevation increases, the mean temperature will slightly decrease. β_{1l}, the annual temperature increase of the lth month in Celsius degree, is positive in all 12 months, which suggests a mildly increasing trend of temperature change over years. The findings perfectly align with our expectations.

We are interested in both subject extrapolation (predicting temperature for a new station based on geographical information and time) and time extrapolation (predicting temperature for a future time). In practice, the former allows us to predict temperatures for locations without a station and the latter allows us to forecasting future temperatures. For subject extrapolation, we predict temperatures for the 5 stations in the test group from January 1978 to December 2008, of which the results are provided in Section 4.3 in Supplementary Materials. For time extrapolation, we predict for 37 stations in the training group from January 2009 to December 2018. There are five stations closed after 2008 and data from January 2009 to December 2018 are not available. We are interested in short-term, mid-term and long-term prediction. For short-term prediction, the prediction for the lth month is made based on information from previous $l-1$ months in the same year and the prediction of the first months is using the marginal distribution; in other words, this is prediction for the next month. For mid-term prediction, the prediction for the lth month is made based on the temperature in the first season (months 1-3) in the same year, for $l = 4, \ldots, 12$; in other words, this is the prediction made for the rest of the year. For long-term prediction, we are predicting the change of the temperature in a decade.

We compare the prediction performance of VINE4 with MRM, LRM and AR using the evaluation metrics MAE and Percentage Outperformance as we did in the simulation studies:

- *MRM*: The monthly marginal regression model (2.12) and (2.13) without considering the dependence structure.
- *LRM*: A linear regression model includes `month` x_5 as a covariate to account for the variation across months. The LRM model is selected by the AIC criterion and fitted to be

$$\begin{aligned} Y_{ikl} =& \beta_0 + \beta_1 \cdot \texttt{latitude} + \beta_2 \cdot \texttt{longitude} + \beta_3 \cdot \texttt{elevation} \\ &+ \beta_4 \cdot \texttt{year} + \sum_{j=1}^{2} \beta_{5j} \cdot \texttt{month}^j + \varepsilon_{ikl}, \end{aligned}$$

  where $\varepsilon_{ikl} \sim N(0, \sigma^2)$.
- *AR*: A time series model, which is selected and fitted to be

$$\begin{aligned} Y_{it} =& \beta_0 + \beta_1 \cdot \texttt{latitude} + \beta_2 \cdot \texttt{longitude} + \beta_3 \cdot \texttt{elevation} \\ &+ \beta_4 \cdot \texttt{year} + \beta_5 sin\left(\frac{\pi t}{6}\right) + \beta_6 cos\left(\frac{\pi t}{6}\right) + \varepsilon_{it}, \end{aligned}$$

  where $\varepsilon_{it} \sim AR(2)$ for $t = 1, \ldots, 360$.
- *SARIMA*: A seasonal autoregressive integrated moving average (SARIMA) model, which is commonly used for seasonal time series data prediction:

$$Y_{it} = \beta_0 + \beta_1 \cdot \texttt{latitude} + \beta_2 \cdot \texttt{longitude} + \beta_3 \cdot \texttt{elevation} + \varepsilon_{it},$$

  where $\varepsilon_{it} \sim \text{SARIMA}(3, 1, 3)(1, 0, 1, 12)$ for $t = 1, \ldots, 360$.

Prediction Results

We evaluate the prediction performance of our proposed method for short-term, mid-term and long-term prediction. Figure 2.5 contains two subfigures, which corresponds to the prediction performance for short-term (on the left) and mid-term (on the right) prediction, respectively. The mid-term prediction was made for months 4-12, but the short-term prediction was made for all 12 months, little previous information is available for months 1-3 and it tends to have large prediction errors in the first three months. Therefore, the short-term prediction has larger median MAEs across all methods.

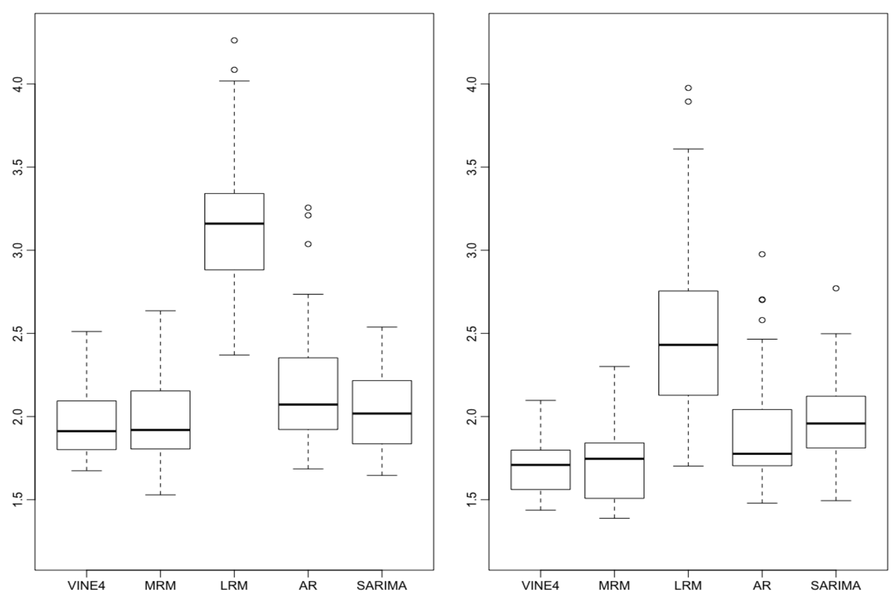


Figure 2.5: Boxplot of MAEs for the short-term (on the left) and mid-term (on the right) time extrapolation

From the boxplots of both short-term and mid-term predictions, the VINE4 has a smaller or comparable median MAEs compared to the other methods, and the MAEs of VINE4 are the least variant. Since the dependence between months within each year is moderate, the advantage of the VINE4 method versus the marginal model (MRM) is limited, which agrees with our findings in Section 2.5.4.

The prediction results for subject extrapolation are summarized in Table 2.9. Both MAEs and percentage outperformances suggest that the proposed R-Vine model estimated using the composite likelihood method can provide a lot more precise prediction than the other three conventional models.

Table 2.9: Prediction results for subject extrapolation (prediction standard error in the brackets)

Name	MAE				Percentage Outperformance		
	VINE4	MRM	LRM	AR	VINE4 vs MRM	VINE4 vs LRM	VINE4 vs AR
BIG TROUT LAKE	**1.916 (1.979)**	2.085 (2.185)	4.127 (4.056)	3.144(2.673)	0.604	0.760	0.708
SIOUX LOOKOUT	**1.908 (2.013)**	2.243 (2.185)	3.840 (4.056)	2.901 (2.857)	0.642	0.717	0.725
BEATRICE	**1.441 (1.949)**	1.555 (2.185)	2.641 (4.056)	1.753 (2.557)	0.625	0.708	0.646
HARROW	**1.568 (1.939)**	1.658 (2.185)	2.798 (4.056)	1.683 (2.629)	0.563	0.667	0.542
ATITOKAN	**1.685 (1.975)**	1.923 (2.185)	3.582 (4.056)	2.304 (2.729)	0.646	0.792	0.646
Average	**1.704 (1.971)**	1.893 (2.185)	3.398 (4.056)	2.357 (2.689)	0.616	0.729	0.653

The prediction results of the 37 stations for time extrapolation in 2018 are summarized in Table 2.10. The VINE4 method provides the smallest MAE for 14 stations, MRM for 16 stations and AR for 2 stations. The VINE4 has the smallest average MAE for all the stations. Since the dependence between months within each year is moderate, the advantage of the VINE4 method versus other models is limited, which agrees with our findings in Section 5. We also find that the MAEs of VINE4 are less variant. However, the MAEs based on other methods give prediction with extremely large MAEs in some occasions (results not shown here).

In addition, we also try to predict the temperature value of the last three seasons in a year, given the temperature values in the first season, i.e., months 1-3. The results for time extrapolation and subject extrapolation are summarized in Tables 2.11 and 2.12, respectively. For the prediction of temperature in the month $l > 4$, we plug in our prediction in months from 4 to $l - 1$ and combine with the true temperature in months 1-3 to form the temperature information in the previous months.

Table 2.10: Prediction result for time extrapolation in year 2018 (prediction standard error in the brackets)

	MAE					Percentage Outperformance			
Name	VINE4	MRM	LRM	AR	SARIMA	VINE4 vs MRM	VINE4 vs LRM	VINE4 vs AR	VINE4 vs SARIMA
LANSDOWNE HOUSE	**2.132 (2.064)**	2.222 (2.193)	4.262 (4.013)	3.210 (2.479)	2.335 (2.482)	0.583	0.740	0.688	0.602
PICKLE LAKE	2.252 (2.015)	**2.218 (2.193)**	4.085 (4.013)	3.256 (2.901)	2.406 (2.767)	0.583	0.726	0.726	0.627
RED LAKE	2.233 (1.998)	**2.154 (2.193)**	3.825 (4.013)	3.037 (2.551)	2.297 (2.590)	0.548	0.702	0.702	0.560
FORT FRANCES	2.511 (1.956)	2.636 (2.193)	3.339 (4.013)	2.541 (2.957)	**2.057 (2.596)**	0.667	0.648	0.537	0.430
MINE CENTRE	2.239 (1.961)	**2.218 (2.193)**	3.341 (4.013)	2.267 (2.649)	2.320 (2.431)	0.575	0.658	0.525	0.565
DRYDEN	2.117 (1.986)	**2.027 (2.193)**	3.785 (4.013)	2.735 (2.816)	2.387 (2.578)	0.491	0.713	0.620	0.600
KENORA	**2.096 (1.997)**	2.112 (2.193)	3.684 (4.013)	2.573 (3.128)	2.538 (2.603)	0.567	0.725	0.600	0.594
CAMERON FALLS	**1.994 (1.954)**	2.054 (2.193)	3.110 (4.013)	2.092 (2.578)	2.479 (2.303)	0.611	0.648	0.509	0.702
GERALDTON	2.184 (2.037)	**2.164 (2.193)**	3.502 (4.013)	2.532 (2.722)	2.374 (2.545)	0.583	0.694	0.542	0.560
THUNDER BAY	1.925 (1.940)	2.006 (2.193)	3.255 (4.013)	2.060 (3.162)	**1.779 (2.258)**	0.630	0.676	0.528	0.475
SAULT STE MARIE	1.961 (1.988)	2.048 (2.193)	3.211 (4.013)	2.311 (2.643)	**1.799 (2.082)**	0.567	0.592	0.617	0.520
WAWA	**1.976 (1.938**	2.240 (2.193)	2.882 (4.013)	2.353 (2.802)	2.452 (2.705)	0.702	0.583	0.619	0.635
CHAPLEAU	1.852 (1.960)	**1.832 (2.193)**	3.022 (4.013)	1.944 (2.760)	1.958 (2.169)	0.597	0.667	0.528	0.550
SUDBURY	1.894 (2.038)	1.831 (2.193)	3.118 (4.013)	**1.823 (2.559)**	2.055 (2.176)	0.467	0.692	0.467	0.642
EARLTON	**1.912 (2.030)**	1.919 (2.193)	3.203 (4.013)	2.014 (2.594)	2.097 (2.302)	0.542	0.650	0.567	0.584
KAPUSKASING	1.953 (2.032)	**1.888 (2.193)**	3.523 (4.013)	2.192 (2.821)	2.104 (2.413)	0.508	0.700	0.517	0.535
MOOSONEE	**1.974 (2.069)**	2.174 (2.193)	4.018 (4.013)	2.676 (2.991)	2.075 (2.280)	0.643	0.702	0.607	0.552
TIMMINS	1.985 (2.024)	**1.913 (2.193)**	3.342 (4.013)	2.072 (3.023)	2.189 (2.385)	0.500	0.675	0.458	0.550
MADAWASKA	2.221 (1.998)	2.472 (2.193)	3.189 (4.013)	2.305 (2.589)	**1.646 (2.053)**	0.667	0.650	0.547	0.395
NORTH BAY	1.850 (1.921)	**1.712 (2.193)**	2.675 (4.013)	1.737 (2.945)	1.836 (2.050)	0.467	0.583	0.450	0.520
GORE BAY	**1.740 (2.002)**	1.776 (2.193)	3.242 (4.013)	2.139 (2.601)	2.196 (1.976)	0.536	0.667	0.631	0.675
BROCKVILLE	**1.674 (1.938)**	1.737 (2.193)	2.814 (4.013)	1.889 (2.796)	1.952 (2.137)	0.512	0.690	0.528	0.550
CORNWALL	1.676 (2.019)	**1.632 (2.193)**	2.909 (4.013)	1.919 (2.614)	2.018 (2.248)	0.491	0.667	0.602	0.646
KINGSTON	**1.776 (2.000)**	1.805 (2.193)	2.904 (4.013)	1.930 (2.769)	1.788 (1.905)	0.611	0.611	0.565	0.510
OTTAWA	1.737 (1.986)	**1.687 (2.193)**	2.728 (4.013)	1.734 (2.689)	1.725 (2.231)	0.509	0.648	0.463	0.476
RIDGETOWN	**2.094 (1.996)**	2.201 (2.193)	3.329 (4.013)	2.485 (2.491)	2.216 (2.010)	0.573	0.677	0.604	0.570
VINELAND	1.804 (2.003)	**1.682 (2.193)**	3.204 (4.013)	2.024 (2.712)	1.752 (1.768)	0.476	0.690	0.548	0.510
WELLAND	1.891 (2.017)	**1.817 (2.193)**	2.985 (4.013)	2.203 (2.657)	1.821 (1.871)	0.533	0.617	0.583	0.557
WINDSOR	2.041 (2.031)	1.914 (2.193)	2.678 (4.013)	**1.864 (2.748)**	1.927 (2.338)	0.500	0.537	0.491	0.520
LONDON	**1.800 (2.023)**	1.898 (2.193)	2.700 (4.013)	1.920 (2.624)	2.000 (1.925)	0.529	0.593	0.565	0.574
WOODSTOCK	1.722 (1.961)	**1.529 (2.193)**	2.370 (4.013)	1.685 (2.731)	1.828 (2.018)	0.472	0.556	0.536	0.545
BELLEVILLE	**1.885 (2.053)**	2.023 (2.193)	3.160 (4.013)	1.922 (2.600)	1.949 (1.911)	0.556	0.583	0.667	0.630
HAMILTON	1.801 (2.037)	**1.778 (2.193)**	2.899 (4.013)	2.009 (2.450)	1.788 (1.902)	0.542	0.583	0.575	0.542
ORANGEVILLE	**1.748 (1.984)**	1.831 (2.193)	2.767 (4.013)	1.983 (2.306)	2.035 (2.412)	0.639	0.556	0.625	0.642
TORONTO	1.788 (1.849)	**1.755 (2.193)**	2.828 (4.013)	2.016 (2.728)	1.958 (2.387)	0.556	0.648	0.546	0.520
HALIBURTON	**1.880(2.104)**	1.955 (2.193)	2.888 (4.013)	1.922 (2.512)	1.939 (2.104)	0.594	0.594	0.565	0.580
PETERBOROUGH	**1.880 (2.038)**	2.010 (2.193)	2.861 (4.013)	2.085 (2.366)	1.998 (2.029)	0.583	0.575	0.550	0.545
Average	**1.951 (2.014)**	1.969 (2.193)	3.179 (4.013)	2.202 (2.707)	2.056 (2.242)	0.560	0.646	0.568	0.562

Table 2.11: Prediction results for subject extrapolation of month 4-12, given the first 3 months (prediction standard error in the brackets)

Name	MAE				Percentage Outperformance		
	VINE4	MRM	LRM	AR	VINE4 vs MRM	VINE4 vs LRM	VINE4 vs AR
BIG TROUT LAKE	**1.721 (1.599)**	1.858 (1.907)	3.838 (4.056)	2.688 (2.739)	0.525	0.764	0.736
SIOUX LOOKOUT	**1.836 (1.616)**	2.010 (1.907)	3.629 (4.056)	2.777 (2.797)	0.578	0.711	0.789
BEATRICE	**1.322 (1.584)**	1.388 (1.907)	1.962 (4.056)	1.605 (2.629)	0.569	0.625	0.708
HARROW	1.586 (1.590)	**1.524 (1.907)**	2.151 (4.056)	1.571 (2.592)	0.458	0.611	0.472
ATITOKAN	**1.500 (1.595)**	1.612 (1.907)	3.055 (4.056)	2.027 (2.447)	0.556	0.792	0.681
Average	**1.593 (1.597)**	1.679 (1.907)	2.927 (4.056)	2.134 (2.641)	0.537	0.701	0.677

Table 2.12: Prediction results for time extrapolation of month 4-12, given the first 3 months (prediction standard error in the brackets)

Name	MAE					Percentage Outperformance			
	VINE4	MRM	LRM	AR	SARIMA	VINE4 vs MRM	VINE4 vs LRM	VINE4 vs AR	VINE4 vs SARIMA
LANSDOWNE HOUSE	**1.900 (1.564)**	1.945 (2.541)	3.894 (4.013)	2.580 (2.710)	2.498 (2.567)	0.514	0.764	0.694	0.665
PICKLE LAKE	2.027 (1.638)	**1.999 (2.541)**	3.976 (4.013)	2.976 (3.010)	2.357 (2.578)	0.508	0.746	0.730	0.654
RED LAKE	1.867 (1.549)	**1.827 (2.541)**	3.581 (4.013)	2.701 (2.597)	2.250 (2.487)	0.524	0.730	0.683	0.625
FORT FRANCES	1.865 (1.550)	2.149 (2.541)	2.751 (4.013)	**1.842 (2.878)**	2.094 (2.733)	0.704	0.654	0.481	0.580
MINE CENTRE	1.764 (1.562)	1.852 (2.541)	2.826 (4.013)	**1.724 (2.659)**	2.122 (2.559)	0.633	0.689	0.478	0.657
DRYDEN	1.952 (1.568)	**1.897 (2.541)**	3.609 (4.013)	2.704 (3.207)	2.441 (2.715)	0.556	0.741	0.679	0.634
KENORA	**1.823 (1.566)**	1.861 (2.541)	3.386 (4.013)	2.246 (3.155)	2.408 (2.735)	0.578	0.789	0.589	0.628
CAMERON FALLS	1.795 (1.572)	1.826 (2.541)	2.500 (4.013)	1.740 (2.635)	**1.730 (2.399)**	0.580	0.580	0.506	0.480
GERALDTON	1.798 (1.679)	**1.785 (2.541)**	3.074 (4.013)	1.878 (2.337)	2.145 (2.329)	0.593	0.685	0.444	0.575
THUNDER BAY	1.736 (1.567)	1.841 (2.541)	2.573 (4.013)	1.776 (2.655)	**1.627 (2.387)**	0.605	0.642	0.516	0.420
SAULT STE MARIE	**1.709 (1.572)**	1.771 (2.541)	2.412 (4.013)	2.104 (2.937)	1.754 (2.229)	0.600	0.533	0.633	0.547
WAWA	**1.912 (1.701)**	1.991 (2.541)	2.131 (4.013)	2.001 (2.816)	2.125 (2.248)	0.603	0.476	0.572	0.625
CHAPLEAU	1.538 (1.492)	1.496 (2.541)	2.498 (4.013)	**1.479 (3.068)**	1.895 (2.174)	0.611	0.685	0.426	0.615
SUDBURY	1.696 (1.577)	1.634 (2.541)	2.531 (4.013)	**1.592 (2.870)**	2.064 (2.257)	0.556	0.656	0.456	0.585
EARLTON	1.654 (1.572)	1.665 (2.541)	2.564 (4.013)	**1.583 (2.869)**	2.013 (2.367)	0.522	0.622	0.422	0.735
KAPUSKASING	1.739 (1.569)	**1.641 (2.541)**	2.984 (4.013)	1.751 (2.668)	2.097 (2.367)	0.500	0.733	0.500	0.685
MOOSONEE	**1.822 (1.564)**	1.980 (2.541)	3.283 (4.013)	2.089 (2.623)	2.771 (2.405)	0.651	0.730	0.540	0.694
TIMMINS	1.727 (1.574)	1.653 (2.541)	2.755 (4.013)	**1.639 (3.061)**	2.201 (2.425)	0.533	0.644	0.467	0.605
MADAWASKA	2.097 (1.571)	2.301 (2.541)	2.431 (4.013)	2.111 (2.696)	**1.749 (2.098)**	0.644	0.589	0.512	0.450
NORTH BAY	1.577 (1.603)	**1.508 (2.541)**	2.282 (4.013)	1.622 (2.808)	1.967 (1.857)	0.556	0.622	0.533	0.585
GORE BAY	**1.486 (1.574)**	1.524 (2.541)	2.366 (4.013)	1.979 (2.679)	1.811 (2.035)	0.524	0.587	0.698	0.628
BROCKVILLE	**1.481 (1.562)**	1.513 (2.541)	1.984 (4.013)	1.711 (2.974)	1.823 (1.987)	0.528	0.604	0.540	0.580
CORNWALL	1.470 (1.594)	**1.403 (2.541)**	2.274 (4.013)	1.771 (2.908)	1.896 (2.036)	0.457	0.654	0.593	0.632
KINGSTON	**1.561 (1.576)**	1.583 (2.541)	2.214 (4.013)	1.801 (3.096)	1.626 (1.972)	0.544	0.617	0.630	0.560
OTTAWA	1.493 (1.570)	**1.388 (2.541)**	2.081 (4.013)	1.488 (2.853)	1.649 (2.105)	0.469	0.617	0.481	0.554
RIDGETOWN	**1.740 (1.569)**	1.784 (2.541)	2.608 (4.013)	2.465 (2.655)	1.869 (2.136)	0.569	0.667	0.694	0.580
VINELAND	1.598 (1.563)	**1.491 (2.541)**	2.498 (4.013)	1.838 (2.691)	1.563 (1.841)	0.429	0.746	0.540	0.510
WELLAND	1.570 (1.597)	1.503 (2.541)	2.160 (4.013)	2.042 (3.126)	**1.494 (1.924)**	0.489	0.600	0.667	0.450
WINDSOR	1.740 (1.620)	**1.547 (2.541)**	1.872 (4.013)	1.585 (2.718)	1.847 (1.975)	0.407	0.444	0.506	0.550
LONDON	**1.604 (1.578)**	1.746 (2.541)	1.915 (4.013)	1.719 (2.798)	1.820 (2.004)	0.644	0.580	0.556	0.575
WOODSTOCK	1.437 (1.491)	**1.411 (2.541)**	1.702 (4.013)	1.725 (3.003)	1.928 (1.985)	0.481	0.444	0.593	0.695
BELLEVILLE	**1.687 (1.576)**	1.755 (2.541)	2.261 (4.013)	1.704 (2.853)	2.026 (1.967)	0.519	0.556	0.593	0.610
HAMILTON	1.538 (1.577)	**1.486 (2.541)**	2.098 (4.013)	1.827 (2.662)	1.769 (1.969)	0.467	0.567	0.611	0.585
ORANGEVILLE	**1.438 (1.520)**	1.475 (2.541)	1.887 (4.013)	1.692 (2.728)	1.958 (2.154)	0.537	0.556	0.630	0.654
TORONTO	1.527 (1.595)	**1.466 (2.541)**	1.983 (4.013)	1.685 (2.614)	2.035 (1.936)	0.543	0.630	0.556	0.620
HALIBURTON	**1.651 (1.578)**	1.750 (2.541)	2.105 (4.013)	1.760 (2.890)	2.066 (2.156)	0.569	0.556	0.500	0.615
PETERBOROUGH	**1.797 (1.588)**	1.813 (2.541)	2.128 (4.013)	1.930 (3.258)	1.843 (2.079)	0.500	0.522	0.511	0.505
Average	**1.698 (1.578)**	1.710 (2.541)	2.545 (4.013)	1.915 (2.832)	1.979 (2.221)	0.547	0.629	0.561	0.593

2.7 General Remarks

In this chapter, we develop a regression model with a specific R-Vine structure to analyze longitudinal data with a time span. One of the challenge in using vine copula model to describe temporal dependence is that the number of parameters increases quadratically with the time length. Use of composite likelihood can help avoid the heavy computation and provide model robustness at the price of some loss in efficiency. Moreover, the R-Vine model can also provide a convenient prediction procedure to incorporate information from the previous time points.

In simulation studies, the parameters are shown to be consistently estimated with moderate efficiency loss using the composite likelihood procedure. In terms of prediction, the prediction results of the proposed R-Vine model under both the full likelihood and the composite likelihood have little difference, which further illustrate the advantage of the composite likelihood procedure in computation.

Chapter 3

A Bayesian Hierarchical Copula Model

3.1 Introduction

In this chapter, we are interested in the scenario with hierarchical structured data as illustrated in Figure 3.1. The nodes at the subject level represent subjects and those at the intermediate level represent clusters which form the population level in the top level. Data of this hierarchical structure arises commonly in practice. Examples include multi-center medical studies conducted at m sites, meta-analyses of m studies, spatially configured data of m locations, longitudinal data from m subjects, time series with time varying dependence structures of m periods, etc. The Bayesian hierarchical approach can adopt these complex data structure naturally, as reviewed in Section 1.6 of Chapter 1, and our interest in this chapter is to study dependence modeling under the Bayesian hierarchical framework.

To account for a more complex hierarchical structure, the three-level structure can be easily extended by including more intermediate levels. Suppose that multivariate data are collected from each subject and the dependence modeling of the subject-level multivariate structure is of interest. We propose a Bayesian hierarchical copula model (BHCM) to model the subject-level dependence by a copula-based model; and such a model accounts for the hierarchical structure by allowing random dependence parameters and specifying multiple layers of prior and hyperprior distributions. This model combines the ideas of the Bayesian hierarchical approach and the copula-based dependence modeling, and it offers us great flexibility in facilitating various association structures and carrying out inference in a straightforward manner.

The rest of the chapter is organized as follows. In Section 3.2, we describe the model formulation of the proposed BHCM. In Section 3.3, we examine issues concerning inferences, the sampling scheme, and the asymptotic properties of the resultant estimators. In Section 3.4, we discuss the selection of transformation functions and associated scaling parameters. In Section 3.5, we perform simulation studies to evaluate the finite sample performance of the proposed methods. In Section 3.6, we analyze the Vertebral Column Data (Dua and Graff, 2017) using the proposed BHCM.

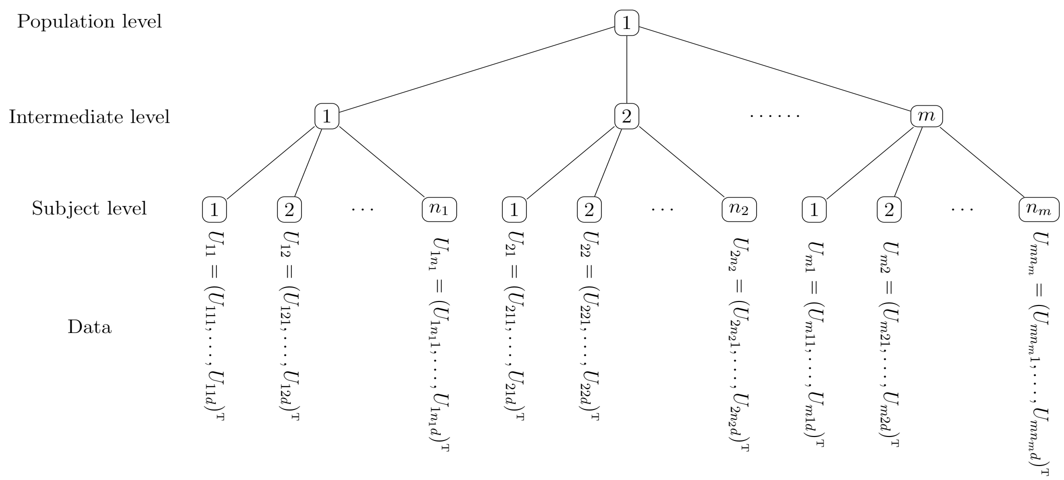


Figure 3.1: A three-level hierarchical structure

3.2 Model Formulation

We consider a three-level hierarchical structure as illustrated in Figure 3.1. The single node at the top level represents the population level. The bottom level is the subject level in which each node corresponds to the data from a subject. The intermediate level contains m clusters to which the bottom-level subjects belong. Let $U_{ji} = (U_{ji1}, \ldots, U_{jid})^{\mathrm{T}}$ be the vector of d features, which are collected from the ith subject of the jth cluster, where $i = 1, \ldots, n_j$, $j = 1, \ldots, m$, and n_j is a positive integer that may depend on j. Let $U_j = (U_{j1}^{\mathrm{T}}, \ldots, U_{jn_j}^{\mathrm{T}})^{\mathrm{T}}$ and $U = (U_1^{\mathrm{T}}, \ldots, U_m^{\mathrm{T}})^{\mathrm{T}}$. Let u_{jik}, u_{ji}, u_j and u represent the observed counterparts of U_{jik}, U_{ji}, U_j and U, respectively, for $i = 1, \ldots, n_j$, $j = 1, \ldots, m$, and $k = 1, \ldots, d$.

The copula formulation is advantageous in its separation of modeling marginal distributions and dependence structures, and much attention has been directed to modeling the dependence structures with a standard treatment of marginal distributions. Consistent with many authors (e.g. Aas et al., 2009; Okhrin et al., 2013a,b), we assume that U_{jik} follows a uniform distribution on $[0,1]$ marginally and focus on dependence modeling of the subject-level data U_{ji} using copula-based models. In Section 3.2.1, we first use a copula-based approach to model the dependence structure among the d features of each subject and allow different structures for different clusters. In Section 3.2.2, we account for the hierarchical structure and continue our discussion in the framework of Bayesian hierarchical models.

3.2.1 Copula-based Dependence Models

According to Sklar (1959), any joint cumulative distribution function (CDF) can be written as a copula function of its univariate marginal CDFs. A copula function on $[0,1]^d$, denoted by C, is defined as $C(u_1,\ldots,u_d) = P(U_1 \leq u_1,\ldots,U_d \leq u_d)$, for uniformly distributed random variables $U_1,\ldots,U_d$ on [0,1]. If the marginal distributions are all continuous, the copula C always exists and is unique. Here we assume that the joint distribution of d features in cluster j is governed by a multivariate copula function C_j. Then the joint CDF F_j of U_{ji} can be written as

$$F_j(u_{ji1},\ldots,u_{jid};\theta_j) = C_j(u_{ji1},\ldots,u_{jid};\theta_j) \tag{3.1}$$

for $i = 1,\ldots,n_j$, where $\theta_j = (\theta_{j1},\ldots,\theta_{jp_j})^{\mathrm{T}}$ is a vector of parameters indexing the copula function C_j, p_j is the number of parameters, and $j = 1,\ldots,m$. Let $\theta = (\theta_1^{\mathrm{T}},\ldots,\theta_m^{\mathrm{T}})^{\mathrm{T}}$ denote the vector of all copula parameters. Common choices of multivariate copula C_j include multivariate Gaussian copula and multivariate t-copula from the elliptical copula family (Frahm et al., 2003), and multivariate Clayton, Frank and Gumbel copulas from the Archimedean copula family (Genest and MacKay, 1986a,b). Copula functions in the Archimedean family contain only one parameter, while those in the elliptical family may contain multiple parameters. Let f_j and c_j denote the density functions corresponding to F_j and C_j, respectively, for $j = 1,\ldots,m$.

3.2.2 Bayesian Hierarchical Models

We construct a Bayesian hierarchical model to account for the 3-level hierarchical structure as illustrated in Figure 3.1 through the following 3-stage specifications of prior and hyperprior distributions (Gustafson et al., 2006; Lindley and Smith, 1972). The first stage of the hierarchical model facilitates the vector $U_{ji} = (U_{ji1}, \ldots, U_{jid})^{\mathrm{T}}$ by a copula-based dependence model as described in Section 3.2.1, where θ_j is of dimension p_j. As we allow the dependence structures to be distinct and governed by the functions across clusters, the association parameters θ_j may have different ranges for $j = 1, \ldots, m$. Before we specify a prior distribution for θ_j, we map each component θ_{jl} of θ_j into the range $\mathbb{R}$ through a proper transformation. A natural way of reparameterizing the parameters θ_{jl} is to invoke the Kendall's τ, together with the Fisher z-transformation (Schamberger et al., 2017), and this is especially the case when there is an explicit expression of Kendall's τ. In the development here, we take an alternative by writing $\gamma_{jl} = \alpha_{jl} g_{jl}(\theta_{jl})$ for $l = 1, \ldots, p_j$ and $j = 1, \ldots, m$, where the transformation function $g(\cdot)$ is a monotonic function mapping the parameter space, $\mathcal{A}$, of the dependence parameter θ to $\mathbb{R}$, and α_{jl} is a non-zero scaling parameter, whose inclusion helps reflect the magnitude of the variability across clusters.

The form of the transformation functions and the rational behind rescaling are discussed in details in Section 3.4. Let $\gamma_j = (\gamma_{j1}, \ldots, \gamma_{jp_j})^{\mathrm{T}}$ denote the vector of transformed and scaled dependence parameters in cluster j, and let $\gamma = (\gamma_1^{\mathrm{T}}, \ldots, \gamma_m^{\mathrm{T}})^{\mathrm{T}}$.

At the second stage of the hierarchical model, we specify the prior distribution for the parameters γ_{jl} as

$$\gamma_{jl}|(\mu_{jl}, \sigma_{jl}) \sim N(\mu_{jl}, \sigma_{jl}^2), \tag{3.2}$$

where μ_{jl} and σ_{jl} indicate the cluster location and variability of γ_{jl}, respectively, for $l = 1, \ldots, p_j$ and $j = 1, \ldots, m$. Let $\mu_j = (\mu_{j1}, \ldots, \mu_{jp_j})^{\mathrm{T}}$ be the vector of mean parameters, let $\sigma_j = (\sigma_{j1}, \ldots, \sigma_{jp_j})^{\mathrm{T}}$ be a vector of standard deviations (s.d.) of the jth cluster, and let $\mu = (\mu_1^{\mathrm{T}}, \ldots, \mu_m^{\mathrm{T}})^{\mathrm{T}}$ and $\sigma = (\sigma_1^{\mathrm{T}}, \ldots, \sigma_m^{\mathrm{T}})^{\mathrm{T}}$. We further specify the prior distributions for cluster-level location parameters μ_{jl} as

$$\mu_{jl}|(\varphi_l, \delta_l) \sim N(\varphi_l, \delta_l^2), \tag{3.3}$$

and the hyperprior distributions for cluster-level variability parameters σ_{jl} as

$$\sigma_{jl} \sim \pi_\sigma,$$

for $l = 1, \ldots, p_j$ and $j = 1, \ldots, m$, where φ_l and δ_l indicate the population location and variability of μ_{jl} and π_σ is the prior distribution of σ_{jl}.

Let $\varphi = (\varphi_1, \ldots, \varphi_{p^*})^{\mathrm{T}}$ and $\delta = (\delta_1, \ldots, \delta_{p^*})^{\mathrm{T}}$, where $p^* = \max(p_1, \ldots, p_m)$. This stage characterizes the cluster-level parameters, which corresponds to the intermediate level of the hierarchical structure in Figure 3.1.

At the third stage, we specify the hyperprior distribution for the population-level parameters φ and δ as

$$\varphi_l \sim \pi_\varphi \quad \text{and} \quad \delta_l \sim \pi_\delta \tag{3.4}$$

for $l = 1, \dots, p^*$, where π_φ and π_δ are prior distributions for φ_l and δ_l, respectively.

Combining (3.2) and (3.3) gives

$$\gamma_{jl}|(\varphi_l, \delta_l, \sigma_{jl}) \sim N(\varphi_l, \sigma_{jl}^2 + \delta_l^2) \tag{3.5}$$

for $l = 1, \dots, p_j$ and $j = 1, \dots, m$, where the variance of γ_{jl} includes the within-cluster variability σ_{jl}^2 and between-cluster variability δ_l^2.

For parameters $(\varphi^{\mathrm{T}}, \delta^{\mathrm{T}})^{\mathrm{T}}$ at the population level and σ_j at the cluster level, we select a weak-informative prior, such as an Inverse Gamma$(\varepsilon, \varepsilon)$ with small ε, or a non-informative prior, such as an improper uniform prior (Jeffreys, 1946). For the construction of the Bayesian hierarchical model, we assume exchangeability for all levels of specification.

3.3 Bayesian Inference

Here we aim to make Bayesian inference for the vector of the dependence parameters $\theta = (\theta_1^{\mathrm{T}}, \dots, \theta_m^{\mathrm{T}})^{\mathrm{T}}$. Since we have worked with the transformed and scaled dependence parameters γ in Section 3.2.2, we will continue our discussion in terms of γ and transform them back to their original scale θ. We first consider the posterior distribution of γ

$$f(\gamma|u) \propto f(u|\gamma)f(\gamma),$$

where $f(u|\gamma)$ stands for the copula density function with the data u and the transformed parameters γ specified as in Section 3.2.1, and $f(\gamma)$ is the prior distribution of γ, given as in Section 3.2.2. The distribution of $f(\gamma)$ can be obtained by integrating the joint distribution of $f(\gamma, \sigma, \varphi, \delta)$ with respect to σ, φ and δ, where $f(\gamma, \sigma, \varphi, \delta)$ is determined by $f(\gamma|\sigma, \varphi, \delta)\pi(\sigma)\pi(\varphi)\pi(\delta)$. This calculation involves integration of dimension $\sum_{j=1}^m p_j + 2p^*$, which is generally difficult to implement. To overcome this difficulty, we employ an alternative strategy and sample from the joint posterior distribution $f(\gamma, \sigma, \varphi, \delta|u)$. The posterior distributions that are used in the sampling algorithm is provided in Section 3.3.1 and sampling algorithm is introduced in Section 3.3.2.

3.3.1 Posterior Distributions

We start with the joint posterior distribution of $(\gamma^{\mathrm{T}}, \sigma^{\mathrm{T}}, \varphi^{\mathrm{T}}, \delta^{\mathrm{T}})^{\mathrm{T}}$,

$$\begin{aligned} f(\gamma, \sigma, \varphi, \delta | u) &\propto f(u|\gamma) f(\gamma|\sigma, \varphi, \delta)\pi(\sigma)\pi(\varphi)\pi(\delta) \\ &= \prod_{j=1}^{m} \left[\prod_{i=1}^{n_j} f_j(u_{ji}; \gamma_j) \prod_{l=1}^{p_j} \phi(\gamma_{jl}|\varphi_l, \sigma_{jl}^2 + \delta_l^2) \right] \pi_\sigma \pi_\varphi \pi_\delta, \end{aligned} \tag{3.6}$$

where $\phi(\cdot|a, b^2)$ is the density function of the normal distribution with mean a and variance b^2.

The joint posterior distribution of $(\varphi^{\mathrm{T}}, \delta^{\mathrm{T}}, \sigma^{\mathrm{T}})^{\mathrm{T}}$ can be obtained by integrating (3.6) with respect to γ,

$$\begin{aligned} f(\sigma, \varphi, \delta | u) &= \int f(\gamma, \sigma, \varphi, \delta | u) d\gamma \\ &= \prod_{j=1}^{m} \int \prod_{i=1}^{n_j} f_j(u_{ji}; \gamma_j) \prod_{l=1}^{p_j} \phi(\gamma_{jl}|\varphi_l, \sigma_{jl}^2 + \delta_l^2) \pi_\sigma \pi_\varphi \pi_\delta d\gamma_j. \end{aligned} \tag{3.7}$$

Finally, the conditional posterior distribution of parameters γ_j, given the all hyperprior parameters and $\gamma_{(-j)} = (\gamma_1^{\mathrm{T}}, \ldots, \gamma_{j-1}^{\mathrm{T}}, \gamma_{j+1}^{\mathrm{T}}, \ldots, \gamma_m^{\mathrm{T}})$, is of the form

$$\begin{aligned} f(\gamma_j|\sigma, \varphi, \delta, \gamma_{(-j)}, u) &= f(\gamma_j|\sigma, \varphi, \delta, u) \\ &\propto f(u_j|\gamma_j) f(\gamma_j|\varphi, \delta, \sigma) \\ &= \prod_{i=1}^{n_j} f_j(u_{ji}; \gamma_j) \prod_{l=1}^{p_j} \phi(\gamma_{jl}|\varphi_l, \sigma_{jl}^2 + \delta_l^2), \end{aligned} \tag{3.8}$$

where the first equality comes from that given $(\varphi^{\mathrm{T}}, \delta^{\mathrm{T}}, \sigma^{\mathrm{T}})^{\mathrm{T}}$, γ_j is independent of $\gamma_{(-j)}$.

3.3.2 Sampling Scheme

To utilize the joint posterior distribution $f(\gamma, \sigma, \varphi, \delta|u)$ in (3.6), we let $\zeta = (\gamma^{\mathrm{T}}, \sigma^{\mathrm{T}}, \varphi^{\mathrm{T}}, \delta^{\mathrm{T}})^{\mathrm{T}}$ denote the vector of all the parameters. The Metropolis-Hasting (M-H) algorithm (Metropolis et al., 1953; Hastings, 1970) can be employed, in principle, to sample from $f(\zeta|u)$ directly. In the instance with a high dimensional ζ, directly applying M-H algorithm to the joint posterior distribution (3.6) is challenging because it is not always straightforward to choose an appropriate proposal density function and tune the parameters in the proposal density to get a good acceptance rate, and therefore the M-H can be inefficient or not even converge. Directly invoking a Gibbs sampler (Geman and Geman, 1987; Gelman et al., 2013) to (3.6) is not a valid option here, since the conditional distribution of hyper-parameters does not depend on the data, i.e.,

$$f(\sigma, \varphi, \delta | \gamma^{(t-1)}, u) \propto f(\gamma^{(t-1)} | \sigma, \varphi, \delta) \pi_\sigma \pi_\varphi \pi_\delta.$$

To cope with the issue, we consider the following "layer by layer" sampling procedure.

1. Sample hyperprior parameters $(\sigma^{\mathrm{T}}, \varphi^{\mathrm{T}}, \delta^{\mathrm{T}})^{\mathrm{T}}$ from the posterior distribution $f(\sigma, \varphi, \delta | u)$ in (3.7) using the M-H algorithm.

2. Calculate the sample means of the sampled vectors in Step 1 as Bayesian estimates for σ, φ, and δ, denoted by $\hat{\sigma}$, $\hat{\varphi}$, and $\hat{\delta}$, respectively.

3. Sample parameters γ_j from the conditional posterior distribution $f(\gamma_j | \hat{\sigma}, \hat{\varphi}, \hat{\delta}, u)$ in (3.8) with the Bayesian estimates for the hyperprior parameters obtained from Step 2 plugged in. Applying the M-H algorithm to $f(\gamma_j | \hat{\sigma}, \hat{\varphi}, \hat{\delta}, u)$. Repeat this step for $j = 1, \ldots, m$.

4. Transform $\gamma_{jl}^{(t)}$ back to obtain $\theta_{jl}^{(t)}$ through a division by α_{jl} and the inverse transformation function $g_{jl}^{-1}(\cdot)$, for $l = 1, \ldots, p_j$, $j = 1, \ldots, m$, and $t = 1, \ldots, N$.

5. Compute the quantities of interest that are related to the parameters θ_{jl}, such as the posterior mean.

In Steps 1 and 3, we apply the random walk Metropolis algorithm, of which the proposal distribution is a normal distribution with mean determined as the sampled value from the previous iteration of the M-H algorithm. Besides normal distribution, other distributions can also be considered for proposal distributions. For variance parameters, σ and δ, a truncated normal or a Gamma distributions can be good options as well Gelman et al. (2013). If a range $[a, b]$ of each parameter can be determined beforehand, a truncated normal proposal can stabilize performance of the sampling procedure when the dependence is extremely strong. Schamberger et al. (2017) and Schepsmeier et al. (2018) contain some guidelines on determining the ranges for copula parameters.

In situations where the dimension of the parameters $(\sigma^{\mathrm{T}}, \varphi^{\mathrm{T}}, \delta^{\mathrm{T}})^{\mathrm{T}}$ is high and/or the convergence of the sampling algorithm is a concern, one may adopt a Gibbs Sampler

(Geman and Geman, 1987; Gelman et al., 2013) in Step 1 and further decompose the joint posterior distribution (3.7) in the tth iteration as

$$\begin{aligned}
&f(\sigma_j|\sigma_{(-j)}^{(t-1)}, \varphi^{(t-1)}, \delta^{(t-1)}, u) = f(\sigma_j|\varphi^{(t-1)}, \delta^{(t-1)}, u) \\
&\propto \int \prod_{i=1}^{n_j} f_j(u_{ji}; \gamma_j) \prod_{l=1}^{p_j} \phi(\gamma_{jl}|\varphi_l^{(t-1)}, \sigma_{jl}^2 + (\delta_l^{(t-1)})^2)\pi_{\sigma_j}\pi_\varphi\pi_\delta d\gamma_j, \\
&f(\varphi|\sigma^{(t)}, \delta^{(t-1)}, u) \propto f(\sigma^{(t)}, \varphi, \delta^{(t-1)}|u), \\
&f(\delta|\sigma^{(t)}, \varphi^{(t-1)}, u) \propto f(\sigma^{(t)}, \varphi^{(t-1)}, \delta|u),
\end{aligned} \tag{3.9}$$

where $\sigma_{(-j)} = (\sigma_1^{\mathrm{T}}, \ldots, \sigma_{j-1}^{\mathrm{T}}, \sigma_{j+1}^{\mathrm{T}}, \ldots, \sigma_m^{\mathrm{T}})^{\mathrm{T}}$, for $j = 1, \ldots, m$. Instead of sampling from the joint posterior (3.7), sampling from each of the conditional distributions in (3.9) improves the sampling efficiency in the sense that it facilitates a lower rejection rate yet a larger effective sample size. This gain is at the price of increasing the computation time which is basically caused by the calculation of the integration over γ.

While a large dimension of γ can considerably increase the computation time of the sampling procedure, Step 3 of the sampling procedure does not require an appreciable computation time, as the sampling from (3.8) is conducted within each cluster j which does not involve any integration. Although most of the computation time is consumed by Step 1 for the case with a large number of parameters, applications of our sampling algorithm are still feasible, because the most frequently-used copulas from Archimedean and Extreme-value families contain one or two parameters; even for copulas from the Elliptical family, such as Gaussian copula, which contain a high dimension of parameters, it is often common to impose certain correlation structures to the copula to facilitate a parsimonious model.

The evaluation of posterior density distribution in (3.7) involves the integrals which generally do not have an analytically close form. To handle this issue, we suggest to use the random walk Metropolis algorithm (Gilks et al., 1995) instead of the MCMC algorithms which require the gradient of the posterior distribution, such as Langevin MCMC or Hamiltonian MC (Radford et al., 2010).

3.3.3 Asymptotic Properties

The asymptotic properties of posterior distributions in Bayesian theory have been thoroughly discussed, see, for example, LeCam (1953), DeGroot (2005) and Shen and Wasserman (2001). We consider the posterior distribution of γ_j taking the form,

$$f(\gamma_j|u_j) \propto \left[\prod_{i=1}^{n_j} f(u_{ji}|\gamma_j)\right] f(\gamma_j), \tag{3.10}$$

where $f(\gamma_j)$ is the marginal prior distribution of parameters γ_j, and $f(u_{ji}|\gamma_j)$ is the density function of the data u_{ji} and the parameter γ_j. If we let $f_T(u_{ji})$ denote the true distribution of U_{ji}, we can define the *Kullback-Leibler divergence* (K-L) at γ_j as,

$$\mathrm{KL}(\gamma_j) = \int \log\left(\frac{f_T(u_{ji})}{f(u_{ji}|\gamma_j)}\right) f_T(u_{ji}) du_{ji}, \tag{3.11}$$

to quantify the discrepancy between the model distribution $f(u_{ji}|\gamma_j)$ and the true distribution $f_T(u_{ji})$. The value that minimizes the K-L divergence is labeled as $\gamma_j^\dagger$.

Under certain regularity conditions, we have the following asymptotic results for the posterior distribution (see, for example, Gelman et al. (2013)):

- Consistency: For every cluster $j = 1, \ldots, m$, the probability over any given neighborhood of $\gamma_j^\dagger$ under the posterior distribution $f(\gamma_j|u_{j1}, \ldots, u_{jn_j})$ converges to 1 as $n_j \to \infty$.
- Asymptotic Normality: As $n_j \to \infty$, the posterior distribution $f(\gamma_j|u_{j1}, \ldots, u_{jn_j})$ approaches the normal distribution with mean $\gamma_j^\dagger$ and covariance matrix $J(\gamma_j^\dagger)^{-1}$, where $J(\cdot)$ is the Fisher information function defined as (Gelman et al., 2013)

$$J(\gamma_j) = -n_j E\left(\frac{\partial^2 \log f(U_{ji}|\gamma_j)}{\partial \gamma_j \partial \gamma_j^{\mathrm{T}}}\middle|\gamma_j\right),$$

  where the conditional expectation is taken with respect to U_j.

3.4 Transformation of the Dependence Parameters

3.4.1 Transformation Function

In this subsection, we discuss the selection of the transformation function $g(\cdot)$, which is a monotonic function mapping $\mathcal{A}$ to $\mathbb{R}$, where $\mathcal{A}$ is the parameter space for the dependence parameter θ. In Table 3.1, we give examples of transformation functions for some commonly-used copula functions, where L and U are the lower and upper bounds of $\mathcal{A}$, respectively.

Table 3.1: Transformation functions for copula parameters

$\mathcal{A}$	Example of Copula Function	Tranformation Function
$[L, U]$	Gaussian Copula	$g(x) = \log\left(\frac{x-L}{U-x}\right)$
$[L, \infty)$	Clayton Copula	$g(x) = \log(x - L)$
$(-\infty, U]$	Rotated Clayton Copula	$g(x) = \log(U - x)$
$(-\infty, \infty) \setminus \{0\}$	Frank Copula	$g(x) = x$

For copula functions with an infinite range, we can impose a certain finite range $[L^*, U^*]$ and use the transformation function $g(x) = \log\left(\frac{x-L^*}{U^*-x}\right)$. For example, for the Frank copula, we may impose the range $[-100, 100]$ to cover the Kendall's τ from -0.96 to 0.96. In simulation section, we compare the identity transformation function and the logit transformation function with end points as $[-100, 100]$ for the Frank copula.

3.4.2 Choice of Scaling Parameter

In this subsection, we discuss the choice of scaling parameter α_{jl}. First, we define $\gamma^*_{jl} = g_{jl}(\theta_{jl})$ as the dependence parameter mapped into $\mathbb{R}$ without scaling and write $\gamma^* = (\gamma^*_{j1}, \ldots, \gamma^*_{jp_j})^{\mathrm{T}}$. Then the scaled and unscaled parameters have the relationship $\gamma_{jl} = \alpha_{jl}\gamma^*_{jl}$, for $l = 1, \ldots, p_j$ and $j = 1, \ldots, m$.

We impose a normal prior on γ_{jl} in Section 3.2.2 in the form of

$$\gamma_{jl} \sim N(\mu_{jl}, \sigma^2_{jl}),$$

and further impose a normal prior on the cluster mean μ_{jl} as

$$\mu_{jl} \sim N(\varphi_l, \delta^2_l),$$

which is equivalent to imposing a normal prior on γ^*_{jl} of the form

$$\gamma^*_{jl} \sim N\left(\frac{\mu_{jl}}{\alpha_{jl}}, \frac{\sigma^2_{jl}}{\alpha^2_{jl}}\right),$$

together with the prior distribution for cluster mean

$$\frac{\mu_{jl}}{\alpha_{jl}} \sim N\left(\frac{\varphi_l}{\alpha_{jl}}, \frac{\delta_l^2}{\alpha_{jl}^2}\right).$$

As $|\alpha_{jl}|$ gets larger, both the within-cluster and between-cluster variances assumed in the prior distributions become smaller. In other words, as $|\alpha_{jl}|$ increases, we impose a stronger prior on γ_{jl}^*.

Next we describe a method of choosing suitable values of the α_{jl}. Suppose that we obtain the maximum likelihood estimate (MLE) of γ_j^*, denoted by $\tilde{\gamma}_j^*$, by maximizing the likelihood function

$$L(\gamma_j^*|u_j) = \prod_{i=1}^{n_j} c_j(u_{ji}|\gamma_j^*).$$

The asymptotic covariance matrix of $\tilde{\gamma}_j^*$ can be estimated by $I^{-1}(\tilde{\gamma}_j^*)$, where $I(\tilde{\gamma}_j^*)$ is the observed information matrix

$$I(\tilde{\gamma}_j^*) = -\frac{\partial^2}{\partial\gamma_j^*\partial\gamma_j^{*\mathrm{T}}} \log L(\gamma_j^*|u_j)\bigg|_{\gamma_j^*=\tilde{\gamma}_j^*}.$$

Let $\widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}^*)$ denote the estimated asymptotic standard deviation of $\tilde{\gamma}_{jl}^*$, which is calculated as the square root of the lth diagonal element of $I^{-1}(\tilde{\gamma}_j^*)$. By the invariance property of MLE, the MLE of $\gamma_{jl} = \alpha_{jl}\gamma_{jl}^*$, denoted by $\tilde{\gamma}_{jl}$, is $\alpha_{jl}\tilde{\gamma}_{jl}^*$, and its estimated asymptotic s.d. is $\widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}) = |\alpha_{jl}|\,\widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}^*)$. We aim to choose the α_{jl} such that resultant 95% confidence intervals of the $\tilde{\gamma}_{jl}$ are of the same length, say, L, for all $l = 1, \ldots, p_j$ and $j = 1, \ldots, m$, where $L = 2 \times 1.96 \times \widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}) = 2 \times 1.96 \times |\alpha_{jl}| \times \widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}^*)$. Therefore, we set

$$\alpha_{jl} = \frac{L}{3.92 \times \widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}^*)} \times \mathrm{sign}(\tilde{\gamma}_{jl}^*),$$

which has the same sign as $\tilde{\gamma}_{jl}^*$; α_{jl} is the ratio of the target width of a 95% confidence interval of $\tilde{\gamma}_{jl}$ to the width of the 95% confidence interval of $\tilde{\gamma}_{jl}^*$. Consequently, the within-cluster mean can be approximated by

$$\tilde{\gamma}_{jl} = \alpha_{jl}\tilde{\gamma}_{jl}^* = \mathrm{sign}(\tilde{\gamma}_{jl}^*)\frac{L}{3.92 \times \widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}^*)}\tilde{\gamma}_{jl}^*,$$

and the within-cluster s.d. can be approximated by

$$\widehat{\mathrm{sd}}(\tilde{\gamma}_{jl}) = |\alpha_{jl}|\,\widehat{\mathrm{sd}}(\tilde{\gamma}^*_{jl}) = \frac{L}{3.92}, \tag{3.12}$$

a constant value shared by all clusters. The population mean can be approximated by

$$\bar{\gamma}_l := \frac{1}{m}\sum_{j=1}^{m}\tilde{\gamma}_{jl} = \sum_{j=1}^{m}\mathrm{sign}(\tilde{\gamma}^*_{jl})\frac{L}{m \times 3.92 \times \widehat{\mathrm{sd}}(\tilde{\gamma}^*_{jl})}\tilde{\gamma}^*_{jl}, \tag{3.13}$$

and the between-cluster s.d. can be approximated by

$$\frac{1}{m-1}\sum_{j=1}^{m}(\tilde{\gamma}_{jl} - \bar{\gamma}_l)^2 = \frac{1}{m-1}\left[\sum_{j=1}^{m}\alpha^2_{jl}(\tilde{\gamma}^*_{jl})^2 - m\bar{\gamma}^2_{jl}\right]. \tag{3.14}$$

Scaling the transformed dependence parameters has the following effects. First, it standardizes how much the subjects within the same cluster vary from the cluster mean. As we derive in (3.12), all clusters share the same within-cluster s.d.. Secondly, the population mean in (3.13) can be viewed as a weighted average of the unscaled γ^*_{jl}'s. If a cluster has a larger within-cluster variability in terms of γ^*_{jl}, which has things to do with the sample size, the shape of the copula function and the true parameter value (see Appendix B.1 for a detailed discussion), a smaller weight is then assigned to this cluster. Therefore, the population mean will be less affected by the clusters with large variabilities and then becomes more stable. The same argument applies to the calculation of between-cluster variance in (3.14). Thirdly, the term $\mathrm{sign}(\tilde{\gamma}^*_{jl})$ in α_{jl} makes sure that all estimates of scaled parameters are positive, which reduces the between-cluster variability. Based on the simulation results in Section 3.5, we suggest to use $L = 4$ as "a rule of thumb" to avoid an overwhelmingly strong or weak prior distribution.

3.5 Simulation Studies

In this section, we conduct simulation studies to examine the finite sample performance of the Bayesian estimators of the dependence parameter θ under the proposed BHCM; the examination is taken in contrast to the performance of the likelihood-based estimators, conventional estimators for the parameters of copula models. Though the interpretation for Bayesian and likelihood estimators is not the same, such comparisons can shed lights on the performance of our proposed BHCM, because with the noninformative priors for the parameters, the Bayesian estimators would be numerically close to the likelihood estimators.

3.5.1 Simulation Settings

We consider a three-level hierarchical structure with $m = 4$ clusters at the intermediate level, and the sample size is taken as $n = 200$ or 400. A vine copula structure (Bedford and Cooke, 2002; Aas et al., 2009) is utilized to simulate dependent hierarchical data. While various dependence structures can be obtained by choosing different types of vines, changing the order of the nodes in the vine structure, and adopting different bivariate copulas on different levels of the vine structure, here we generate data from a D-Vine copula structure as illustrated in Figure 3.2, where the bivariate copulas in vine structure higher than level 1 are all assumed to be independent. In Figure 3.2, the dependence strength between U_{ji1} and U_{ji2} is of interest. The bivariate copula between U_{1i2} and U_{2i1} is the connecting structure between clusters 1 and 2. Similarly, $C(u_{2i2}, u_{3i1})$ connects clusters 2 and 3, and $C(u_{3i2}, u_{4i1})$ connects clusters 3 and 4.

We consider five simulation settings. The copula forms and the parameter values are summarized in Table 3.2. Settings 3.1 and 3.2 have the same copula forms for different clusters, and Settings 3.3, 3.4 and 3.5 allow different dependence structures. In Settings 3.1 and 3.3, the difference between the strength of dependence is moderate across clusters, while the difference is more obvious in Settings 3.2, 3.4 and 3.5. To demonstrate the capability of our proposed BHCM in handling the setting with multiple copula parameters, in Setting 3.5, we further consider copulas with a single parameter in clusters 1 and 2 and copulas with two parameters in clusters 3 and 4. A moderate dependence between clusters is introduced in all settings and the linking copulas are set to be Gaussian(0.71).

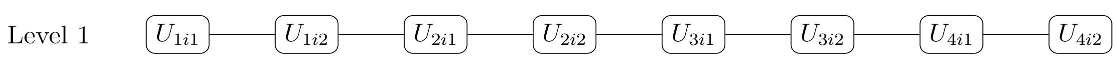


Figure 3.2: The top level of a D-Vine structure

Table 3.2: Simulation settings: copula forms and parameters

	Setting 3.1	τ[1]	Setting 3.2	τ	Setting 3.3	τ	Setting 3.4	τ	Setting 3.5	τ
Cluster 1	Clayton(1.33)	0.40	Clayton(1.33)	0.40	Clayton(3.00)	0.60	Clayton(3.00)	0.60	Gumbel(2.50)	0.60
Cluster 2	Clayton(1.64)	0.45	Clayton(2.00)	0.50	Gumbel(2.50)	0.60	Gumbel(4.00)	0.75	Joe(2.50)	0.45
Cluster 3	Clayton(2.00)	0.50	Clayton(3.00)	0.60	Gaussian(0.81)	0.60	Gaussian(0.60)	0.41	BB1(5.00,3.00)[2]	0.90
Cluster 4	Clayton(2.44)	0.55	Clayton(4.67)	0.70	Frank(7.93)	0.60	Frank(13.00)	0.73	BB7(3.00,5.00)[3]	0.73
Between-cluster	Gaussian(0.71)	0.50	Gaussian(0.71)	0.50	Gaussian(0.71)	0.50	Gaussian(0.71)	0.50	Gaussian(0.71)	0.50

[1] Kendall's τ
[2] Clayton-Gumbel Copula
[3] Joe-Clayton Copula

We construct the following BHCM. For $i = 1, \ldots, n$, $j = 1, 2, 3, 4$ and $l = 1, 2$ (for setting 3.5), assume that

$$
\begin{aligned}
U_{ji} = (U_{ji1}, U_{ji2}) &\sim C_j(u_{ji1}, u_{ji2}; \theta_j) \\
\gamma_{jl} &= \alpha_{jl} g_{jl}(\theta_{jl}), \\
\gamma_{jl}|\mu_{jl}, \sigma_{jl} &\sim N(\mu_{jl}, \sigma_{jl}^2), \\
\mu_{jl}|\varphi_l, \delta_l &\sim N(\varphi_l, \delta_l^2),
\end{aligned}
$$

and all the hyperprior parameters have non-informative uniform priors. Sampling $N = 6000$ from the posterior distribution and setting the Normal density with mean $\zeta^{(t-1)}$ and variance as the stepsize, as the proposal density $q(\zeta'|\zeta^{(t-1)})$, we use the M-H algorithm and the layer-by-layer sampling strategy described in Section 3.3.2 to sample θ. The posterior sample mean is used as the point Bayesian estimators for the parameters. In comparison, we also obtain the MLE of θ by maximizing the likelihood function

$$
L(\theta) = \prod_{i=1}^{n} \left[\prod_{j=1}^{4} c_j(u_{ji1}, u_{ji2}; \theta_j) \cdot \prod_{k=1}^{3} c_{k,k+1}(u_{ki2}, u_{k+1,i1}) \right],
$$

where c_j is the copula density governing the subject-dependence within cluster j for $j = 1, \ldots, 4$, and $c_{k,k+1}$ denotes the copula densities that connect between clusters for $k = 1, 2, 3$.

While the sampling algorithm is implemented on the `R` platform, we handle the integrals in the posterior distribution (3.7) by employing C++ through Monte Carlo approximations of size 15000, which is computationally fast yet the resulting approximation is fairly accurate. Simulations are repeated 200 times for each setting.

3.5.2 Evaluation Metrics

We use the following metrics to evaluate the Bayesian estimators and MLEs.

1. *Empirical Bias (EBias)*: The EBias is calculated as the average of the point estimates obtained from 200 simulations subtracting the true parameter values.

2. *Empirical Standard Error (ESE)*: The sample standard deviation of the 200 estimates.

3. *Asymptotic Standard Error (ASE)*: The average of the estimated asymptotic standard deviations obtained from the 200 simulations. The estimated asymptotic s.d. for a Bayesian estimator is calculated as the sample s.d. of the sampled sequence, and that of a maximum likelihood estimator is calculated from the inversion of the observed information matrix.

4. *95% Interval*: Left and right endpoints of an equal-tailed 95% Bayesian credible interval are computed as the 2.5th percentile and the 97.5th percentile of a sampled sequence, respectively. A 95% confidence interval for the MLE is computed by MLE$\pm$1.96$\times$the estimated asymptotic s.d.. 95% Interval is computed by averaging the left and right endpoints of 200 simulations (Chen and Shao, 1999).

5. *Empirical Coverage Probability (ECP)*: ECP is the percentage of the 95% credible intervals or 95% confidence intervals that contain the true value of the parameter out of 200 simulations.

3.5.3 Simulation Results

We summarize the simulation results for Setting 3.5 in Table 3.3, and those for Settings 3.1-3.4 in Tables B.2-B.5 in the Appendix.

Table 3.3: Simulation results for Setting 3.5

				n=200					n=400				
Cluster	Copula	Parameter	L	Ebias	ESE	ASE	95% interval	ECP	Ebias	ESE	ASE	95% interval	ECP
						Bayesian Estimation							
1	Gumbel	2.5	4	0.020	0.150	0.133	(2.264,2.789)	0.940	0.003	0.093	0.095	(2.321,2.694)	0.950
2	Joe	2.5	4	-0.029	0.166	0.158	(2.172,2.795)	0.950	-0.024	0.118	0.111	(2.263,2.701)	0.940
3	BB1	5.0	4	0.111	0.617	0.478	(4.116,5.998)	0.920	-0.141	0.383	0.316	(4.227,5.474)	0.925
		3.0	4	0.083	0.247	0.270	(2.552,3.617)	0.970	0.021	0.172	0.171	(2.689,3.364)	0.960
4	BB7	3.0	4	0.005	0.279	0.253	(2.551,3.550)	0.940	0.043	0.232	0.185	(2.717,3.403)	0.905
		5.0	4	0.038	0.486	0.497	4.079,6.036)	0.970	0.021	0.364	0.347	(4.359,5.727)	0.940
1	Gumbel	2.5	20	-0.037	0.176	0.113	(2.249,2.694)	0.820	-0.015	0.087	0.072	(2.346,2.631)	0.870
2	Joe	2.5	20	-0.075	0.191	0.133	(2.175,2.699)	0.845	-0.013	0.111	0.090	(2.315,2.669)	0.910
3	BB1	5.0	20	0.091	0.835	0.407	(4.207,5.811)	0.765	-0.152	0.321	0.218	(4.401,5.262)	0.820
		3.0	20	0.048	0.233	0.208	(2.647,3.469)	0.890	0.021	0.109	0.083	(2.860,3.188)	0.900
4	BB7	3.0	20	0.042	0.442	0.228	(2.632,3.529)	0.780	0.113	0.198	0.126	(2.872,3.369)	0.810
		5.0	20	0.051	0.470	0.451	(4.178,5.956)	0.950	0.053	0.322	0.250	(4.565,5.552)	0.920
						Maximum Likelihood Estimation							
1	Gumbel	2.5	-	0.024	0.141	0.147	(2.236,2.812)	0.960	0.020	0.106	0.104	(2.317,2.723)	0.950
2	Joe	2.5	-	0.036	0.169	0.178	(2.187,2.885)	0.960	0.030	0.126	0.131	(2.284,2.776)	0.940
3	BB1	5.0	-	-0.373	0.541	0.863	(2.936,6.318)	0.940	-0.289	0.448	0.627	(3.483,5.940)	0.930
		3.0	-	0.234	0.365	0.472	(2.309,4.159)	0.960	0.178	0.300	0.332	2.527,3.830)	0.930
4	BB7	3.0	-	0.060	0.285	0.296	(2.479,3.640)	0.930	0.062	0.252	0.209	(2.654,3.471)	0.910
		5.0	-	0.072	0.520	0.547	(4.005,6.149)	0.980	0.060	0.389	0.384	(4.308,5.812)	0.920

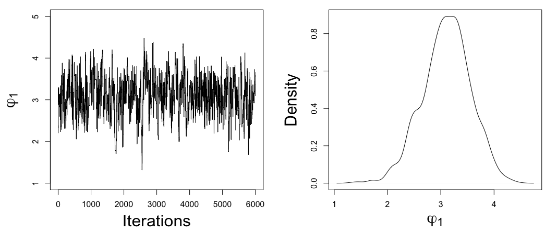
The findings for all the settings reveal the consistent patterns, as commented below. We tune L, the target length of a 95% confidence interval of $\tilde{\gamma}_{jl}$, to be 1, 4, 10 and 20 for comparison (results for $L = 1$ and 10 not shown). For the point estimates of the copula parameters under all simulation settings, the EBias of estimates obtained from the proposed BHCM are compatible with or smaller than those from the likelihood-based estimates. The Bayesian estimators with $L = 1$ have similar ESE's and ASE's to those of the likelihood-based estimates; the standard error of the Bayesian estimators gets smaller, as L gets larger. For interval estimates of the copula parameters, 95% Bayesian credible interval of the proposed BHCM are shorter than the likelihood-based 95% intervals when L is set to be 4, 10 or 20. When L is set to be a large number, there are unignorable gaps between the ESE's and ASE's, and ECP deviates from the 95% nominal level. This is attributed to the strong prior imposed on γ^*_{jl} as we discussed in Section 3.4.2, so that the posterior distribution may be highly peaked and deviated from the normal distribution. We recommend against choosing L to be too small (close to 1) or too large (greater than 10). The former imposes a weak prior and leads to results similar to maximum likelihood estimates, and the latter imposes a too strong prior and leads to an underestimated standard deviation and a possibly inflated bias.

As the sample size increases from 200 to 400, both the proposed BHCM and MLE provide estimates with smaller bias, a better agreement between ESE's and ASE's and, the coverage rates closer to the 95% nominal level. The improvement in the standard error of BHCM estimates, compared to the likelihood-based estimates, is reduced, since Bayesian estimation tends to perform better with a smaller sample size and the two estimation methods have the same limiting distribution, which, therefore, have similar performance as the sample size gets larger. The gaps between ESE's and ASE's of Bayesian estimates with a large L are getting closer as the sample size increases, showing that the posterior distributions get closer to normality with a larger sample.

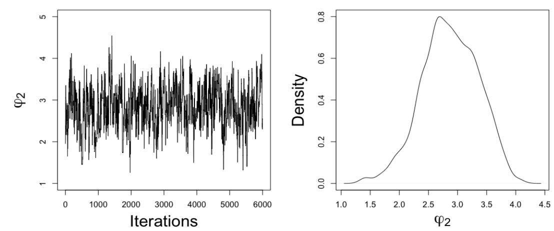
For the Frank copula with the range $(-\infty, \infty)\backslash\{0\}$ in Settings 3.3 and 3.4, we report the results of two different choices of transformation functions in Tables B.4 and B.5 in Appendix, respectively. The identity transformation function $g(\theta) = \theta$ performs poorly with a small sample size, compared to the logit transformation function $g(\theta) = \log(\frac{100+\theta}{100-\theta})$. As the sample size increases from 200 to 400, the two transformation functions seem to work equally well.

Above all, with $L = 4$ across all settings, the BHCM provides reasonable point estimates and interval estimates of copula parameters, and smaller EBias and shorter 95% intervals than those from maximum likelihood method. The benefit of using BHCM is more obvious if the clusters share more similarity in the subject-level dependence structures (e.g., Setting 3.1). The BHCM exhibits capability of handling settings with copula structures containing both one- and two-parameter copulas and large differences in dependence strength.

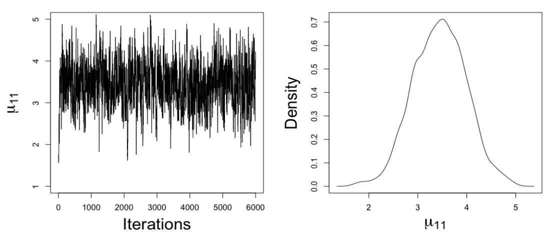
For the BHCM with $L = 4$ in Setting 3.5, we also report the sample trace plots and sample density plots for the results of the mean parameters φ_l and μ_{jl} and those for the copula parameters θ_{jl} for $j = 1, 2, 3, 4$ and $l = 1, 2$, respectively, in Figures 3.3 and 3.4. In all the sample trace plots, the samples of mean parameters and copula parameters vary closely around the posterior mean, and the sample densities are all close to a bell shape, indicating the convergence of the M-H algorithm.



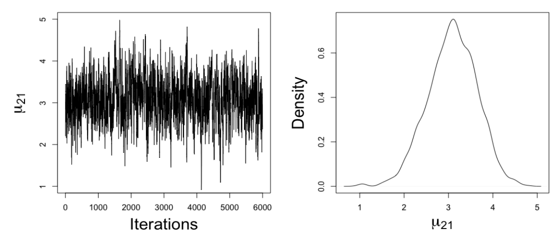
(a) (1) φ_1



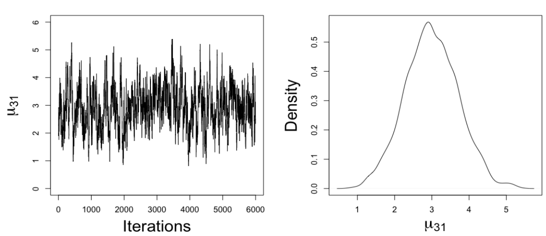
(b) (2) φ_2



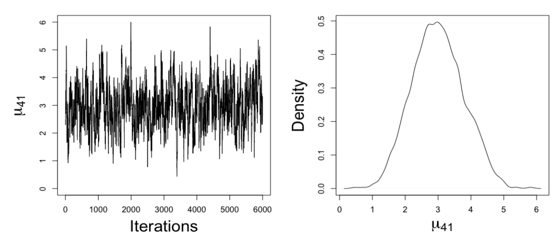
(c) (3) μ_{11}



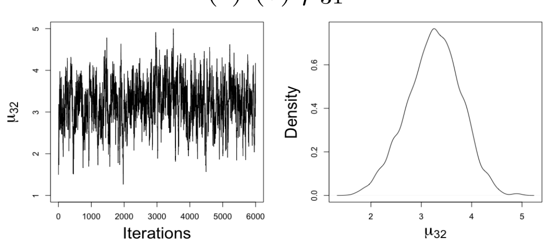
(d) (4) μ_{21}



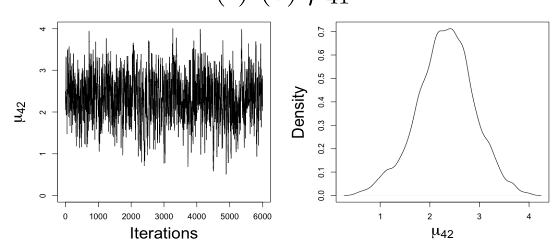
(e) (5) μ_{31}



(f) (6) μ_{41}

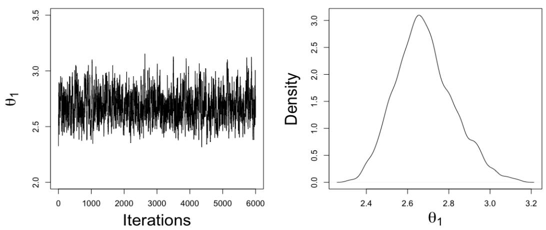


(g) (7) μ_{32}

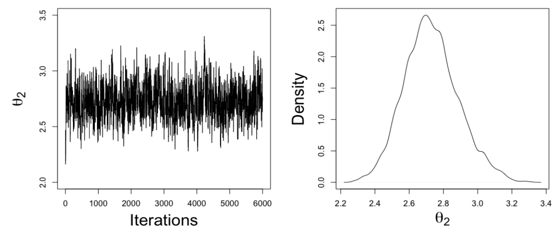


(h) (8) μ_{42}

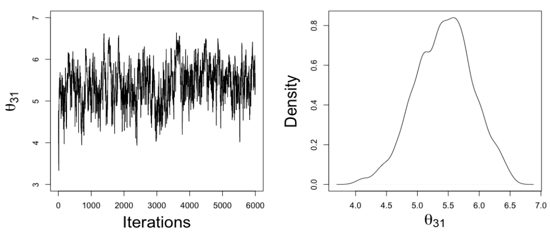
Figure 3.3: Sample trace plots and sample density plots of mean parameters φ_l and μ_{jl} for $j = 1, 2, 3, 4$ and $l = 1, 2$ of the BHCM with $L = 4$ in Setting 3.5



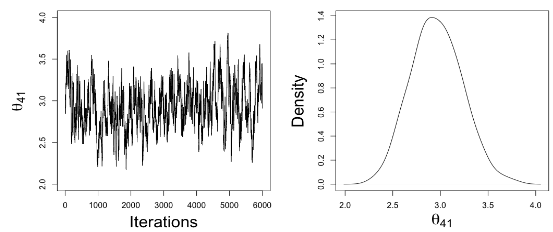
(a) (1) θ_1



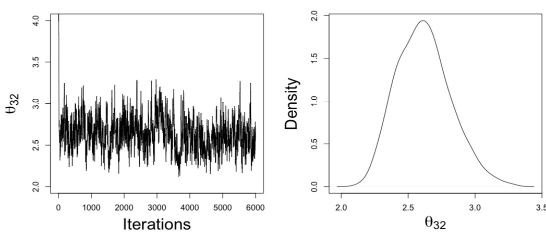
(b) (2) θ_2



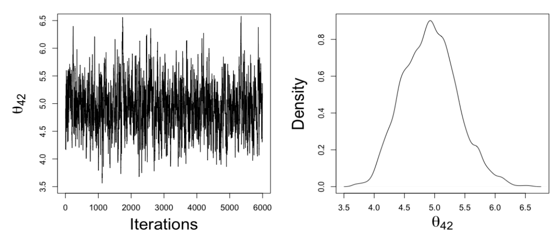
(c) (3) θ_{31}



(d) (4) θ_{41}



(e) (5) θ_{32}



(f) (6) θ_{42}

Figure 3.4: Sample trace plots and sample density plots of copula parameters θ_{jl} for $j = 1, 2, 3, 4$ and $l = 1, 2$ of the BHCM with $L = 4$ in Setting 3.5

3.6 Data Analysis

We now apply the proposed BHCM to analyze the Vertebral Column dataset from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/vertebral+column). This is a biomedical dataset collected by Dr. Henrique da Mota during a medical residence at Lyon, France. The dataset contains the biomedical features of 60 patients with disk hernia, 150 patients with spondylolisthesis and 100 healthy volunteers. The three groups of people are labeled as $j = 1, 2, 3$, respectively. Six biomechanical features are collected, including `angle of pelvic incidence (PI)`, `angle of pelvic tilt (PT)`, `lumbar lordosis angle (LL)`, `sacral slope (SS)`, `pelvic radius (PR)`, and `degree of spondylolisthesis (DS)`, which are labeled as $k = 1, 2, 3, 4, 5$ and 6, respectively. For $j = 1, 2, 3$, $i = 1, \ldots, n_j$, and $k = 1, 2, 3, 4, 5$, let Y_{ijk} denote the kth biomedical features of the ith subject from the jth group of people,where $n_1 = 60$, $n_2 = 150$, and $n_3 = 100$.

In medical research, PR describes pelvic lordosis angle and, PI, PT and SS describe the shape and orientation of the pelvis. They represent two different approaches to characterize the pelvis. For the latter one, PI is defined as "the angle between a line perpendicular to the sacral plate and a line joining the sacral plate to the axis of the femoral heads" and is the arithmetic summation of PT and SS (Berthonnaud et al., 2005). We are interested in examining the dependence of PI versus PT and of PI versus SS. DS is the degree of slipping and can take negative values. We are interested in understanding its dependence with characteristics of pelvis including PI, PT and PR, and that of lumbar LL.

3.6.1 Marginal Model

The histograms of the six biomedical features in three groups are displayed in Figure B.2 in Appendix B.3.1, all showing uni-modal but possibly skewed distributions. As a result, we use a generalized skewed-t distribution to model the marginal distributions of the features to account for the possible skewness.

The estimates of the marginal parameters are obtained by maximizing the marginal likelihood function, and the results are summarized in Table B.6 in the Appendix B.3. The six biomedical features are transformed to copula data $u_{jik} \in [0, 1]$ by applying the fitted marginal CDF to the observed values of the corresponding feature. Let U_{jik} denote the transformed uniformed random variable for of the kth feature of the ith subject in group j for $j = 1, 2, 3$, $i = 1, \ldots, n_j$ and $k = 1, \ldots, 6$.

3.6.2 Dependence Model

We are interested in studying the dependence between the following 6 pairs of variables: PI versus PT, PI versus SS, DS versus PI, DS versus PT, DS versus PR, and DS versus LL. The scatter plots for those pairs are displayed in Figure B.4 in Appendix B.3.2.

We construct a set of parametric copula functions, including the commonly-used copulas in the Archimedean family (Clayton, Gumbel, Frank and Joe copula), Gaussian copula and their rotated versions. The specific copula function forms are selected based on the AIC criterion (Akaike, 1998), which is conducted using the BiCopSelect function in the R package `VineCopula` Schepsmeier et al. (2018). For each bivariate feature, we construct a BHCM for three groups of individuals.

For comparison, we consider two benchmark models. The first one is a multivariate copula model (MCopula), which takes the same marginal and dependence models as the BHCM, i.e., the marginals are generalized skewed-t distributions and copula models are selected using AIC as reported in Table 3.5. The second one is a multivariate Gaussian model (MVN), in which the marginal distributions are all specified as Gaussian distribution and the copulas of the interested six pairs are also specified as Gaussian copula. The parameters in both benchmark models are estimated using the maximum likelihood method.

3.6.3 Results

We compare the performance of the three models, BHCM, MCopula and MVN, in terms of log-likelihood values and the Deviance Information Criterion (DIC) (Spiegelhalter et al., 2014), and summarize the results in Table 3.4. The BHCM has the smallest overall DIC, thus being the best to fit the data. For the clusters of patients with Spondilolisthesis and being healthy, the marginal distributions of some features, for instance, DS, are highly skewed as shown in Figure B.2, MVN provides a poor fit of the data, yielding the smallest log-likelihood and the largest DIC. For the cluster of patients with Disk Hernia, the skewness in the marginal distributions is mild and most of the bivariate copulas selected are Gaussian copula as shown in Table 3.5. The BHCM and MCopula produce log-likelihood values similar to that of MVN but smaller DIC than MVN does, which is partially attributed to the fact that BHCM and MCopula are penalized by extra parameters in their marginal generalized skewed-t distributions.

Table 3.4: Log-likelihood and DIC of three models for each cluster

	Disk Hernia		Spondilolisthesis		Healthy		Total	
	log-likelihood	DIC	log-likelihood	DIC	log-likelihood	DIC	log-likelihood	DIC
BHCM	-1209.79	2464.05	-3639.85	7322.6	-2060.90	4166.29	-6910.54	13952.96
MCopula	-1209.90	2467.78	-3637.70	7323.46	-2062.70	4173.45	-6910.34	13964.68
MVN	-1212.80	2461.66	-3686.70	7409.31	-2079.30	4194.61	-6978.79	14065.58

Tables 3.5 shows the point estimates and interval estimates under the proposed BHCM with $L = 4$ together with the results obtained from the likelihood-based method. Once a Frank copula selected, we use the logit transformation function, which leads to more stable results than the identity transformation function when the sample size is small, shown in

the simulation studies. It is seen that PI has a positive dependence on PT and SS, which aligns with the medical literature (Berthonnaud et al., 2005). Across different groups, the dependence strengths of PI versus PT and PI versus SS show similar Kendall's τ ranging from 0.4 to 0.6. The dependence between DS and other pelvic and lumbar characteristics show an obvious distinction across groups. For patients with disease disk hernia and healthy people, DS has a weak dependence on other four features. However, for patients with Spondylolisthesis, DS has a much stronger positive dependence on the four features.

BHCM with $L = 4$ produces similar point estimates to those obtained from the likelihood-based method, but smaller standard errors. The 95% credible interval of BHCM with $L = 4$ are narrower than 95% confidence intervals obtained from the likelihood-based method. For the cluster of patients with Spondilolisthesis, the DS feature is highly right-skewed as shown in Figure B.2, thus MVN model fails to fit the data well.

3.7 General Remarks

In this chapter, the Bayesian hierarchical copula model (BHCM) is proposed to model correlated data with a hierarchical structure, in which the copula model accounts for the subject level dependence and the Bayesian hierarchical model is used to feature the hierarchical structure. In forming the copula models here, the marginal distributions are assumed to be uniform over the unit interval $[0, 1]$. However, this assumption is not essential. Other parametric models, such as the normal distribution and generalized skewed-t distribution, can be considered to reflect various data features. Furthermore, nonparametric models can also be considered as robust alternatives. It is interesting to extend the proposed method to accommodate these settings.

We comment that our BHCM differs from the Hierarchical Archimedean Copula (HAC) proposed by Okhrin et al. (2013a). Since an Archimedean copula function can be defined through the generator function of the copula (e.g. Nelsen, 2007), an HAC is built by applying the generator function to a lower level HAC in a recursive manner. An HAC overcomes some disadvantages of a regular Archimedean copula. However, it is not designed to handle a hierarchical structure as the one in Figure 3.1. Though our proposed BHCM does not necessarily feature an HAC as the fundamental building block, our proposed framework is general enough to cover the structures that the HAC can handle.

It is noteworthy that our proposed method has multiple sources of regularization. In particular, the estimates of copula parameters are regularized by the tuning parameter L and the estimates of the hyperprior parameters $\hat{\sigma}$, $\hat{\varphi}$, and $\hat{\delta}$. While the hyperprior parameters bring in information "borrowed" from other clusters, the tuning parameter L controls

Table 3.5: Copula functions and estimates for six interested dependence of 3 health groups

Group	Dependence Relations	Copula	BHCM with $L = 4$ Estimates	s.d.	95% Interval	MCopula Estimates	s.d.	95% Interval	MVN Copula	Estimates	s.d.	95% Interval
Disk Hernia	PI v.s. PT	Gaussian	0.696	0.046	(0.599,0.775)	0.694	0.055	(0.586,0.801)	Gaussian	0.710	0.052	(0.608,0.812)
	PI v.s. SS	Gaussian	0.726	0.040	(0.633,0.793)	0.766	0.042	(0.683,0.849)	Gaussian	0.756	0.044	(0.670,0.842)
	DS v.s. PI	Gaussian	0.161	0.098	(-0.031,0.339)	0.150	0.125	(-0.095,0.395)	Gaussian	0.144	0.125	(-0.101,0.389)
	DS v.s. PT	Frank	-0.511	0.577	(-1.489,0.522)	-0.226	0.753	(-1.702,1.250)	Gaussian	0.044	0.129	(-0.209,0.297)
	DS v.s. LL	Gaussian	0.244	0.103	(0.031,0.435)	0.246	0.118	(0.015,0.477)	Gaussian	0.231	0.119	(-0.002,0.464)
	DS v.s. PR	Gaussian	-0.055	0.113	(-0.263,0.175)	-0.060	0.128	(-0.312,0.191)	Gaussian	-0.051	0.129	(-0.304,0.202)
Spondilolisthesis	PI v.s. PT	Frank	5.718	0.505	(0.599,0.775)	5.594	0.622	(4.375,6.814)	Gaussian	0.601	-	-
	PI v.s. SS	Gumbel	1.729	0.099	(1.554,1.943)	1.736	0.113	(1.515,1.958)	Gaussian	0.665	-	-
	DS v.s. PI	Frank	3.427	0.431	(2.552,4.245)	3.453	0.535	(2.404,4.502)	Gaussian	0.533	-	-
	DS v.s. PT	S Clayton[1]	0.887	0.143	(0.608,1.174)	0.905	0.153	(0.605,1.206)	Gaussian	0.439	-	-
	DS v.s. LL	Frank	3.230	0.426	(2.437,4.104)	3.155	0.527	(2.121,4.189)	Gaussian	0.324	-	-
	DS v.s. PR	Joe	1.466	0.115	(1.265,1.698)	1.481	0.123	(1.239,1.723)	Gaussian	0.329	-	-
Healthy	PI v.s. PT	Gaussian	0.633	0.038	(0.555,0.699)	0.636	0.051	(0.537,0.735)	Gaussian	0.634	0.051	(0.534,0.734)
	PI v.s. SS	Gumbel	2.574	0.178	(2.239,2.910)	2.599	0.214	(2.179,3.018)	Gaussian	0.839	0.023	(0.794,0.884)
	DS v.s. PI	Frank	1.822	0.430	(0.936,2.632)	1.714	0.628	(0.483,2.945)	Gaussian	0.200	0.094	(0.016,0.384)
	DS v.s. PT	Gaussian	0.242	0.080	(0.085,0.401)	0.244	0.091	(0.065,0.423)	Gaussian	0.182	0.095	(-0.004,0.368)
	DS v.s. LL	Frank	1.409	0.570	(0.335,2.538)	1.511	0.600	(0.334,2.687)	Gaussian	0.261	0.090	(0.085,0.437)
	DS v.s. PR	Gaussian	-0.111	0.093	(-0.289,0.065)	-0.107	0.098	(-0.299,0.086)	Gaussian	-0.058	0.099	(-0.252,0.136)

[1] Survival Clayton Copula

the strength that the hyperprior parameters can influence the copula parameters. As discussed in Section 3.4 and shown in Section 3.5, with a larger value of L, the parameters θ are more strongly regularized.

Chapter 4

Grouping Dependence Structure and Selection of Copula-Based Models Using Bayesian Nonparametric Methods

4.1 Introduction

The selection of copula forms and the estimation of corresponding parameters of dependent data are highly related to the size of data. A small sample size can lead to inaccurate model selection and parameter estimation. In real life, dependent data that arises from multiple sources may exhibit a similar dependence structure, thus it is feasible to pool the similar dependent data together. A Dirichlet process (DP) is a stochastic process whose realizations are probability distributions. In other words, a DP is "a distribution of distributions". Due to its discrete nature, the DP approach is widely applied to solve clustering problems, see Kim et al. (2006); Dahl (2006); Vlachos et al. (2009); Yu et al. (2010). In Chapter 4, we consider using DP, in combination with copula-based models, to identify similar dependence structures and group them together. We propose a copula-based model with copula selection indicators and dependence parameters following a DP prior, and we call this model the mixture of DPM copula model (M-DPM-CM). The M-DPM-CM is able to group the clusters with similar dependence structures together. The grouping of clusters sharing similar dependence relations can benefit the copula selection and parameter estimation by facilitating a larger sample size.

It is worth clarifying that the commonly-used terminologies "covariance-based clustering" and "copula-based clustering" in the literature are different from what we propose here. The "covariance clustering" is a clustering method that distinguishes data by their variability and quantifies the distances between two groups through the covariance matrices to determine whether they should be put into the same cluster (see Ieva et al., 2016, for example). Some works with the keyword "copula-based clustering" are further categorized in Di Lascio et al. (2017) as "Dissimilarity-based clustering" and "likelihood-based clustering". The former measures the similarity between groups using concordance, tail-dependence or risk measures, which all can be seen as a function of copula parameters. The later makes the grouping based on the maximum likelihood estimation and the Bayesian information criterion (BIC). Furthermore, Fern et al. (2005) used a mixture of local Canonical Correlation Analysis (CCA) model to cluster different local correlations, which focused primarily on the linear dependence relation. Klami and Kaski (2007) proposed a DP prior Gaussian mixture model for dependency-seeking clustering, which "suffers from a severe model mismatch problem" if the data is not normally distributed, as commented by Klami et al. (2012). Rey and Roth (2012) proposed to use the DPM model to perform dependence clustering with the dependence structure described by a Gaussian copula, but their study is restricted to Gaussian copulas for bivariate case and does not involve copula selection. Although their approach allows more general marginal models, they should suffer from the same mismatch problem if the dependent structure is not Gaussian. To our best knowledge, this is the first research that considers the clustering and copula model selection simultaneously.

The rest of the chapter is organized as follows. In Section 4.2, we discuss the model formulation and the construction of the DP prior. In Section 4.3, we describe the sampling scheme and the sampling algorithm. In Section 4.4, we perform simulation studies to evaluate the performance of the proposed M-DPM-CM and compare it with the model selection procedure using AIC. Section 4.5 contains an application to the Vertebral Column dataset.

4.2 Model Formulation

In this section, we introduce the formulation of the mixture of Dirichlet process mixture copula model (M-DPM-CM), a model formulation different from the traditional copula-based mixture models (e.g., Tewari et al., 2011; Rajan and Bhattacharya, 2016; Kosmidis and Karlis, 2016; Kasa et al., 2020) which are mainly concerned with the characterization of multimodal distributions and data clustering.

Suppose that data arises from a hierarchical structure as illustrated in Figure 3.1. Let U_{ji} represent the data from the ith subject of jth cluster for $i = 1, \ldots, n_j$ and $j = 1, \ldots, m$. Suppose a vector of d features are collected for each subject, i.e., $U_{ji} = (U_{ji1}, \ldots, U_{jid})^{\mathrm{T}}$ for $i = 1, \ldots, n_j$ and $j = 1, \ldots, m$. Let $U_j = (U_{j1}^{\mathrm{T}}, \ldots, U_{jn_j}^{\mathrm{T}})^{\mathrm{T}}$ and $U = (U_1^{\mathrm{T}}, \ldots, U_m^{\mathrm{T}})^{\mathrm{T}}$. Furthermore, let u_{jik}, u_{ji}, u_j and u denote the observed counterparts of U_{jik}, U_{ji}, U_j and U, respectively, for $i = 1, \ldots, n_j$, $j = 1, \ldots, m$, and $k = 1, \ldots, d$.

In copula-related literature, the inference functions for margin (IFM) method (Joe and Xu, 1996) is commonly adopted to construct separate models for the marginal distributions and the dependence structure, and focus on the dependence structure modeling in a margin-free framework. Consistent with the copula-related literature (e.g. Aas et al., 2009; Joe, 2014), we assume that U_{jik} marginally follows a uniform distribution on $[0, 1]$, and focus our discussion on dependence modeling of the subject-level data U_{ji} using copula-based models.

Assume that for each cluster $j = 1, \ldots, m$, U_j follows a distribution, which is postulated by the copula model $C_j(\cdot)$ with the associated dependent parameters suppressed in the notation. Let $\{C_r(\cdot) : r \in \mathcal{F}\}$ denote the set of distinct copula functions with $\mathcal{F} = \{1, \ldots, R\}$ recording the labels of distinct copula models. Common choices of copula functions include Clayton, Frank and Gumbel copulas from the Archimedean family (Genest and MacKay, 1986a,b) and Gaussian and t copula from the elliptical family (Frahm et al., 2003). To highlight the idea, we restrict our attention to the single-parameter copula functions and the bivariate scenario (i.e., $d = 2$).

To give a unified presentation for all clustered data, for $j = 1, \ldots, m$, let λ_{jr} denote the binary indicator taking value 1 if U_j is modeled by the copula model $C_r(\cdot)$ and taking value 0 otherwise. Clearly, the constraint $\sum_{r=1}^{R} \lambda_{jr} = 1$ holds for $j = 1, \ldots, m$. We let θ_{jr} represent the associated dependence parameter for the copula model $C_r(\cdot)$ when referring to the modeling for U_j. Specifically, the joint cumulative distribution function (CDF) of the random vector U_{ji} can be expressed as

$$F(U_{ji1}, U_{ji2}) = \sum_{r=1}^{R} \lambda_{jr} C_r(u_{ji1}, u_{ji2}; \theta_{jr}), \tag{4.1}$$

for $i = 1, \ldots, n_j$.

Depending on the function form of the copula, the parameter range for θ_{jr} is often one of the following forms: (1) a bounded interval $[L, U]$, (2) an interval $[L, \infty)$, (3) an interval $(-\infty, U]$ and (4) an infinite interval $(-\infty, \infty) \setminus \{0\}$. To introduce a convenient prior for the dependence parameter θ_{jr}, we reparameterize θ_{jr} via a linear transformation function

$g_r(\cdot)$: $\gamma_{jr} = g_r(\theta_{jr})$. The third column in Table 4.1 summarizes the recommended forms of transformation functions for four types of copula parameters.

For $j = 1, \ldots, m$, let $\lambda_j = (\lambda_{j1}, \ldots, \lambda_{jR})^{\mathrm{T}}$, $\gamma_j = (\gamma_{j1}, \ldots, \gamma_{jR})^{\mathrm{T}}$, and $\psi_j = (\lambda_j^{\mathrm{T}}, \gamma_j^{\mathrm{T}})^{\mathrm{T}}$. Write $\lambda = (\lambda_1^{\mathrm{T}}, \ldots, \lambda_m^{\mathrm{T}})^{\mathrm{T}}$, $\gamma = (\gamma_1^{\mathrm{T}}, \ldots, \gamma_m^{\mathrm{T}})^{\mathrm{T}}$ and $\psi = (\psi_1^{\mathrm{T}}, \ldots, \psi_m^{\mathrm{T}})^{\mathrm{T}}$.

4.2.1 Bayesian Hierarchical Model with Dirichlet Process Prior

We construct a Bayesian hierarchical model for the random vector U_{ji} as follows

$$\begin{aligned} U_{ji}|\psi_j &\sim F(U_{ji1}, U_{ji2}; \psi_j) \\ \psi_j|G &\sim G, \\ G|\eta, a &\sim \mathrm{DP}(a, G_\eta), \\ (\eta, a) &\sim \pi(\eta, a), \end{aligned} \tag{4.2}$$

for $j = 1, \ldots, m$. In this model, ψ_j has a prior distribution G, a discrete probability measure, which is generated from a Dirichlet Process (DP) with a scale parameter $a > 0$ and a base probability measure G_η indexed by parameters η. G_η can be understood as the "center" of the DP and a indicates how much the DP concentrates around G_η (Müller et al., 2015). The hyper-prior parameters $(\eta^{\mathrm{T}}, a)^{\mathrm{T}}$ have the joint density function $\pi(\cdot, \cdot)$.

More specifically, we assume that the base measure of the DP is of the form

$$G_\eta = G_{\eta_\lambda} \cdot G_{\eta_\gamma}, \tag{4.3}$$

in which G_{η_λ}, indexed by parameter η_λ, corresponds to the indicator vector $\lambda_j = (\lambda_{j1}, \ldots, \lambda_{jR})^{\mathrm{T}}$, and G_{η_γ} takes the form $G_{\eta_\gamma} = \prod_{r=1}^{R} G_{\eta_{\gamma r}}$, in which $G_{\eta_{\gamma r}}$, indexed by $\eta_{\gamma r}$, corresponds to the dependence parameters γ_{jr} for $r = 1, \ldots, R$. Let $\eta = (\eta_\lambda^{\mathrm{T}}, \eta_{\gamma_1}^{\mathrm{T}}, \ldots, \eta_{\gamma_R}^{\mathrm{T}})^{\mathrm{T}}$. Specifically, we assume G_{η_λ} to be a measure corresponding to a Dirichlet-multinomial distribution with the total number of trial being 1, denoted by Dir-Mul(η_λ), where $\eta_\lambda = (\eta_{\lambda 1}, \ldots, \eta_{\lambda R})^{\mathrm{T}}$ is a vector of positive real parameters indexing the Dirichlet-multinomial distribution. The last column of Table 4.1 provides recommended distributions for $G_{\eta_{\gamma r}}$, corresponding to the four types of transformed parameters γ. We let $\eta_{\gamma_r} = (\alpha_r, \beta_r)^{\mathrm{T}}$, for $r = 1, \ldots, R$.

Copulas	Range of θ_{jr}	Tranformation Function	Range of γ_{jr}	Distribution for $G_{\eta_{\gamma_r}}$
Gaussian	$[L, U]$	$g(x) = \frac{1}{U-L}x - \frac{L}{U-L}$	[0,1]	Beta(α_r, β_r)
Clayton	$[L, \infty)$	$g(x) = x - L$	$[0, \infty)$	Gamma(α_r, β_r)
Rotated Clayton	$(-\infty, U]$	$g(x) = U - x$	$[0, \infty)$	Gamma(α_r, β_r)
Frank	$(-\infty, \infty) \setminus \{0\}$	$g(x) = x$	$(-\infty, \infty)$	$N(\alpha_r, \beta_r^2)$

Table 4.1: Transformation functions and distributions for $G_{\eta_{\gamma_r}}$

For hyperprior distribution of a, we assume $a \sim$ gamma(c, d) with mean c/d and variance c/d^2. The hyperprior distribution for η is assumed to be $\pi(\cdot)$. To select weak-informative or noninformative hyperprior distributions, small values of c, d are used for a and uniform priors are set for η.

4.2.2 Model Selection and Grouping under Dirichlet Process Prior

While the formulation of copula model in (4.1) and the Bayesian hierarchical model in (4.2) is natural to reflect our interest in using suitable copula models to group similar dependence structures among different clusters, the derivation of the posterior distribution of ψ is not straightforward. Alternatively, we consider an equivalent formulation of ψ, which will lead to convenient derivations of the posterior distribution of ψ.

The DP prior distribution for ψ_j, G, is discrete in nature. Such a property allows a positive probability that two or more clusters can be modeled by the same copula function with the same dependence parameter. We assume that there are h unique values of ψ_j for $j = 1, \ldots, m$, with $h \leq m$. Let $\psi^* = (\psi_1^*, \ldots, \psi_h^*)^{\mathrm{T}}$. For $l = 1, \ldots, h$, let $S_l = \{j : \lambda_j = \lambda_l^*, \gamma_j = \gamma_l^*\}$ be the index set of the clusters with the lth unique parameter vector, and let n_{S_l} denote the number of elements in S_l. Then the collection $\{S_1, \ldots, S_h\}$ is a partition of $\{1,2,\ldots,m\}$.

We further let $z_j = l$ if $j \in S_l$, h_j denote the number of unique models in the first j clusters and h_{jl} denote the number of clusters which select the lth unique model in the first j clusters, for $j = 1, \ldots, m$. Let $z = (z_1, \ldots, z_m)^{\mathrm{T}}$, and we have $h_j = \sum_{l=1}^{h} h_{jl}$, for $j = 1, \ldots, m$. When $a = 0$, all clusters take the same model. By the Pólya Urn sampling scheme (Blackwell and MacQueen, 1973), the conditional distribution of z_j, given

$z_1, \ldots, z_{j-1}$ and a is

$$p(z_j = l | z_1, \ldots, z_{j-1}, a) = \begin{cases} \dfrac{h_{j-1,l}}{a + j - 1}, & l = 1, \ldots, h_{j-1}, \\ \dfrac{a}{a + j - 1}, & l = h_{j-1} + 1, \end{cases} \tag{4.4}$$

for $j = 1, \ldots, m$. This suggests that when the model assignment is completed for the first $j-1$ clusters, the probability that the jth cluster is modeled by the lth model is proportional to the number of clusters already being assigned to this model with $l = 1, \ldots, h_{j-1}$, and the probability of assigning cluster j to a new model is proportional to a. This sampling scheme is a "winner-gets-more" mechanism. By the fact that $z_1, \ldots, z_m$ are exchangeable, the conditional distribution for z_j, given z_{-j} and a, is

$$p(z_j = l | z_{-j}, a) = \begin{cases} \dfrac{h_{-j,l}}{a + m - 1}, & l = 1, \ldots, h_{-j}, \\ \dfrac{a}{a + m - 1}, & l = h_{-j} + 1, \end{cases} \tag{4.5}$$

where $z_{-j} = (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_j)^{\mathrm{T}}$, h_{-j} is the number of unique values in $\psi_{-j} = (\psi_1^{\mathrm{T}}, \ldots, \psi_{j-1}^{\mathrm{T}},\ \psi_{j+1}^{\mathrm{T}}, \ldots, \psi_m^{\mathrm{T}})^{\mathrm{T}}$ and $h_{-j,l}$ is the number of the lth unique value in ψ_{-j}, for $j = 1, \ldots, m$.

The proposed model has several advantages over the conventional copula model selection and estimation methods. First, the M-DPM-CM performs model selection for m clusters simultaneously. In contrast, conventional methods select the copula forms for each cluster separately. Second, through the grouping effect of the DP prior, the clusters with similar dependence relation will be postulated by the same model, thus reducing the number of parameters to be estimated and increasing the size for estimation of the associated parameters, and eventually yielding more efficient inference results.

4.3 Bayesian Inference Process

4.3.1 Posterior and Hyper-Posterior Distribution

The proposed M-DPM-CM is a non-conjugate mixture of DPM model. The conditional posterior distribution of z_j, given $\{z_{-j},\ \psi^*,\ a,\ \eta,\ u_j\}$ is

$$p(z_j = l | z_{-j}, \psi^*, a, \eta, u_j) \propto f(u_j | z_j = l, \psi_l^*) p(z_j = l | z_{-j}, a)$$

$$= \begin{cases} \frac{h_{-j,l}}{a+m-1} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}), & \text{if } l = 1, \ldots, h_{-j}, \\ \frac{a}{a+m-1} \prod_{i=1}^{n_j} \int \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}) dG_\eta(\psi_l^{*}), & \text{if } l = h_{-j}+1, \end{cases} \tag{4.6}$$

where $\theta_{lr}^{*} = g_r^{-1}(\gamma_{lr}^{*})$. Since the posterior distribution (4.6) involves analytically intractable integrals, we take the approach of Neal (2000) to generate augmented parameters to obtain a posterior distribution of no integration. We augment the sequence of unique parameters by considering b additional latent parameters $\psi_b = (\psi_{h+1}^{\mathrm{T}}, \ldots, \psi_{h+b}^{\mathrm{T}})^{\mathrm{T}}$, in which ψ_{h+v} is independently generated from G_η for $v = 1, \ldots, b$. After the augmentation, the conditional prior distribution in (4.5) becomes

$$p(z_j = l | z_{-j}, a) = \begin{cases} \dfrac{h_{-j,l}}{a+m-1}, & \text{if } l = 1, \ldots, h_{-j}, \\ \dfrac{a}{b(a+m-1)}, & \text{if } l = h_{-j}+1, \ldots, h_{-j}+b. \end{cases} \tag{4.7}$$

In other words, when model l is not one of the h_{-j} unique models that has already been taken, instead of taking a new model generated from the DP process, we randomly choose a model from one of the b augmented models with equal chances. The posterior distribution of z_j can be derived as

$$\begin{aligned} & p(z_j = l | z_{-j}, \psi^{*}, \psi_b, a, u_j) \\ \propto & \begin{cases} \frac{h_{-j,l}}{a+m-1} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}), & \text{if } l = 1, \ldots, h_{-j}, \\ \frac{a}{b(a+m-1)} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}), & \text{if } l = h_{-j}+1, \ldots, h_{-j}+b, \end{cases} \\ \propto & \begin{cases} h_{-j,l} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}), & \text{if } l = 1, \ldots, h_{-j}, \\ \frac{a}{b} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}), & \text{if } l = h_{-j}+1, \ldots, h_{-j}+b. \end{cases} \end{aligned} \tag{4.8}$$

Then the parameter ψ_l^{*}, conditional on z, η and u, is

$$p(\psi_l^{*} | z, \eta, u) \propto G_\eta(\psi_l^{*}) \prod_{j \in S_l} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}).$$

Since $\psi_l^{*} = (\lambda_l^{*}, \gamma_l^{*})$ contains both discrete and continuous parameters, we further decompose the conditional distribution as

$$\begin{aligned} p(\lambda_l^{*} | z, \eta, u, \gamma_l^{*}) &\propto G_\eta(\psi_l^{*}) \prod_{j \in S_l} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}). \\ p(\gamma_l^{*} | z, \eta, u, \lambda_l^{*}) &\propto G_\eta(\psi_l^{*}) \prod_{j \in S_l} \prod_{i=1}^{n_j} \sum_{r=1}^{R} \lambda_{lr}^{*} c_r(u_{ji1}, u_{ji2}; \theta_{lr}^{*}). \end{aligned} \tag{4.9}$$

For hyperparameter η, its conditional posterior distribution given ψ^*, is

$$p(\eta|\psi^*) \propto \pi(\eta) \prod_{l=1}^{h} G_\eta(\psi_l^*). \tag{4.10}$$

4.3.2 Sampling Scheme

The following Gibbs sampler (Geman and Geman, 1987; Neal, 2000) algorithm is used to obtain a sample of $(z^{\mathrm{T}}, \psi^{*\mathrm{T}}, \eta^{\mathrm{T}}, a)^{\mathrm{T}}$ from the joint posterior distribution $f(z, \psi^*, \eta, a|u)$. Let $z^{(t)}$, $\psi^{*(t)}$, $\eta^{(t)}$ and $a^{(t)}$ denote the sampled values of the corresponding parameters in the tth iteration. Let $h^{(t)}$ denote the number of unique values in $\psi^{*(t)}$. To simplify the notation, we let $z_{1:(j-1)} = (z_1, \ldots, z_{j-1})^{\mathrm{T}}$ and $z_{(j+1):m} = (z_{j+1}, \ldots, z_m)^{\mathrm{T}}$.

In the algorithm, the posterior distribution of a, given $h^{(t)}$ and the auxiliary parameter $\phi^{(t)}$, is a mixture of two gamma density functions with probabilities $\pi^{(t)}$ and $1 - \pi^{(t)}$, respectively. For details, refer to Escobar and West (1995). After the algorithm converges, ψ^* provides the results of grouping and model selection.

We conclude this section with comments. The method developed here is scalable to accommodating an increasing dimension of the features and the number of clusters. When modeling data with more than two features, the commonly adopted copula forms from Archimedean or Extreme-value families (Joe, 1997) contain only one or two parameters, whereas copulas from the Elliptical family, such as Gaussian copula, often involve with a larger dimension of parameters; certain correlation structures are usually imposed to facilitate a parsimonious model. As a result, an increase in the dimension of features does not necessarily lead to a dramatic increase in the dimension of copula parameters, thus not bringing much challenge to the implementation of the algorithm. Moreover, when dealing with a large number of clusters, the convergence of the algorithm is not compromised as the sampling procedures for each cluster in Steps 2 and 3 are conducted separately. The main paid price with a large number of clusters is the increase of the computation time due to more iterations in each loop of the algorithm.

Table 4.2: Sampling algorithm for M-DPM-CM

Algorithm: Gibbs sampler for sampling from the posterior distribution $f(z, \psi^*, \eta, a|u)$

Input: Initial values of parameters $z^{(0)}$, $\psi^{*(0)}$, $\eta^{(0)}$, $a^{(0)}$, $h^{(0)}$, m, the number of augmented parameters b and the prior parameters c, d for the scale parameter a

1. Generate the additional latent parameters $\psi_b^{(t)}$.

for v=1,...,b **do**

Sample $\psi_{h+v}^{(t)}$ independently from $G_{\eta^{(t-1)}}$.

end

2. Generate the grouping indicator $z^{(t)}$.

for j=1,...,m **do**

Sample $z_j^{(t)}$ from the posterior distribution $p(z_j|z_{1:(j-1)}^{(t)}, z_{(j+1):m}^{(t-1)}, \psi^{*(t-1)}, \psi_b^{(t)}, a^{(t-1)}, u_j)$ as given in (4.8)

end

3. Generate the unique parameters $\psi^{*(t)}$

for l=1,...,$h^{(t)}$ **do**

Sample ψ_l^* using a Gibbs sampler from the conditional posterior distribution $p(\lambda_l^*|z, \eta, u, \gamma_l^*)$ and $p(\gamma_l^*|z, \eta, u, \lambda_l^*)$ as given in (4.9).

end

4. Update hyperparameters $\eta^{(t)}$ through $p(\eta|\psi^{*(t)})$ in (4.10)

5. Update the scale parameter $a^{(t)}$ using an auxiliary sampler proposed by Escobar and West (1995).

(1) Generate $\phi^{(t)} \sim \text{Beta}(a^{(t-1)} + 1, m)$.

(2) Solve $\pi/(1-\pi) = (c + h^{(t)} - 1)/\{m(d - \log(\phi^{(t)}))\}$ to get $\pi^{(t)}$.

(3) Generate

$$a|\phi^{(t)}, h^{(t)} = \begin{cases} \text{gamma}(c + h^{(t)}, d - \log(\phi^{(t)})) & \text{with probability } \pi^{(t)} \\ \text{gamma}(c + h^{(t)} - 1, d - \log(\phi^{(t)})) & \text{with probability } 1 - \pi^{(t)} \end{cases}$$

4.4 Simulation Studies

In this section, the performance of the M-DPM-CM is investigated through finite sample studies from multiple perspective, in comparison with the conventional copula selection using AIC (Akaike, 1998). Specifically, the AIC for each cluster $j = 1, \dots, m$ and each candidate model $r = 1, \dots, R$ is calculated as

$$\text{AIC}_{jr} = 2 - 2 \ln \left[\prod_{i=1}^{n_j} c_r(u_{ji1}, u_{ji2}, \hat{\theta}_{jr}) \right],$$

where $\hat{\theta}_{jr}$ is the maximum likelihood estimate of θ_{jr} obtained under the assumption that the dependence structure is governed by the rth copula function from the candidate pool. The model yielding the minimum AIC value is selected. Since only copulas with one parameter are considered, the penalty term simplifies to a constant in the AIC formula.

4.4.1 Simulation Settings

Consider the case where we have $m = 12$ clusters and n_j subjects in each cluster for $j = 1, \dots, m$. We generate

$$U_{ji} = (U_{ji1}, U_{ji2}) \sim C_j(u_{ji1}, u_{ji2}; \theta_j),$$

independently for $i = 1, \dots, n$ and $j = 1, \dots, m$. Four simulation settings are considered here.

The first setting is a "high signal" setting in the sense that there are large differences across clusters in terms of their dependence structures. We assume that the bivariate variables (U_{ji1}, U_{ji2}) are positively dependent in some clusters but negatively dependent in others. In this setting, clusters with different dependence structures tend to be easily differentiated and are postulated with different models. The second setting is a "low signal" settings in which we assume that the bivariate variables (U_{ji1}, U_{ji2}) hold positive dependence in all m clusters. It is more challenging to differentiate dependence structures across clusters. In the third setting, we let some clusters have the same parametric copula form, but with different strength of dependence, i.e., different copula parameters. The fourth setting facilitates a "nearly independent" structure where Kendall's τ's of all copulas considered take the value of 0.1 or -0.1, characterizing an eminently weak dependence. The copula forms C_j and the corresponding parameters θ_j are summarized in Table 4.3.

Table 4.3: Copula Forms and Parameter Values in Each Cluster in the Simulation Set-ups

High Signal Setting

Cluster	1	2	3	4	5	6
Copula(θ_j)	Clayton(3)	R. Gumbel(-2.5)[1]	Clayton(3)	Gaussian(-0.6)	Frank(6)	Clayton(3)
Cluster	7	8	9	10	11	12
Copula(θ_j)	Clayton(3)	R. Gumbel(-2.5)[1]	Gaussian(-0.6)	Frank(6)	Clayton(3)	Gaussian(-0.6)

Low Signal Setting

Cluster	1	2	3	4	5	6
Copula(θ_j)	Clayton(3)	Gumbel(2.5)	Clayton(3)	Gaussian(0.6)	Frank(6)	Clayton(3)
Cluster	7	8	9	10	11	12
Copula(θ_j)	Clayton(3)	Gumbel(2.5)	Gaussian(0.6)	Frank(6)	Clayton(3)	Gaussian(0.6)

Common Copula Form Setting

Cluster	1	2	3	4	5	6
Copula(θ_j)	Clayton(2)	Clayton(4)	Clayton(2)	Frank(8)	Frank(5)	Clayton(4)
Cluster	7	8	9	10	11	12
Copula(θ_j)	Clayton(2)	Clayton(4)	Frank(8)	Frank(5)	Clayton(2)	Frank(8)

Nearly Independent Setting

Cluster	1	2	3	4	5	6
Copula(θ_j)	Clayton(0.22)	Gumbel(1.11)	Clayton(0.22)	Frank(0.91)	Frank(-0.91)	Clayton(0.22)
Cluster	7	8	9	10	11	12
Copula(θ_j)	Clayton(0.22)	Gumbel(1.11)	Frank(0.91)	Frank(-0.91)	Clayton(0.22)	Frank(0.91)

[1] Rotated Gumbel Copula with 90 degrees

In the first, second, and the fourth settings, we assume that there are 4 unique dependence models. Clusters 1, 3, 6, 7 and 11 share a common dependence structure (in blue), clusters 2 and 8 share one (in red), clusters 4, 9 and 12 have a common model (in yellow), and clusters 5 and 10 share another one (in green). In the third setting, clusters 1, 2, 3, 6, 7, 8 and 11 share the same copula form, but clusters 1, 3, 7, 11 have relatively weak dependence (in blue), and clusters 2, 6, 8 have stronger dependence (in red). Clusters 4, 5, 9, 10 and 12 have a common copula model, but clusters 4, 9, 12 have a strong dependence (in yellow), and clusters 5 and 10 share a weak dependence (in green).

For the first three settings, we perform simulations with the identical cluster size ($n_j = n$) and $n = 50, 100$, 200, 400 or 1000. For the fourth setting, since every cluster holds a highly similar and nearly independent dependence structure, it is challenging to conduct model selection using the proposed method or other existing methods for cases with a small sample size. Therefore, we perform simulations on sample sizes $n = 100, 200, 400$ and 1000 in the fourth setting. A scenario with varied sample sizes across clusters is also considered for the Common Copula Form Setting, with $n_1 = n_2 = n_3 = 50$, $n_4 = n_5 = n_6 = 100$, $n_7 = n_8 = n_9 = 200$ and $n_{10} = n_{11} = n_{12} = 400$, to demonstrate the capability of M-DPM-CM to handle distinct cluster sizes.

We take b to be 2 in the Gibbs sampler, shown by Neal (2000) with simulations to be sufficient for exploring the parameter space. The set of copula functions $\mathcal{F}$ includes the one-parameter copulas in the Archimedean family (Clayton, Gumbel, Frank and Joe copula), Gaussian copula and their rotated versions of copulas. In total, $R = 14$ copulas can be selected for each cluster. The hyperprior distribution for a is set to be a weakly informative prior, gamma$(0.01, 0.01)$, and the hyperprior distribution of η is assumed to be a non-informative uniform prior. Three hundred simulations are repeated for each setting.

4.4.2 Evaluation Metrics

We consider different metrics to evaluate different aspects of the proposed M-DPM-CM.

1. Grouping Effects: For m clusters, there are $\binom{m}{2}$ pairs. Let TP (true positive) denote the number of pairs that belong to the same group under the true model and are assigned to the same group by M-DPM-CM; let TN (true negative) denote the number of pairs that do not belong to the same group under the true model and are assigned to different groups by M-DPM-CM; let FN (false negative) denote the number of pairs that belong to the same group under the true model but are assigned with the different models by M-DPM-CM; and let FP (false positive) denote the number of pairs that do not belong to the same group under the true model and are allocated to the same group by M-DPM-CM. Consequently,

$$\text{TP+TN+FP+FN} = \binom{m}{2}.$$

   (a) Rate of False Positive (RFP) : To quantify how bad the grouping may have been done, we give special attention to false positive rate, calculated as RFP=FP/$\binom{m}{2}$. We report the average RFP of 300 simulations.

   (b) *Rand Index (RI)* : Rand Index (named after Willam M. Rand) is a measure of the similarity between two ways of grouping. Under the true model described in Section 4.4.1, the 12 clusters are grouped into 4 sets. In each set, the clusters share the same dependence model (the same copula function and the same parameter values). After applying M-DPM-CM, there are h unique models which group the clusters. Here Rand Index is used to compare the groupings under the true model and the M-DPM-CM. The Rand Index is computed as

$$\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\binom{m}{2}},$$

with the range $0 \leq \mathrm{RI} \leq 1$. The greater the RI, the better the grouping resulted from the M-DPM-CM. We report the average RI of 300 simulations.

(c) *Correct Grouping Percentage (CGP)* : If the M-DPM-CM gives the correct partition of the 12 clusters, we say that the DPMCM leads to a "correct grouping". We report the percentage of correct groupings for those 300 simulations.

2. Copula Selection:

   Mis-selected Percentage (MSP) : If the copula function selected by the M-DPM-CM or AIC is different from the one under the true model for a particular cluster, we say that the M-DPM-CM leads to a "mis-selected" copula function for the cluster. We report the percentage of mis-selected copula functions for those 300 simulations for each cluster.

3. Parameter Estimation: We perform parameter estimation based on the grouping and copula selection results. If the copula form governing U_{ji1} and U_{ji2} is correctly selected, the dependence parameter θ_j of the copula function is then estimated from maximum likelihood estimation (MLE) using the grouped data obtained from M-DPM-CM. We also implement the conventional copula selection method AIC to select copula function for each cluster and use MLE to estimate dependence parameters in each cluster separately. For both methods, we consider the following four metrics computed based on the simulations with correct selection of copula forms:

   (a) *Empirical Bias (EBias)*: The difference between the average of the estimated values from simulations with correct selection of copula forms and the true value of the parameters;

   (b) *Empirical Standard Error (ESE)*: The sample standard deviation of the estimates;

   (c) *Asymptotic Standard Error (ASE)*: The average of estimated asymptotic standard deviations of the estimators;

   (d) *Empirical Coverage Probability (ECP)*: The proportion of the confidence intervals that contain the true parameter values.

4.4.3 Simulation Results

The results for grouping effects are summarized in Table 4.4. For the first three settings and different sample sizes, RFPs are less than 4.4%, and RIs are all greater than 0.89. When the sample size is equal to or greater than 200, the RFP gets close to 0, suggesting that the M-DPM-CM rarely groups two clusters belonging to different groups as a single one; the RI becomes close to 1, showing correct grouping. The CGP is over 84% when the sample size reaches 200, suggesting that the M-DPM-CM nearly perfectly recovers the true grouping of clusters with a moderate sample size. With a given sample size, the RFP is the smallest, the RI and the CGP are the largest in the high signal setting. Unsurprisingly, all grouping metrics are unsatisfying in the Nearly Independent Setting, since all clusters hold highly similar structures, which are all close to independence. This is a challenging scenario of a very low signal where the dependence structures are barely distinguishable, and the M-DPM-CM tends to group the clusters together, especially in the cases of small sample sizes, when information from data is too little to differentiate clusters. As the sample size increases, the RFP shows an obviously decreasing trend with a dramatic jump in the RI and CGP, demonstrating the capability of the M-DPM-CM to pick the weak signals if fed with sufficient information.

Table 4.4: Simulation results for grouping effects

	High Signal Setting						Low Signal Setting				
Sample Size	50	100	200	400	1000		50	100	200	400	1000
RFP	0.891%	0.030%	0.000%	0.000%	0.000%		4.371%	2.460%	0.586%	0.015%	0.000%
RI	97.008%	99.439%	99.747%	99.808%	99.863%		91.606%	96.455%	99.056%	99.793%	99.947%
CGP	55.779%	84.667%	93.333%	95.333%	96.000%		20.500%	50.667%	84.667%	94.667%	97.500%
	Common Copula Form Setting							Nearly Independent Setting			
Sample Size	50	100	200	400	1000	Varied Size		100	200	400	1000
RFP	4.106%	1.136%	0.096%	0.000%	0.000%	0.598%		58.076%	45.432%	31.667%	16.886%
RI	89.008%	96.949%	99.293%	99.343%	99.447%	96.848%		38.978%	50.742%	64.318%	81.242%
CGP	10.000%	59.667%	84.667%	87.333%	91.000%	57.500%		0.000%	0.500%	3.500%	38.500%

The results for copula selection and parameter estimation in the Common Copula Form Setting are shown in Table 4.5, and those for High and Low Signal Settings and Nearly Independent Setting are provided in Appendix C. We report the results of the proposed M-DPM-CM and the conventional copula selection method using AIC. The results suggest that the proposed M-DPM-CM has significantly lower MSP for all clusters in all signal settings with all sample sizes than the AIC method does. For parameter estimation, the MLEs under the model selected by the M-DPM-CM generally have smaller EBiases, ESEs and ASEs than those produced by AIC. In the case of varied sample sizes, the M-DPM-CM handles this challenging scenario well by providing competitive model selection and parameter estimation results, and the clusters with small sample sizes (clusters 1-6) obviously have more substantial efficiency gain than the clusters with large sample sizes (clusters 7-12).

The advantages of the M-DPM-CM in copula selection and parameter estimation is largely attributed to its excellent grouping performance. The M-DPM-CM has more data to work with and therefore has a greater chance to select the right copula form and obtain more efficient estimates. The improvement in EBias and the efficiency gain are similar in all settings, but more obvious when the sample size is smaller. The model selection and estimation results in the Nearly Independent Setting deteriorate as the cluster size gets smaller, as expected, due to the high RFP, but the M-DPM-CM provides competitive results when the sample size is greater than 400. It is interesting that the results in the Nearly Independent Setting does not compromise the usefulness of the M-DPM-CM. In practice, it is usually of less interest to characterize dependence structure when it is "nearly independent". Moreover, one major motivation of dependence analysis is to improve statistical efficiency of marginal analysis by borrowing information from associated data. When data are "nearly independent", the benefit of dependence modeling is fading out as little information can be used to assist efficiency gain of marginal analysis.

The computation time and complexity of the M-DPM-CM depend on multiple factors, including the number of clusters, the cluster sizes, and the candidate pool of copulas. For the simulation studies considered, the convergence speed of the Gibbs sampler described in Section 4.3 is fast and becomes faster as the sample size increases. For simulations with the sample size 50, the Gibbs sampler converges within 200 iterations, and for those with the sample size 1000, the algorithm converged within 50 iterations.

In summary, the simulation studies show that the M-DPM-CM can efficiently group clusters with similar dependence relations when the within cluster dependence is not weak, even with a small sample size, and thus, benefit model selection and parameter estimation, especially for clusters with small sample sizes.

Table 4.5: Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Common Copula Form Setting

M-DPM-CM

	$n=50$					$n=100$					$n=200$					$n=400$					$n=1000$					Varied Sample Size				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	56.000%	0.125	0.358	0.227	0.915	44.667%	0.024	0.180	0.143	0.952	39.333%	0.006	0.107	0.100	0.934	27.667%	-0.014	0.081	0.070	0.918	12.500%	-0.005	0.049	0.045	0.954	47.500%	0.029	0.235	0.118	0.895
2	43.000%	-0.026	0.453	0.356	0.925	31.333%	0.002	0.259	0.257	0.937	26.667%	-0.003	0.192	0.180	0.936	30.333%	-0.008	0.137	0.127	0.928	26.000%	-0.004	0.084	0.081	0.932	29.500%	-0.033	0.261	0.238	0.950
3	57.000%	0.128	0.382	0.226	0.915	44.333%	0.016	0.213	0.144	0.952	39.000%	< 0.001	0.102	0.099	0.945	29.333%	-0.011	0.082	0.070	0.920	13.000%	-0.002	0.046	0.044	0.954	45.000%	0.035	0.275	0.126	0.905
4	7.500%	-0.055	1.205	0.987	0.900	0.333%	-0.111	0.727	0.593	0.910	0.000%	-0.022	0.418	0.380	0.937	0.000%	0.017	0.318	0.271	0.913	0.000%	0.001	0.172	0.172	0.960	0.000%	0.001	0.529	0.416	0.920
5	12.000%	0.318	1.318	0.816	0.930	3.333%	0.099	0.735	0.574	0.905	0.333%	-0.005	0.385	0.368	0.950	0.000%	-0.018	0.252	0.261	0.967	0.000%	0.011	0.160	0.164	0.955	1.500%	0.112	0.448	0.372	0.910
6	42.500%	-0.055	0.474	0.355	0.935	31.000%	-0.002	0.261	0.257	0.950	26.667%	-0.006	0.194	0.180	0.936	30.333%	-0.007	0.133	0.127	0.933	26.000%	-0.001	0.081	0.081	0.939	28.000%	-0.053	0.234	0.235	0.951
7	56.000%	0.165	0.348	0.248	0.948	46.000%	-0.005	0.153	0.144	0.969	37.667%	0.006	0.118	0.099	0.936	29.000%	-0.011	0.077	0.070	0.918	13.500%	-0.005	0.045	0.044	0.960	44.500%	-0.013	0.132	0.110	0.930
8	43.000%	-0.040	0.480	0.357	0.925	32.000%	-0.007	0.314	0.256	0.937	26.667%	-0.005	0.190	0.181	0.936	30.333%	-0.007	0.133	0.127	0.933	25.500%	< 0.001	0.081	0.081	0.940	27.500%	-0.044	0.234	0.235	0.952
9	4.000%	-0.134	1.161	0.910	0.915	0.333%	-0.098	0.705	0.596	0.920	0.000%	0.014	0.443	0.385	0.933	0.000%	0.006	0.290	0.271	0.933	0.000%	-0.002	0.194	0.173	0.940	0.500%	-0.024	0.455	0.373	0.925
10	17.500%	0.149	1.019	0.894	0.938	2.333%	0.127	0.728	0.582	0.910	0.667%	-0.001	0.414	0.367	0.946	0.333%	-0.008	0.252	0.261	0.970	0.000%	0.008	0.159	0.164	0.955	0.000%	0.018	0.350	0.333	0.940
11	56.000%	0.151	0.351	0.232	0.900	45.667%	0.019	0.207	0.156	0.935	38.333%	< 0.001	0.121	0.099	0.958	28.333%	-0.011	0.081	0.070	0.920	13.000%	-0.003	0.047	0.044	0.954	44.500%	-0.001	0.125	0.106	0.950
12	7.000%	-0.062	1.157	0.884	0.925	0.667%	-0.087	0.696	0.599	0.915	0.000%	-0.035	0.450	0.382	0.954	0.000%	0.028	0.315	0.272	0.923	0.000%	0.003	0.187	0.171	0.955	0.000%	-0.038	0.411	0.355	0.915

AIC

	$n=50$					$n=100$					$n=200$					$n=400$					$n=1000$					Varied Sample Size				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	56.667%	0.029	0.412	0.389	0.956	48.333%	0.030	0.270	0.276	0.968	40.000%	0.035	0.190	0.196	0.967	29.667%	-0.005	0.139	0.138	0.934	19.500%	-0.014	0.080	0.087	0.963	54.500%	0.029	0.412	0.389	0.956
2	47.333%	0.123	0.565	0.629	0.962	42.667%	0.118	0.519	0.449	0.936	38.000%	-0.021	0.324	0.309	0.935	38.000%	-0.022	0.216	0.219	0.919	31.000%	-0.004	0.147	0.139	0.942	47.000%	0.123	0.565	0.629	0.962
3	59.000%	0.136	0.426	0.406	0.963	43.667%	0.038	0.299	0.277	0.947	44.333%	0.024	0.175	0.197	0.970	40.333%	-0.004	0.139	0.138	0.950	26.5%	-0.003	0.089	0.087	0.952	60.000%	0.136	0.426	0.406	0.963
4	47.000%	0.403	1.607	1.335	0.890	22.667%	0.061	0.908	0.920	0.961	3.000%	-0.039	0.625	0.645	0.962	0.667%	0.065	0.476	0.460	0.926	0.000%	-0.004	0.270	0.289	0.955	19.500%	0.128	0.899	0.925	0.975
5	58.333%	0.238	1.092	1.056	0.956	36.000%	-0.010	0.726	0.733	0.948	15.333%	< 0.001	0.489	0.518	0.961	3.000%	-0.001	0.349	0.366	0.973	0.000%	0.017	0.239	0.231	0.950	33.000%	0.104	0.744	0.741	0.955
6	46.000%	0.037	0.559	0.621	0.962	48.000%	0.030	0.461	0.441	0.942	44.000%	0.019	0.304	0.311	0.940	38.000%	-0.003	0.235	0.219	0.941	35.500%	-0.006	0.143	0.139	0.977	39.500%	0.007	0.436	0.441	0.950
7	55.000%	0.078	0.0.368	0.396	0.967	47.333%	-0.015	0.266	0.273	0.949	40.333%	0.005	0.188	0.195	0.961	32.667%	0.003	0.145	0.138	0.926	26.000%	0.011	0.086	0.088	0.966	40.000%	< 0.001	0.199	0.194	0.958
8	51.667%	0.084	0.677	0.631	0.952	41.667%	0.044	0.467	0.442	0.937	35.667%	0.065	0.326	0.315	0.927	35.667%	0.018	0.229	0.221	0.933	31.000%	0.022	0.134	0.140	0.964	40.000%	-0.047	0.309	0.308	0.950
9	43.000%	0.216	1.143	1.319	0.974	19.333%	0.025	0.853	0.918	0.975	4.333%	0.050	0.631	0.650	0.965	0.000%	0.003	0.428	0.457	0.963	0.000%	0.014	0.312	0.290	0.935	7.000%	-0.021	0.603	0.646	0.952
10	55.000%	0.166	1.089	1.048	0.967	32.333%	0.086	0.701	0.738	0.966	15.333%	0.003	0.491	0.518	0.965	2.333%	-0.015	0.344	0.365	0.956	0.000%	0.007	0.228	0.231	0.960	2.000%	0.021	0.376	0.366	0.949
11	60.000%	0.028	0.414	0.387	0.910	51.667%	0.046	0.310	0.278	0.931	45.333%	-0.012	0.205	0.193	0.951	37.000%	0.009	0.142	0.138	0.926	21.500%	-0.002	0.081	0.087	0.962	33.500%	-0.006	0.135	0.138	0.955
12	42.000%	0.404	1.415	1.341	0.925	19.000%	0.071	0.947	0.921	0.951	5.000%	0.061	0.685	0.650	0.958	0.000%	0.019	0.471	0.458	0.953	0.000%	0.005	0.301	0.289	0.950	0.000%	-0.009	0.503	0.457	0.925

4.5 Data Analysis

We continue to analyze the Vertebral Column dataset from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/vertebral+column) as we do in Chapter 3. We consider the same marginal models as we do in Section 3.6 of Chapter 3 and we are still interested in studying the dependence of same 6 pairs of features: `PI` versus `PT`, `PI` versus `SS`, `DS` versus `PI`, `DS` versus `PT`, `DS` versus `PR`, and `DS` versus `LL`. Here we use the M-DPM-CM to identify common dependence structures out of the 18 pairs of features (6 pairs of features in 3 health groups) and select copula functions for the identified groups. To do so, we imagine our data coming from a hierarchical structure with 18 clusters in the intermediate level.

4.5.1 Marginal Model

The histograms of the five biomechanical features in the three groups are displayed in Figure B.2 in Appendix B.3.1, all showing unimodal but possibly skewed distributions. As a result, we use a generalized skewed-t distribution to model the marginal distributions of the features to account for the possible skewness. The estimates of the marginal parameters are obtained by maximizing the marginal likelihood function, and the results are summarized in Table B.6 in the Appendix B.3.1. The five biomechanical features are transformed to copula data $u_{jik} \in [0, 1]$ through applying the fitted marginal CDF to the observed values of the corresponding feature.

4.5.2 Dependence Model

We consider the same set of copula functions used for the simulation studies in Section 4.4. We compare the performance of the M-DPM-CM and the AIC for copula selection and conduct MLE under selected models. The results are reported in Table 4.6.

Empirical results for Kendall's τ of each pair of features from every health group are reported in the last column in Table 4.6. Generally speaking, `DS` has mild dependence versus the other four features in the patients with Disk Hernia and healthy people, but stronger dependence in the group of patients with Spondilolisthesis.

M-DPM-CM divides the 12 pairs of features into three groups. The dependence structures of `DS` versus the other four features (`PI`, `PT`, `PR` and `LL`) for patients with Disk Hernia and healthy people are identified to be the same by M-DPM-CM, and the common copula

Table 4.6: Selected copula functions and estimated parameters for the dependence of six pairs of interest in three health groups

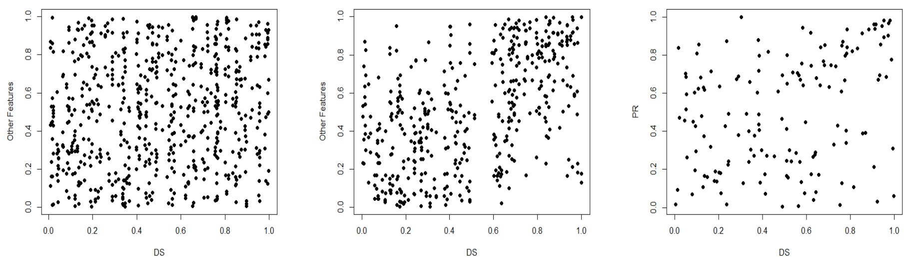
Health Group	Pairs of Features	M-DPM-CM Group	M-DPM-CM Copula	M-DPM-CM Estimates	M-DPM-CM s.d.	AIC Copula	AIC Estimates	AIC s.d.	Empirical Kendall's τ
Disk Hernia	`DS v.s. PI`	1	Gaussian	0.128	0.024	Gaussian	0.150	0.125	0.076
	`DS v.s. PT`	1	Gaussian	0.128	0.024	Frank	-0.226	0.753	-0.010
	`DS v.s. LL`	1	Gaussian	0.128	0.024	Gaussian	0.246	0.118	0.149
	`DS v.s. PR`	1	Gaussian	0.128	0.024	Gaussian	-0.060	0.128	-0.023
Spondilolisthesis	`DS v.s. PI`	2	Gumbel	1.437	0.029	Frank	3.453	0.535	0.355
	`DS v.s. PT`	2	Gumbel	1.437	0.029	S Clayton[1]	0.905	0.153	0.365
	`DS v.s. LL`	2	Gumbel	1.437	0.029	Frank	3.155	0.527	0.328
	`DS v.s. PR`	3	Joe	1.481	0.123	Joe	1.481	0.123	0.215
Healthy	`DS v.s. PI`	1	Gaussian	0.128	0.024	Frank	1.714	0.628	0.179
	`DS v.s. PT`	1	Gaussian	0.128	0.024	Gaussian	0.244	0.091	0.172
	`DS v.s. LL`	1	Gaussian	0.128	0.024	Frank	1.511	0.600	0.157
	`DS v.s. PR`	1	Gaussian	0.128	0.024	Gaussian	-0.107	0.098	-0.095

[1] Survival Clayton Copula

form selected is a Gaussian copula. When pooling the data of four bivariate features (`DS` versus `PI`, `DS` versus `PT`, `DS` versus `PR`, and `DS` versus `LL`) from two health group (patients with Disk Hernia and healthy people) together, the empirical Kendall's τ is calculated as 0.082 (with standard error 0.026), which suggests that `DS` is barely dependent to the features characterizing pelvis and lumbar for patients with Disk Hernia and healthy people. The dependence structures of `DS` versus `PI`, `PT` and `LL` in the group of patients with Spondilolisthesis are grouped together by the M-DPM-CM with the selected copula form as Gumbel and the empirical Kendall's τ is around 0.35 (with standard error 0.027). The dependence of `DS` and `PR` is identified as a group itself with the selected copula form as Joe copula.

In Figure 4.1, we report the scatter plots of `DS` versus other features based on the combined datasets of the three groups identified by the M-DPM-CM. Subfigure (a) corresponds to four pairs of features in patients with Disk Hernia and healthy people and exhibits pure randomness; subfigures (b) and (c) correspond to the dependence in the group of patients with Spondilolisthesis and show moderate positive dependence. The empirical findings, grouping by M-DPM-CM and graphics tell the same story, which is also consistent with the medical interpretation (Berthonnaud et al., 2005).

In summary, M-DPM-CM provides some insights of the dependence between different features of three types of people. The estimation of dependence parameters is also more efficient due to the grouping effect of M-DPM-CM.



(a) Group 1 (b) Group 2 (c) Group 3

Figure 4.1: Scatter plots for the three groups identified by the M-DPM-CM

4.6 General Remarks

In this chapter, the mixture of DPM copula model (M-DPM-CM) is developed to identify similar dependence structures for correlated data and group similar data together to obtain better inference results. The M-DPM-CM can perform grouping and copula selection simultaneously. The numerical results show that the M-DPM-CM can accurately recover the true grouping structure with a moderate sample size, and in turn achieve a more accurate model selection and more efficient parameter estimation than the conventional AIC method. Moreover, the M-DPM-CM requires little tuning or user-specified parameters compared with other commonly used models, such as Gaussian mixture model (Lindsay, 1995; McLachlan and Peel, 2004), so that it is easy to be applied in practice.

Chapter 5

Polya Tree Monte Carlo Method

5.1 Introduction

Sampling from a distribution has been an important research topic in statistics and enjoys broad applications in different contexts, including the Bayesian framework and the machine learning paradigm(e.g., Goodfellow et al., 2014).

When the inverse of a cumulative distribution function (CDF) is available, sampling from the distribution is commonly conducted through the "inversion method" (Devroye, 1986). In most situations where the explicit inverse of CDF is unavailable, Markov Chain Monte Carlo (MCMC) methods are commonly invoked (Gelfand and Smith, 1990; Gilks et al., 1995; Brooks et al., 2011; Craiu and Rosenthal, 2014). Commonly-used MCMC algorithms include Metropolis-Hasting (MH) algorithm (Hastings, 1970), and Gibbs sampler (Geman and Geman, 1987).

While MCMC algorithms are useful in applications, they have several limitations. Samples generated by MCMC can be highly correlated and may not be diverse enough to reasonably reflect the domain space of the target distribution (Brooks et al., 2011). To reach convergence, MCMC algorithms may require a large number of iterations (Gelman and Rubin, 1992; Cowles and Carlin, 1996) and carefully tuned stepsizes to achieve an suitable acceptance rate (Graves, 2011). Furthermore, MCMC algorithms can be inefficient in sampling from multi-modal distributions (Gelman and Rubin, 1992; Geyer and Thompson, 1995; Neal, 1996).

To overcome these limitations of the MCMC algorithms, various methods have been proposed. For instance, to address the issues of correlated samples, it was suggested to use independent proposal density to approximate the target distribution, where approximations may be conducted through multivariate normal distributions (Haario et al., 2001), finite mixture distributions (e.g., Cappé et al., 2008; Keith et al., 2008; Holden et al., 2009; Giordani and Kohn, 2010), piecewise approximating functions (Cai et al., 2008), or neural networks (Neklyudov et al., 2018). Adaptive algorithms (Haario et al., 2001; Atchadé and Rosenthal, 2005) and the Delayed Rejection Adaptive Metropolis (DRAM) (Haario et al., 2006) were developed to achieve efficient sampling procedures. Methods concerning the step-size tuning in MCMC were discussed by Graves (2011) and Kleppe (2016). Methods of efficiently exploring the domain space were considered by Gelman and Rubin (1992), Geyer and Thompson (1995), Neal (1996), Richardson and Green (1997), and Kou et al. (2006) for sampling from multi-modal distribution. Under the adaptive MCMC framework, Giordani and Kohn (2010), Andrieu and Thoms (2008), Craiu et al. (2009), Bai et al. (2011) and Zhang et al. (2019) also extended the algorithms naturally to handle multi-modal distributions.

As a complement to available methods, in this chapter, we propose a novel sampling method, called Polya tree Monte Carlo (PTMC), to address the aforementioned limitations of MCMC algorithms. Our proposed PTMC method approximates the posterior Polya tree by the Monte Carlo method and it can be established theoretically that the approximated Polya tree posterior converges to the target distribution under regularity conditions. We further propose a series of simple and efficient sampling algorithms which are useful for different scenarios. It is noteworthy that our proposed algorithm is completely different from the "Polya tree sampler" discussed by Hanson et al. (2011). This method posteriorly updates the Polya tree with a simulated sample via time-consuming iterative procedures, while our PTMC method approximates the posterior Polya tree using the Monte Carlo method in a fast and straightforward manner.

The rest of the chapter is organized as follows. In Section 5.2, we describe the proposed Polya tree Monte Carlo (PTMC) method and several sampling algorithms. In Section 5.3, we perform simulation studies to evaluate the finite sample performance of the proposed PTMC method and compare it with the MCMC algorithm. In Section 5.4, we analyze two heterogeneous datasets based on Gaussian mixture models, and examine the capacity of the PTMC algorithms for sampling from complex multi-modal distributions.

5.2 Polya Tree Monte Carlo Method

In this section, we introduce a novel sampling method, the Polya tree Monte Carlo (PTMC) method. The detailed review of the Polya tree can be found on Section 1.7.2. In Section

5.2.1, we propose the Polya Tree Monte Carlo (PTMC) method, and provide theoretical results. In Section 5.2.2, we develop a variety of sampling algorithms based on the theoretical results of the PTMC method.

5.2.1 Polya Tree Monte Carlo Method

As reviewed in Section 1.7.2, Polya trees are conventionally used as Bayesian nonparametric priors when a random sample is available to make inference about the unknown distribution. However, our interest is to sample from a known target distribution, and it can be difficult under certain circumstances (e.g., when the target distribution has no explicit inverse form of CDF). To resolve the difficulty, we consider sampling from the empirical counterpart of the target distribution via the PT posterior distribution. Since the PT posterior distribution is obtained from the samples from the target distribution, we propose the Polya Tree Monte Carlo (PTMC) method to approximate the PT posterior using the Monte Carlo (MC) method.

Suppose that we are interested in sampling from the distribution of the random variable Y with domain $\mathcal{S}$, probability measure $\mathcal{F}$ and density function f. As we eventually sample from the empirical counterpart (a histogram) of the target distribution, it is convenient to focus on a bounded sub-region. To this end, we define a "high probability region" $\mathcal{S}^*$, a bounded space such that $\mathcal{F}(\mathcal{S}^*) = 1 - \delta$ with a small $0 \leq \delta < 1$; if $\mathcal{S}$ is bounded, then we set $\mathcal{S}^* = \mathcal{S}$ and $\delta = 0$. Further, we consider a random variable Y^* with domain $\mathcal{S}^*$, the scaled probability measure $\mathcal{F}/(1-\delta)$ and the density function $f/(1-\delta)$. A PT model is assumed for Y^*, such that the prior G^* and the posterior $G^*|Y^*$ follow PT distributions defined on $\mathcal{S}^*$, i.e.,

$$
\begin{aligned}
Y^*|G^* &\sim G^*, \\
G^* &\sim PT(\Pi^*, \mathcal{A}^*), \\
G^*|Y^* &\sim PT(\Pi^*, \mathcal{A}^*(Y^*)),
\end{aligned}
$$

where $\Pi^* = \{\pi^*_m : m \in N^+\}$ is a collection of nested partitions of the space $\mathcal{S}^*$ with $\pi^*_m = \{\mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_m} : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$ being the m-level partition of space $\mathcal{S}^*$; $\mathcal{A}^* = \{\mathcal{A}^*_m : m \in N^+\}$ is a collection of positive parameters indexing the prior distribution with $\mathcal{A}^*_m = \{\alpha^*_{\varepsilon_1 \ldots \varepsilon_m} : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$. $\mathcal{A}^*(Y^*) = \{\mathcal{A}^*_m(Y^*) : m \in N^+\}$ is a collection of parameters indexing the posterior distribution with $\mathcal{A}^*_m(Y^*) = \{\alpha^*_{\varepsilon_1 \ldots \varepsilon_m}(Y^*) : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$ and

$$
\alpha^*_{\varepsilon_1 \ldots \varepsilon_m}(Y) = \begin{cases} \alpha^*_{\varepsilon_1 \ldots \varepsilon_m} + 1 & \text{if } Y^* \in \mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_m}, \\ \alpha^*_{\varepsilon_1 \ldots \varepsilon_m} & \text{otherwise.} \end{cases}
$$

Let n^* denote a user-specified positive integer related to the MC approximation to be discussed later. Suppose $(Y_1^*, \ldots, Y_{n^*}^*)$ is a i.i.d. random sample having the same distribution as that of Y^*. Analogous to (1.10), the random conditional probabilities, given $(Y_1^*, \ldots, Y_{n^*}^*)$, are of the form

$$G^*_{\varepsilon_1\ldots\varepsilon_{m-1}0}|(Y_1^*, \ldots, Y_{n^*}^*) \sim \text{Beta}\left(\alpha^*_{\varepsilon_1\ldots\varepsilon_{m-1}0} + N^*_{\varepsilon_1\ldots\varepsilon_{m-1}0}, \alpha^*_{\varepsilon_1\ldots\varepsilon_{m-1}1} + N^*_{\varepsilon_1\ldots\varepsilon_{m-1}1}\right), \quad (5.1)$$

where $N^*_{\varepsilon_1\ldots\varepsilon_m}$ is the number of sample points in $(Y_1^*, \ldots, Y_{n^*}^*)$ that falls in the subset $\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m}$.

Next, we consider a probability measure to approximate the posterior PT distribution $PT(\Pi^*, \mathcal{A}^*(Y^*))$. Suppose that $\mathcal{G}_U \sim PT(\Pi^*, \mathcal{A}^\dagger(U))$ is a PT based on the same collection of nested partitions Π^* of $\mathcal{S}^*$, but indexed by a different set of parameters $\mathcal{A}^\dagger(U) = \{\mathcal{A}^\dagger_m(U) : m \in N^+\}$ with $\mathcal{A}^\dagger_m(U) = \{\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_m}(U) : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$ and

$$\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_m}(U) = \begin{cases} \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_m} + f(U) & \text{if } U \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m} \text{ and } m \leq M, \\ \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_m} & \text{otherwise,} \end{cases}$$

where U is a uniform random variable on $\mathcal{S}^*$, and $M \in N^+$ is a pre-specified "truncated level" to approximate $PT(\Pi^*, \mathcal{A}^*(Y^*))$ up to a finite level. Suppose that $\tilde{U} = (U_1, \ldots, U_{n^*})$ includes i.i.d. uniform random variables on $\mathcal{S}^*$. Let $\mathcal{G}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}$ be the random conditional probabilities from different levels based on $\tilde{U}$, which are assumed to be independent Beta random variables with

$$\mathcal{G}_{\varepsilon_1\ldots\varepsilon_{m-1}0}|\tilde{U} \sim \text{Beta}\left(\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{m-1}0} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{m-1}0}) f(U_i), \right.$$
$$\left. \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{m-1}1} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{m-1}1}) f(U_i)\right) \quad (5.2)$$

if $m \leq M$, and

$$\mathcal{G}_{\varepsilon_1\ldots\varepsilon_{m-1}0} \sim \text{Beta}\left(\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{m-1}0}, \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{m-1}1}\right) \quad (5.3)$$

if $m > M$. For the target distribution with dimension higher than 1, the PTMC method can be constructed in a similar manner by replacing the Beta random variable in (5.2) and (5.3) with the Dirichlet random variable. Let $\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})$ denote the random probability of the subset $\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m}$ from the PT constructed based on $\tilde{U}$. Then the expected value of

$\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})$, given the uniform sample $\tilde{U}$, is

$$E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})] = E\left[\prod_{j=1}^{m}\mathcal{G}_{\varepsilon_1\ldots\varepsilon_j}|\tilde{U}\right]$$

$$= \begin{cases} \prod_{j=1}^{m} \frac{\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_j}+\sum_{i=1}^{n^*} I(U_i\in\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_j})f(U_i)}{\sum_{l=0}^{1}\left[\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_{j-1}l}+\sum_{i=1}^{n^*} I(U_i\in\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(U_i)\right]} & \text{if } m \le M, \\ \prod_{j=1}^{M} \frac{\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_j}+\sum_{i=1}^{n^*} I(U_i\in\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_j})f(U_i)}{\sum_{l=0}^{1}\left[\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_{j-1}l}+\sum_{i=1}^{n^*} I(U_i\in\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(U_i)\right]} \prod_{j=M+1}^{m} \frac{\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_j}}{\sum_{l=0}^{1}\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_{j-1}l}} & \\ & \text{if } m > M. \end{cases} \tag{5.4}$$

To see the rationale of using $PT(\Pi^*, \mathcal{A}^{\dagger}(U))$ to approximate $PT(\Pi^*, \mathcal{A}^*(U))$, we note that the probability for Y^* to fall in ${\mathcal{B}^*}_{\varepsilon_1\ldots\varepsilon_m}$ is

$$P(Y^* \in {\mathcal{B}^*}_{\varepsilon_1\ldots\varepsilon_m}) = \frac{\mathcal{F}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})}{1-\delta} = \frac{1}{1-\delta}\int_{\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m}} f(y)dy \tag{5.5}$$

$$= \frac{1}{1-\delta}\cdot\frac{w_{\mathcal{S}^*}}{n^*}\sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})f(U_i) + O_p(\frac{1}{\sqrt{n^*}}), \tag{5.6}$$

where $w_{\mathcal{S}^*}$ is the volume of $\mathcal{S}^*$ and (5.6) is a Monte Carlo (MC) approximation of (5.5) (Gilks et al., 1995; Brooks et al., 2011). Since $N^*_{\varepsilon_1\ldots\varepsilon_m} = \sum_{i=1}^{n^*} I(Y_i^* \in \mathcal{B}^*)$ with $I(\cdot)$ being the indicator function, and

$$E[I(Y_i^* \in \mathcal{B}^*)] = P(Y^* \in \mathcal{B}^*)$$
$$Var[I(Y_i^* \in \mathcal{B}^*)] = P(Y^* \in \mathcal{B}^*)[1 - P(Y^* \in \mathcal{B}^*)],$$

by the Central Limit Theorem,

$$\frac{N^*_{\varepsilon_1\ldots\varepsilon_m} - n^*P(Y^* \in \mathcal{B}^*)}{\sqrt{n^*P(Y^* \in \mathcal{B}^*)[1-P(Y^* \in \mathcal{B}^*)]}} \xrightarrow{d} N(0,1) \qquad \text{as } n^* \to \infty \tag{5.7}$$

Combining (5.6) and (5.7) yields

$$N^*_{\varepsilon_1\ldots\varepsilon_m} = n^*P(Y^* \in {\mathcal{B}^*}_{\varepsilon_1\ldots\varepsilon_m}) + O_p(\sqrt{n^*})$$
$$= n^*\frac{\mathcal{F}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m})}{1-\delta} + O_p(\sqrt{n^*})$$

$$= \quad \frac{w_{\mathcal{S}^*}}{1-\delta}\sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m}) f(U_i) + O_p(\sqrt{n^*}),$$

thus the quantity $N^*_{\varepsilon_1\ldots\varepsilon_m}$ in (5.1) can be approximated using $\sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_m}) f(U_i)$ to derive the distribution of the random conditional probabilities concerning $\widetilde{U}$ in (5.2) naturally. The theoretical results stay unaffected if the constant term $w_{\mathcal{S}^*}/(1-\delta)$ is omitted as the proof in Appendix D.1.

The following theorems show the theoretical results for the Polya Tree Monte Carlo method.

Theorem 5.1. *(Pointwise Convergence of Polya-Tree Monte Carlo) For any measurable set $B \in \pi^*_m$ with $m = 1, \ldots, M$, and $n^* = O(M^{3+\eta})$ with $\eta > 0$, then*

(1) $E[\mathcal{G}_{\tilde{U}}(B)] \xrightarrow{p} \mathcal{F}(B)/(1-\delta)$ *as* $M \to \infty$*;*

(2) $Var[\mathcal{G}_{\tilde{U}}(B)] = O_p(\frac{M}{n^*})$*;*

(3) $\mathcal{G}_{\tilde{U}}(B) \xrightarrow{p} \mathcal{F}(B)/(1-\delta)$ *as* $M \to \infty$*.*

Theorem 5.2. *(Consistency of Polya-Tree Monte Carlo) Suppose $\mathcal{F}$ is the Lebesgue measure with the absolute continuous density f on $\mathcal{S}^*$. Let $\mathfrak{S}^* = \{B \subset \mathcal{S}^* : B \text{ is measurable}\}$, $\mathfrak{V} = \{\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M} | \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M} \in \pi^*_M; \mathcal{F}(\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M}) > 0\}$ and let $\gamma(M) = \min_{B\in\mathfrak{V}} \mathcal{F}(B)/(1-\delta)$. Then the following properties hold:*

(1) $\sup_{B\in\mathfrak{S}^*} E[\mathcal{G}_{\tilde{U}}(B)] - \mathcal{F}(B)/(1-\delta) = max\Big(O_p\big(\frac{M}{\sqrt{n^*}\gamma(M)}\big), O_p\big(\frac{M^3}{n^*\gamma(M)}\big)\Big)$*;*

(2) $\sup_{B\in\mathfrak{S}^*} Var[\mathcal{G}_{\tilde{U}}(B)] = O_p(\frac{M}{n^*\gamma(M)})$*;*

(3) Let $g_{\tilde{U}}$ be the density of $\mathcal{G}_{\tilde{U}}$, and let $D(\mathcal{G}_{\tilde{U}}, \mathcal{F}/(1-\delta)) = \int_{\mathcal{S}^*} \big|g_{\tilde{U}}(x) - f(x)/(1-\delta)\big| dx$ denote the distance between two probability measures $\mathcal{G}_{\tilde{U}}$ and $\mathcal{F}$. If $n^* = O(2^{5M} M^{3+\eta})$ with $\eta > 0$, then as $M \to \infty$,

$$P[D(\mathcal{G}_{\tilde{U}}, \mathcal{F}/(1-\delta)) \geq \epsilon] \to 0$$

for any $\epsilon > 0$.

The proofs of both theorems are provided in Appendix D.1. Theorems 5.1 and 5.2 show that $\mathcal{G}_{\tilde{U}}$ converges to the scaled target distribution on the high probability region $\mathcal{S}^*$. If the target distribution is defined on a bounded space $\mathcal{S}$, then $\delta = 0$ and $\mathcal{G}_{\tilde{U}}$ approximates the target distribution well with a sufficiently large uniform sample. If the target distribution has unbounded space $\mathcal{S}$, the high probability space $\mathcal{S}^*$ can be constructed with an ignorably small δ, and $\mathcal{G}_{\tilde{U}}$ can still provide a reasonably good approximation to the target distribution. While the theoretical results are presented for one-dimension case for notation simplicity, the results can be extended to settings with a higher dimension. The relevant proof is analogous to the proofs regarding the Polya tree with a higher dimension (Ning and Shephard, 2018).

5.2.2 Sampling Algorithms

To sample from $\mathcal{G}_{\tilde{U}}$, we describe four algorithms. Algorithm 5.1 is outlined in Table 5.1. The first M levels of the Polya tree correspond to a sequence of histograms with increasingly finer bins. With the histogram corresponding to the M-level partition of the Polya tree, Step 4 of Algorithm 5.1 elects the bin for generating samples. For any $m > M$, it is easily seen from (5.3) that

$$E(\mathcal{G}_{\varepsilon_1 \ldots \varepsilon_{m-1} 0}) \quad = \quad \frac{\alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_{m-1} 0}}{\alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_{m-1} 0} + \alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_{m-1} 1}} \quad = \quad \frac{1}{2},$$

when $\alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_m}$ takes its default value ϕm^2 with $\phi > 0$. In other words, a sample falling in $\mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_M}$ is uniformly distributed on $\mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_M}$. Thus, Step 5 further generates a uniform random variable on a selected bin as a sample point.

Compared to the usual MCMC, which provides correlated samples, Algorithm 5.1 provides independent samples. Algorithm 5.1 requires to evaluate the density function $f(\cdot)$ for a fixed number of times n^*, whereas in MCMC, the number of evaluations of $f(\cdot)$ depends on the convergence speed and the target sample size n. As a result, Algorithm 5.1 is superior to the MCMC algorithm in terms of computational time when the evaluation of the density function $f(\cdot)$ is time-consuming and/or a large sample needs to be generated. Furthermore, the evaluation of the density function $f(\cdot)$ in Algorithm 5.1 can be accelerated through parallel computing techniques, which is another advantage over the MCMC algorithm.

Our discussion so far has been focused on the single-dimensional scenario. When Y is a random vector of the dimension $k \geq 2$, as discussed in Section 1.7.2, the M-level partition

Table 5.1: Algorithm 5.1: Polya Tree Monte Carlo algorithm

Input: Sample size n from $f(y)$, the number of uniform samples n^* and $M = 9$ (default).

1. Generate i.i.d. samples $u_1, \ldots, u_{n^*}$ from a uniform distribution on $\mathcal{S}^*$.
2. Evaluate the density value of the target distribution $f(u_i)$, for $i = 1, \ldots, n^*$.
3. For all $\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M} \in \pi^*_M$, calculate the expected probability $E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M})]$ using (5.4).
4. Sample n subspaces $\mathcal{B}^{*(i)}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M}$ with replacement based on $E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M})]$, for $i = 1, \ldots, n$.
5. **for** i *in* $1 : n$ **do**
    Generate y_i from a uniform distribution on $\mathcal{B}^{*(i)}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M}$.

end

of the Polya tree splits $\mathcal{S}^*$ into 2^{kM} subsets. When k is large, it is computationally intensive to evaluate the expected probabilities for 2^{kM} times in Step 4 of Algorithm 5.1. To cope with this problem, we design Algorithm 5.2, which takes the strategy to examine the multi-dimensional space dimension by dimension and sample directly from the marginal density,

$$f(y_\ell) = \int f(y_1, \ldots, y_k) dy_{(-\ell)}, \tag{5.8}$$

where $y_\ell \in \mathcal{S}^{*\{\ell\}}$ and $y_{(-\ell)} = (y_1, \ldots, y_{\ell-1}, y_{\ell+1}, \ldots, y_k)^{\mathrm{T}}$ for $\ell = 1, \ldots, k$.

To be specific, we define $\Pi^{*\{\ell\}} = \{\pi^{*\{\ell\}}_m : m \in N^+\}$ as a collection of nested and equal-sized partitions of the space $\mathcal{S}^{*\{\ell\}}$, where $\pi^{*\{\ell\}}_m = \{\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m} : \varepsilon_j \in \{0,1\}, j = 1, \ldots, m\}$. For subset $\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}$ at the m-level partition of $\mathcal{S}^{*\{\ell\}}$, the probability that Y_ℓ falls in $\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}$ is

$$\int_{\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}} f(y_\ell) dy_\ell = \int_{\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}} \int f(y_1, \ldots, y_k) dy_{(-\ell)} dy_\ell,$$

which can be approximated using the MC method by

$$\frac{w_{\mathcal{S}^*}}{n^*} \sum_{i=1}^{n^*} I(U_{i\ell} \in \mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}) f\left(U_{i1}, U_{i2}, \ldots, U_{ik}\right),$$

where $U_{i\ell}$ denotes the ℓth element of U_i and U_i represents a uniform sample from $\mathcal{S}^*$ for $i = 1, \ldots, n^*$.

We propose a marginal measure $\mathcal{G}^{\{\ell\}}_{U_\ell} \sim PT(\Pi^{*\{\ell\}}, \mathcal{A}^{\{\ell\}}(U_\ell))$, where $\mathcal{A}^{\{\ell\}}(U_\ell) = \{\mathcal{A}^{\{\ell\}}_m(U_\ell) : m \in N^+\}$, $\mathcal{A}^{\{\ell\}}_m(U_\ell) = \{\alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}(U_\ell) : \varepsilon_j \in \{0,1\}, j = 1,\ldots,m\}$, and

$$\alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m}(U_\ell) = \begin{cases} \alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m} + f(U_\ell) & \text{if } U_\ell \in \mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m} \text{ and } m \le M, \\ \alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m} & \text{otherwise.} \end{cases}$$

The random conditional probabilities $\mathcal{G}^{\{\ell\}}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_{m-1}0}$ from different levels are assumed to be mutually independent Beta random variables with

$$\mathcal{G}^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}0}|\tilde{U} \sim \text{Beta}\Bigg(\alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}0} + \sum_{i=1}^{n^*} I(U_{i\ell} \in \mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}0}) f(U_i), \\ \alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}1} + \sum_{i=1}^{n^*} I(U_{i\ell} \in \mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}1}) f(U_i)\Bigg)$$

if $m \le M$, and

$$\mathcal{G}^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}0} \sim \text{Beta}\left(\alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}0}, \alpha^{\{\ell\}}_{\varepsilon_1\ldots\varepsilon_{m-1}1}\right)$$

if $m > M$. The expected value of $\mathcal{G}^{\{\ell\}}_{\tilde{U}}(\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_m})$ can be similarly derived as in (5.4).

Table 5.2: Algorithm 5.2: Polya-Tree Monte Carlo algorithm for $k \ge 2$

Input: Sample size n from $f(y)$, the number of uniform samples n^* and $M = 9$ (default).

1. Generate i.i.d. samples $u_1, \ldots, u_{n^*}$ from a uniform distribution on $\mathcal{S}^*$.
2. Evaluate the density value of the target distribution $f(u_i)$, for $i = 1, \ldots, n^*$.
3. **for** ℓ *in* $1 : k$ **do**
   (1) For all $\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\ldots\varepsilon_M} \in \pi^{*\{\ell\}}_M$, calculate the expected probability $E[\mathcal{G}^{\{\ell\}}_{\tilde{U}}(\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M})]$.
   (2) Sample n subspaces $\mathcal{B}^{*\{\ell\}(i)}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M}$ with replacement based on $E[\mathcal{G}^{\{\ell\}}_{\tilde{U}}(\mathcal{B}^{*\{\ell\}}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M})]$, for $i = 1, \ldots, n$.
   (3) **for** i *in* $1 : n$ **do**
      Generate $y_{i\ell}$ from a uniform distribution on $\mathcal{B}^{*\{\ell\}(i)}_{\varepsilon_1\varepsilon_2\ldots\varepsilon_M}$.
   end
end

In Algorithm 5.2, we sample from $\mathcal{G}_{\tilde{U}}^{\{\ell\}}$ for $l = 1, \ldots, k$ sequentially. Algorithm 5.2 reduces the computational burden from 2^{kM} times in Algorithm 5.1 to $k \cdot 2^M$ calculations of the expected probabilities. However, due to the "curse of dimensionality" suffered by the Polya tree, both algorithms require a large uniform sample, i.e., a large n^*, to achieve an accurate approximation of the target distribution when k is large. Therefore, we further propose Algorithm 5.3 to combine the PTMC with Gibbs sampler in high-dimensional settings to avoid massive computation in Algorithm 5.2.

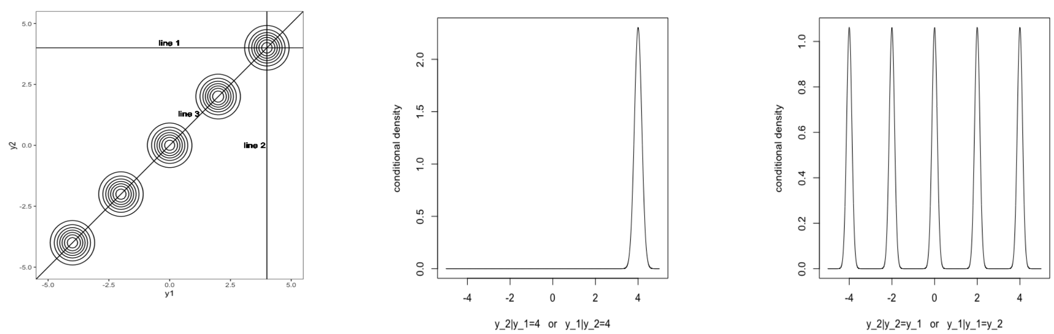
In Algorithm 5.3, sampling from the conditional distribution in each iteration of Gibbs sampler is conducted through the PTMC Algorithm 5.1, which is powerful in single-dimensional scenario. Compared to the MCMC algorithm, the PTMC Gibbs sampler is free of tuning parameters and enjoys high sampling efficiency and convergence rate to the target distribution as illustrated through simulation studies.

Table 5.3: Algorithm 5.3: PTMC Gibbs sampler for a high-dimensional distribution

Input: Sample size n from $f(y)$, the number of uniform samples $n^* = 500$ (default), burn-in sample size b_1, initial values $y^1 = (y_1^1, \ldots, y_k^1)^{\mathrm{T}}$ and $M = 9$ (default)

for t *in* $2 : (n + b_1)$ **do**
  for ℓ *in* $1 : k$ **do**
    Generate one sample from a single-dimensional distribution $f(y_\ell | y_1^t, \ldots, y_{\ell-1}^t, y_{\ell+1}^{t-1}, \ldots, y_k^{t-1})$ for $y_\ell \in \mathcal{S}^{*\{\ell\}}$, which is proportional to $f(y_1^t, \ldots, y_{\ell-1}^t, y_\ell, y_{\ell+1}^{t-1}, \ldots, y_k^{t-1})$ using Algorithm 5.1 and set it to be y_ℓ^t.
  end
end

The PTMC Gibbs sampler (Algorithm 5.3) cannot handle some complex multi-modal distributions. A bivariate normal-mixture distribution with five modes can be considered as an example, with a contour plot given in Figure 5.1 (a). The PTMC Gibbs sampler updates the sample values through the horizontal and vertical lines (e.g., lines 1 and 2 in Figure 5.1 (a), respectively). The conditional densities of the PTMC Gibbs sampler is provided in Figure 5.1 (b) and apparently, the algorithm is trapped in this mode. This motivates us to propose Algorithm 5.4, which considers searching values through a general linear line $y_2 = ay_1 + b$. If the line is $y_2 = y_1$ (i.e., $a = 1$ and $b = 0$) as line 3 in Figure 5.1 (a) with the conditional density along the line provided in Figure 5.1 (c), the other modes can be easily discovered.



(a) Contour of the target distribution (b) Conditional density of the PTMH Gibbs Sampler (c) Conditional density if updating along line 3

Figure 5.1: An example of a multimodal distribution

Algorithm 5.4 basically combines the PTMC and Metropolis-Hasting algorithms, and we call it the PTMC-MH algorithm. For a k-dimensional target distribution with the density $f(y_1, \ldots, y_k)$, we select y_1 as the reference dimension, and assume a linear relationship between y_ℓ and y_1 for $\ell = 2, \ldots, k$. In the tth iteration of the PTMC-MH algorithm, we assume that y_ℓ can be written as a linear transformation of y_1 with $y_\ell^t = a_\ell^t y_1^t + b_\ell^t$, where a_ℓ^t and b_ℓ^t denote the slope and intercept of the line, respectively. The slope a_ℓ^t will be randomly generated in each iteration so that different directions of the domain space will be explored. As the line $y_\ell^t = a_\ell^t y_1^t + b_\ell^t$ is required to cross the point $(y_1^{t-1}, y_\ell^{t-1})$ from iteration $(t-1)$, the value of b_ℓ^t is determined as $b_\ell^t = y_\ell^{t-1} - a_\ell^t y_1^{t-1}$. We let $A_\ell^t = \{y_1^t \in \mathcal{S}^{*\{1\}} | y_\ell^t = a_\ell^t y_1^t + b_\ell^t \text{ for } y_\ell^t \in \mathcal{S}^{*\{\ell\}}\}$ denote a set of values that y_1 can take, restricted by the values that y_ℓ can take for $\ell = 2, \ldots, k$; and let $A^t = \bigcap_{\ell=2}^k A_\ell^t$ denote the values that y_1 can take jointly determined by the $k-1$ lines and the space $\mathcal{S}^*$ in the tth iteration.

In the tth iteration, the proposal distribution for $y_1 \in A^t$ is proposed to be

$$q(y|y^{t-1}) = \frac{f(y_1, a_2^t y_1 + b_2^t, \ldots, a_k^t y_1 + b_k^t)}{\int_{A^t} f(y_1, a_2^t y_1 + b_2^t, \ldots, a_k^t y_1 + b_k^t) dy_1}, \tag{5.9}$$

$$\propto f(y_1, a_2^t y_1 + b_2^t, \ldots, a_k^t y_1 + b_k^t). \tag{5.10}$$

The proposed PTMC-MH algorithm searches for an update of $y_1, \ldots, y_k$ along the line $\{(y_1, \ldots, y_k) : y_\ell = a_\ell^t y_1 + b_\ell^t, \text{ for } y_\ell \in \mathcal{S}^{*\{\ell\}} \text{ and } \ell = 2, \ldots, k\}$. The denominator of (5.9) is included so that (5.9) is a proper density. In each iteration, the PTMC-MH algorithm draws a new value of y_1 from the proposal distribution and $y_2, \ldots, y_k$ are determined by their linear relationship with y_1. As the proposal distribution in (5.10) is a single-dimensional distribution, the sampling procedure can be implemented using Algorithm 5.1. The acceptance rate of the PTMC-MH algorithm is

$$\frac{f(y)q(y^{t-1}|y)}{f(y^{t-1})q(y|y^{t-1})} = \frac{f(y)}{f(y^{t-1})} \cdot \frac{f(y_1^{t-1}, a_2^t y_1^{t-1} + b_2^t, \ldots, a_k^t y_1^{t-1} + b_k^t)}{f(y_1, a_2^t y_1 + b_2^t, \ldots, a_k^t y_1 + b_k^t)} = 1.$$

Therefore, we always accept the proposal values drawn from the proposal distribution.

Table 5.4: Algorithm 5.4: PTMC-MH algorithm for a high-dimensional distribution

Input: Sample size n from $f(y)$, the number of uniform samples $n^* = 500$ (default), burn-in sample size b_1, initial values $y^1 = (y_1^1, \ldots, y_k^1)^{\mathrm{T}}$ and $M = 9$ (default)

for t *in* $2 : (n + b_1)$ **do**

  1. **for** ℓ *in* $2 : k$ **do**

    (1) Generate $\theta_\ell^t \sim \text{Uniform}([-\frac{\pi}{2}, \frac{\pi}{2}])$.

    (2) Calculate $a_\ell^t = tan(\theta_\ell^t)$ and $b_\ell^t = y_\ell^{t-1} - a_\ell^t y_1^{t-1}$.

  end

  2. Determine the set A^t.

  3. Generate one sample y_1' from a single dimensional distribution $f(y_1, a_2^t y_1 + b_2^t, \ldots, a_k^t y_1 + b_k^t)$ for $y_1 \in A^t$ using Algorithm 5.1, and set $y^t = (y_1', a_2^t y_1' + b_2^t, \ldots, a_k^t y_1' + b_k^t)^{\mathrm{T}}$.

end

The PTMC-MH algorithm is computationally faster and more powerful than the PTMC Gibbs Sampler. More impressively, the PTMC-MH algorithm works well with complex multi-dimensional distributions as the sample points from each iteration of the PTMC-MH algorithm move according to lines with different slopes, and eventually reach all possible modes of the distribution with sufficient iterations. It is noteworthy that although the density function $f(y)$ is assumed to be known in the algorithm, the four algorithms are still working when the density function is partially known, such as a unnormalized density function.

5.3 Simulation Studies

We conduct extensive simulation studies to compare the performance of PTMC-based algorithms with MCMC algorithms in terms of the capability of recovering the target distribution, sampling efficiency, computational speed and inference performance.

5.3.1 Setting 5.1

In the first simulation, we compare the performance of the proposed PTMC algorithms with the random walk MCMC algorithm and the Langevin Monte Carlo (LMC) algorithm (Welling and Teh, 2011) when sampling from complex distributions. The simulation is repeated $nsim = 500$ times. The "burn-in" sample size b_1 is set to be 1000.

Simulation Setting

We draw $n = 5000$ samples from each of the following three distributions:

(1) The dog bowl distribution with the density function:

$$f(y_1, y_2) = \frac{1}{(2\pi)^{\frac{3}{2}}} \exp\left[-0.5\left(\sqrt{y_1^2 + y_2^2} - 10\right)^2\right](y_1^2 + y_2^2)^{-1/2} \text{ for } (y_1, y_2) \in \mathbb{R}^2.$$

(2) A 25-normal mixture distribution with the mixture density:

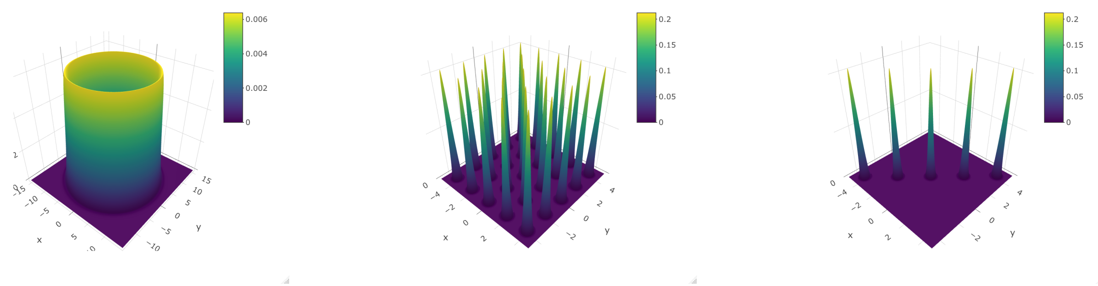
$$f(y_1, y_2) = \frac{1}{25}\sum_{\mu\in\Omega} \phi(y_1, y_2; \mu, \Sigma),$$

where $\phi(\cdot; \mu, \Sigma)$ is the density of a bivariate normal distribution with mean vector μ and covariance matrix $\Sigma = \begin{pmatrix} 0.03 & 0 \\ 0 & 0.03 \end{pmatrix}$, and $\Omega = \{(\mu_1, \mu_2) : \mu_j \in \{-4, -2, 0, 2, 4\}, j = 1, 2\}$.

(3) A 5-normal mixture distribution illustrated in Figure 5.1 (a), having the density:

$$f(y_1, y_2) = \frac{1}{5}\sum_{i=1}^{5} \phi(y_1, y_2; \mu_i, \Sigma),$$

where the mean vectors are set as $\mu_1 = (-4,-4)^{\mathrm{T}}, \mu_2 = (-2,-2)^{\mathrm{T}}, \mu_3 = (0,0)^{\mathrm{T}}, \mu_4 = (2,2)^{\mathrm{T}}, \mu_5 = (4,4)^{\mathrm{T}}$, and the covariance matrix is the same as the one in setting (2). The shapes of three target distributions are illustrated by the 3D density plots in Figure 5.2.



(a) dog bowl distribution (b) 25-normal mixture (c) 5-normal mixture

Figure 5.2: The 3-D density plots of target distributions

Evaluation Metrics

The following metrics are used to evaluate the performance of the PTMC algorithms versus MCMC and LMC algorithms:

1. *Quantiles*: We calculate the average of the 2.5%, 50% and 97.5% empirical quantiles of the sample points obtained from 500 simulations, and compare them to their theoretical counterparts. This metric reflects how well the samples are representative of the target distribution.

2. *Effective Sample Size (ESS)*: The effective sample size for the jth dimension of the target distribution is defined as

$$\mathrm{ESS}_j = \frac{n}{1 + 2\sum_{s=1}^{S} \rho_s},$$

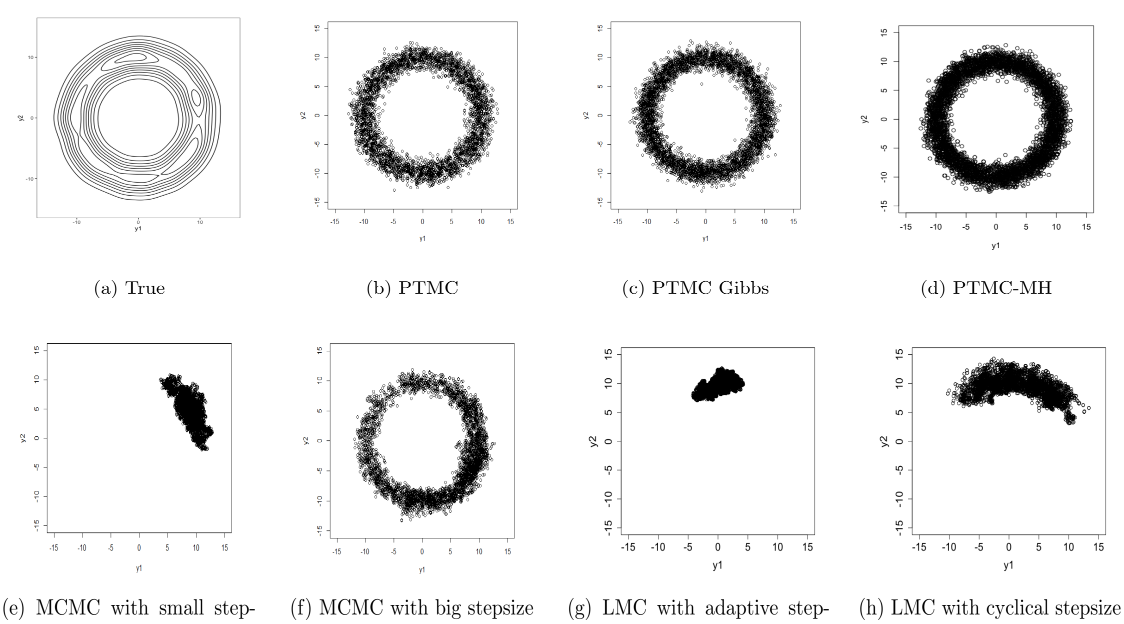
   where ρ_s is the correlation coefficient between y_j^t and y_j^{t+s} at lag s and $S = \min\{s : \rho_s < 0.05\}$. ESS is calculated as the average effective sample sizes across the 500 simulations. This metric indicates the sampling efficiency of the algorithm.

3. *Computation Time (CT)*: CT is the average computation time for generating 5000 samples from an algorithm across the 500 simulations.

All simulations are done in the `R` environment on Dell PowerEdge R630 computers with two Intel Xeon E5-2667v4 8-core 3.2 GHz CPUs and 64G memory to ensure comparable computation time across algorithms. For the PTMC algorithms, we use parallel computing on density evaluations to achieve faster computation.

Simulation Results

We provide the scatter plots of the sample points drawn from the dog bowl, 25-normal mixture and 5-normal mixture distributions in Figures 5.3, 5.4 and 5.5, respectively, each figure including 8 subfigures. Subfigure (a) gives a contour plot of the target distribution; subfigures (b), (c) and (d) correspond to the proposed Algorithms 5.1, 5.3 and 5.4, respectively; subfigures (e) - (f) correspond to random walk MCMC with small and big stepsizes, and subfigures (g)-(h) correspond to LMC algorithms with adaptive stepsize (stepsize is set to be proportional to $0.05t^{-0.55}$ with t to be the sampling iteration) and cyclical stepsize (Zhang et al., 2019), respectively. We also report the empirical quantiles versus theoretical quantiles, ESS and CT for the three distributions in Tables D.1, D.2 and D.3, respectively, in Appendix D.2.

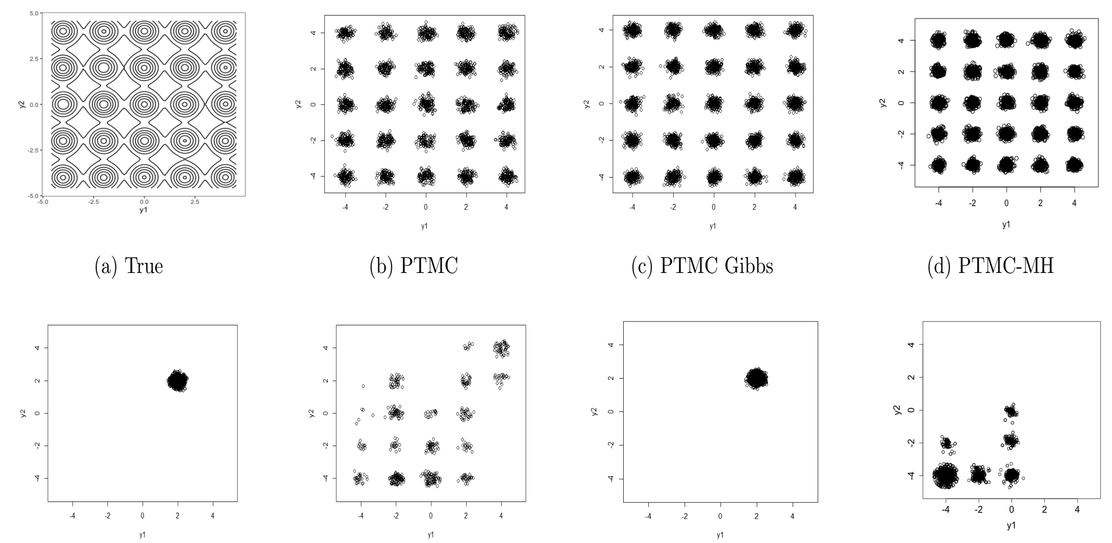


(a) True (b) PTMC (c) PTMC Gibbs (d) PTMC-MH

(e) MCMC with small stepsize (f) MCMC with big stepsize (g) LMC with adaptive stepsize (h) LMC with cyclical stepsize

Figure 5.3: Plots of samples from the dog bowl distribution using various algorithms

Subfigures (b), (c) and (d) in Figure 5.3 give the scatter plots of samples drawn from the dog bowl distribution obtained from the proposed Algorithms 5.1, 5.3 and 5.4, respectively. The sample points in subfigures (b), (c) and (d) evenly form a donut shape, exhibiting consistent patterns with the target distribution as shown in subfigure (a). However, for the MCMC with small stepsizes and LMC algorithms in subfigures (e), (g) and (h), the sample points fail to recover the circle shape; for the MCMC with big stepsize in subfigures (f), the sample points in the bottom-right corner are obviously denser than the top-left corner, suggesting that the algorithm fails to "walk through" the domain space. Table D.1 in SWA D.2 suggests that the MCMC or LMC algorithms have an extremely low ESS (≤ 10), indicating an inappropriately high rejection rate and low sampling efficiency.



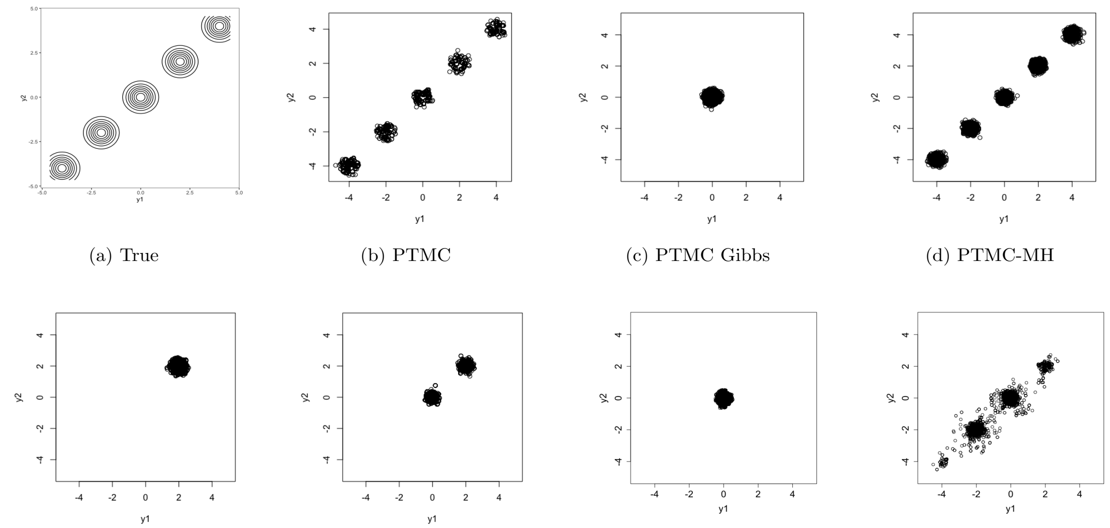
(a) True (b) PTMC (c) PTMC Gibbs (d) PTMC-MH

(e) MCMC with small stepsize (f) MCMC with big stepsize (g) LMC with small stepsize (h) LMC with cyclical stepsize

Figure 5.4: Plots of samples from the 25-normal mixture using various algorithms

Similar findings are obtained from Figure 5.4 with scatter plots of samples from the 25-normal mixture distribution. All the three proposed algorithms recover all 25 modes but MCMC with small stepsize and LMC algorithm with adaptive stepsize only successfully recover one mode and MCMC with big stepsize or LMC with cyclical stepsize also perform unsatisfactorily in terms of mode recovery.

In Figure 5.5, Algorithms 5.1 and 5.4 are the only ones that recover all five modes of the 5-normal mixture distribution. The conclusion is corroborated by the results in Tables D.1-D.3 in SWA D.2, in which the empirical quantiles of the PTMC and PTMC-MH



(a) True (b) PTMC (c) PTMC Gibbs (d) PTMC-MH

(e) MCMC with small stepsize (f) MCMC with big stepsize (g) LMC with adpative stepsize (h) LMC with cyclical stepsize

Figure 5.5: Plots of samples from the 5-normal mixture using various algorithms

algorithms are closer to the theoretical quantiles of the corresponding target distributions. As commented in Section 5.2, PTMC Gibbs sampler also fails to recover all modes in this scenario as it iterates through coordinates and can easily get stuck in one mode.

Overall, PTMC (Algorithm 5.1) and PTMC-MH (Algorithm 5.4) have superior performance in recovering complex distributions with multiple modes. The PTMC Algorithms 5.1 and 5.2 generate independent samples, providing samples with ESS much larger than those of the MCMC and LMC algorithms. Finally, PTMC-MH (Algorithm 5.4) has computational advantages over other PTMC algorithms for a large dimension k.

5.3.2 Setting 5.2

In the second simulation, we consider sampling from the posterior distribution under the Bayesian inference framework and compare the inference performance between the PTMC-based algorithms and random walk MCMC.

Simulation Setting

Suppose that $x^{(1)}, \ldots, x^{(400)}$ are i.i.d. samples from the distribution with density $f(x|\beta)$ indexed by the parameter vector $\beta = (\beta_1, \ldots, \beta_q)^{\mathrm{T}}$. To make Bayesian inference aboutβ, we aim to sample β from the posterior distribution

$$f(\beta|x^{(1)}, \ldots, x^{(400)}) \propto \prod_{i=1}^{400} f(x^{(i)}|\beta).$$

We draw samples from one of the following different density functions of $f(x|\beta)$:

(1) Setting 5.2.1: Low-dimensional distributions. We consider the following the one-dimensional and two-dimensional distributions in Table 5.5 (i.e., $q = 1$ or 2):

Table 5.5: One- and two-dimensional distributions $f(x|\beta)$ with parameter values

Distribution	β_1	β_2	Distribution	β_1	β_2
Geometric	0.5	-	Poisson	3	-
Gaussian copula	0.5	-	Clayton copula	3	-
Beta	3	4	Gamma	3	4
Joe-Gumbel copula	3	4	Clayton-Gumbel copula	3	4
Joe-Clayton copula	3	4	Tawn Type I copula	4	0.5
Tawn Type II copula	4	0.5			

The simulation is repeated $nsim = 500$ times for this setting, and we compare Algorithm 5.1 with random walk MCMC (MH algorithm) where the sample size from the target distribution is taken as $n = 5000$ or 8000, and the uniform sample size of the PTMC method is set as $n^* = 500$ or 1000 for dimension $k = 1$ and $n^* = 1000$ or 2000 for dimensions $k = 2$. For the MCMC algorithms, we set the "burn-in" sample size to be 500 for the one-dimensional case, and 1000 for the two-dimensional scenario.

(2) Setting 5.2.2: Multi-dimensional distributions. We consider two distributions with 5 or 6 dimensions, respectively:

(i) Gamma-Normal mixture distribution: The density $f(x|\beta)$ is

$$f(x|\beta) = \beta_1 \cdot \frac{\beta_3^{\beta_2}}{\Gamma(\beta_2)} x^{\beta_2 - 1} e^{-\beta_3 x} + (1 - \beta_1) \cdot \frac{1}{\sqrt{2\pi}\beta_5} e^{-(x-\beta_4)^2/2\beta_5^2},$$

which is essentially a mixture of a gamma distribution and a normal distribution with $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^{\mathrm{T}} = (0.5, 4, 2, -5, 3)^{\mathrm{T}}$.

(ii) D-vine distribution: The density $f(x|\beta)$ follows a D-Vine distribution, whose structure is illustrated in Figure 5.6.

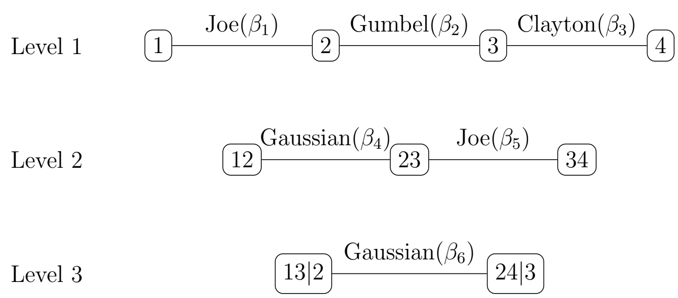


Figure 5.6: D-Vine structure and copula functions

The parameters of the D-Vine are given in Table 5.6.

Table 5.6: D-Vine copulas and the corresponding parameters

	$1,2$	$2,3$	$3,4$	$1,3\|2$	$2,4\|3$	$1,4\|2,3$
Copula	Joe	Gumbel	Clayton	Gaussian	Joe	Gaussian
Parameters	$\beta_1 = 3.83$	$\beta_2 = 2.50$	$\beta_3 = 3.00$	$\beta_4 = 0.70$	$\beta_5 = 2.86$	$\beta_6 = 0.59$
Kendall's τ	0.60	0.60	0.60	0.50	0.50	0.40

The simulation is repeated $nsim = 100$ times, and the comparison is conducted for the following five algorithms: i) the PTMC Algorithm 5.2, ii) Algorithm 5.3, iii) Algorithm 5.4, iv) MCMC(0), and v) MCMC(500). For the two MCMC algorithms, MH algorithm is embedded in each iteration of a Gibbs sampler used for sampling from the conditional density $f(\beta_\ell|\beta_{(-\ell)}, x^{(1)}, \ldots, x^{(400)})$ for $\ell = 1 \ldots, k$. We consider no burn-in samples and a 500 burn-in samples for MCMC(0) and MCMC(500) for the embedded MH algorithm, respectively. We set b_1 of the PTMC Algorithms 5.3 and 5.4, the "burn-in" sample size for the Gibbs sampler iterations of the MCMC(0) and MCMC(500) to be 1000. The sample size simulated from the posterior distributions is set to be 5000.

Evaluation Metrics

We consider the following metrics to evaluate the inference performance of the PTMC and MCMC algorithms:

1. *Empirical Bias (EBias)*
2. *Empirical Standard Error (ESE)*
3. *Average Standard Error (ASE)*
4. *Empirical Coverage Probability (ECP)*
5. *Ratio of Computation Time (RCT)*: RCT is calculated as

$$\mathrm{RCT} = \frac{\text{CT of MCMC}}{\text{CT of PTMC}},$$

where the details for the first four metrics can be found in Section 3.5.2.

Simulation Results

Simulation Setting 5.2.1 considers various low-dimensional distributions. The RCTs of the MCMC MH algorithm versus PTMC Algorithm 5.1 are illustrated in Figure 5.7, where the simulated sample size is set to be 5000 versus 8000, and the number of uniform samples in our proposed Algorithm 5.1 is set to be 500 versus 1000 for single-dimensional distributions (when $k = 1$) and 1000 versus 2000 for two-dimensional distributions (when $k = 2$). A larger RCT suggests a greater advantage of PTMC Algorithm 5.1 over MCMC MH in terms of computation speed. The RCTs are always greater than 2 except for the gamma distribution, and they are larger than 10 for Geometric, Poisson, Gaussian copula, Clayton Copula, Tawn Type I copula when n^* is set as a more conservative value with $n^* = 500$ for $k = 1$ and 1000 for $k = 2$. The RCTs become larger if sample size n increases from 5000 to 8000 as the number of evaluations of the target density is pre-determined and fixed as n^* for Algorithm 5.1, however, it increases as n gets larger for the MCMC algorithm. Generally speaking, the advantage of PTMC Algorithm 5.1 over MCMC MH in computational time reduces when sampling from a distribution with higher dimension, because a large n^* is usually required to guarantee valid inference performance.

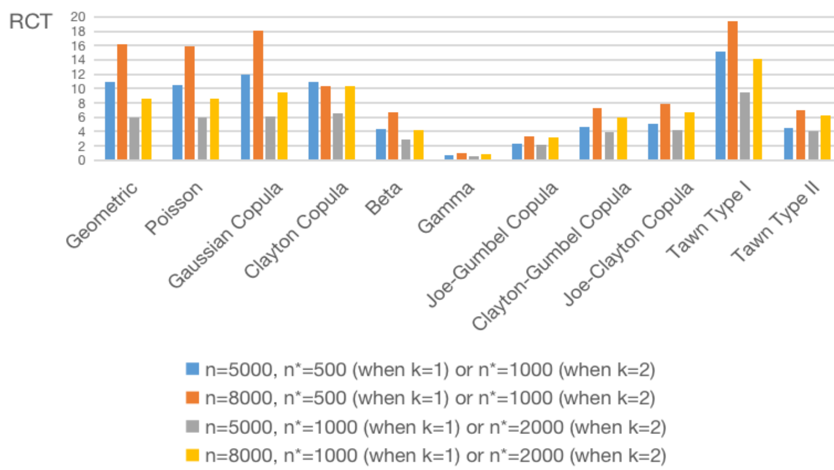


Figure 5.7: RCTs for different distributions in Setting 5.2.1

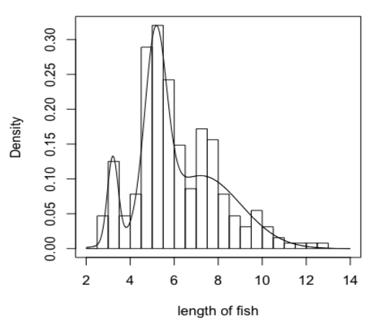
Table D.4 in Appendix D.2.2 reports the EBias, ESE, ASE, ECP, ESS and CT (in minutes) of the PTMC Algorithm 5.1 with $n^* = 500$, Algorithm 5.1 with $n^* = 1000$ and MCMC MH algorithm for sample sizes $n = 5000$ and 8000 from four one-dimensional distributions. The same metrics of the PTMC Algorithm 5.1 with $n^* = 1000$, Algorithm 5.1 with $n^* = 2000$ and MCMC MH algorithm from seven two-dimensional distributions are summarized in Table D.5. The PTMC Algorithm 5.1 generates independent samples, therefore gives much larger ESS's than the MCMC MH algorithm, suggesting that the PTMC Algorithm 5.1 is a more efficient algorithm for low-dimensional sampling. The Bayesian estimates of parameters β have ignorable biases, their ESE's and ASE's have a reasonable match and the ECP's are close to the 95% nominal level for all algorithms under all scenarios. The PTMC Algorithm 5.1 with the more conservative uniform sample size n^* provides valid inference results and works as well as the one with larger n^* in most scenarios, except that the one with $n^* = 2000$ leads to a better match between ESE's and ASE's for the Gamma distribution.

In summary, the PTMC Algorithm 5.1 exhibits great advantages over the MCMC MH algorithms in terms of both the sampling efficiency and the computational speed and provides comparable inference performance in low-dimensional scenarios.

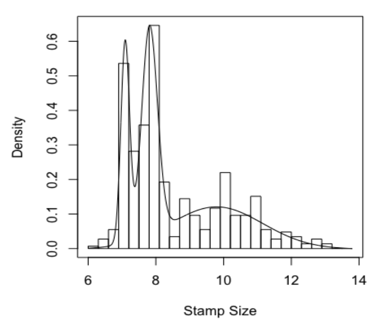
Simulation Setting 5.2.2 considers two multi-dimensional distributions. The numerical results containing the same set of evaluation metrics for the Gamma-Normal mixture distribution and D-vine are provided in Tables D.6 and D.7, respectively, in Appendix D.2.2. Five algorithms, PTMC Algorithms 5.2, 5.3, 5.4, MCMC(0) and MCMC(500), are compared. The PTMC Algorithm 5.2 provides independent samples and has the largest ESS's close to the sample size. The PTMC Gibbs sampler (Algorithm 5.3) and MCMC(500) have comparable ESS's, which are significantly larger than the ESS's of the PTMC-MH (Algorithm 5.4) and MCMC(0). The PTMC Algorithm 5.2 requires an incredibly large size of uniform samples to achieve a precise approximation of the multi-dimensional target distribution. In the example of D-vine, a six-dimensional distribution, the PTMC Algorithm 5.2 with $n^* = 1500000, 2500000$ and 5000000 fails to provide valid sampling results due to large discrepancies between ASE's and ESE's as well as notably lower ECP's than the 95% nominal level. When the uniform sample size increases to $n = 12500000$, the results from the PTMC Algorithm 5.2 look valid. As commented in Section 5.2, the PTMC method is a Bayesian nonparametric approach and suffers from the "curse of dimensionality" which motivates our Algorithms 5.3 and 5.4.

5.4 Data Analysis

We apply the proposed PTMC-based algorithms to analyze two heterogeneous datasets having multiple modes: the Fishery data (Titterington et al., 1986; Frühwirth-Schnatter, 2006) consisting of the lengths of 256 snappers, and the Hidalgo Stamp data (Izenman and Sommer, 1988) consisting of the thickness of 485 stamps. Based on the studies of Titterington et al. (1986) and Izenman and Sommer (1988), both datasets can be reasonably fitted by a Gaussian mixture model with the number of modes $J = 3$. Figure 5.8 displays the histograms overlaid with Gaussian mixture densities.



(a) Fishery Data



(b) Hidalgo Stamps Data

Figure 5.8: Histograms and 3-component Gaussian mixture density of two datasets

5.4.1 Models

Suppose that there is a univariate dataset with i.i.d. data points $x^{(i)}$ for $i = 1, \ldots, n_1$ from a Gaussian mixture model with J components, such that the density is

$$f(x_i|\theta) = \sum_{j=1}^{J} \lambda_j \phi(x_i|\mu_j, \sigma_j),$$

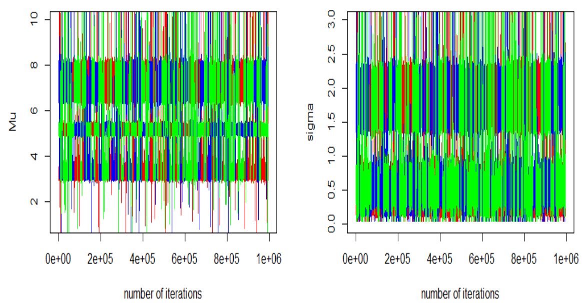
where $\theta = (\lambda, \mu, \sigma)^{\mathrm{T}}$ with $\lambda = (\lambda_1, \ldots, \lambda_{J-1})^{\mathrm{T}}$, $0 < \lambda_j < 1$, $j = 1, \ldots, J-1$, $\sum_{j=1}^{J} \lambda_j = 1$, $\mu = (\mu_1, \ldots, \mu_J)^{\mathrm{T}}$, $\sigma = (\sigma_1, \ldots, \sigma_J)^{\mathrm{T}}$, and $\phi(\cdot|\mu_j, \sigma_j)$ denotes the normal density with mean μ_j and standard deviation σ_j for $j = 1, \ldots, J$. The posterior distribution is

$$f(\theta|x) \propto f(\theta) \prod_{i=1}^{n_1} f(x_i|\theta).$$

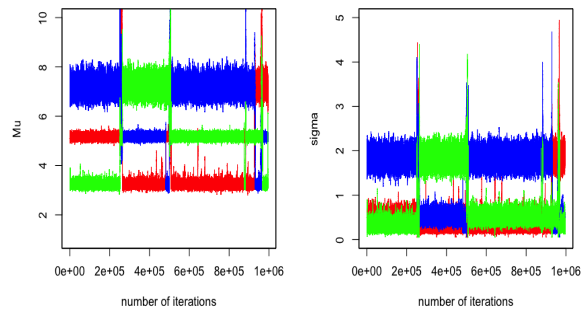
The Gaussian mixture model has identifiability issues, as the model is invariant under the exchange of the J components. As a result, the posterior distribution $f(\theta|x)$ is always a multi-modal distribution. For simplicity, we use noninformative uniform priors for all parameters in θ. We sample from the posterior distribution using the following algorithms: PTMC Gibbs Sampler (Algorithm 5.3), PTMC-MH (Algorithm 5.4), and MCMC with a big or small stepsize. Each algorithm runs 10^6 iterations with the first 5000 iterations removed as the burn-in period.

5.4.2 Sampling Results

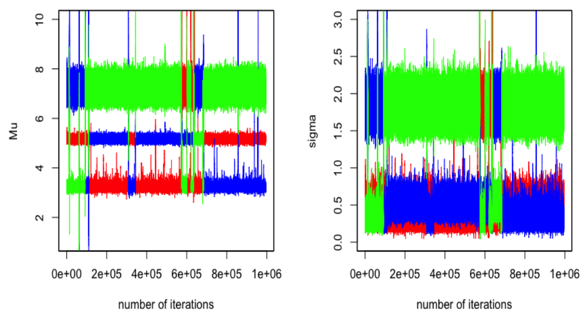
For the Fishery data, the dataset is relatively small with 256 data points, which results in a less peaked posterior distribution. The sampling results of the mean and the standard deviation of the PTMC Gibbs, PTMC-MH, MCMC with big stepsize and with small stepsize, and LMC with adaptive stepsize and cyclical stepsize, are provided in panels (a)-(f), respectively, in Figure 5.9. As can be seen in subfigure (a) in Figure 5.8, the data contains three modes roughly at 3.2, 5.2 and 7.2, therefore the number of modes in the proposed Gaussian mixture models is set to be J=3. The posterior distribution of the Gaussian mixture model should have $3! = 6$ modes. In the left subfigures in Figure 5.9, the mean values are roughly located at 3.2, 5.2 and 7.2, and lines in three colors represent the trajectories of the population means from the three modes.



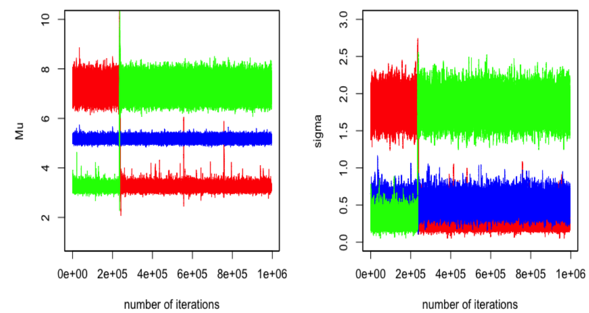
(a) PTMC Gibbs



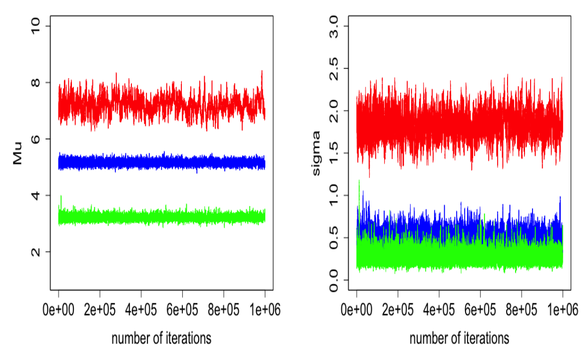
(b) PTMC-MH



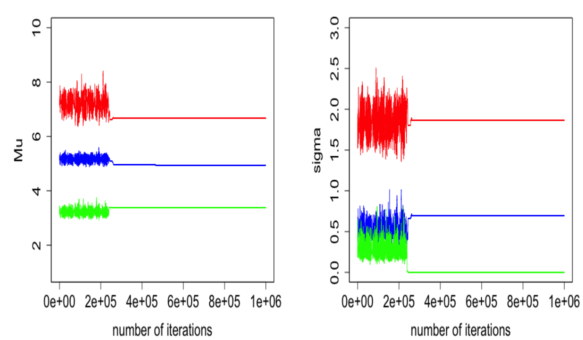
(c) MCMC with big stepsize



(d) MCMC with small stepsize



(e) LMC with adaptive stepsize



(f) LMC with cyclical stepsize

Figure 5.9: The Fishery data: Sample plots of the means and standard deviations of the Gaussian mixture model

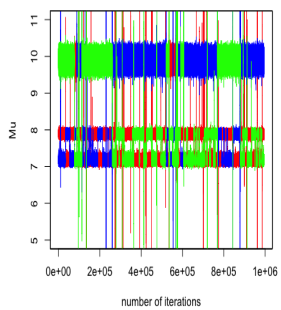
If an algorithm is able to recover all six modes, we expect all three colored lines transit frequently between and eventually get a reasonable large number of iterations at each location (i.e., around one of 3.2, 5.2 and 7.2). From Figure 5.9, PTMC Gibbs can frequently transit between modes, PTMC-MH and MCMC with a big stepsize can occasionally transit between different modes, however, MCMC with a small stepsize and LMC with adaptive stepsize get stuck in one local modes. Similar patterns can be observed from the right subfigures. Note that the LMC with cyclical stepsize cyclically explores the regions far away from the current sample values, so that sometimes the sample values may enter the regions with extremely low density values, especially for the parameters σ and λ with some boundary values. After entering the low density regions, the algorithm keeps rejecting the new sample values, as illustrated by the straight horizontal lines in subfigure (f).

The Hidalgo Stamp Dataset contains 485 data points, for which the posterior distribution of the Gaussian mixture model is more peaked. In this dataset, both MCMC algorithms with a big or small stepsize fail to transit between modes and can no longer discover all modes from the mixture models. The results of the LMC algorithms are similar to those of the LMC algorithms in the Fishery dataset. However, both PTMC-based Algorithm 5.3 and 5.4 perform very well in discovering possible modes.

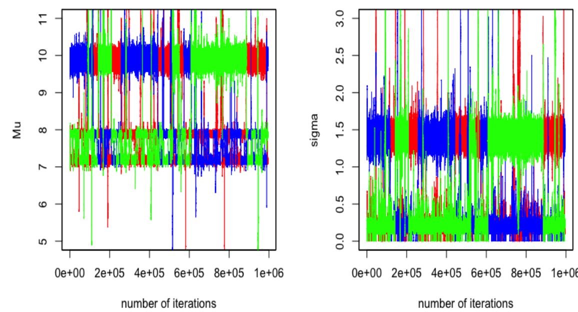
Additional numerical sampling results including the Bayesian estimates (Estimate), standard errors (SE), and ESS of the means and standard deviations of all modes for the Fishery data are summarized in Table D.8 and those of the Hidalgo Stamp data are given in Table D.9 in Appendix D.3.

5.5 General Remarks

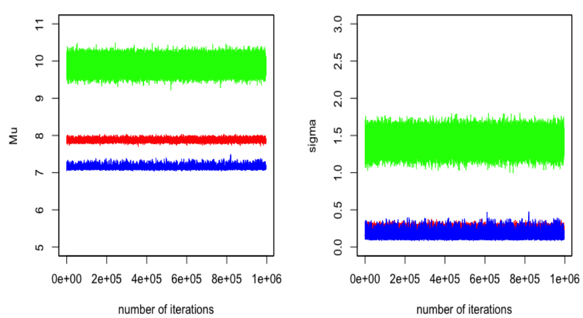
In this chapter, multiple sampling algorithms are proposed based on the Polya tree Monte Carlo method (PTMC) to sample from potentially multi-modal distributions. Compared with the MCMC algorithms, the PTMC algorithms have several advantages in terms of sampling efficiency and mode discovery. More specifically, for distributions in low dimensions, the PTMC algorithm 5.1 provides independent samples and fast computation speed. For high-dimensional distributions, the Algorithm 5.4 is powerful in discover multiple modes. The proposed algorithms also require little tuning or user-specified parameter, thus enjoying broad applications.



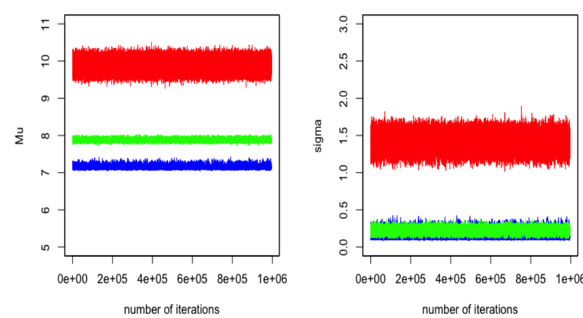
(a) PTMC Gibbs



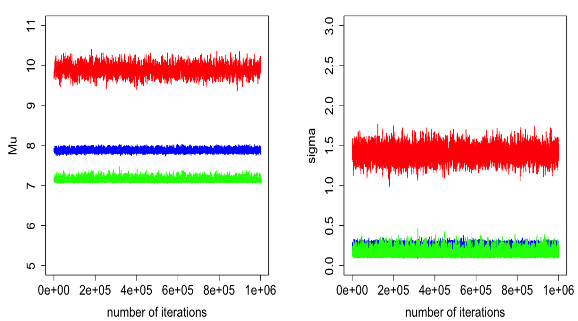
(b) PTMC-MH



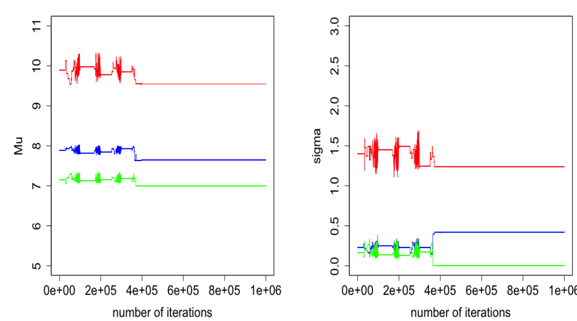
(c) MCMC with big stepsize



(d) MCMC with small stepsize



(e) LMC with adaptive stepsize



(f) LMC with cyclical stepsize

Figure 5.10: The Hidalgo Stamp data: Sample plots of of the means and standard deviations of the Gaussian mixture model

Chapter 6

Polya Tree Based Nearest Neighbor Regression

6.1 Introduction

Regression analysis is a powerful statistical method for delineating the relationship between responses and covariates of interest. As more and more data with irregular distributions emerge, parametric or semi-parametric regression models are under the risk of model misspecification. In this chapter, we introduce a new fully nonparametric regression model, called the Polya tree based nearest neighbor (PTNN) regression, which constructs a PT-distributed probability measures of the responses in a "nearest" neighborhood of the covariates of interest. Here "a nearest neighbor" is loosely used in the same way as the nearest neighbor method (Cover and Hart, 1967; Beyer et al., 1999), though strictly speaking, there is no "nearest" neighborhood of a center in a continuous metric (unless the center itself is taken as its nearest neighborhood). The constructed probability measure well approximates the true probability measure of the response given covariates, and the resulting nonparametric estimates are easy to obtain based on a sample from the constructed PT distribution. The model enjoys several merits including simple formulation, consistent estimates of the conditional distribution G_x and computational efficiency. The proposed method does not require any parametric model assumption and thus possesses the robustness property.

The rest of the chapter is organized as follows. We describe the Polya tree based nearest neighbor regression model (PTNN) in Section 6.2. In Section 6.3, we provide the asymptotic properties of the PTNN, and in Section 6.4.1, the selection of the tuning parameter and the sampling procedure of the PTNN are discussed. In Section 6.5, we conduct simulation studies to compare the proposed PTNN method with some benchmark nonparametric models. In Section 6.6, we apply the PTNN to the Combined Cycle Power Plant dataset.

6.2 Model Formulation

Suppose that we have i.i.d. random variables $Z_i = (Y_i, X_i^{\mathrm{T}})^{\mathrm{T}}$, where for $i = 1, \ldots, n$, $Y_i \in \mathcal{S} \subset \mathbb{R}$ is a response variable and $X_i = (X_{i1}, \ldots, X_{ip})^{\mathrm{T}} \in \mathcal{S}_x \subset \mathbb{R}^p$ is a vector of covariates. Let $z_i = (y_i, x_i^{\mathrm{T}})^{\mathrm{T}}$ denote the observed counterparts of Z_i for $i = 1, \ldots, n$. We now consider a formulation of a fully nonparametric regression model as indicated by (1.12). Taking PT priors as the building blocks, we describe a strategy for connecting the probability measure of Y to x.

To extend the PT prior reviewed in Section 1.7.2 to a regression setting, one may attempt to assume a PT prior for G_x and update the posterior random splitting probabilities in (1.10) if repeated measurements of Y at some specific covariate value x are available. This consideration is natural, especially when dealing with discrete covariates. However, this procedure is not doable if some covariates are continuous. As a remedy, we develop a nearest neighbor regression model based on creating a neighborhood of the covariate value of interest, and abbreviate it as PTNN.

Suppose F_x is the true probability measure of response Y given covariate value x and our objective is to obtain a fully nonparametric estimate of F_x. The basic idea of PTNN regression is to construct a PT-distributed probability measure given data $Z_i = (Y_i, X_i^{\mathrm{T}})^{\mathrm{T}}$, $i = 1, \ldots, n$, to provide a good approximation the true probability measure F_x, and then to obtain a nonparametric estimate of F_x using the samples from the constructed PT probability measure. The PT-distributed probability measure is constructed in a manner similar to the posterior PT in (1.10) but $N_{\varepsilon_1 \ldots \varepsilon_m}$ is updated as the summation of weighted samples of which the response falls in the subset $B_{\varepsilon_1 \ldots \varepsilon_m}$ with covariates being in the "nearest neighborhood" of x. The detailed formulation is as follows.

We consider a probability measure $G_{x|Z}$, the probability measure of response given covariate x obtained based on the data $Z = (Y, X^{\mathrm{T}})^{\mathrm{T}}$, which is assumed to follow a PT distribution

$$G_{x|Z} \sim PT(\Pi, \mathcal{A}_{x|Z}), \tag{6.1}$$

where Π is a collection of partitions of $\mathcal{S}$ as defined in Section 1.7.2. Here $\mathcal{A}_{x|Z} = \bigcup_{m\in N^+} \mathcal{A}_{m,x}(Z)$ with $\mathcal{A}_{m,x}(Z) = \{\alpha_{\varepsilon_1\ldots\varepsilon_m,x}(Z) : \varepsilon_j \in \{0,1\}, j = 1,\ldots,m\}$ and

$$\alpha_{\varepsilon_1\ldots\varepsilon_m,x}(Z) \quad = \quad \begin{cases} \alpha_{\varepsilon_1\ldots\varepsilon_m} + \prod_{j=1}^{p} w(X_j) & \text{if } Y \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}, X \in \mathcal{S}_{x,h} \text{ and } m \leq M \\ \alpha_{\varepsilon_1\ldots\varepsilon_m} & \text{otherwise,} \end{cases}$$

where $\alpha_{\varepsilon_1\ldots\varepsilon_m} > 0$, and its default choice is $\alpha_{\varepsilon_1\ldots\varepsilon_m} = \phi m^2$ with $\phi > 0$; $w(\cdot)$ is a weight function; $\mathcal{S}_{x,h} = \{(x'_1,\ldots,x'_p)^{\mathrm{T}} : x'_j \in [x_j - h_j, x_j + h_j], j = 1,\ldots,p\}$ is the "nearest neighbor" of x, $h_j > 0$ quantifies the width of neighborhood of x_j, $j = 1,\ldots,p$; and $M \in N^+$ is a pre-specified "truncated level", suggesting that the PT tree approximates the true probability measure only to a finite level. If we have n i.i.d. copies of Z, denoted $\tilde{Z} = (Z_1^{\mathrm{T}},\ldots,Z_n^{\mathrm{T}})^{\mathrm{T}}$, the conditional random splitting probabilities of this PT distribution given $\tilde{Z}$, denoted $G_{\varepsilon_1\ldots\varepsilon_{m-1}0,x}(\tilde{Z})$, are

$$G_{\varepsilon_1\ldots\varepsilon_{m-1}0,x}(\tilde{Z}) \sim \text{Beta}\left(\alpha_{\varepsilon_1\ldots\varepsilon_{m-1}0} + N_{\varepsilon_1\ldots\varepsilon_{m-1}0,x}(\tilde{Z}), \alpha_{\varepsilon_1\ldots\varepsilon_{m-1}1} + N_{\varepsilon_1\ldots\varepsilon_{m-1}1,x}(\tilde{Z})\right),$$

if $m \leq M$;and

$$G_{\varepsilon_1\ldots\varepsilon_{m-1}0,x}(\tilde{Z}) \sim \text{Beta}\left(\alpha_{\varepsilon_1\ldots\varepsilon_{m-1}0}, \alpha_{\varepsilon_1\ldots\varepsilon_{m-1}1}\right),$$

if $m > M$, where $N_{\varepsilon_1\ldots\varepsilon_m}(\tilde{Z})$ is a function of $\tilde{Z}$ of the form:

$$N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) = \sum_{i=1}^{n}\prod_{j=1}^{p} w(X_{ij}) I\big(Y_i \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\big)\left[\prod_{j=1}^{p} I\left(X_{ij} \in [x_j - h_j, x_j + h_j]\right)\right]. \tag{6.2}$$

In (6.2), $I(X_{ij} \in [x_j - h_j, x_j + h_j])$ indicates whether X_{ij} belongs to the subset (the "nearest" neighbor) $[x_j - h_j, x_j + h_j]$ of the covariate value x_j for $j = 1,\ldots,p$, and $I\big(Y_i \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}\big)$ indicates whether Y_i belongs to $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}$. The weight function $w(\cdot)$ is built according to the principle that larger weights should be assigned to the individuals whose covariate values X_{ij} are closer to the target value x_j, and hence satisfied the following conditions for $j = 1,\ldots,p$:

1. $w(\cdot)$ is positive and bounded on $[x_j - h_j, x_j + h_j]$;
2. $w(\cdot)$ is symmetric around x_j on $[x_j - h_j, x_j + h_j]$, i.e., $w(x_j - t) = w(x_j + t)$ for $t \in [0, h_j]$;
3. For $x_{ij}, x_{kj} \in [x_j - h_j, x_j + h_j]$ with $i \neq k$, if $||x_{ij} - x_j||_2 \leq ||x_{kj} - x_j||_2$, then $w(x_{ij}) \geq w(x_{kj})$.

Let $w_{\max}$ and $w_{\min}$ denote the maximum and minimum values of $w(\cdot)$ over $[x_j - h_j, x_j + h_j]$, respectively, for $j = 1, \ldots, p$. As a result, the weight function reaches the maximum at x_j, i.e., $w_{\max} = w(x_j)$, and comes to the minimum at $x_j - h_j$ and $x_j + h_j$, i.e., $w_{\min} = w(x_j - h_j) = w(x_j + h_j)$. Obviously, the uniform weight function, $w(x) = 1$, can be an option, and symmetric kernel functions, such as Gaussian kernel function, can be considered. For the prior parameters $\alpha_{\varepsilon_1 \ldots \varepsilon_m} = \phi m^2$, we suggest to take $\phi = w_{\min}$ to formulate a weak prior of the Polya tree.

Let $G_{x|\tilde{Z}}(\mathcal{B}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_m})$ denote the random probability of the subset $\mathcal{B}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_m}$ given the data $\tilde{Z}$ and

$$G_{x|\tilde{Z}}(\mathcal{B}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_m}) = \prod_{k=1}^{m} G_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_k, x}(\tilde{Z}).$$

Then the expected value of $G_{x|\tilde{Z}}(\mathcal{B}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_m})$ is

$$\begin{aligned} E\left[G_{x|\tilde{Z}}(\mathcal{B}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_m})\right] &= E\left[\prod_{k=1}^{m} G_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_k, x}(\tilde{Z})\right] \\ &= \begin{cases} \displaystyle\prod_{k=1}^{m} \frac{\alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} \varepsilon_k} + N_{\varepsilon_1 \ldots \varepsilon_{k-1} \varepsilon_k, x}(\tilde{Z})}{\sum_{l=0}^{1}\left[\alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} l} + N_{\varepsilon_1 \ldots \varepsilon_{k-1} l, x}(\tilde{Z})\right]} & \text{if } m \leq M \\ \displaystyle\prod_{k=1}^{M} \frac{\alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} \varepsilon_k} + N_{\varepsilon_1 \ldots \varepsilon_{k-1} \varepsilon_k, x}(\tilde{Z})}{\sum_{l=0}^{1}\left[\alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} l} + N_{\varepsilon_1 \ldots \varepsilon_{k-1} l, x}(\tilde{Z})\right]} \cdot \prod_{k=M+1}^{m} \frac{\alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} \varepsilon_k}}{\sum_{l=0}^{1} \alpha_{\varepsilon_1 \ldots \varepsilon_{k-1} l}} & \text{if } m > M. \end{cases} \end{aligned} \tag{6.3}$$

It is worth to clarify that the proposed PTNN is not a "posterior" distribution of Polya tree, but a constructed PT distribution with good approximation to the true probability measure. The theoretical properties of the PTNN are provided in the Section 6.3.

6.3 Asymptotic Properties

In this section, we provide the asymptotic properties of the proposed PT (6.1), which forms the theoretical foundation of the PTNN. The following two theorems prove the pointwise convergence and consistency of the proposed the proposed PT (6.1), and the proofs of the theorems are provided in Appendix E.1.

Theorem 6.1. *If the following conditions are satisfied:*

1. $h_j = O(n^{-\eta/p})$ *for* $\eta \in (0,1)$ *and* $j = 1, \ldots, p$;

2. $g_{\varepsilon_1 \ldots \varepsilon_M}(x) = F_x(\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M})$ *is a smooth function with derivative* $g'_{\varepsilon_1 \ldots \varepsilon_M}(x)$;

then for any $x \in \mathcal{S}_x$ *and any subset* $\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_m} \in \pi_m$ *with* $m = 1, \ldots, M$,

$$\frac{1}{w_x} N_{\varepsilon_1 \ldots \varepsilon_m, x}(\tilde{Z}) \xrightarrow{p} F_x(\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_m}) \qquad \text{as } n \to \infty,$$

where $w_x = \sum_{i=1}^n \prod_{j=1}^p w(X_{ij}) I(X_{ij} \in [x_j - h_j, x_j + h_j])$ *is the summation of the weights in the "nearest" neighbor of* x.

The first condition in Theorem 6.1 states that the window width of the nearest neighbors decreases in a lower order as the sample size n increases, which serves as the criterion of selecting h in Section 6.4.1. The second condition in Theorem 6.1 assumes the smoothness of $F_x(\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M})$ with respect to x, which is commonly made in the nonparametric literature. Theorem 6.1 can further lead to the following asymptotic results regarding the Polya tree in PTNN.

Theorem 6.2 (Asymptotic Properties of the Polya Tree in PTNN)**.** *Assume that the conditions in Theorem 6.1 hold and the joint density* $f(y, x)$ *of* $(Y, X^T)^T$ *is smooth. Then the following results hold for any* $x \in \mathcal{S}_x$:

(1) Let $\mathfrak{S} = \{B \in \mathcal{S} : B \text{ is measurable}\}$. *If* $n = max\left\{O(M^{\frac{3}{1-\eta}+\xi}), O(M^{1/\eta+\xi})\right\}$ *for* $\xi > 0$, *then for any* $B \in \mathfrak{S}$,

$$G_{x|\tilde{Z}}(B) \xrightarrow{p} F_x(B) \qquad \text{as } M \to \infty$$

(2) If $n = O(2^{\frac{5M}{\eta^*}} M^{\frac{3}{\eta^*}})$ *for* $\eta^* = min\{\eta, 1-\eta\}$, *then for any* $\delta > 0$,

$$P\left[D(G_{x|\tilde{Z}}, F_x) \geq \delta\right] \longrightarrow 0 \qquad \text{as } M \to \infty$$

where $D(G_{x|\tilde{Z}}, F_x) = \int_{\mathcal{S}} \left| g_{x|\tilde{Z}}(y) - f(y \mid x) \right| dy$ *with* $g_{x|\tilde{Z}}$ *representing the density function of* $G_{x|\tilde{Z}}$.

6.4 Inference Procedures

6.4.1 Selection of Tuning Parameter h

In this subsection, we discuss the selection of the tuning parameters $h = (h_1, \ldots, h_p)^{\mathrm{T}}$, which play a role similar to the bandwidth in the kernel method. The condition 1 of Theorem 6.1 states that $h_j = O(n^{-\eta/p})$, we propose to select h_j as $h_j = c_j n^{-\eta/p}$, where c_j is a positive constant for $j = 1, \ldots, p$.

We first discuss the selection of η. In the proof of Theorem 6.1 provided in the Appendix E.1, we have

$$\sup\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_M,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M})\right| \leq 2^p \prod_{j=1}^{p} h_j \sup_{x\in\mathcal{S}_x}|g'_{\varepsilon_1\ldots\varepsilon_M}(x)| + O_p\left(\frac{1}{\sqrt{N_x}}\right) \tag{6.4}$$

where $N_x = \sum_{i=1}^n \prod_{j=1}^p I\left(X_{ij} \in [x_j - h_j, x_j + h_j]\right)$ denotes the number of data points falling in the "nearest" neighbor of x. In the Appendix E.1, it is shown that $N_x = O_p(n^{1-\eta})$ with $\eta \in (0, 1)$. Applying $h_j = c_j n^{-\eta/p}$ to (6.4), we get

$$\begin{aligned}
& \sup\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_M,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M})\right| \\
\leq\ & \left[\prod_{j=1}^{p} 2c_j\right] n^{-\eta} \sup_{x\in\mathcal{S}_x} \mid g'_{\varepsilon_1\ldots\varepsilon_M}(x) \mid + O_p\left(\frac{1}{\sqrt{N_x}}\right) & (6.5) \\
=\ & \max\left\{O_p\left(\frac{1}{n^\eta}\right), O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right)\right\}. & (6.6)
\end{aligned}$$

η should be selected to minimize the upper bound of the difference of conditional probabilities $\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_M,x}(\tilde{Z})$ and $F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M})$ in (6.6), which can be achieved when $n^\eta = n^{(1-\eta)/2}$, i.e., $\eta = 1/3$. Namely, the optimal pointwise convergence rate of PTNN is $O_p(n^{-1/3})$.

We next discuss the choice of the constant c_j for $j = 1, \ldots, p$. For a random sample of covariate $x_{1j}, \ldots, x_{nj}$, the sample range, denoted by r_j, is defined as the $\max\{x_{1j}, \ldots, x_{nj}\}-\min\{x_{1j}, \ldots, x_{nj}\}$, for $j = 1, \ldots, p$. The data with a larger range is more sparse than a sample with a smaller range, thus h_j should be set proportionally to the sample range r_j so that the number of data points in the nearest neighbor maintains at a similar scale. We suggest to use $c_j = r_j/2$, for $j = 1, \ldots, p$.

Using (6.5) to find a value of η to minimize the upper bound, the value of $\sup_{x \in \mathcal{S}_x} \mid g'_{\varepsilon_1 \dots \varepsilon_M}(x) \mid$ is basically needed. However, calculating $\sup_{x \in \mathcal{S}_x} \mid g'_{\varepsilon_1 \dots \varepsilon_M}(x) \mid$ requires knowledge about the unknown true underlying conditional probability. As a remedy, we suggest to adopt a cross-validation procedure to select the optimal η that minimizes the average absolute difference between the predicted responses and the true responses from a set of candidate values. It is also worth to mention that $\sup_{x \in \mathcal{S}_x} \mid g'_{\varepsilon_1 \dots \varepsilon_M}(x) \mid$ is not always small in practice as the response y can vary dramatically with the change of some covariate value x. As a result, when $\sup_{x \in \mathcal{S}_x} \mid g'_{\varepsilon_1 \dots \varepsilon_M}(x) \mid$ is large, suggesting that the change in the outcome variable is very sensitive to the change in x, η should take values greater than 1/3, thus leading to a narrower neighborhood of x.

6.4.2 Sampling Algorithm

In this subsection, we provide an algorithm to sample from the constructed PT distribution given in Section 6.2, which has limiting distribution F_x as shown in Section 6.3.

We briefly describe the steps of sampling algorithm here and provide the detailed pseudo code in Table 6.1. For data $\tilde{z} = (z_1^{\mathrm{T}}, \dots, z_n^{\mathrm{T}})^{\mathrm{T}}$ with $z_i = (y_i, x_{i1}, \dots, x_{ip})^{\mathrm{T}}$ for $i = 1, \dots, n$, the data points in the "nearest neighbor" of a given covariate value of interest x are identified by calculating the the product of the indicator functions $\prod_{j=1}^{p} I\left(x_{ij} \in [x_j - h_j, x_j + h_j]\right)$ and reserving the ones with value 1. In the next step, update $N_{\varepsilon_1 \dots \varepsilon_M, x}(\tilde{z})$ using (6.2). After updating for all $N_{\varepsilon_1 \dots \varepsilon_M, x}(\tilde{z})$, $m \leq M$, the expected probabilities, $E[G_{x|\tilde{z}}(\mathcal{B}_{\varepsilon_1 \dots \varepsilon_M})]$ are calculated following (6.3). For any $m > M$, it is easily seen from (6.3) that $E(\mathcal{G}_{\varepsilon_1 \dots \varepsilon_{m-1} 0, x}(\tilde{z})) = \alpha_{\varepsilon_1 \dots \varepsilon_{m-1} 0} / (\alpha_{\varepsilon_1 \dots \varepsilon_{m-1} 0} + \alpha_{\varepsilon_1 \dots \varepsilon_{m-1} 1}) = 1/2$, when $\alpha_{\varepsilon_1 \dots \varepsilon_m}$ takes its default value ϕm^2 with $\phi > 0$. In other words, for the proposed Polya tree beyond level M, a sample falling in $\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M}$ is uniformly distributed on $\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M}$. Thus, to sample from the constructed PT distribution, a subset $\mathcal{B}_{\varepsilon_1 \dots \varepsilon_M}$ must be first sampled based on $E[G_{x|\tilde{z}}(\mathcal{B}_{\varepsilon_1 \dots \varepsilon_M})]$, and then a sample of response is generated uniformly on $\mathcal{B}_{\varepsilon_1 \dots \varepsilon_M}$.

Table 6.1: Sampling algorithm of PTNN Model

Input: Dataset $\tilde{z} = (z_1^{\mathrm{T}}, \ldots, z_n^{\mathrm{T}})^{\mathrm{T}}$ with $z_i = (y_i, x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ for $i = 1, \ldots, n$, covariate value x, tuning parameters $h = (h_1, \ldots, h_p)^{\mathrm{T}}$, weight function $w(x)$, size of the sample to be sampled from the constructed PT distribution n_1;

Initiation: $N_{\varepsilon_1 \ldots \varepsilon_M, x}(\tilde{z}) = 0$ for $\varepsilon_k \in \{0, 1\}$, $k = 1, \ldots, M$;

Output: A sample from the constructed PT distribution of size n_1.

1. **for** i *in* $1 : n$ **do**
  - **for** j *in* $1 : p$ **do**
    - Calculate the indicator function $I_{ij}(x) = I\left(x_{ij} \in [x_j - h_j, x_j + h_j]\right)$ for the jth covariate of the ith data point.
  - **end**

end

2. Identify the index set of the data points in the nearest neighbor of x: $\mathcal{I}_x = \{i \in \{1, \ldots, n\} : \prod_{j=1}^{p} I_{ij}(x) = 1\}$.

for $\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M}$ *in* π_M **do**
  - **for** k *in* $\mathcal{I}_x$ **do**

$$N_{\varepsilon_1 \ldots \varepsilon_M, x}(\tilde{z}) \leftarrow N_{\varepsilon_1 \ldots \varepsilon_M, x}(\tilde{z}) + I(y_k \in \mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M}) \prod_{j=1}^{p} w(x_{kj})$$

  - **end**

end

3. Set $m = M - 1$

while $1 \leq m \leq M - 1$ **do**
  - **for** $\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_m}$ *in* π_m **do**

$$N_{\varepsilon_1 \ldots \varepsilon_m, x}(\tilde{z}) = \sum_{l=m+1}^{M} \sum_{\varepsilon_l = 0}^{1} N_{\varepsilon_1 \ldots \varepsilon_l, x}(\tilde{z}).$$

  - **end**
  - $m \leftarrow m - 1$

end

4. **for** $\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M}$ *in* π_M **do**
  - Calculate the expected probability $E[G_{x|\tilde{z}}(\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M})]$ using (6.3) by setting $m = M$.

end

5. Sample n_1 subspaces $\mathcal{B}^{(i)}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_M}$ with replacement based on $E[G_{x|\tilde{z}}(\mathcal{B}_{\varepsilon_1 \ldots \varepsilon_M})]$, for $i = 1, \ldots, n_1$.

6. **for** i *in* $1 : n_1$ **do**
  - Generate $Y^{(i)} \sim \mathrm{Uniform}(\mathcal{B}^{(i)}_{\varepsilon_1 \varepsilon_2 \ldots \varepsilon_M})$.

end

6.5 Simulation Studies

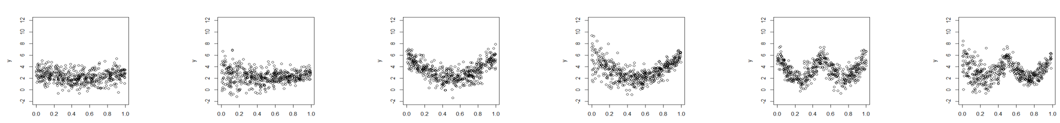
6.5.1 Simulation Settings

We consider four simulation settings, each containing six scenarios, as summarized in Table 6.2. Five hundred simulations are repeated for each setting. Sample sizes $n = 100, 250, 500, 1000$ and 2500 are considered.

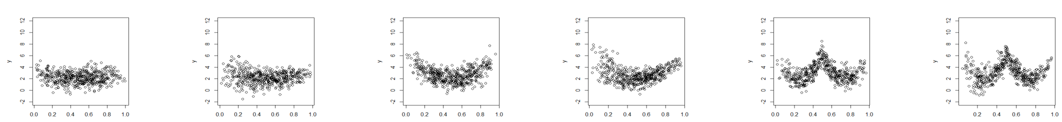
Suppose that the true regression model is of the form

$$Y_i = \phi(x_i) + \varepsilon_i$$

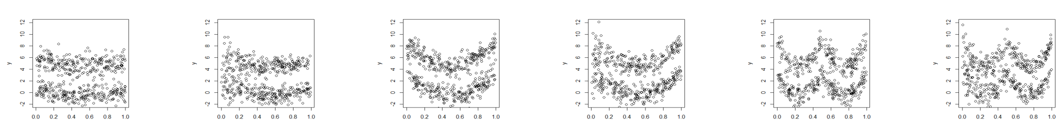
for $i = 1, \ldots, n$, where $\phi(x)$ is a regression function of covariates and $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. random errors. The covariate variable(s) x_i, $i = 1, \ldots, n$, are generated independently from the distributions in the second column of Table 6.2, the random errors are simulated independently according to those in the last column, and the response is obtained as the $\phi(\cdot)$ functions in the third column evaluated at the generated x_i plus the generated random error term.



(a) Setting 6.1.1 (b) Setting 6.1.2 (c) Setting 6.1.3 (d) Setting 6.1.4 (e) Setting 6.1.5 (f) Setting 6.1.6



(g) Setting 6.2.1 (h) Setting 6.2.2 (i) Setting 6.2.3 (j) Setting 6.2.4 (k) Setting 6.2.5 (l) Setting 6.2.6



(m) Setting 6.3.1 (n) Setting 6.3.2 (o) Setting 6.3.3 (p) Setting 6.3.4 (q) Setting 6.3.5 (r) Setting 6.3.6

Figure 6.1: A scatterplot of the response versus the covariate in Settings 6.1-6.3 ($n = 500$)

Table 6.2: The distributions of the covariate(s), the regression function and the distributions of random errors of six scenarios in each of the four simulation settings

Setting	X	$\phi(x)$	ε
6.1.1	Unif([0,1])	$(2x-1)^2+2$	$N(0,1)$
6.1.2			$N(0,(x+0.5)^{-1}))$
6.1.3		$(4x-2)^2+2$	$N(0,1)$
6.1.4			$N(0,(x+0.5)^{-1}))$
6.1.5		$\begin{cases}(8x-2)^2+2, & x \le 0.5\\(8x-6)^2+2, & x > 0.5\end{cases}$	$N(0,1)$
6.1.6			$N(0,(x+0.5)^{-1}))$
6.2.1	Beta(2,2)	$(2x-1)^2+2$	$N(0,1)$
6.2.2			$N(0,(x+0.5)^{-1}))$
6.2.3		$(4x-2)^2+2$	$N(0,1)$
6.2.4			$N(0,(x+0.5)^{-1}))$
6.2.5		$\begin{cases}(8x-2)^2+2, & x \le 0.5\\(8x-6)^2+2, & x > 0.5\end{cases}$	$N(0,1)$
6.2.6			$N(0,(x+0.5)^{-1}))$
6.3.1	Unif([0,1])	$(2x-1)^2+2$	$0.5(N(2.5,1)+N(-2.5,1))$
6.3.2			$0.5(N(2.5,(x+0.5)^{-1})+N(-2.5,(x+0.5)^{-1}))$
6.3.3		$(4x-2)^2+2$	$0.5(N(2.5,1)+N(-2.5,1))$
6.3.4			$0.5(N(2.5,(x+0.5)^{-1})+N(-2.5,(x+0.5)^{-1}))$
6.3.5		$\begin{cases}(8x-2)^2+2, & x \le 0.5\\(8x-6)^2+2, & x > 0.5\end{cases}$	$0.5(N(2.5,1)+N(-2.5,1))$
6.3.6			$0.5(N(2.5,(x+0.5)^{-1})+N(-2.5,(x+0.5)^{-1}))$
6.4.1	Unif($[0,1]^2$)	$(2x_1-1)^2+(2x_2-1)^2+2$	$N(0,1)$
6.4.2			$N(0,(x_1+x_2+0.3)^{-1})$
6.4.3		$(4x_1-2)^2+(4x_2-2)^2+2$	$N(0,1)$
6.4.4			$N(0,(x_1+x_2+0.3)^{-1})$
6.4.5		$\begin{cases}(8x_1-2)^2+(8x_2-2)^2+2, & x_1 \le 0.5 \text{ and } x_2 \le 0.5\\(8x_1-2)^2+(8x_2-6)^2+2, & x_1 \le 0.5 \text{ and } x_2 > 0.5\\(8x_1-6)^2+(8x_2-2)^2+2, & x_1 > 0.5 \text{ and } x_2 \le 0.5\\(8x_1-6)^2+(8x_2-6)^2+2, & x_1 > 0.5 \text{ and } x_2 > 0.5\end{cases}$	$N(0,1)$
6.4.6			$N(0,(x_1+x_2+0.3)^{-1})$

We consider a single covariate ($p = 1$) for Settings 6.1-6.3 and two covariates ($p = 2$) for Setting 6.4. Setting 6.1 considers a uniform covariate and a normal distributed random error term, Setting 6.2 considers a non-uniform covariate and random errors following the same distributions as those in Setting 6.1, and Setting 6.3 considers random errors following a mixture of normal distributions with uniform distributed covariates. Moreover, the regression function is assumed to take a quadratic form for Scenarios 1-4 and a non-smooth check function for Scenarios 5-6 in all settings. The distributions of the random error term are set to be a normal distribution with a fixed variance in Scenarios 1, 3, and 5, and a normal distribution with a covariate-dependent variance in Scenarios 2, 4, and 6 across settings. Figure 6.1 contains a scatterplot of the response versus the covariate in Settings 6.1-6.3 to show the shape of the data.

We compare the proposed PTNN model with the kernel density estimation (Kernel), Polya tree density estimation (PT), and the Linear Dependent Tail Free Process (LDTFP) method (Jara and Hanson, 2011). The bandwidth of the kernel method is selected using the Silverman's rule of thumb (Silverman, 1986). The PT density estimation is conducted directly to the joint density, $f(y, x)$, of $(Y, X^{\mathrm{T}})^{\mathrm{T}}$, and the truncation level M is set as 9. The LDTFP model is implemented using the `R` package `DPpackage`. We consider linear predictors and quadratic predictors, labeled as LDTFP1 and LDTFP2, respectively.

6.5.2 Evaluation Metrics

We employ the following two metrics to evaluate the performance of the proposed PTNN model:

1. *Kullback-Leibler Divergence (K-L)*: K-L divergence measures the difference between the true conditional density $f(y|x)$ and the nonparametric density estimate, denoted as $\hat{f}(y|x)$, by the formula

$$KL = \int_{\mathcal{S}_x} \int_{\mathcal{S}} \log \frac{f(y|x)}{\hat{f}(y|x)} f(y|x) f(x) dy dx. \tag{6.7}$$

2. *Mean Integrated Squared Error (MISE)*: To measure of the difference between the true conditional density $f(y|x)$ and the corresponding nonparametric density estimate $\hat{f}(y|x)$, the L_2 norm can be considered and the quantity

$$\int_{\mathcal{S}} [f(y|x) - \hat{f}(y|x)]^2 dy \tag{6.8}$$

is evaluated at the covariate value x. MISE integrates the quantity (6.8) over the distribution of covariates

$$MISE = \int_{\mathcal{S}_x} \int_{\mathcal{S}} [f(y|x) - \hat{f}(y|x)]^2 dy f(x) dx. \tag{6.9}$$

6.5.3 Simulation Results

We report the curves of the K-L divergences and the square roots of MISEs for various nonparametric regression models as the sample size increases across the designed simulation settings as given in Table 6.2.

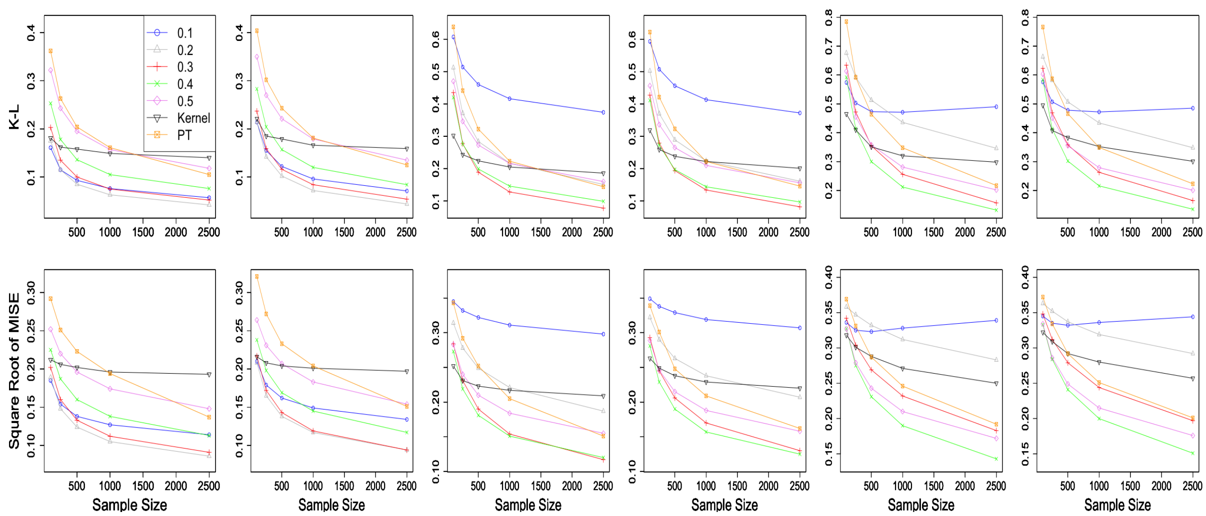
Monte Carlo Based Results

In this subsection, we calculate the K-L divergences in (6.7) and the MISEs in (6.9) using the Monte Carlo method and report the results for Settings 6.1-6.4 in Figures 6.2-6.5, respectively. Each figure contains 12 sub-figures. The 6 top figures correspond to the K-L divergences and the 6 bottom ones correspond to the square root of MISEs. In each subfigure, we report the desired metric changing with respect to the sample size for PTNN using a Gaussian weight function with $\eta = 0.1$ (round symbol and blue line), 0.2 (triangle up symbol with grey line), 0.3 (plus sign symbol with red line), 0.4 (X symbol with green line), 0.5 (diamond with pink line), kernel density estimation (triangle down symbol with black line) and PT density estimation (square with orange line). The numerical values of K-L divergences and the square root of MISEs and their standard errors for PTNN using both the uniform weight function and Gaussian kernel weight function with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel and PT density estimations are given in Sections E.2.1, E.2.3, E.2.4 and E.2.6 in Appendix for Settings 6.1-6.4, respectively.

The PTNN models based on the Gaussian weight function are constantly better than those based on uniform weight function in all settings as shown in Section E.2 in Appendix, therefore, we only report the curves of the PTNN with Gaussian weights in Figures 6.2-6.5. For the PTNN method, the K-L divergences and MISEs always decrease as the sample size increases, corroborating the consistency results of the PTNN model proved in Section 6.3. The performance of PTNN method varies with different choices of the tuning parameters η. For scenarios 1 and 2 across settings, the response changes rarely as the covariate changes as seen in Figure 6.1, therefore, the best approximation to the true density occurs when $\eta = 0.1$ or 0.2, a value smaller than the optimal value 1/3. For scenarios 3 and 4, the response changes moderately as the covariate changes, PTNN with $\eta = 0.3$ or 0.4 usually gives top performance. In scenarios 5 and 6 when the response changes more dramatically as the covariate changes, the MISEs tend to identify PTNN with $\eta = 0.4$ or 0.5 as the best performed PTNN method. These simulation results are consistent with our assessment in Section 6.4.1 on the selection of the tuning parameter η. As the performance of the PTNN model highly depends on the choice of η, it motivates a procedure to select η in practice. We discuss the use of a cross-valuation procedure in details when applying the propose PTNN to analyze a real dataset in Section 6.6.

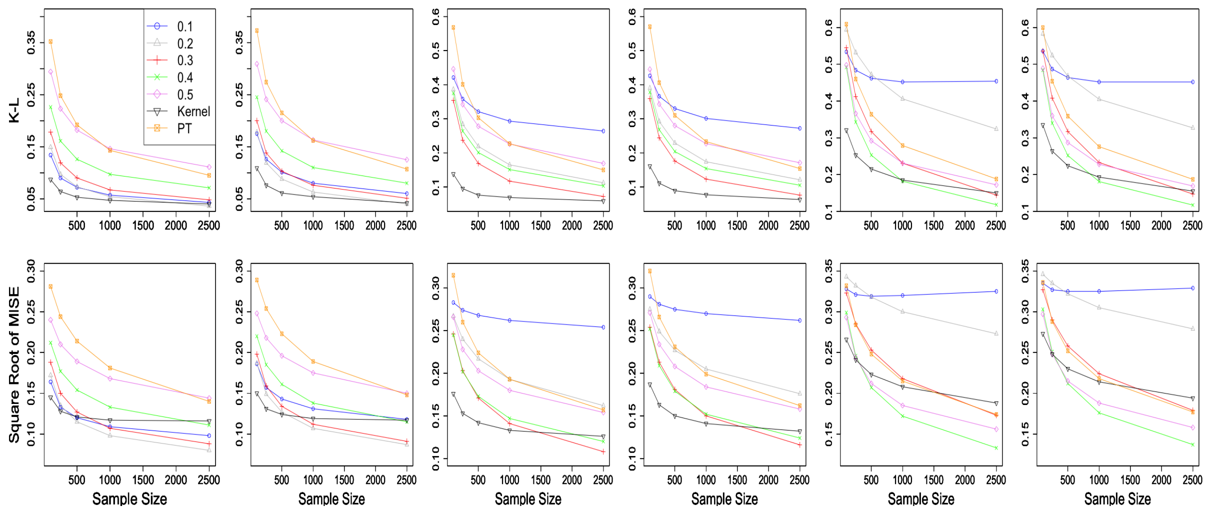
Comparing with the kernel density estimation, which is represented by the black curve in each subfigure, the PTNN decreases faster than the kernel method as the sample size increases, which suggests that the PTNN has a faster convergence rate than the kernel method. The Polya tree density estimation generally performs poorly in most settings and tends to provide the worst or the second worst results comparing with the other considered nonparametric methods. It is worth to note that the kernel density estimation performs better than the PTNN method, especially with smaller sample size, in Setting 6.2, where the covariate values are generated from a Beta distribution and hence more concentrated to the midpoint of the [0, 1] interval. In this scenario, the PTNN method is undermined by the sparsity of covariate values near 0 and 1. However, the PTNN method approximates the true conditional distribution better than the kernel density estimation in all other three settings, including Setting 6.3, of which the error distribution is a mixture of two normal distributions, and Setting 6.4 with two covariates.

In summary, the consistency property the PTNN model is confirmed and the PTNN generally outperforms the kernel and PT density estimation in terms of accuracy and convergence rate with a well-selected tuning parameter η. However, the kernel density estimation may provide more accurate estimation of the true conditional density when data are sparse near the boundary values of covariates and the sample size is not large.



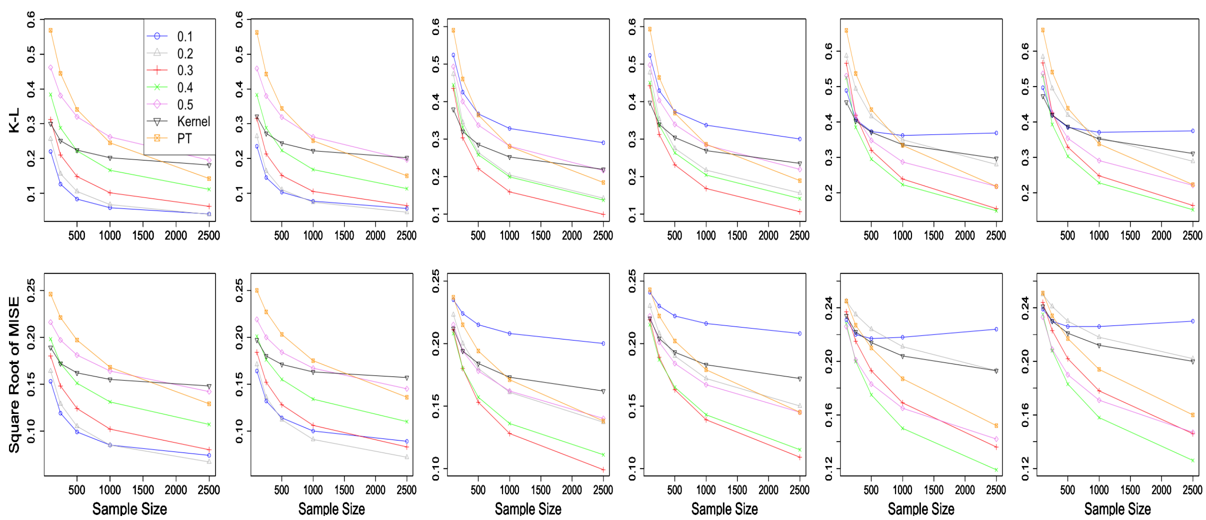
(a) Setting 6.1.1 (b) Setting 6.1.2 (c) Setting 6.1.3 (d) Setting 6.1.4 (e) Setting 6.1.5 (f) Setting 6.1.6

Figure 6.2: K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.1



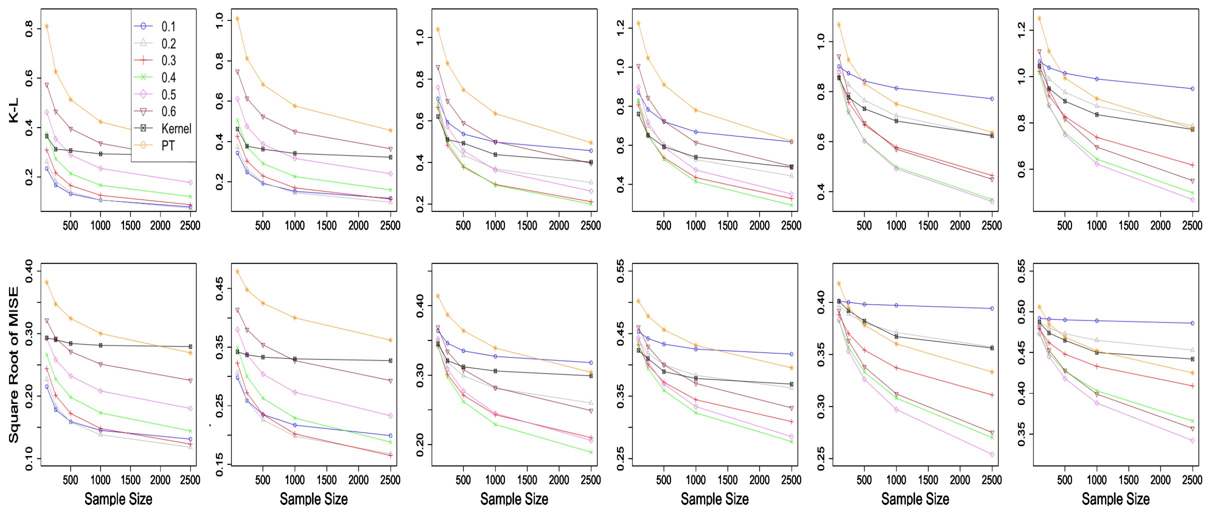
(a) Setting 6.2.1 (b) Setting 6.2.2 (c) Setting 6.2.3 (d) Setting 6.2.4 (e) Setting 6.2.5 (f) Setting 6.2.6

Figure 6.3: K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.2



(a) Setting 6.3.1 (b) Setting 6.3.2 (c) Setting 6.3.3 (d) Setting 6.3.4 (e) Setting 6.3.5 (f) Setting 6.3.6

Figure 6.4: K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.3



(a) Setting 6.4.1 (b) Setting 6.4.2 (c) Setting 6.4.3 (d) Setting 6.4.4 (e) Setting 6.4.5 (f) Setting 6.4.6

Figure 6.5: K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method and Polya tree density estimation for Setting 6.4

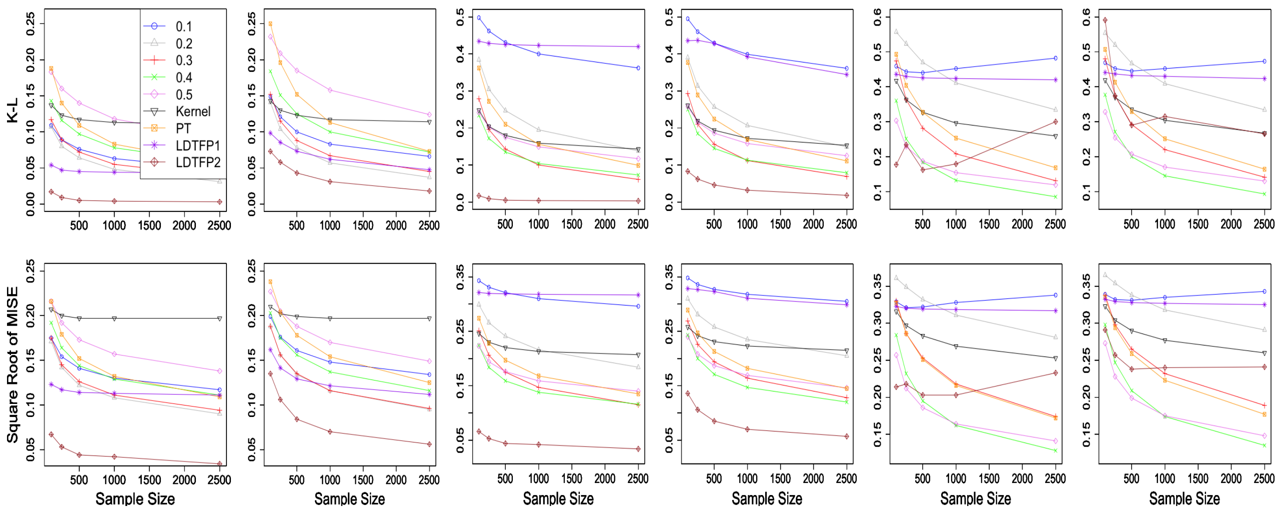
Grid-Based Results

The LDTFP method is implemented using `R` package `DPpackage`, of which the output of $\hat{f}(y|x)$ is only available at some grid points of x. Therefore, to compare the performance of PTNN with different values of η, kernel density estimation, PT density estimation with LDTFP method, we compute the K-L divergences in (6.7) and MISEs in (6.9) using the grid-based methods, which are conducted in the following way: (i) 100 evenly distributed values are selected on $[0, 1]$ for the covariate and another 100 evenly distributed values are selected on the domain of response, which are combined to yield 10,000 grid points; (ii) the K-L divergences or MISEs are obtained by evaluating their integrands at the 10,000 grid points and taking an empirical average.

We plot the grid-based K-L divergences and MISEs of the PTNN models with $\eta = 0.1$, 0.2, 0.3, 0.4, and 0.5, kernel density estimation, Polya tree density estimation, LDTFP1 (with linear predictor) and LDTFP2 (with quadratic predictor) for simulation Settings 6.1 and 6.3 in Figures 6.6 and 6.7. Figures are arranged in a similar manner as Figures 6.2-6.5, and the detailed numerical results are provided in Appendix E.2, with Setting 6.1 in Section E.2.2 and Setting 6.3 in Section E.2.5.

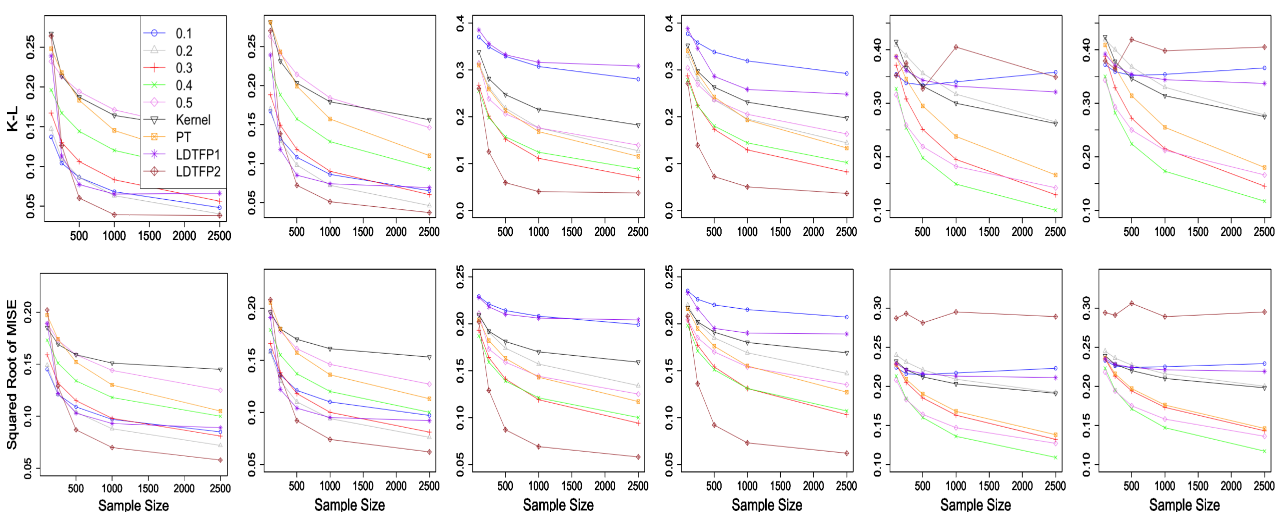
The results of the PTNN models, the kernel and PT density estimation are similar to those in Section 6.5.3, thus our discussion here mainly focuses on the results of LDTFP models to avoid redundancy. As introduced in Section 6.1, the LDTFP method models the random splitting probabilities by a logistic regression, in other words, a logistic transformation of the random splitting probabilities is linked to a regression function of covariates. The regression function can assume a linear form of covariates (LDTFP1) or a quadratic form (LDTFP2), which, in nature, makes the LDTFP not fully nonparametric. For Scenarios 1-4 of Settings 6.1 and 6.3, the true underlying regression function is quadratic, thus the LDTFP2 with a correctly specified regression function outperforms all other nonparametric methods in terms of estimation accuracy, as expected. The LDTPF1 behaves reasonably well in Scenarios 1 and 2 in both settings when the response changes gently with respect to the changes of the covariate and a linear regression function is sensible. However, both LDTFP methods fail for Scenarios 5-6 in both settings when the true underlying relationship is a segmented model and the regression functions in LDTFP1 and LDTFP2, assumed to be linear or quadratic, respectively, are deemed as misspecified models.

In summary, the LDTFP surely exhibits higher estimation accuracy when the parametric regression function is correctly specified, but suffers brutally when the parametric assumption is violated. However, our proposed PTNN model enjoys the advantage of model robustness compared to the LDTFP models. The proposed PTNN sacrifices some efficiency in estimation to obtain robust performance under complicated regression relations.



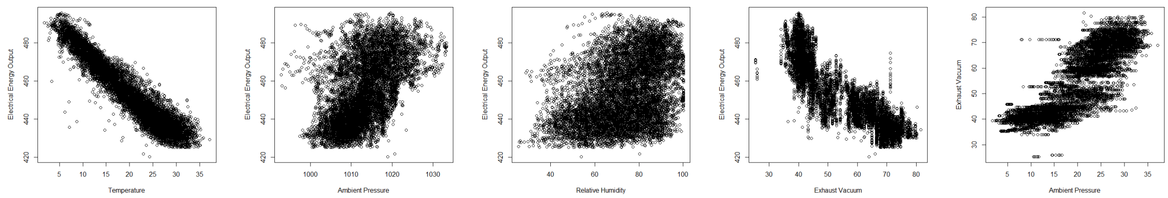
(a) Setting 6.1.1 (b) Setting 6.1.2 (c) Setting 6.1.3 (d) Setting 6.1.4 (e) Setting 6.1.5 (f) Setting 6.1.6

Figure 6.6: Grid-based K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method, Polya tree density estimation, LDTFP1 with linear predictor and LDTFP2 with quadratic predictor for Setting 6.1



(a) Setting 6.3.1 (b) Setting 6.3.2 (c) Setting 6.3.3 (d) Setting 6.3.4 (e) Setting 6.3.5 (f) Setting 6.3.6

Figure 6.7: Grid-based K-L divergences and square root of MISEs versus sample size for PTNN (Gaussian kernel weight) when $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel method, Polya tree density estimation, LDTFP1 with linear predictor and LDTFP2 with quadratic predictor for Setting 6.3



(a) y v.s. x_1 (b) y v.s. x_2 (c) y v.s. x_3 (d) y v.s. x_4 (e) x_4 v.s. x_2

Figure 6.8: Scatter plots of the Net Hourly Electrical Energy Output (y) versus Temperature (x_1), Ambient Pressure (x_2), Relative Humidity (x_3) and Exhaust Vacuum (x_4), respectively, and the plot of Exhaust Vacuum (x_4) versus Ambient Pressure (x_2)

6.6 Data Analysis

6.6.1 Dataset Description

We apply the proposed PTNN to analyze the Combined Cycle Power Plant dataset from the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant). This is an electricity dataset, containing 9568 data points collected from a Combined Cycle Power Plant, which works on full load over 6 years (2006-2011). The response of interest is the Net Hourly Electrical Energy Output (y) of the plant. There are four features in the dataset: Temperature (x_1), Ambient Pressure (x_2), Relative Humidity (x_3), and Exhaust Vacuum (x_4). We aim to build a regression model to understand the relationship between the electrical energy output and the four features. Figure 6.8 contains the scatter plots of the electrical energy output versus each of the four covariates in subfigures (a)-(d), respectively, and that of Exhaust Vacuum versus Ambient Pressure in subfigure (e), all showing a linear relationship.

We aim at evaluating the prediction accuracy of our proposed PTNN regression model and compare it to other benchmark nonparametric regression methods using the Combined Cycle Power Plant dataset. We divide the dataset into two subsets: the training set of the first 6000 data points $\tilde{z}_i = (y_i, x_{i1}, x_{i2}, x_{i3}, x_{i4})^{\mathrm{T}}$ for $i = 1, \ldots, 6000$, which is used to fit the nonparametric models, and the test set of the last 3568 data points, which is used to calculate the prediction errors and evaluate the prediction performance of the fitted models.

Let y_i and $\hat{y}_i$ be the true value and the predicted value for the ith subject in the test set, respectively. The predicted value of the ith subject is calculated using the expected value of the response given covariate value x_i based on the fitted nonparametric model. We use two metrics to report the prediction performance: the Mean Absolute Error (MAE), $MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i|$, and the Root Mean Square Error (RMSE), $RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}$, where m is the size of the test set.

6.6.2 Selection of Tuning Parameter η

The extensive simulation studies in Section 6.5 suggest that the performance of the PTNN depends on the choice of η to a large degree. It is vital to develop a procedure to select an "optimal" η when the underlying relationship between the response and covariates is unknown. We propose to use the V-fold cross-validation procedure to select an optimal value from a set of candidate values $\eta_1, \ldots, \eta_L$, by minimizing the mean absolute error which measures the distance between the true and predicted responses as described below.

In the V-fold cross-validation procedure, the training set is randomly divided into V mutually exclusive subsets with an equal or nearly equal size; common choices of V range from 5 to 10. For $v = 1, \ldots, V$, we fit PTNN models with different values of η using the training data with subset v excluded. Thereby, for each $v = 1, \ldots, V$, a sequence of PTNN models $\hat{f}_v^{(\eta_l)}(y|x)$ is obtained for $l = 1, \ldots, L$. Next, we define the cross-validated estimator of the mean absolute error as

$$RCV(\eta_l) = \frac{1}{n}\sum_{v=1}^{V}\sum_{i=1}^{n} I(S_{i,v} = 1)|y_i - \hat{y}_v^{(\eta_l)}(x_i)|,$$

where $S_{i,v}$ indicates whether subject i belongs to subset v and $\hat{y}_v^{(\eta_l)}(x_i)$ is the predicted value from model $\hat{f}_v^{(\eta_l)}(y|x)$ at the covariate value x_i. For $l = 1, \ldots, L$, calculate $RCV(\eta_l)$ and the "optimal" η is the value which minimizes $RCV(\eta_l)$.

For the Combined Cycle Power Plant dataset, the size of training set is $n = 6000$ and the 5-fold cross-validation procedure is conducted to select an "optimal" η from candidate values $\{0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$. Figure 6.9 displays how the cross-validated mean absolute error changes with respect to the value of η. The cross-validation error is minimized at $\eta = 0.85$. The value of η is fairly large as the variability of the response for some particular values of x_2, x_3 or x_4 can be quite large as shown in Figure 6.8.

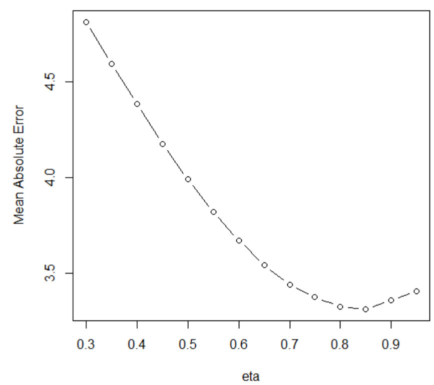


Figure 6.9: The cross-validated mean absolute error changes with respect to the value of η

6.6.3 Models to Compare

We compare the prediction performance of the PTNN with $\eta = 0.85$ with the following models:

1. (*KDE*) Kernel density estimation: the bandwidth is selected using the Silverman's rule of thumb (Silverman, 1986);
2. (*KR*) Kernel regression: the multivariate Nadaraya-Watson estimator (Ruppert and Wand, 1994) is used;
3. (*LDTFP*) Linear Dependent Tail Free Process.
4. (*LM1*) Linear model I: a simple linear regression model of the response over the four features of interest in the dataset;

$$Y_i = \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \beta_4 x_{i3} + \beta_5 x_{i4} + \varepsilon_i \tag{6.10}$$

   for $i = 1, \ldots, n$, where $\varepsilon_i \sim N(0, \sigma^2)$.

5. (*LM2*) Linear model II: a simple linear regression with the interaction between x_2 and x_4.

$$Y_i = \beta_1 + \beta_2 x_{i1} + \beta_3 x_{i2} + \beta_4 x_{i3} + \beta_5 x_{i4} + \beta_6 x_{i2} x_{i4} + \varepsilon_i \tag{6.11}$$

   for $i = 1, \ldots, n$, where $\varepsilon_i \sim N(0, \sigma^2)$.

6. (*PT*) Polya tree density estimation: the truncation level M is set to be 8.

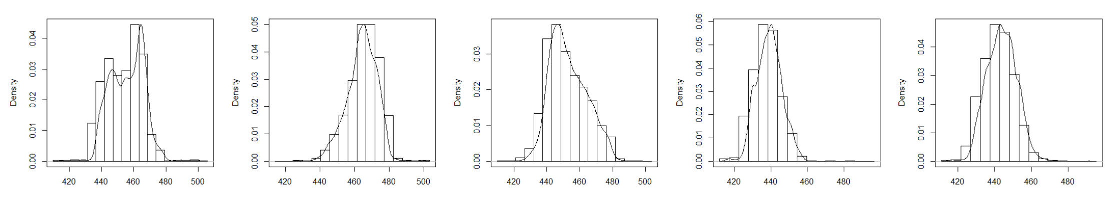
6.6.4 Results

Table 6.3 summarizes the prediction results of MAEs and RMSEs for the PTNN versus the kernel density estimation (KDE), kernel regression (KR), linear dependent tail free process with a linear regression function (LDTFP1), linear model I (LM1) (6.10), linear model II (LM2) (6.11) and Polya tree density estimation (PT) methods described in Section 6.6.3. The PTNN provides the smallest MAE and RMSE, suggesting the best prediction performance.

Table 6.3: Prediction performance of PTNN versus the kernel density estimation (KDE), kernel regression (KR), linear dependent tail free process (LDTFP1), linear model I (LM1) (6.10), linear model II (LM2) (6.11) and Polya tree density estimation (PT) methods

	PTNN	KDE	KR	LDTFP1	LM1	LM2	PT
MAE	3.203	7.570	3.292	3.590	3.500	3.603	15.215
RMSE	4.115	9.569	4.193	4.496	4.379	4.493	17.397

The histograms overlaid with the curves of the conditional density estimated by PTNN model at 5 different covariate values are provided in Figure 6.10. With different covariate values, the densities of conditional density exhibit a diversity of shapes, including a bimodal distribution in subfigure (a), and skewed shapes in subfigures (b) - (e). The proposed PTNN regression, as a fully nonparametric approach, provides decent fits to a variety of complicated distributions.



(a) (b) (c) (d) (e)

Figure 6.10: Histograms with superimposed curves of the estimated conditional densities by PTNN model at 5 different covariate values: (a) $(20, 40.36, 1007.89, 40.56)^{\mathrm{T}}$, (b) $(20, 40.36, 1018.30, 85.16)^{\mathrm{T}}$, (c) $(19.69, 54.28, 1013.20, 73.21)^{\mathrm{T}}$, (d) $(22.11, 66.56, 1007.89, 40.56)^{\mathrm{T}}$, (e) $(22.11, 66.56, 1018.30, 85.16)^{\mathrm{T}}$

6.7 General Remarks

In this chapter, we propose a fully nonparametric regression model, a Polya tree based nearest neighbor regression, which provides consistent and robust performance in characterizing complex relationship between responses and covariates. Since the PTNN requires no parametric assumption on both the regression function or the error distribution, it provides robust performance in different irregular regression relations, as illustrated in the numerical studies. Furthermore, in the simulation studies, the results show that the PTNN has a faster convergence rate than kernel density estimation. Generally speaking, using PTNN to model the conditional density $f(y|x)$ provides a comprehensive overview of the regression relations, and many common interested quantities can be derived from the conditional distribution, such as the response expectation, variance or confidence interval. Compared with the nonparametric methods to model the regression functions, such as the spline method or the wavelet method, the PTNN model can characterize the variations in the response better.

It is noteworthy that the PTNN is different from the Bayesian nonparametric smoothed density estimation method proposed by Hanson et al. (2018), in which attention was given particularly on the spatial data. Our proposed PTNN features a "nearest neighbor" of covariate values considered, which reduces the computational burden and facilitates desirable theoretical results of convergence. In Hanson et al. (2018), the weight function was specified to be the Mahalanobis distance, a common choice in spatial analysis, while our PTNN allow the weight function to adopt more flexible forms. Finally, Hanson et al. (2018) can be extended to censored data, which is an interesting future direction for PTNN.

Chapter 7

Discussion and Future Work

In this chapter, we present a summary and briefly mention some potential future work.

Chapter 2:

In this chapter, we propose a R-Vine based regression model for analyzing periodic longitudinal data. We introduce composite likelihood methods which outperform the likelihood-based methods in terms of robustness and computational efficiency. We conduct extensive simulation studies to evaluate the performance of the proposed methods. The numerical studies suggest that the (conditional) bivariate copulas can still be accurately selected and the parameters of interest can be consistently estimated with moderate efficiency loss when simultaneous procedure is used. Moreover, the model provides more precise prediction results than the conventional models in both the simulation studies and the real data analysis. Time extrapolation is what we usually care about in prediction problems, while subject extrapolation is valuable for imputing missing response values.

Chapter 3:

In this chapter, we propose a Bayesian hierarchical copula model to characterize the subject-level dependency for data with a hierarchical structure. The model is flexible enough to account for data coming from multiple sources with different sample sizes. We use a "layer by layer" sampling scheme, combined with the Metropolis Hasting algorithm to sample from the posterior distribution. Simulation studies and data analysis are conducted to compare the estimators obtained from our proposed BHCM to the likelihood-based estimators. The results show that the BHCM outperforms the maximum likelihood methods and this advantage gets more obvious when the sample size is small. The proposed model captures the between-cluster variability and facilitates information sharing across clusters through delineating the hierarchical structures.

Our analysis under BHCM was conducted under correctly specified models. It will be useful to understand the robustness of the proposed model. The effects of misspecification of the copula function in one cluster on the estimation of parameters in another cluster should be explored.

Chapter 4:

In this chapter, we propose a M-DPM-CM to identify similar dependence structures for dependent data coming from a hierarchical structure. We construct a mixed copula model for the subject-level dependence, in which the copula selection indicators and copula parameters follow a DP prior. We can make inference on our proposed model by introducing a Gibbs sampler algorithm with augmented parameters. The M-DPM-CM can perform grouping and copula selection simultaneously. Simulation studies and data analysis are conducted to compare the M-DPM-CM to the conventional copula selection method using AIC. The results show that the M-DPM-CM can accurately recover the true grouping structure with a moderate sample size, and in turn achieve a more accurate model selection and more efficient parameter estimation than the conventional AIC method.

The M-DPM-CM can also be used for copula selection in vine copula models. Working with a given vine structure, a tree-by-tree selection of bivariate copulas can be done by using M-DPM-CM and regarding each pair of bivariate data as a cluster. A more sophisticated M-DPM-CM with an extra indicator corresponding to different vine structures can be developed to select the vine structure, copula functions and parameter values simultaneously.

When performing model selection, it is common to introduce some penalty terms to penalize on the number of parameters in the model, such as AIC or BIC. If we consider including copula functions with different number of parameters in the set $\mathcal{F}$, the proposed M-DPM-CM can be easily extended by including some penalty terms in the Gibbs sampler.

Chapter 5:

In this chapter, we propose a Polya tree Monte Carlo method which utilizes the Polya tree distribution in an innovative way. We describe multiple sampling algorithms to sample from potentially complex multi-modal distributions. Our proposed PTMC algorithms have several merits compared to the MCMC algorithms. When sampling from low-dimensional distributions, our proposed Algorithms 5.1 is superior in computational speed and sampling efficiency. When sampling from multi-dimensional distributions, the proposed Algorithm 5.4 is free of the hassle to tune the stepsize and can recover multiple modes of the target distribution. The proposed algorithms enjoy a broad scope of applications.

In the PTMC algorithms, only the density function of the target distribution is evaluated. However, in the PTMC MH algorithm, the gradient information of density function can also be incorporated to improve the sampling efficiency in a similar way as Langevin MC (Radford et al., 2010), which is an interesting direction for the PTMC algorithms.

Chapter 6:

In this chapter, we propose a fully nonparametric regression model, a Polya tree based nearest neighbor regression, which provides consistent and robust performance in characterizing complex relationship between responses and covariates. Some fully nonparametric regression methods, such as the LDTFP model (Jara and Hanson, 2011), make assumptions about the form of predictors, therefore, fail to provide desirable results when the form of predictors is misspecified. Our proposed PTNN model literally makes no parametric assumptions about the regression function or the error distribution, thus it exhibits more robust performance across various designed simulation settings in terms of different forms of covariate distribution, regression functions and error distributions. Another merit of our proposed PTNN model is its faster convergence performance than other compared nonparametric methods. As demonstrated by the simulation studies that the K-L divergence and MISE are reduced faster for PTNN than for kernel density estimation as the sample size increases. Moreover, the inference procedure of PTNN is computationally simple and efficient. For future directions, the PTNN model can be extended to different types of data, such as censored data, or data with mixed types.

References

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44:182–198.

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*. Springer, New York.

Albert, P. S. (2000). A transitional model for longitudinal binary data subject to nonignorable missing data. *Biometrics*, 56:602–608.

Andersen, E. W. (2005). Two-stage estimation in copula models used in family studies. *Lifetime Data Analysis*, 11:333–350.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing*, 18:343–373.

Atchadé, Y. F. and Rosenthal, J. S. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, 11:815–828.

Bai, Y., Craiu, R. V., and Di Narzo, A. F. (2011). Divide and conquer: A mixture-based approach to regional adaptation for MCMC. *Journal of Computational and Graphical Statistics*, 20:63–79.

Becker, M. and Balagtas, C. C. (1993). Marginal modeling of binary cross-over data. *Biometrics*, 49:997–1009.

Bedford, T. and Cooke, R. M. (2002). Vines: a new graphical model for dependent random variables. *The Annals of Statistics*, 30:1031–1068.

Berthonnaud, E., Dimnet, J., Roussouly, P., and Labelle, H. (2005). Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters. *Clinical Spine Surgery*, 18:40–47.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is “nearest neighbor” meaningful? In *International Conference on Database Theory*. Springer, New York.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355.

Boehm, L., Reich, B. J., and Bandyopadhyay, D. (2013). Bridging conditional and marginal inference for spatially referenced binary data. *Biometrics*, 69:545–554.

Bogaerts, K. and Lesaffre, E. (2008). Modeling the association of bivariate interval-censored data using the copula approach. *Statistics in Medicine*, 27:6379–6392.

Borsuk, M. E., Higdon, D., Stow, C. A., and Reckhow, K. H. (2001). A Bayesian hierarchical model to predict benthic oxygen demand from organic matter loading in estuaries and coastal zones. *Ecological Modelling*, 143:165–181.

Braekers, R. and Veraverbeke, N. (2005). A copula-graphic estimator for the conditional survival function under dependent censoring. *Canadian Journal of Statistics*, 33:429–447.

Brechmann, E. C., Czado, C., and Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40:68–85.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. CRC Press, Florida.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, 88:9–25.

Breslow, N. E. and Lin, X. (1995). Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika*, 82:81–91.

Broët, P., Richardson, S., and Radvanyi, F. (2002). Bayesian hierarchical model for identifying changes in gene expression from microarray experiments. *Journal of Computational Biology*, 9:671–683.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC Press, Florida.

Brown, E. R. and Ibrahim, J. G. (2003). A Bayesian semiparametric joint hierarchical model for longitudinal and survival data. *Biometrics*, 59:221–228.

Cai, B., Meyer, R., and Perron, F. (2008). Metropolis–Hastings algorithms with adaptive proposals. *Statistics and Computing*, 18:421–433.

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18:447–459.

Chen, M.-H. and Shao, Q.-M. (1999). Monte Carlo estimation of Bayesian credible and HPD intervals. *Journal of Computational and Graphical Statistics*, 8:69–92.

Chen, X. and Fan, Y. (2005). Pseudo-likelihood ratio tests for semiparametric multivariate copula model selection. *Canadian Journal of Statistics*, 33:389–414.

Chen, X. and Fan, Y. (2006a). Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics*, 135:125–154.

Chen, X. and Fan, Y. (2006b). Estimation of copula-based semiparametric time series models. *Journal of Econometrics*, 130:307–335.

Cheon, K., Thoma, M. E., and Kong, X. (2014). A mixture of transition models for heterogeneous longitudinal ordinal data: with applications to longitudinal bacterial vaginosis data. *Statistics in Medicine*, 33:3204–3213.

Cherubini, U., Luciano, E., and Vecchiato, W. (2004). *Copula Methods in Finance*. John Wiley & Sons.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4:266–298.

Chollete, L., Heinen, A., and Valdesogo, A. (2009). Modeling international financial returns with a multivariate regime-switching copula. *Journal of Financial Econometrics*, 7:437–480.

Chui, C. K. (2016). *An Introduction to Wavelets*. Elsevier, Toronto.

Chung, Y. and Dunson, D. B. (2011). The local Dirichlet process. *Annals of the Institute of Statistical Mathematics*, 63:59–80.

Congdon, P. (2014). *Applied Bayesian Modelling*. John Wiley & Sons, London.

Congdon, P. D. (2010). *Applied Bayesian Hierarchical Methods*. CRC Press, Florida.

Cook, R. J. (1999). A mixed model for two-state Markov processes under panel observation. *Biometrics*, 55:915–920.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21–27.

Cowles, M. K. and Carlin, B. P. (1996). Markov Chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91:883–904.

Craiu, R. V., Duchesne, T., Fortin, D., and Baillargeon, S. (2011). Conditional logistic regression with longitudinal follow-up and individual-level random coefficients: A stable and efficient two-step estimation method. *Journal of Computational and Graphical Statistics*, 20:767–784.

Craiu, R. V., Rosenthal, J., and Yang, C. (2009). Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association*, 104:1454–1466.

Craiu, R. V. and Rosenthal, J. S. (2014). Bayesian computation via Markov Chain Monte Carlo. *Annual Review of Statistics and Its Application*, 1:179–201.

Dahl, D. B. (2006). Model-based clustering for expression data via a Dirichlet process mixture model. *Bayesian Inference for Gene Expression and Proteomics*, 4:201–218.

Davis, R. A., Lii, K.-S., and Politis, D. N. (2011). Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*. Springer, New York.

De Boor, C. (1978). *A Practical Guide to Splines*. Springer, New York.

De Iorio, M., Müller, P., Rosner, G. L., and MacEachern, S. N. (2004). An ANOVA model for dependent random measures. *Journal of the American Statistical Association*, 99(465):205–215.

DeGroot, M. H. (2005). *Optimal Statistical Decisions*. John Wiley & Sons, New Jersey.

Demuth, H. B., Beale, M. H., De Jess, O., and Hagan, M. T. (2014). *Neural Network Design*. PWS Publishing, Boston.

Devroye, L. (1986). Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*.

Di Lascio, F. M. L., Durante, F., and Pappada, R. (2017). Copula–based clustering methods. In *Copulas and Dependence Models with Applications*. Springer, New York.

Diggle, P., Heagerty, P., Heagerty, P. J., Liang, K.-Y., and Zeger, S. (2002). *Analysis of Longitudinal Data*. Oxford University Press, North York, Canada.

Dissmann, J., Brechmann, E., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69.

Domma, F., Giordano, S., and Perri, P. (2009). Statistical modelling of temporal dependence in financial data via a copula function. *Communications in Statistics*, 38:703–728.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Duchateau, L. and Janssen, P. (2005). Understanding heterogeneity in generalized mixed and frailty models. *The American Statistician*, 59:143–146.

Dunson, D. B., Pillai, N., and Park, J. H. (2007). Bayesian density regression. *Journal of the Royal Statistical Society: Series B*, 69:163–183.

Erhardt, T. M., Czado, C., and Schepsmeier, U. (2015). R-Vine models for spatial time seires with an application to daily mean temperature. *Biometrics*, 71:323–332.

Escobar, M. D. (1994). Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89:268–277.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588.

Eubank, R. L. (1999). *Nonparametric Regression and Spline Smoothing*. CRC Press, Florida.

Fei-Fei, L. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2. Institute of Electrical and Electronics Engineers.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. In *Recent Advances in Statistics*. Elsevier, New York.

Fermanian, J.-D. (2005). Goodness-of-fit tests for copulas. *Journal of Multivariate Analysis*, 95:119–152.

Fern, X. Z., Brodley, C. E., and Friedl, M. A. (2005). Correlation clustering for learning mixtures of canonical correlation models. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics.

Fitzmaurice, G., Davidian, M., Verbeke, G., and Molenberghs, G. (2009). *Longitudinal Data Analysis*. CRC Press, Florida.

Frahm, G., Junker, M., and Szimayer, A. (2003). Elliptical copulas: applicability and limitations. *Statistics & Probability Letters*, 63:275–286.

Frees, E. and Wang, P. (2006). Copula credibility for aggregate loss models. *Insurance: Mathematics and Economics*, 38:360–373.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer, New York.

Garson, G. D. (2012). *Hierarchical Linear Modeling: Guide and Applications*. Sage, California.

Gasser, T. and Müller, H.-G. (1979). Kernel estimation of regression functions. In *Smoothing Techniques for Curve Estimation*. Springer, New York.

Geerdens, C., Claeskens, G., and Janssen, P. (2016). Copula based flexible modeling of associations between clustered events times. *Lifetime Data Analysis*, 22:363–381.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. CRC Press, Florida.

Geman, S. and Geman, D. (1987). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *Readings in Computer Vision*. Elsevier, Toronto.

Genest, C. and MacKay, J. (1986a). The joy of copulas: bivariate distributions with uniform marginals. *The American Statistician*, 40:280–283.

Genest, C. and MacKay, R. J. (1986b). Copules Archimédiennes et families de lois bidimensionnelles dont les marges sont données. *Canadian Journal of Statistics*, 14:145–159.

Genest, C., Quessy, J.-F., and Rémillard, B. (2006). Goodness-of-fit procedures for copula models based on the probability integral transformation. *Scandinavian Journal of Statistics*, 33:337–366.

Genest, C., Rémillard, B., and Beaudoin, D. (2009). Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and Economics*, 44:199–213.

George, D. and Hawkins, J. (2005). A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 3. Institute of Electrical and Electronics Engineers.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov Chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90:909–920.

Ghidey, W., Lesaffre, E., and Eilers, P. (2004). Smooth random effects distribution in a linear mixed model. *Biometrics*, 60:945–953.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Giordani, P. and Kohn, R. (2010). Adaptive independent Metropolis–Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics*, 19:243–259.

Goldstein, H. (2011). *Multilevel Statistical Models*. John Wiley & Sons, London.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*.

Graves, T. L. (2011). Automatic step size selection in random walk Metropolis algorithms. *arXiv preprint arXiv:1103.5986*.

Griffin, J. E. and Steel, M. F. J. (2006). Order-based dependent Dirichlet processes. *Journal of the American Statistical Association*, 101:179–94.

Gruber, L. and Czado, C. (2015). Sequential Bayesian model selection of regular vine copulas. *Bayesian Analysis*, 10:937–963.

Gruber, L. and Czado, C. (2018). Bayesian model selection of regular vine copulas. *Bayesian Analysis*, 13:1107–1131.

Gustafson, P., Hossain, S., and Macnab, Y. C. (2006). Conservative prior distributions for variance parameters in hierarchical models. *Canadian Journal of Statistics*, 34:377–390.

Haario, H., Laine, M., Mira, A., and Saksman, E. (2006). DRAM: efficient adaptive MCMC. *Statistics and Computing*, 16:339–354.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7:223–242.

Hans, M. (2007). Estimation and model selection of copulas with an application to exchange rates. Research Memorandum 056, Maastricht University, Maastricht Research School of Economics of Technology and Organization (METEOR).

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001.

Hanson, T. and Johnson, W. O. (2004). A Bayesian semiparametric AFT model for interval-censored data. *Journal of Computational and Graphical Statistics*, 13:341–361.

Hanson, T., Zhou, H., and de Carvalho, V. I. (2018). Bayesian nonparametric spatially smoothed density estimation. In *New Frontiers of Biostatistics and Bioinformatics*. Springer, New York.

Hanson, T. E., Monteiro, J. V., and Jara, A. (2011). The Polya tree sampler: Toward efficient and automatic independent Metropolis–Hastings proposals. *Journal of Computational and Graphical Statistics*, 20:41–62.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

He, J., Li, H., Edmondson, A. C., Rader, D. J., and Li, M. (2012). A Gaussian copula approach for the analysis of secondary phenotypes in case-control genetic asscoiation studies. *Biostatistics*, 13:497–508.

Heagerty, P. J. (2002). Marginalized trasition models and likelihood inference for longitudinal categorical data. *Biometrics*, 58:342–351.

Heagerty, P. J. and Zeger, S. L. (2000). Marginalized multilevel models and likelihood inference (with discussion). *Statistical Science*, 15:1–26.

Hedeker, D. and Gibbons, R. D. (2006). *Longitudinal Data Analysis*. John Wiley & Sons, New Jersey.

Holden, L., Hauge, R., and Holden, M. (2009). Adaptive independent Metropolis–Hastings. *The Annals of Applied Probability*, 19:395–413.

Hu, L. (2006). Dependence patterns across financial markets: a mixed copula approach. *Applied Financial Economics*, 16:717–729.

Huang, X. and Zhang, N. (2008). Regression survival analysis with an assumed copula for dependent censoring: A sensitivity analysis approach. *Biometrics*, 64:1090–1099.

Ieva, F., Paganoni, A. M., and Tarabelloni, N. (2016). Covariance-based clustering in multivariate and functional data analysis. *Journal of Machine Learning Research*, 17:4985–5005.

Izenman, A. J. and Sommer, C. J. (1988). Philatelic mixtures and multimodal densities. *Journal of the American Statistical Association*, 83:941–953.

Jara, A. and Hanson, T. E. (2011). A class of mixtures of dependent tail-free processes. *Biometrika*, 98:553–566.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186:453–461.

Jiang, H., Fine, J. P., and Chappell, R. (2005). Semiparametric analysis of survival data with left truncation and dependent right censoring. *Biometrics*, 61:567–575.

Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. CRC Press, Florida.

Joe, H. (2005). Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94:401–419.

Joe, H. (2014). *Dependence Modeling with Copulas*. CRC Press, Florida.

Joe, H. and Xu, J. J. (1996). The estimation method of inference functions for margins for multivariate models.

Jondeau, E. and Rockinger, M. (2006). The copula-GARCH model of conditional dependencies: An international stock market application. *Journal of International Money and Finance*, 25:827–853.

Kasa, S. R., Bhattacharya, S., and Rajan, V. (2020). Gaussian mixture copulas for high-dimensional clustering and dependency-based subtyping. *Bioinformatics*, 36:621–628.

Keith, J. M., Kroese, D. P., and Sofronov, G. Y. (2008). Adaptive independence samplers. *Statistics and Computing*, 18:409–420.

Kenward, M., Lesaffre, E., and Molenberghs, G. (1994). An application of maximum likelihood and generalized estimating equations to the analysis of ordinal data from a longitudinal study with cases missing at random. *Biometrics*, 50:945–953.

Killiches, M. and Czado, C. (2018). A D-Vine copula-based model for repeated measurements extending linear mixed models with homogeneous correlation structure. *Biometrics*, 74:997–1005.

Kim, S., Tadesse, M. G., and Vannucci, M. (2006). Variable selection in clustering via Dirichlet process mixture models. *Biometrika*, 93:877–893.

Klami, A. and Kaski, S. (2007). Local dependent components. In *Proceedings of the 24th International Conference on Machine Learning*. Association for Computing Machinery.

Klami, A., Virtanen, S., and Kaski, S. (2012). Bayesian exponential family projections for coupled data sources. *arXiv preprint arXiv:1203.3489*.

Kleinbaum, D. G. and Klein, M. (2002). *Logistic Regression*. Springer, New York.

Kleppe, T. S. (2016). Adaptive step size selection for Hessian-based manifold Langevin samplers. *Scandinavian Journal of Statistics*, 43:788–805.

Kolev, N., Anjos, U. d., and Mendes, B. V. d. M. (2006). Copulas: A review and recent developments. *Stochastic Models*, 22:617–660.

Koru-Sengul, T., Stoffer, D. S., and Day, N. L. (2007). A residuals based transition model for longitudinal analysis with estimation in the presence of missing data. *Statistics in Medicine*, 26:3330–3341.

Kosmidis, I. and Karlis, D. (2016). Model-based clustering using copulas with applications. *Statistics and Computing*, 26:1079–1099.

Kou, S., Zhou, Q., and Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics. *The Annals of Statistics*, 34:1581–1619.

Kraft, C. H. (1964). A class of distribution function processes which have derivatives. *Journal of Applied Probability*, 1:385–388.

Krupskii, P. and Genton, M. G. (2017). Factor copula models for data with spatio-temporal dependence. *Spatial Statistics*, 22:180–195.

Lambert, P. and Vandenhende, F. (2002). A copula-based model for multivariate non-normal longitudinal data: analysis of a dose titration safety study on a new antidepressant. *Statistics in Medicine*, 21:3197–3217.

Lavine, M. (1992). Some aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, 20:1222–1235.

Lavine, M. (1994). More aspects of Polya tree distributions for statistical modelling. *The Annals of Statistics*, 22:1161–1176.

Lawson, A. B. (2013). *Bayesian Disease Mapping: Hierarchical Modeling in Spatial Epidemiology*. CRC Press, Florida.

LeCam, L. (1953). On some asymptotic properties of maximum likelihood estimates and related Bayes estimates. *University of California Public Statistics*, 1:277–330.

Lee, E. W. and Kim, M. Y. (1998). The analysis of correlated panel data using a continuous-time Markov model. *Biometrics*, 54:1638–1644.

Lee, Y., Nelder, J. A., and Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. CRC Press, Florida.

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

Lindley, D. V. and Smith, A. F. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society: Series B*, 34:1–18.

Lindsay, B. G. (1988). Composite likelihood methods. *Contemporary Mathematics*, 80:220–239.

Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*.

Lindsay, B. G., Yi, G. Y., and Sun, J. (2011). Issues and strategies in the selection of composite likelihoods. *Statistica Sinica*, 21:71–105.

Lipsitz, S. R., Kim, K., and Zhao, L. P. (1994). Analysis of repeated categorical data using generalized estimating equations. *Statistics in Medicine*, 13:1149–1163.

Lipsitz, S. R., Laird, N., and Harrington, D. P. (1991). Generalized estimating equations for correlated binary data: Using the odds ratio as a measure of association. *Biometrika*, 78:153–160.

Litière, S., Alonso, A., and Molenberghs, G. (2008). The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in Medicine*, 27:3125–44.

Lo, A. (1984). On a class of Bayesian nonparametric estimates. I: density estimates. *The Annals of statistics*, 12:351–357.

MacEachern, S. N. (1999). Dependent nonparametric processes. In *ASA Proceedings of the Section on Bayesian Statistical Science*, volume 1. American Statistical Association.

Madsen, L. and Fang, Y. (2011). Joint regression analysis for discrete longitudinal data. *Biometrics*, 67:1171–1176.

Mauldin, R. D., Sudderth, W. D., and Williams, S. (1992). Polya trees and random distributions. *The Annals of Statistics*, 20:1203–1221.

McCulloch, C. E. (1997). Maximum likelihood algorithm for generalized linear mixed models. *Journal of the American Statistical Association*, 92:162–170.

McLachlan, G. J. and Peel, D. (2004). *Finite Mixture Models*. John Wiley & Sons, New York.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092.

Miller, M. E., Davis, C. S., and Landis, J. R. (1993). The analysis of longitudinal polytomous data: Generalized estimating equations and connections with weighted least squares. *Biometrics*, 49:1033–1044.

Min, A. and Czado, C. (2010). Bayesian inference for multivariate copulas using pair-copula constructions. *Journal of Financial Econometrics*, 8:511–546.

Molenbergh, G. and Lesaffre, E. (1994). Marginal modeling of correlated ordinal data using a multivariate Plackett distribution. *Journal of the American Statistical Association*, 89:633–644.

Monticino, M. (2001). How to construct a random probability measure. *International Statistical Review*, 69:153–167.

Muenz, L. R. and Rubinstein, L. V. (1985). Markov models for covariate dependence of binary sequences. *Biometrics*, 41:91–101.

Müller, P., Quintana, F. A., Jara, A., and Hanson, T. (2015). *Bayesian Nonparametric Data Analysis*. Springer, New York.

Muthén, B. and Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the EM-algorithm. *Biometrics*, 55:463–469.

Natarajan, R. and Kass, R. E. (2000). Reference Bayesian methods for generalized linear mixed models. *Journal of the American Statistical Association*, 95:227–237.

Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6:353–366.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

Neklyudov, K., Egorov, E., Shvechikov, P., and Vetrov, D. (2018). Metropolis-Hastings view on variational inference and adversarial training. *arXiv preprint arXiv:1810.07151*.

Nelsen, R. B. (2007). *An Introduction to Copulas*. Springer, New York.

Ng, E. S. W., Carpenter, J. R., Goldstein, H., and Rasbash, J. (2006). Estimation in generalised linear mixed models with binary outcomes by simulated maximum likelihood. *Statistical Modeling*, 6:23–42.

Ning, S. and Shephard, N. (2018). A nonparametric Bayesian approach to copula estimation. *Journal of Statistical Computation and Simulation*, 88:1081–1105.

O'Hagan, A. (1978). Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society: Series B*, 40:1–24.

Okhrin, O., Okhrin, Y., and Schmid, W. (2013a). On the structure and estimation of hierarchical Archimedean copulas. *Journal of Econometrics*, 173:189–204.

Okhrin, O., Okhrin, Y., and Schmid, W. (2013b). Properties of hierarchical Archimedean copulas. *Statistics & Risk Modeling with Applications in Finance and Insurance*, 30:21–54.

Patton, A. J. (2012). A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4–18.

Qu, A., Lindsay, B. G., and Li, B. (2000). Improving generalised estimating equations using quadratic inference functions. *Biometrika*, 87:823–836.

Radford, N., Brooks, S., Gelman, A., Jones, G., and Meng, X. (2010). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 31:32.

Rajan, V. and Bhattacharya, S. (2016). Dependency clustering of mixed data with Gaussian mixture copulas. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*.

Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage, California.

Raudenbush, S. W., Yang, M., and Yosef, M. (2000). Maximum likelihood for generalized linear models with nested random effects via high-order multivariate Laplace approximation. *Journal of Computational and Graphical Statistics*, 9:141–157.

Reinsch, C. H. (1967). Smoothing by spline functions. *Numerische Mathematik*, 10:177–183.

Rey, M. and Roth, V. (2012). Copula mixture model for dependency-seeking clustering. *arXiv preprint arXiv:1206.6433*.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B*, 59:731–792.

Romeo, J. S., Tanaka, N. I., and de Lima, A. C. P. (2006). Bivariate survival modeling: a Bayesian approach based on copulas. *Lifetime Data Analysis*, 12:205–222.

Ruppert, D. and Wand, M. P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22:1346–1370.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

Ruscone, M. N. and Osmetti, S. A. (2016). Modelling the dependence in multivariate longitudinal data by pair copula decomposition. In *International Conference on Soft Methods in Probability and Statistics*. Springer, New York.

Schamberger, B., Gruber, L. F., and Czado, C. (2017). Bayesian inference for latent factor copulas and application to financial risk forecasting. *Econometrics*, 5:21.

Schepsmeier, U., Stoeber, J., Brechmann, E. C., Graeler, B., Nagler, T., Erhardt, T., Almeida, C., Min, A., Czado, C., and Hofmann, M. (2018). Package 'vinecopula'. *R package version*, 2.

Schervish, M. J. (1995). *Theory of Statistics*. Springer, New York.

Schumaker, L. (2007). *Spline Functions: Basic Theory*. Cambridge University Press, Cambridge.

Seber, G. A. and Lee, A. J. (2012). *Linear Regression Analysis*. John Wiley & Sons, New Jersey.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650.

Shen, X. and Wasserman, L. (2001). Rates of convergence of posterior distributions. *The Annals of Statistics*, 29:687–714.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. CRC Press, Florida.

Sklar, A. (1959). Functions de repartition an dimension set leursmarges. *Publications de L'In-stitut de Statistique de L'Universite de Paris*, 8:229–231.

Smith, M., Min, A., Almeida, C., and Czado, C. (2010). Modeling longitudinal data using a pair-copula decomposition of serial dependence. *Journal of the American Statistical Association*, 105:1467–1479.

Smith, M. S. (2011). Bayesian approaches to copula modelling. *arXiv preprint arXiv:1112.4204*.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2014). The deviance information criterion: 12 years on. *Journal of the Royal Statistical Society: Series B*, 76:485–493.

Staicu, A.-M., Crainiceanu, C. M., Reich, D. S., and Ruppert, D. (2012). Modeling functional data with spatially heterogeneous shape characteristics. *Biometrics*, 68:331–343.

Stober, J. and Schepsemeier, U. (2013). Estimating standard errors in regular vine copula models. *Computational Statistics*, 28:2679–2707.

Tabachnick, B. G., Fidell, L. S., and Ullman, J. B. (2007). *Using Multivariate Statistics*, volume 5. Pearson Boston, MA.

Tewari, A., Giering, M. J., and Raghunathan, A. (2011). Parametric characterization of multimodal distributions with non-Gaussian modes. In *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1986). *Statistical Analysis of Finite Mixture Distribution*. Wiley, New York.

Trippa, L., Müller, P., and Johnson, W. (2011). The multivariate Beta process and an extension of the Polya tree model. *Biometrika*, 98:17–34.

Van Den Goorbergh, R. W., Genest, C., and Werker, B. J. (2005). Bivariate option pricing using dynamic copula models. *Insurance: Mathematics and Economics*, 37:101–114.

Varin, C. (2008). On composite marginal likelihoods. *AStA Advances in Statistical Analysis*, 92:1–28.

Varin, C., Reid, N., and Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21:5–42.

Verbeke, G., Fieuws, S., Molenberghs, G., and Davidian, M. (2014). The analysis of multivariate longitudinal data: A review. *Statistical Methods in Medical Research*, 23:42–59.

Verbeke, G. and Lefaffre, E. (1996). A linear mixed-effects model with heterogeneity in the random-effects population. *Journal of the American Statistical Association*, 91:217–221.

Verbeke, G. and Lefaffre, E. (1997). The effect of misspecifying the random effects distribution in linear mixed models for longitudinal data. *Computational Statistics & Data Analysis*, 23:541–556.

Verbeke, G. and Molenberghs, G. (2009). *Linear Mixed Models for Longitudinal Data*. Springer, New York.

Vincent, L. A., Wang, X. L., Milewska, E. J., Wan, H., Yang, F., and Swail, V. (2012). A second generation of homogenized canadian monthly surface air temperature for climate trend analysis. *Journal of Geophysical Research: Atmospheres*, 117(D18).

Vlachos, A., Korhonen, A., and Ghahramani, Z. (2009). Unsupervised and constrained Dirichlet process mixture models for verb clustering. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*. Association for Computational Linguistics.

Wahba, G. (1990). *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadephia.

Walker, S. and Mallick, B. K. (1999). A Bayesian semiparametric accelerated failure time model. *Biometrics*, 55:477–483.

Walker, S. G. and Mallick, B. K. (1997). Hierarchical generalized linear models and frailty models with Bayesian nonparametric mixing. *Journal of the Royal Statistical Society: Series B*, 59:845–860.

Wand, M. P. and Jones, M. C. (1994). *Kernel Smoothing*. CRC Press, Florida.

Wang, Y. and Carey, V. (2004). Unbiased estimating equations from working correlation models for irregularly timed repeated measures. *Journal of the American Statistical Association*, 99:845–853.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*.

Wikle, C. K., Berliner, L. M., and Cressie, N. (1998). Hierarchical Bayesian space-time models. *Environmental and Ecological Statistics*, 5:117–154.

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001). Spatiotemporal hierarchical Bayesian modeling tropical ocean surface winds. *Journal of the American Statistical Association*, 96:382–397.

Ye, H. and Pan, J. (2006). Modeling of covariance structures in generalised estimating equations for longitudinal data. *Biometrika*, 93:927–941.

Yi, G. Y. (2017a). Composite likelihood/pseudolikelihood. *Wiley StatsRef: Statistics Reference Online*.

Yi, G. Y. (2017b). *Statistical Analysis with Measurement Error or Misclassification: Strategy, Method and Application*. Springer, New York.

Yi, G. Y., He, W., and Li, H. (2017). A class of flexible models for analysis of complex structured correlated data with application to clustered longitudinal data. *STAT*, 6:448–461.

Yu, G., Huang, R., and Wang, Z. (2010). Document clustering via Dirichlet process mixture model with feature selection. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery.

Zeng, L. and Cook, R. J. (2007). Transition models for multivariate longitudinal binary data. *Journal of the American Statistical Association*, 102:211–223.

Zhang, D. and Davidian, M. (2001). Linear mixed models with flexible distributions of random effects for longitudinal data. *Biometrics*, 57:795–802.

Zhang, R., Li, C., Zhang, J., Chen, C., and Wilson, A. G. (2019). Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*.

APPENDICES

Appendix A

Appendix for Chapter 2

A.1 Additional Simulation Results

A.1.1 Efficiency

Table A.1: Simulation results using the four estimation methods: strong dependence and $n = 1000$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	-0.135	-0.069	-0.156	-0.195	-0.013	-0.001	-0.002	0.002	0.003	-0.007	< 0.001	0.031	-0.041	0.006	-0.069	-0.008	0.753	0.077	0.001	1.080	-0.007	0.573
	ESE[2]	0.047	0.048	0.049	0.051	0.002	0.002	0.002	0.004	0.005	0.004	0.005	0.007	0.022	0.022	0.023	0.024	0.131	0.053	0.003	0.163	0.005	0.132
	ASE[3]	0.046	0.047	0.047	0.049	0.002	0.002	0.002	0.004	0.005	0.004	0.005	0.007	0.022	0.022	0.023	0.024	0.125	0.050	0.003	0.155	0.005	0.129
	ECP[4]	0.950	0.948	0.945	0.965	0.945	0.948	0.955	0.935	0.958	0.940	0.955	0.960	0.948	0.955	0.948	0.945	0.945	0.963	0.950	0.955	0.938	0.948
Method 2: Full likelihood Two-stage Estimation	EBias*	-0.098	0.408	-0.317	0.316	0.018	-0.145	0.077	-0.166	-0.008	-0.020	0.045	0.039	-0.153	-0.112	-0.207	-0.133	-4.422	-1.612	-0.096	-11.068	-0.181	-1.911
	ESE	0.087	0.085	0.082	0.086	0.023	0.021	0.021	0.021	0.009	0.008	0.008	0.008	0.040	0.040	0.042	0.043	0.160	0.077	0.004	0.189	0.006	0.142
	ASE	0.084	0.084	0.084	0.084	0.022	0.022	0.022	0.022	0.009	0.008	0.008	0.008	0.040	0.039	0.042	0.042	0.159	0.075	0.004	0.195	0.006	0.154
	ECP	0.943	0.953	0.950	0.948	0.953	0.938	0.943	0.945	0.943	0.940	0.945	0.943	0.940	0.953	0.938	0.950	0.928	0.940	0.935	0.903	0.940	0.948
	Efficiency	0.298	0.306	0.306	0.342	0.012	0.008	0.012	0.036	0.279	0.264	0.423	0.829	0.299	0.329	0.293	0.313	0.617	0.451	0.478	0.633	0.677	0.701
Method 3: Composite likelihood Simultaneous Estimation	EBias*	-0.010	0.058	-0.137	-0.174	-0.017	-0.033	-0.004	0.002	-0.011	-0.006	0.014	0.042	-0.154	-0.114	-0.186	-0.122	0.734	0.034	-0.010	0.743	-0.022	0.463
	ESE	0.064	0.067	0.067	0.069	0.004	0.003	0.004	0.004	0.011	0.012	0.013	0.014	0.029	0.031	0.033	0.035	0.177	0.078	0.004	0.186	0.006	0.134
	ASE	0.063	0.064	0.064	0.066	0.004	0.003	0.004	0.004	0.012	0.013	0.013	0.014	0.029	0.031	0.032	0.034	0.167	0.074	0.004	0.178	0.006	0.130
	ECP	0.945	0.943	0.948	0.948	0.960	0.955	0.945	0.940	0.940	0.935	0.950	0.945	0.938	0.935	0.943	0.945	0.953	0.955	0.935	0.945	0.953	0.958
	Efficiency	0.535	0.528	0.524	0.553	0.349	0.331	0.456	0.996	0.166	0.117	0.144	0.240	0.549	0.509	0.487	0.488	0.562	0.455	0.499	0.760	0.755	0.977
Method 4: Composite likelihood Two-stage Estimation	EBias*	-0.098	0.408	-0.317	0.316	0.018	-0.145	0.077	-0.166	-0.008	-0.020	0.045	0.039	-0.153	-0.112	-0.207	-0.133	-2.564	-0.898	-0.059	-10.215	-0.229	-2.577
	ESE	0.087	0.085	0.082	0.086	0.023	0.021	0.021	0.021	0.009	0.008	0.008	0.008	0.040	0.040	0.042	0.043	0.184	0.091	0.005	0.201	0.007	0.145
	ASE	0.084	0.084	0.084	0.084	0.022	0.022	0.022	0.022	0.009	0.008	0.008	0.008	0.040	0.039	0.042	0.042	0.177	0.086	0.005	0.204	0.007	0.155
	ECP	0.943	0.953	0.950	0.948	0.953	0.938	0.943	0.945	0.943	0.940	0.945	0.943	0.940	0.953	0.938	0.950	0.945	0.943	0.948	0.915	0.935	0.950
	Efficiency	0.298	0.306	0.306	0.342	0.012	0.008	0.012	0.036	0.279	0.264	0.423	0.829	0.299	0.329	0.293	0.313	0.503	0.343	0.396	0.583	0.599	0.694

[1] EBias*=EBias$\times 10^2$

[2] ESE: Empirical Standard Error

[3] ASE: Asymptotic Standard Error

[4] ECP: Empirical Coverage Probability

Table A.2: Simulation results using the four estimation methods: moderate dependence and $n = 500$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1:	EBias*[1]	0.133	0.151	0.363	-0.550	0.012	0.017	0.081	0.050	0.008	0.028	0.007	0.157	-0.386	-0.288	0.520	-0.276	0.230	-0.372	-0.066	-0.272	-0.023	0.279
Full likelihood	ESE[2]	0.093	0.093	0.102	0.113	0.011	0.011	0.013	0.022	0.024	0.022	0.025	0.032	0.034	0.039	0.040	0.036	0.101	0.071	0.018	0.141	0.017	0.139
Simultaneous	ASE[3]	0.093	0.097	0.099	0.115	0.011	0.012	0.012	0.021	0.024	0.023	0.025	0.032	0.036	0.041	0.041	0.036	0.099	0.069	0.016	0.146	0.018	0.140
Estimation	ECP[4]	0.955	0.948	0.953	0.955	0.953	0.945	0.955	0.958	0.948	0.943	0.943	0.945	0.935	0.933	0.953	0.943	0.933	0.943	0.963	0.945	0.950	0.950
Method 2:	EBias*	0.273	0.307	-0.618	-0.291	-0.125	0.033	0.205	-0.021	-0.017	-0.128	0.047	0.147	-0.524	-0.510	-0.720	-0.381	-1.032	-1.162	-0.181	-1.724	-0.097	-0.023
Full likelihood	ESE	0.126	0.127	0.133	0.132	0.031	0.030	0.032	0.033	0.029	0.027	0.029	0.033	0.040	0.045	0.048	0.039	0.106	0.077	0.019	0.142	0.017	0.139
Two-stage	ASE	0.127	0.130	0.131	0.131	0.031	0.031	0.031	0.031	0.029	0.028	0.029	0.033	0.042	0.046	0.047	0.038	0.104	0.074	0.017	0.148	0.018	0.141
Estimation	ECP	0.958	0.958	0.943	0.953	0.950	0.933	0.958	0.943	0.948	0.945	0.940	0.955	0.945	0.935	0.945	0.953	0.938	0.965	0.960	0.943	0.953	0.953
	Efficiency	0.529	0.556	0.573	0.762	0.135	0.146	0.142	0.471	0.661	0.690	0.753	0.938	0.745	0.769	0.756	0.882	0.911	0.881	0.905	0.966	0.977	0.990
Method 3:	EBias*	-0.241	0.246	0.323	-0.576	0.086	-0.048	0.037	0.051	-0.007	-0.025	0.054	0.180	-0.470	-0.452	-0.633	-0.352	0.230	-0.448	-0.096	-0.517	-0.053	0.246
Composite	ESE	0.098	0.100	0.109	0.116	0.013	0.015	0.015	0.022	0.025	0.023	0.026	0.032	0.037	0.043	0.044	0.038	0.111	0.078	0.019	0.145	0.017	0.139
likelihood	ASE	0.100	0.105	0.106	0.117	0.013	0.015	0.014	0.022	0.025	0.025	0.027	0.033	0.039	0.044	0.044	0.037	0.107	0.073	0.017	0.151	0.018	0.141
Simultaneous	ECP	0.940	0.948	0.943	0.960	0.960	0.953	0.940	0.958	0.950	0.938	0.945	0.943	0.950	0.935	0.948	0.940	0.938	0.960	0.963	0.948	0.958	0.950
Estimation	Efficiency	0.861	0.847	0.875	0.952	0.718	0.621	0.700	0.960	0.858	0.854	0.880	0.950	0.878	0.865	0.882	0.942	0.849	0.894	0.927	0.932	0.987	0.996
Method 4:	EBias*	0.273	0.307	-0.618	-0.291	-0.125	0.033	0.205	-0.021	-0.017	-0.128	0.047	0.147	-0.524	-0.510	-0.720	-0.381	-0.741	-1.073	-0.165	-1.885	-0.120	-0.042
Composite	ESE	0.126	0.127	0.133	0.132	0.031	0.030	0.032	0.033	0.029	0.027	0.029	0.033	0.040	0.045	0.048	0.039	0.111	0.079	0.019	0.144	0.017	0.139
likelihood	ASE	0.127	0.130	0.131	0.131	0.031	0.031	0.031	0.031	0.029	0.028	0.029	0.033	0.042	0.046	0.047	0.038	0.108	0.075	0.017	0.151	0.018	0.141
Two-stage	ECP	0.958	0.958	0.943	0.953	0.950	0.933	0.958	0.943	0.948	0.945	0.940	0.955	0.945	0.935	0.945	0.953	0.938	0.965	0.963	0.950	0.953	0.953
Estimation	Efficiency	0.529	0.556	0.573	0.762	0.135	0.146	0.142	0.471	0.661	0.690	0.753	0.938	0.745	0.769	0.756	0.882	0.846	0.853	0.891	0.934	0.980	0.990

[1] EBias*=EBias$\times 10^2$

[2] ESE: Empirical Standard Error

[3] ASE: Asymptotic Standard Error

[4] ECP: Empirical Coverage Probability

Table A.3: Simulation results using the four estimation methods: moderate dependence and $n = 1000$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	-0.185	-0.001	-0.309	-0.253	-0.078	-0.015	-0.002	-0.011	0.066	-0.023	0.030	0.108	-0.005	0.125	-0.077	0.046	0.691	0.044	0.025	0.106	-0.012	0.449
	ESE[2]	0.064	0.070	0.071	0.083	0.008	0.008	0.008	0.016	0.016	0.016	0.018	0.023	0.026	0.029	0.030	0.027	0.074	0.052	0.013	0.107	0.013	0.105
	ASE[3]	0.066	0.069	0.070	0.081	0.008	0.008	0.008	0.015	0.017	0.017	0.018	0.023	0.026	0.029	0.029	0.026	0.070	0.049	0.012	0.103	0.012	0.099
	ECP[4]	0.963	0.940	0.948	0.953	0.950	0.953	0.950	0.940	0.945	0.940	0.953	0.968	0.945	0.955	0.955	0.948	0.945	0.953	0.955	0.945	0.958	0.950
Method 2: Full likelihood Two-stage Estimation	EBias*	-0.666	0.295	-0.742	0.081	0.072	-0.085	0.140	-0.159	0.105	-0.068	0.098	0.127	-0.072	0.078	-0.175	0.023	0.116	-0.306	0.019	-0.601	-0.039	0.328
	ESE	0.085	0.091	0.089	0.095	0.022	0.021	0.021	0.022	0.020	0.020	0.020	0.024	0.030	0.033	0.035	0.029	0.078	0.055	0.013	0.106	0.013	0.105
	ASE	0.090	0.092	0.092	0.093	0.022	0.022	0.022	0.022	0.020	0.020	0.020	0.024	0.030	0.033	0.034	0.027	0.074	0.053	0.012	0.104	0.013	0.099
	ECP	0.955	0.960	0.948	0.958	0.958	0.948	0.955	0.963	0.955	0.950	0.950	0.948	0.948	0.958	0.965	0.945	0.945	0.943	0.955	0.950	0.960	0.948
	Efficiency	0.528	0.553	0.573	0.759	0.134	0.144	0.140	0.470	0.662	0.690	0.753	0.937	0.739	0.760	0.750	0.873	0.898	0.871	0.896	0.973	0.973	0.994
Method 3: Composite likelihood Simultaneous Estimation	EBias*	-0.070	0.241	-0.244	-0.217	-0.071	-0.082	-0.019	-0.011	0.024	-0.057	0.028	0.094	-0.045	0.057	-0.114	0.021	0.693	0.015	0.015	-0.031	-0.023	0.455
	ESE	0.070	0.076	0.077	0.086	0.009	0.010	0.010	0.016	0.017	0.017	0.019	0.024	0.028	0.031	0.033	0.028	0.081	0.056	0.013	0.111	0.013	0.105
	ASE	0.071	0.075	0.075	0.083	0.009	0.011	0.010	0.015	0.018	0.018	0.019	0.023	0.027	0.031	0.031	0.026	0.076	0.052	0.012	0.107	0.013	0.099
	ECP	0.958	0.950	0.953	0.958	0.953	0.948	0.948	0.940	0.950	0.953	0.960	0.968	0.958	0.953	0.955	0.945	0.940	0.953	0.950	0.935	0.958	0.950
	Efficiency	0.864	0.847	0.876	0.949	0.727	0.615	0.699	0.989	0.861	0.854	0.882	0.951	0.873	0.857	0.877	0.934	0.844	0.888	0.918	0.934	0.981	0.996
Method 4: Composite likelihood Two-stage Estimation	EBias*	-0.666	0.295	-0.742	0.081	0.072	-0.085	0.140	-0.159	0.105	-0.068	0.098	0.127	-0.072	0.078	-0.175	0.023	0.244	-0.274	-0.012	-0.696	-0.049	0.321
	ESE	0.085	0.091	0.089	0.095	0.022	0.021	0.021	0.022	0.020	0.020	0.020	0.024	0.030	0.033	0.035	0.029	0.082	0.057	0.013	0.110	0.013	0.105
	ASE	0.090	0.092	0.092	0.093	0.022	0.022	0.022	0.022	0.020	0.020	0.020	0.024	0.030	0.033	0.034	0.027	0.077	0.053	0.012	0.106	0.013	0.099
	ECP	0.955	0.960	0.948	0.958	0.958	0.948	0.955	0.963	0.955	0.950	0.950	0.948	0.948	0.958	0.965	0.945	0.945	0.950	0.955	0.933	0.958	0.948
	Efficiency	0.528	0.553	0.573	0.759	0.134	0.144	0.140	0.470	0.662	0.690	0.753	0.937	0.739	0.760	0.750	0.873	0.835	0.842	0.882	0.939	0.976	0.993

[1] EBias*=EBias$\times 10^2$

[2] ESE: Empirical Standard Error

[3] ASE: Asymptotic Standard Error

[4] ECP: Empirical Coverage Probability

A.1.2 Robustness

Table A.4: Simulation results using the four estimation methods when block-connecting structure is mis-specified: strong dependence and $n = 500$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	4.035	7.165	4.057	4.476	-0.002	-0.020	0.015	0.005	-0.070	-0.034	-0.047	-0.005	-9.798	-7.674	-9.916	-9.548	-59.683	-15.620	-1.371	-27.232	0.025	5.794
	ESE[2]	0.090	0.090	0.094	0.094	0.005	0.004	0.004	0.006	0.010	0.010	0.011	0.013	0.041	0.047	0.047	0.051	0.249	0.115	0.008	0.244	0.008	0.184
	ASE[3]	0.069	0.073	0.072	0.076	0.005	0.004	0.004	0.006	0.011	0.012	0.012	0.014	0.034	0.035	0.036	0.038	0.180	0.083	0.006	0.220	0.008	0.185
	ECP[4]	0.918	0.878	0.933	0.918	0.939	0.933	0.948	0.956	0.939	0.945	0.953	0.939	0.324	0.627	0.420	0.516	0.312	0.723	0.601	0.810	0.950	0.927
Method 2: Full likelihood Two-stage Estimation	EBias*	0.884	0.378	-0.016	0.375	-0.200	0.004	0.105	-0.099	-0.106	-0.059	0.005	0.029	-0.685	-0.643	-0.954	-0.804	-42.505	-11.992	-0.885	-27.852	-0.531	-0.137
	ESE	0.120	0.118	0.122	0.124	0.031	0.029	0.032	0.034	0.015	0.014	0.012	0.013	0.048	0.051	0.056	0.057	0.237	0.121	0.008	0.284	0.009	0.222
	ASE	0.119	0.120	0.120	0.120	0.031	0.031	0.031	0.031	0.015	0.014	0.013	0.013	0.049	0.052	0.057	0.058	0.350	0.186	0.011	0.392	0.015	0.269
	ECP	0.953	0.953	0.953	0.953	0.950	0.935	0.963	0.963	0.945	0.958	0.950	0.960	0.953	0.948	0.960	0.953	0.564	0.820	0.781	0.854	0.906	0.961
Method 3: Composite likelihood Simultaneous Estimation	EBias*	0.354	0.374	0.250	0.097	0.019	-0.027	-0.006	0.012	-0.092	-0.040	-0.035	0.014	-0.580	-0.617	-0.729	-0.662	-0.515	-0.601	-0.050	1.335	-0.028	0.561
	ESE	0.088	0.090	0.091	0.093	0.006	0.005	0.005	0.006	0.017	0.018	0.019	0.020	0.039	0.043	0.045	0.047	0.224	0.107	0.006	0.216	0.007	0.184
	ASE	0.085	0.088	0.091	0.091	0.006	0.005	0.005	0.006	0.016	0.017	0.018	0.020	0.037	0.041	0.043	0.046	0.224	0.101	0.006	0.231	0.008	0.184
	ECP	0.950	0.958	0.953	0.960	0.945	0.960	0.955	0.955	0.953	0.960	0.958	0.958	0.955	0.955	0.948	0.950	0.943	0.955	0.945	0.945	0.945	0.958
Method 4: Composite likelihood Two-stage Estimation	EBias*	0.884	0.378	-0.016	0.375	-0.200	0.004	0.105	-0.099	-0.106	-0.059	0.005	0.029	-0.685	-0.643	-0.954	-0.804	-6.902	-2.448	-0.156	-18.732	-0.418	-7.898
	ESE	0.120	0.118	0.122	0.124	0.031	0.029	0.032	0.034	0.015	0.014	0.012	0.013	0.048	0.051	0.056	0.057	0.239	0.123	0.007	0.251	0.009	0.223
	ASE	0.119	0.120	0.120	0.120	0.031	0.031	0.031	0.031	0.015	0.014	0.013	0.013	0.049	0.052	0.057	0.058	0.240	0.117	0.006	0.288	0.009	0.245
	ECP	0.953	0.953	0.953	0.953	0.950	0.935	0.963	0.963	0.945	0.958	0.950	0.960	0.953	0.948	0.960	0.953	0.938	0.945	0.943	0.893	0.913	0.940

[1] EBias*=EBias$\times 10^2$
[2] ESE: Empirical Standard Error
[3] ASE: Asymptotic Standard Error
[4] ECP: Empirical Coverage Probability

Table A.5: Simulation results using the four estimation methods when block-connecting structure is mis-specified: strong dependence and $n = 1000$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	3.976	7.162	4.118	4.606	-0.005	0.003	-0.005	-0.010	-0.007	-0.011	-0.029	< 0.001	-9.440	-7.182	-9.389	-9.014	-58.638	-14.465	-1.290	-27.151	0.018	5.767
	ESE[2]	0.067	0.069	0.069	0.071	0.003	0.003	0.003	0.004	0.007	0.008	0.008	0.010	0.031	0.036	0.035	0.038	0.187	0.081	0.005	0.188	0.006	0.134
	ASE[3]	0.047	0.049	0.049	0.052	0.003	0.003	0.003	0.004	0.008	0.008	0.008	0.009	0.023	0.024	0.024	0.026	0.121	0.055	0.004	0.155	0.005	0.130
	ECP[4]	0.906	0.828	0.900	0.896	0.955	0.948	0.958	0.942	0.955	0.955	0.955	0.955	0.136	0.447	0.194	0.294	0.123	0.592	0.337	0.712	0.945	0.932
Method 2: Full likelihood Two-stage Estimation	EBias*	-0.180	0.493	-0.280	0.304	0.051	-0.141	0.092	-0.140	0.005	-0.037	0.031	0.036	-0.152	-0.056	-0.246	-0.150	-37.137	-9.330	-0.720	-18.848	0.667	4.154
	ESE	0.087	0.085	0.082	0.087	0.023	0.021	0.021	0.021	0.011	0.010	0.009	0.009	0.036	0.038	0.042	0.043	0.174	0.088	0.005	0.210	0.007	0.146
	ASE	0.085	0.085	0.085	0.085	0.022	0.022	0.022	0.022	0.011	0.010	0.009	0.009	0.035	0.037	0.040	0.041	0.209	0.108	0.006	0.232	0.008	0.159
	ECP	0.945	0.955	0.950	0.943	0.958	0.948	0.948	0.950	0.958	0.955	0.940	0.948	0.955	0.940	0.950	0.953	0.425	0.831	0.731	0.855	0.833	0.952
Method 3: Composite likelihood Simultaneous Estimation	EBias*	0.171	0.196	0.093	0.024	-0.003	-0.027	-0.011	0.002	-0.081	-0.060	-0.058	-0.026	-0.133	-0.093	-0.157	-0.090	0.437	-0.050	-0.012	1.020	-0.007	0.607
	ESE	0.065	0.069	0.069	0.071	0.004	0.003	0.004	0.004	0.011	0.012	0.013	0.014	0.028	0.030	0.032	0.035	0.168	0.074	0.004	0.172	0.006	0.135
	ASE	0.062	0.064	0.064	0.066	0.004	0.003	0.004	0.004	0.012	0.012	0.013	0.014	0.028	0.030	0.032	0.034	0.159	0.071	0.004	0.163	0.005	0.130
	ECP	0.953	0.965	0.960	0.960	0.945	0.950	0.953	0.943	0.950	0.960	0.948	0.953	0.935	0.940	0.955	0.948	0.943	0.945	0.945	0.950	0.955	0.958
Method 4: Composite likelihood Two-stage Estimation	EBias*	-0.180	0.493	-0.280	0.304	0.051	-0.141	0.092	-0.140	0.005	-0.037	0.031	0.036	-0.152	-0.056	-0.246	-0.150	-2.782	-0.952	-0.062	-9.450	-0.200	-2.754
	ESE	0.087	0.085	0.082	0.087	0.023	0.021	0.021	0.021	0.011	0.010	0.009	0.009	0.036	0.038	0.042	0.043	0.176	0.087	0.005	0.191	0.006	0.146
	ASE	0.085	0.085	0.085	0.085	0.022	0.022	0.022	0.022	0.011	0.010	0.009	0.009	0.035	0.037	0.040	0.041	0.170	0.084	0.005	0.190	0.006	0.154
	ECP	0.945	0.955	0.950	0.943	0.958	0.948	0.948	0.950	0.958	0.955	0.940	0.948	0.955	0.940	0.950	0.953	0.948	0.945	0.940	0.930	0.938	0.953

[1] EBias*=EBias$\times 10^2$

[2] ESE: Empirical Standard Error

[3] ASE: Asymptotic Standard Error

[4] ECP: Empirical Coverage Probability

Table A.6: Simulation results using the four estimation methods when block-connecting structure is mis-specified: moderate dependence and $n = 500$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	-1.573	0.412	-1.902	-1.362	0.016	-0.016	0.082	0.044	-0.006	-0.004	-0.024	0.161	-2.284	-0.022	-2.062	-0.834	-3.260	-1.688	-0.382	-4.933	0.004	0.116
	ESE[2]	0.097	0.096	0.107	0.115	0.012	0.013	0.014	0.022	0.026	0.025	0.028	0.033	0.030	0.038	0.037	0.036	0.101	0.067	0.017	0.144	0.017	0.139
	ASE[3]	0.096	0.103	0.103	0.117	0.012	0.013	0.012	0.021	0.027	0.028	0.028	0.034	0.033	0.038	0.038	0.035	0.095	0.065	0.016	0.147	0.018	0.140
	ECP[4]	0.947	0.947	0.947	0.950	0.955	0.942	0.945	0.957	0.947	0.945	0.945	0.942	0.870	0.950	0.915	0.937	0.932	0.940	0.950	0.920	0.967	0.952
Method 2: Full likelihood Two-stage Estimation	EBias*	0.163	0.264	-0.853	-0.428	-0.136	0.041	0.273	0.014	0.005	-0.157	0.057	0.149	-0.396	-0.391	-0.597	-0.384	-3.294	-0.548	-0.211	-5.544	-0.090	0.065
	ESE	0.130	0.130	0.134	0.134	0.031	0.030	0.032	0.033	0.033	0.031	0.032	0.035	0.034	0.040	0.043	0.037	0.101	0.071	0.018	0.143	0.017	0.140
	ASE	0.131	0.134	0.134	0.133	0.031	0.031	0.031	0.031	0.033	0.032	0.033	0.035	0.036	0.041	0.042	0.036	0.099	0.068	0.016	0.149	0.018	0.141
	ECP	0.950	0.958	0.948	0.950	0.958	0.933	0.953	0.940	0.943	0.945	0.950	0.953	0.945	0.943	0.945	0.948	0.930	0.950	0.960	0.910	0.968	0.955
Method 3: Composite likelihood Simultaneous Estimation	EBias*	-0.257	0.144	-0.326	-0.602	0.072	-0.040	0.051	0.051	-0.020	-0.020	0.027	0.175	-0.370	-0.320	-0.534	-0.360	0.312	-0.591	-0.083	-0.635	-0.069	0.289
	ESE	0.098	0.100	0.111	0.117	0.013	0.015	0.016	0.022	0.027	0.026	0.029	0.034	0.032	0.038	0.039	0.036	0.104	0.070	0.018	0.142	0.017	0.140
	ASE	0.101	0.107	0.107	0.118	0.013	0.015	0.014	0.021	0.027	0.028	0.029	0.034	0.033	0.038	0.039	0.035	0.101	0.067	0.016	0.149	0.018	0.140
	ECP	0.945	0.943	0.958	0.950	0.955	0.950	0.940	0.958	0.950	0.933	0.950	0.948	0.945	0.943	0.943	0.950	0.938	0.958	0.955	0.948	0.943	0.953
Method 4: Composite likelihood Two-stage Estimation	EBias*	0.163	0.264	-0.853	-0.428	-0.136	0.041	0.273	0.014	0.005	-0.157	0.057	0.149	-0.396	-0.391	-0.597	-0.384	-0.650	-1.197	-0.150	-2.028	-0.137	0.000
	ESE	0.130	0.130	0.134	0.134	0.031	0.030	0.032	0.033	0.033	0.031	0.032	0.035	0.034	0.040	0.043	0.037	0.104	0.072	0.018	0.141	0.017	0.140
	ASE	0.131	0.134	0.134	0.133	0.031	0.031	0.031	0.031	0.033	0.032	0.033	0.035	0.036	0.041	0.042	0.036	0.101	0.069	0.016	0.149	0.018	0.141
	ECP	0.950	0.958	0.948	0.950	0.958	0.933	0.953	0.940	0.943	0.945	0.950	0.953	0.945	0.943	0.945	0.948	0.940	0.948	0.953	0.940	0.945	0.955

[1] EBias*=EBias$\times 10^2$

[2] ESE: Empirical Standard Error

[3] ASE: Asymptotic Standard Error

[4] ECP: Empirical Coverage Probability

Table A.7: Simulation results using the four estimation methods when block-connecting structure is mis-specified: moderate dependence and $n = 1000$

Methods	Metrics	Marginal Parameters																Dependence Parameters					
		β_{01}	β_{02}	β_{03}	β_{04}	β_{11}	β_{12}	β_{13}	β_{14}	β_{21}	β_{22}	β_{23}	β_{24}	σ_1	σ_2	σ_3	σ_4	$\theta_{k1,k2}$	$\theta_{k1,k3}$	$\theta_{k1,k4}$	$\theta_{k2,k3}$	$\theta_{k2,k4}$	$\theta_{k3,k4}$
Method 1: Full likelihood Simultaneous Estimation	EBias*[1]	-1.607	0.441	-1.853	-1.010	-0.076	-0.019	0.002	-0.015	0.062	-0.018	0.017	0.109	-2.074	0.279	-1.793	-0.548	-3.042	-1.333	-0.318	-0.4.337	0.023	0.288
	ESE[2]	0.067	0.073	0.073	0.086	0.008	0.009	0.008	0.016	0.018	0.018	0.020	0.024	0.023	0.027	0.027	0.026	0.073	0.047	0.012	0.107	0.013	0.105
	ASE[3]	0.068	0.073	0.073	0.083	0.009	0.009	0.009	0.015	0.019	0.020	0.020	0.024	0.024	0.027	0.027	0.025	0.067	0.046	0.011	0.104	0.013	0.099
	ECP[4]	0.947	0.942	0.950	0.950	0.947	0.955	0.957	0.940	0.955	0.950	0.945	0.965	0.857	0.950	0.897	0.942	0.925	0.945	0.942	0.910	0.955	0.947
Method 2: Full likelihood Two-stage Estimation	EBias*	-0.788	0.291	-0.835	0.023	0.082	-0.065	0.156	-0.145	0.133	-0.081	0.120	0.137	-0.079	0.117	-0.201	-0.006	-2.347	0.316	-0.069	-4.106	-0.017	0.344
	ESE	0.088	0.092	0.090	0.097	0.022	0.021	0.021	0.022	0.023	0.022	0.023	0.025	0.026	0.029	0.031	0.027	0.075	0.050	0.012	0.106	0.013	0.105
	ASE	0.093	0.095	0.095	0.095	0.022	0.022	0.022	0.022	0.023	0.023	0.023	0.025	0.026	0.029	0.030	0.026	0.071	0.049	0.012	0.105	0.013	0.099
	ECP	0.948	0.960	0.945	0.955	0.963	0.940	0.958	0.960	0.950	0.948	0.945	0.948	0.963	0.950	0.968	0.955	0.950	0.953	0.960	0.935	0.955	0.948
Method 3: Composite likelihood Simultaneous Estimation	EBias*	-0.059	0.235	-0.164	-0.184	-0.074	-0.066	-0.022	-0.010	0.014	-0.060	0.009	0.087	-0.055	0.069	-0.114	-0.006	0.220	-0.120	0.013	0.058	-0.034	0.467
	ESE	0.069	0.075	0.076	0.087	0.009	0.010	0.010	0.016	0.018	0.019	0.020	0.025	0.024	0.027	0.029	0.027	0.077	0.050	0.012	0.108	0.013	0.105
	ASE	0.071	0.076	0.076	0.084	0.009	0.011	0.010	0.015	0.019	0.020	0.021	0.024	0.023	0.027	0.027	0.025	0.072	0.048	0.011	0.105	0.012	0.099
	ECP	0.958	0.953	0.940	0.958	0.958	0.953	0.945	0.940	0.960	0.955	0.950	0.955	0.955	0.955	0.958	0.950	0.950	0.953	0.960	0.935	0.955	0.948
Method 4: Composite likelihood Two-stage Estimation	EBias*	-0.788	0.291	-0.835	0.023	0.082	-0.065	0.156	-0.145	0.133	-0.081	0.120	0.137	-0.079	0.117	-0.201	-0.006	0.217	-0.415	-0.015	-0.591	-0.057	0.303
	ESE	0.088	0.092	0.090	0.097	0.022	0.021	0.021	0.022	0.023	0.022	0.023	0.025	0.026	0.029	0.031	0.027	0.076	0.051	0.012	0.108	0.013	0.106
	ASE	0.093	0.095	0.095	0.095	0.022	0.022	0.022	0.022	0.023	0.023	0.023	0.025	0.026	0.029	0.030	0.026	0.072	0.049	0.012	0.105	0.013	0.099
	ECP	0.948	0.960	0.945	0.955	0.963	0.940	0.958	0.960	0.950	0.948	0.945	0.948	0.963	0.950	0.968	0.955	0.935	0.955	0.958	0.943	0.955	0.945

[1] EBias*=EBias×10^2
[2] ESE: Empirical Standard Error
[3] ASE: Asymptotic Standard Error
[4] ECP: Empirical Coverage Probability

A.1.3 Prediction

Simulation Results for Prediction

Table A.8: Simulation results for subject extrapolation and time extrapolation in terms of percentage outperformance VINE4 versus the other models

	Subject Extrapolation			Time Extrapolation		
	MRM	LRM	AR	MRM	LRM	AR
Scenario 1(S)	0.618	0.814	0.909	0.868	0.847	0.938
Scenario 1(M)	0.558	0.725	0.874	0.761	0.780	0.937
Scenario 2(S)	0.618	0.744	0.745	0.868	0.776	0.836
Scenario 2(M)	0.558	0.637	0.665	0.761	0.636	0.745
Scenario 3(S)	0.611	0.807	0.945	0.868	0.840	0.919
Scenario 3(M)	0.552	0.719	0.904	0.748	0.764	0.938
Scenario 4(S)	0.611	0.725	0.725	0.868	0.707	0.710
Scenario 4(M)	0.534	0.665	0.665	0.748	0.618	0.620
Scenario 5	0.545	0.547	0.535	0.692	0.693	0.533
Scenario 6	0.545	0.536	0.534	0.692	0.550	0.534

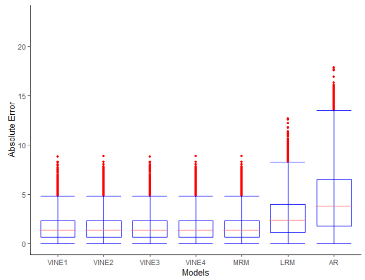
S: strong dependence setting; M: moderate dependence setting

MAEs by Time Points for Subject Extrapolation

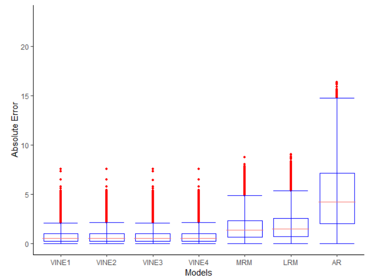
MAE by time points for subject extrapolation for the lth time point is computed by

$$\frac{1}{200 \cdot 50 \cdot 5} \sum_{r=1}^{200} \sum_{i=451}^{500} \sum_{k=1}^{5} |\hat{y}_{ikl}^{(r)} - y_{ikl}^{(r)}|.$$

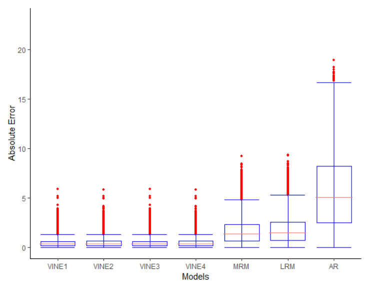
(1) Scenario 1(S)



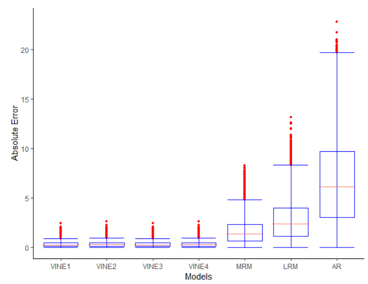
(a) $l = 1$



(b) $l = 2$

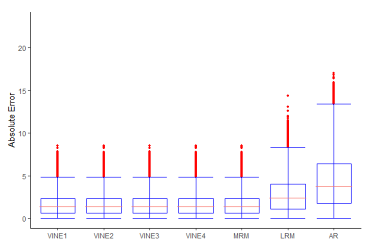


(c) $l = 3$

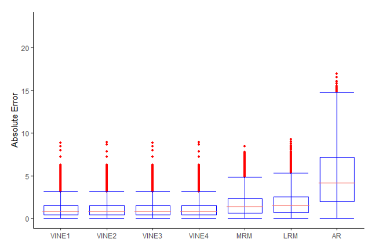


(d) $l = 4$

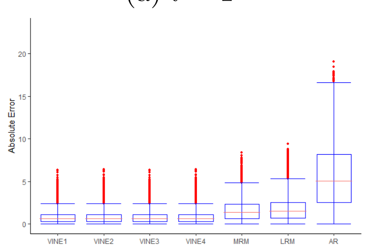
(2) Scenario 1(M)



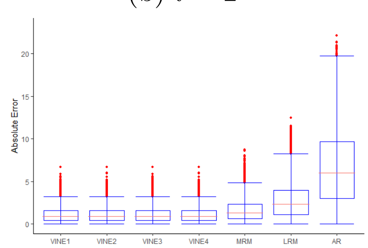
(a) $l = 1$



(b) $l = 2$

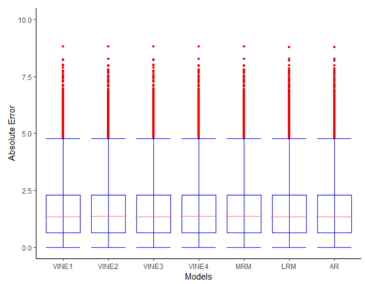


(c) $l = 3$

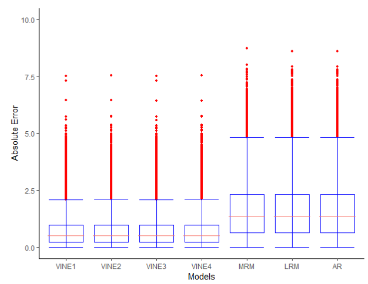


(d) $l = 4$

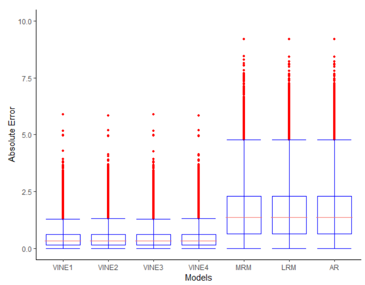
(3) Scenario 2(S)



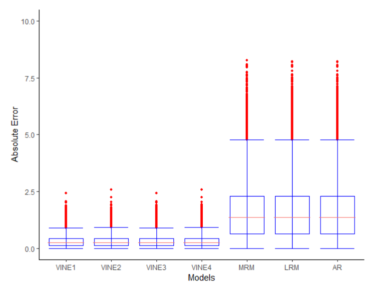
(a) $l = 1$



(b) $l = 2$

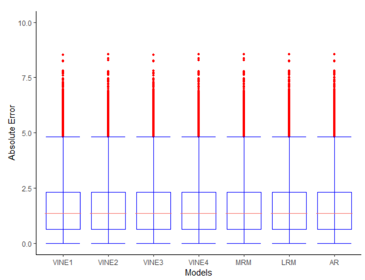


(c) $l = 3$

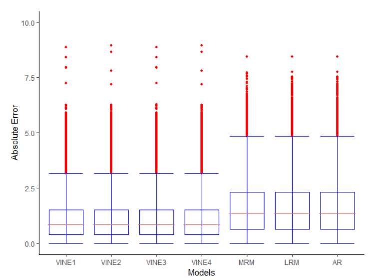


(d) $l = 4$

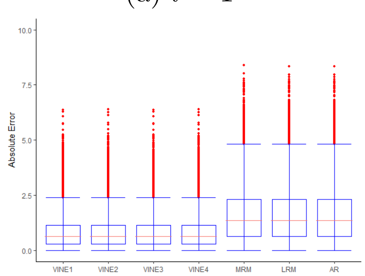
(4) Scenario 2(M)



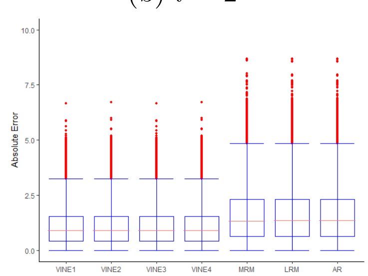
(a) $l = 1$



(b) $l = 2$

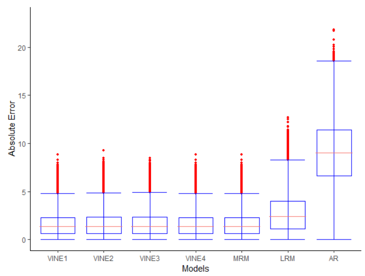


(c) $l = 3$

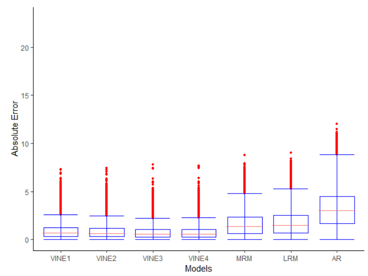


(d) $l = 4$

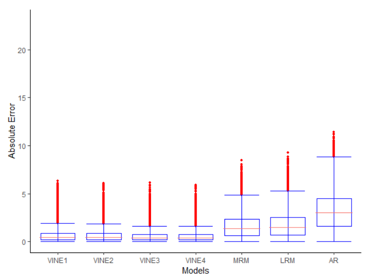
(7) Scenario 3(S)



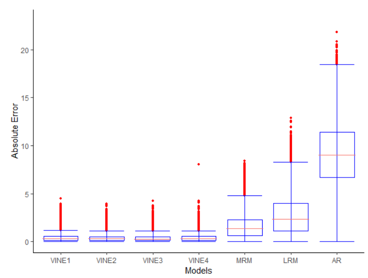
(a) $l = 1$



(b) $l = 2$

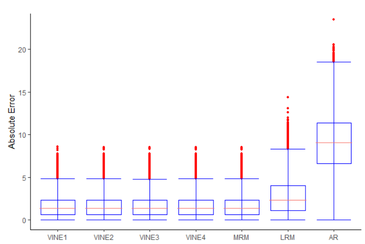


(c) $l = 3$

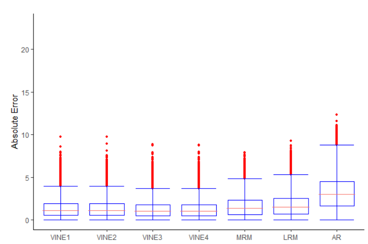


(d) $l = 4$

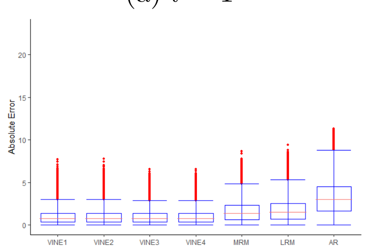
(8) Scenario 3(M)



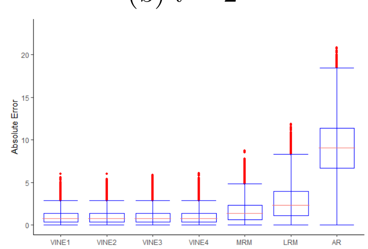
(a) $l = 1$



(b) $l = 2$

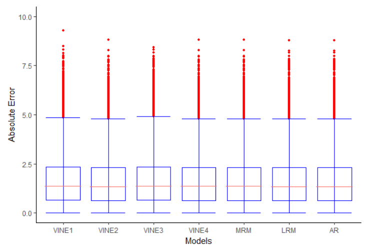


(c) $l = 3$

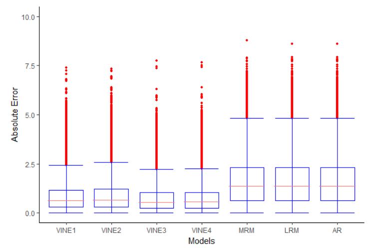


(d) $l = 4$

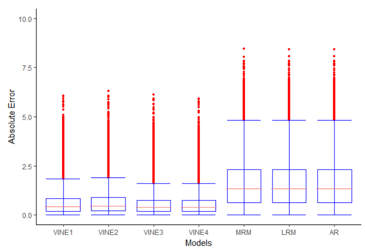
(9) Scenario 4(S)



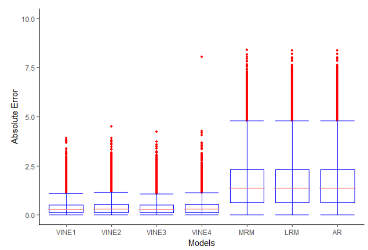
(a) $l = 1$



(b) $l = 2$

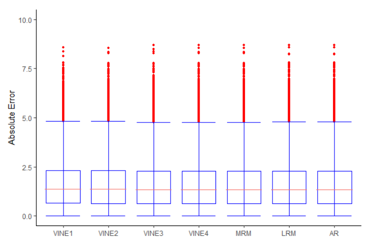


(c) $l = 3$

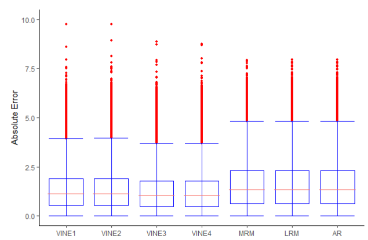


(d) $l = 4$

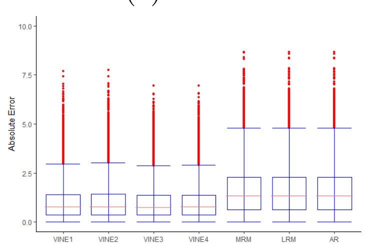
(10) Scenario 4(M)



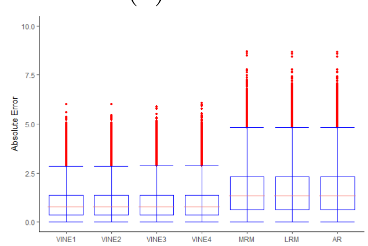
(a) $l = 1$



(b) $l = 2$

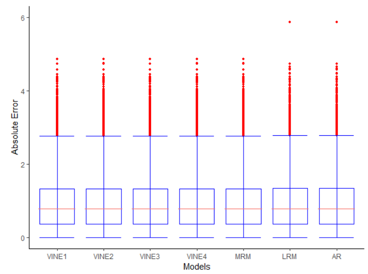


(c) $l = 3$

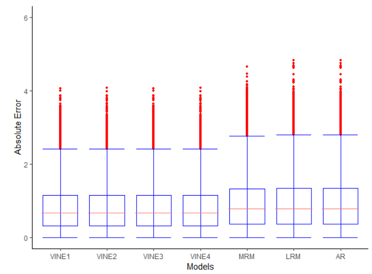


(d) $l = 4$

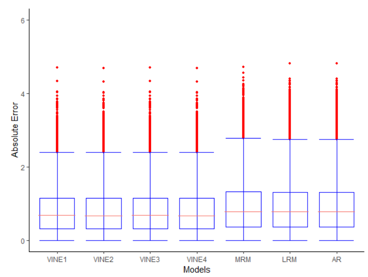
(9) Scenario 5



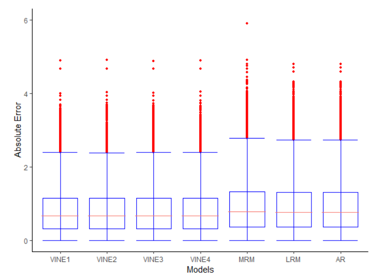
(a) $l = 1$



(b) $l = 2$

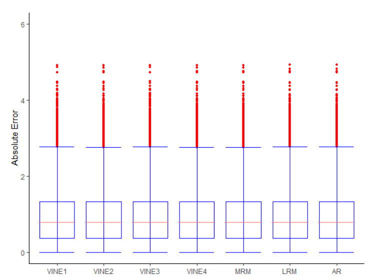


(c) $l = 3$

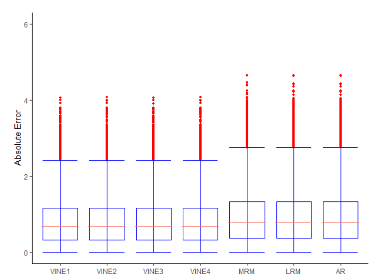


(d) $l = 4$

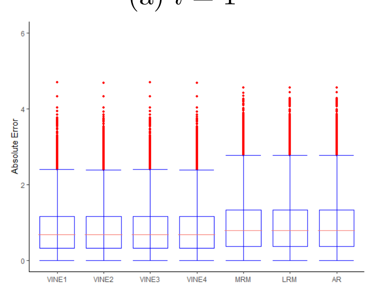
(10) Scenario 6



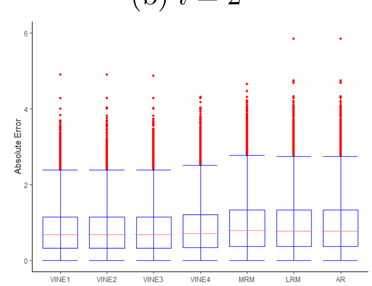
(a) $l = 1$



(b) $l = 2$



(c) $l = 3$



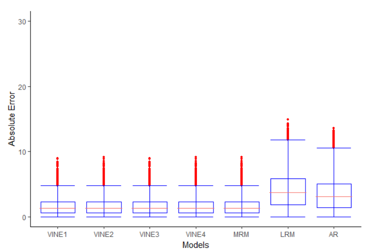
(d) $l = 4$

MAEs by Time Points for Time Extrapolation

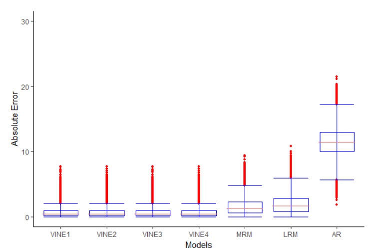
MAE by time points for time extrapolation for the lth time point is computed by

$$\frac{1}{200 \cdot 500} \sum_{r=1}^{200} \sum_{i=1}^{500} |\hat{y}_{i5l}^{(r)} - y_{i5l}^{(r)}|.$$

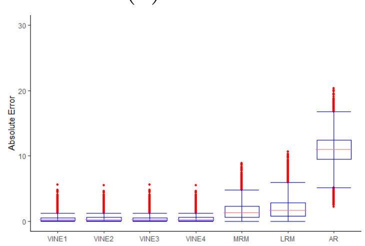
(1) Scenario 1(S)



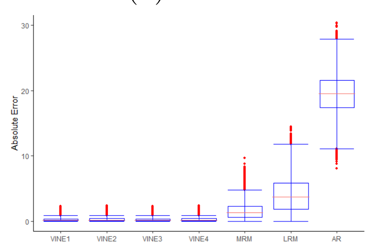
(a) $l = 1$



(b) $l = 2$

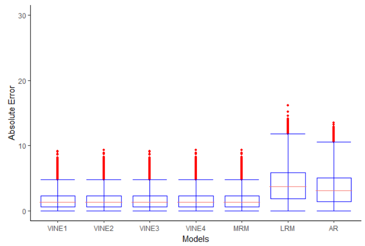


(c) $l = 3$

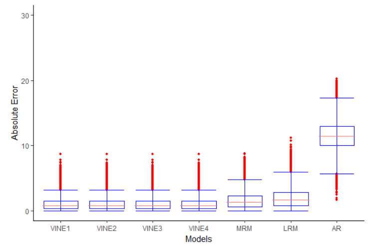


(d) $l = 4$

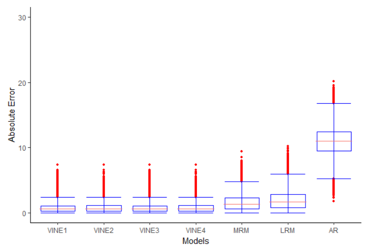
(2) Scenario 1(M)



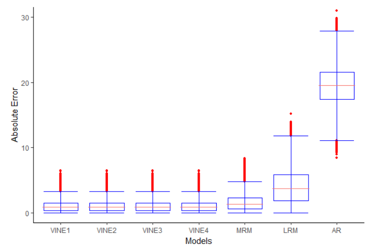
(a) $l = 1$



(b) $l = 2$

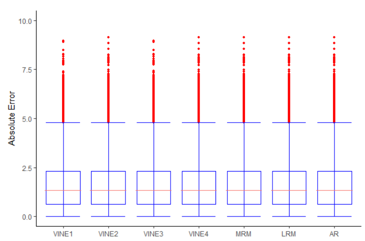


(c) $l = 3$

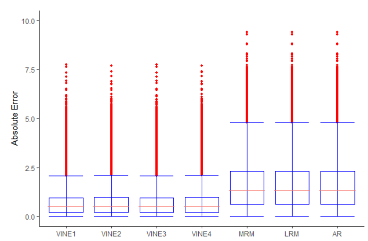


(d) $l = 4$

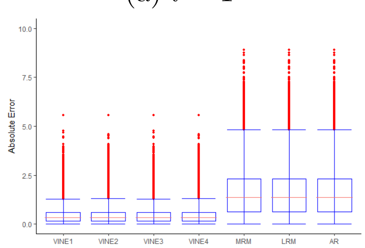
(3) Scenario 2(S)



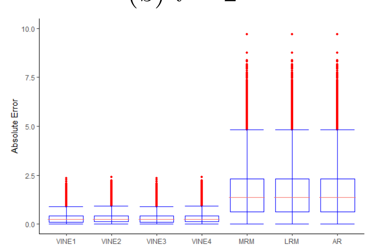
(a) $l = 1$



(b) $l = 2$

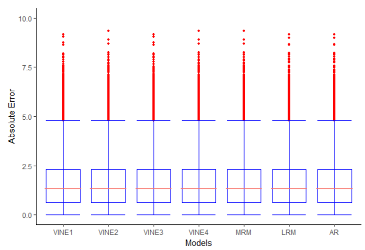


(c) $l = 3$

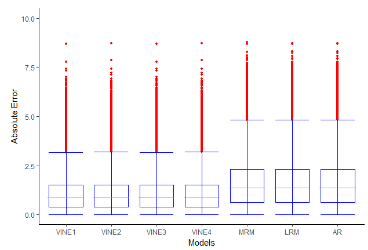


(d) $l = 4$

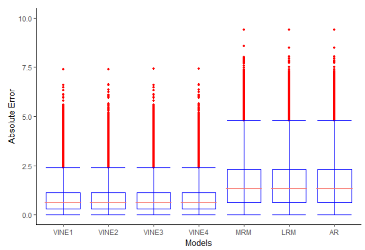
(4) Scenario 2(M)



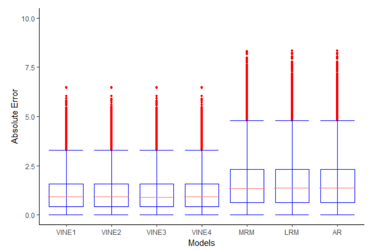
(a) $l = 1$



(b) $l = 2$

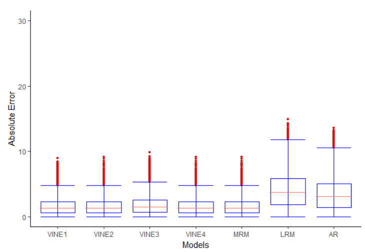


(c) $l = 3$

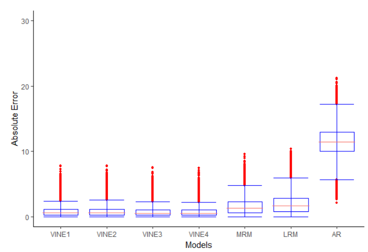


(d) $l = 4$

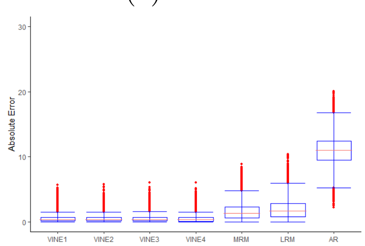
(5) Scenario 3(S)



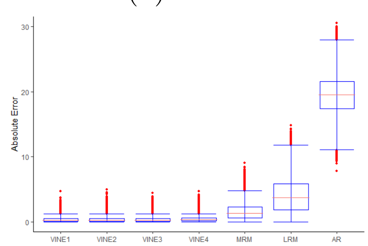
(a) $l = 1$



(b) $l = 2$

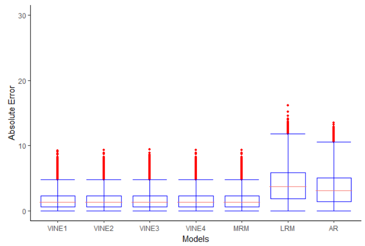


(c) $l = 3$

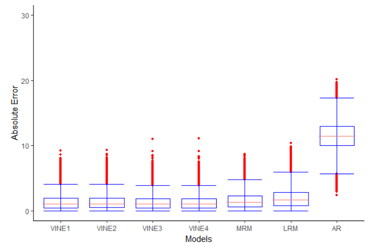


(d) $l = 4$

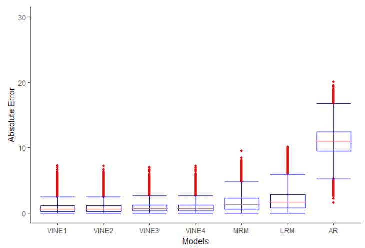
(6) Scenario 3(M)



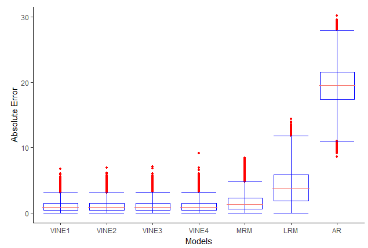
(a) $l = 1$



(b) $l = 2$

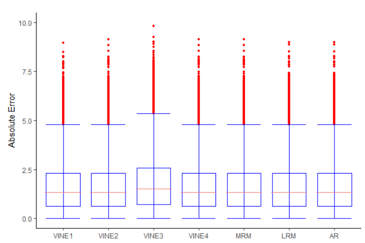


(c) $l = 3$

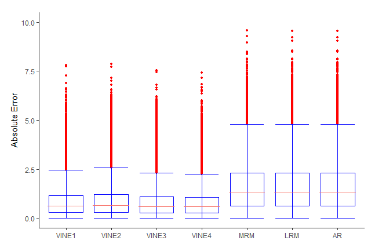


(d) $l = 4$

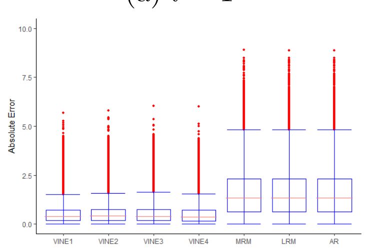
(7) Scenario 4(S)



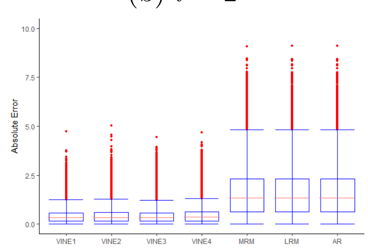
(a) $l = 1$



(b) $l = 2$

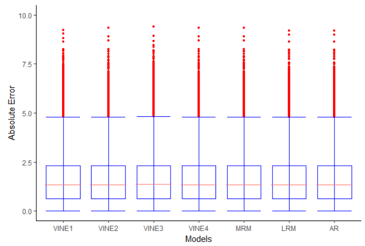


(c) $l = 3$

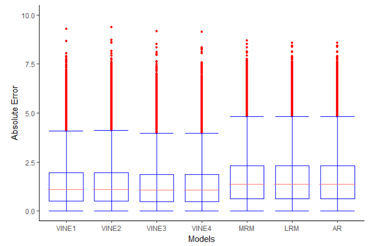


(d) $l = 4$

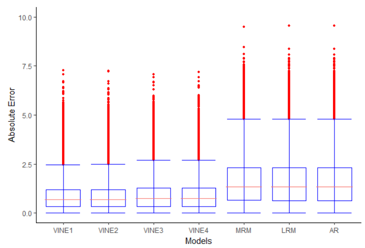
(8) Scenario 4(M)



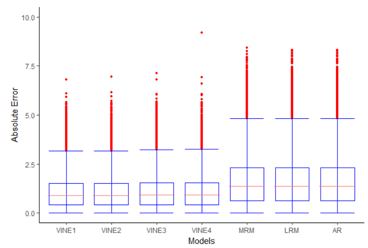
(a) $l = 1$



(b) $l = 2$

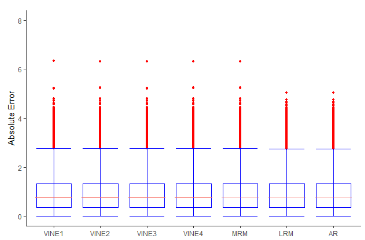


(c) $l = 3$

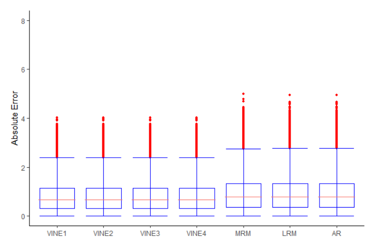


(d) $l = 4$

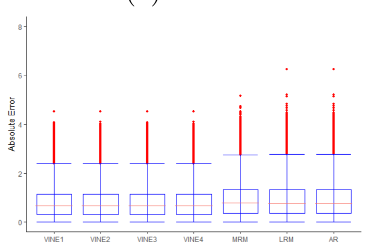
(9) Scenario 5



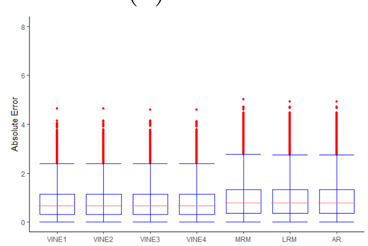
(a) $l = 1$



(b) $l = 2$

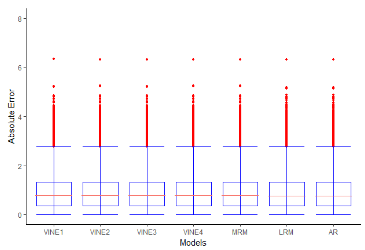


(c) $l = 3$

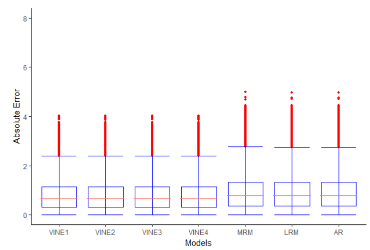


(d) $l = 4$

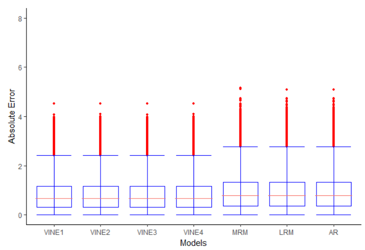
(10) Scenario 6



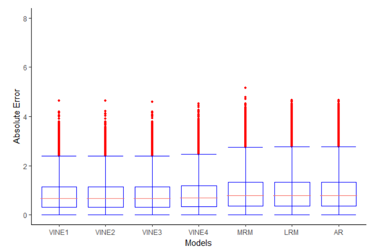
(a) $l = 1$



(b) $l = 2$



(c) $l = 3$



(d) $l = 4$

A.2 Data Analysis

A.2.1 Dataset Description

Table A.9: Location information of 47 observation stations

ID	Name	Latitude	Longitude	Elevation	Group	ID	Name	Latitude	Longitude	Elevation	Group
1	LANSDOWNE HOUSE	52.23	-87.88	255	Training	25	BROCKVILLE	44.60	-75.67	96	Training
2	PICKLE LAKE	51.45	-90.22	386	Training	26	CORNWALL	45.02	-74.75	64	Training
3	RED LAKE	51.07	-93.80	386	Training	27	KINGSTON	44.22	-76.60	93	Training
4	FORT FRANCES	48.65	-93.43	342	Training	28	MORRISBURG	44.92	-75.18	82	Training
5	MINE CENTRE	48.80	-92.60	361	Training	29	OTTAWA	45.38	-75.72	79	Training
6	DRYDEN	49.78	-92.83	413	Training	30	OWEN SOUND	44.58	-80.93	179	Training
7	KENORA	49.78	-94.37	406	Training	31	RIDGETOWN	42.45	-81.88	206	Training
8	CAMERON FALLS	49.15	-88.35	233	Training	32	VINELAND	43.17	-79.42	79	Training
9	GERALDTON	49.78	-86.93	349	Training	33	WELLAND	43.00	-79.27	175	Training
10	THUNDER BAY	48.37	-89.33	199	Training	34	WINDSOR	42.27	-82.97	190	Training
11	HORNEPAYNE	49.20	-84.77	335	Training	35	LONDON	43.03	-81.15	278	Training
12	SAULT STE MARIE	46.48	-84.52	192	Training	36	WOODSTOCK	43.13	-80.77	282	Training
13	WAWA	47.97	-84.78	287	Training	37	BELLEVILLE	44.15	-77.40	76	Training
14	CHAPLEAU	47.82	-83.35	447	Training	38	HAMILTON	43.17	-79.93	238	Training
15	SUDBURY	46.62	-80.80	348	Training	39	ORANGEVILLE	43.92	-80.08	412	Training
16	EARLTON	47.70	-79.85	243	Training	40	TORONTO	43.67	-79.40	113	Training
17	IROQUOIS FALLS	48.75	-80.67	259	Training	41	HALIBURTON	45.03	-78.53	330	Training
18	KAPUSKASING	49.42	-82.47	227	Training	42	PETERBOROUGH	44.23	-78.37	191	Training
19	MOOSONEE	51.27	-80.65	10	Training						
20	SMOKY FALLS	50.07	-82.17	183	Training	43	BIG TROUT LAKE	53.83	-89.87	224	Validation
21	TIMMINS	48.57	-81.38	295	Training	44	SIOUX LOOKOUT	50.12	-91.90	383	Validation
22	MADAWASKA	45.50	-77.98	316	Training	45	BEATRICE	45.13	-79.40	297	Validation
23	NORTH BAY	46.37	-79.42	370	Training	46	HARROW	42.03	-82.90	182	Validation
24	GORE BAY	45.88	-82.57	194	Training	47	ATITOKAN	48.8	-91.58	442	Validation

A.2.2 Model Fitting Results

Table A.10: Estimates of first parameters of the copula functions in the C-Vine structure obtained by the two-stage estimation procedure (standard error in the bracket)

Tree \ Month	2	3	4	5	6	7	8	9	10	11	12
1	3.035(25.076)	1.647(6.797)	0.311(0.063)	0.120(1.253)	-	0.253(0.922)	0.375(0.390)	-0.735(1.636)	-0.248(0.065)	-	-
2		1.804(5.513)	0.544(0.717)	-	0.101(0.136)	0.055(0.449)	0.201(0.049)	1.796(0.275)	1.152(0.267)	1.519(0.268)	1.405(0.201)
3			0.156(5.391)	0.419(1.622)	-	0.100(3.417)	-1.728(2.472)	-	1.059(0.586)	1.392(0.432)	0.023(0.908)
4				1.934(10.769)	0.230(0.636)	0.268(0.254)	-	-	-0.188(0.563)	1.678(1.094)	0.209(0.065)
5					-	0.191(2.059)	1.461(1.274)	-	0.106(0.421)	-	1.080(0.045)
6						0.548(0.077)	1.352(0.254)	1.962(0.795)	0.884(0.583)	-1.113(0.080)	-
7							1.108(0.168)	0.316(0.484)	-1.133(0.222)	1.136(0.335)	-1.037(0.184)
8								-	1.334(0.621)	-	-0.224(0.158)
9									1.216(0.131)	1.581(0.518)	-
10										1.611(2.412)	-0.138(0.171)
11											1.688(0.570)

Table A.11: Estimates of second parameters of the copula functions in the C-Vine structure obtained by the two-stage estimation procedure (standard error in the bracket)

Tree \ Month	2	3	4	5	6	7	8	9	10	11	12
1	0.078(0.952)	0.263(3.607)	-	-	-	-	-	-	-	-	-
2		0.860(1.060)	-	-	-	-	-	0.115(0.069)	-	0.256(0.301)	0.084(0.060)
3			3.674(28.822)	-	-	-	0.075(0.081)	-	0.164(0.888)	0.271(2.377)	5.077(9.447)
4				0.248(2.377)	12.932(56.992)	-	-	-	-	0.819(0.703)	-
5					-	-	0.168(0.578)	-	-	-	-
6						-	0.525(0.329)	-	-	-	-
7							0.310(1.036)	-	-	-	-
8								-	0.348(0.702)	-	10.113(24.569)
9									-	0.152(0.225)	-
10										0.054(0.076)	-
11											0.958(0.080)

Appendix B

Appendix for Chapter 3

B.1 Variability of the Transformed Dependence Parameters

We now explore the within-cluster variability of $\tilde{\gamma}^*_{jl}$, which relates to the choice of rescaling parameter α_{jl}. We use simulations to show how the standard error of the transformed dependence parameter, $\widehat{\mathrm{sd}}(\tilde{\gamma}_{jl})$, varies with respect to the copula form, the sample size and the strength of dependence. Five commonly-used copula forms, Clayton copula, Gumbel copula, Joe copula, Frank copula and Gaussian copula, are considered with Kendall's τ varing from 0.1 to 0.9 and the sample sizes $n = 200$ or 400. In each scenario, simulation is repeated 500 times, and the transformed dependence parameters are estimated using maximum likelihood estimation with standard errors calculated from the inverse of observed information. We report the results in Table B.1.

The results show that the copula form, the true parameter values , the sample size, and the transformation function affect the standard error of the transformed parameter γ^*_{jl}.

Table B.1: Empirical standard error of the MLE of transformed dependence parameter under various copula functions

Kendall's	Clayton		Gumbel		Joe		Gaussian		Frank		Frank†	
τ	$n = 200$	$n = 400$	$n = 200$	$n = 400$	$n = 200$	$n = 400$	$n = 200$	$n = 400$	$n = 200$	$n = 400$	$n = 200$	$n = 400$
0.1	0.590	0.348	0.844	0.413	0.632	0.347	0.142	0.103	0.431	0.307	0.009	0.006
0.2	0.229	0.160	0.291	0.197	0.262	0.175	0.137	0.098	0.436	0.311	0.009	0.006
0.3	0.152	0.108	0.192	0.134	0.177	0.121	0.130	0.093	0.453	0.324	0.009	0.007
0.4	0.118	0.083	0.146	0.103	0.136	0.093	0.123	0.087	0.487	0.350	0.010	0.007
0.5	0.098	0.069	0.119	0.084	0.111	0.076	0.116	0.083	0.548	0.395	0.011	0.008
0.6	0.085	0.061	0.101	0.071	0.096	0.066	0.110	0.079	0.651	0.471	0.013	0.009
0.7	0.076	0.054	0.087	0.061	0.085	0.059	0.106	0.076	0.837	0.604	0.017	0.012
0.8	0.069	0.050	0.077	0.054	0.076	0.053	0.104	0.074	1.225	0.883	0.025	0.018
0.9	0.064	0.046	0.069	0.048	0.068	0.048	0.101	0.073	2.429	1.730	0.057	0.041

† Using transformation function $g(x) = \alpha \log(\frac{x+100}{100-x})$

B.2 Additional Simulation Results

Table B.2: Simulation results for Setting 3.1

			$n = 200$					$n = 400$				
Cluster	Copula	L	EBias	ESE	ASE	95% Interval	ECP	EBias	ESE	ASE	95% Interval	ECP
						Bayesian Estimation						
1	Clayton(1.33)	4	-0.001	0.125	0.127	(1.091,1.589)	0.950	-0.009	0.073	0.089	(1.154,1.503)	0.960
2	Clayton(1.64)	4	-0.002	0.140	0.157	(1.335,1.949)	0.970	-0.015	0.103	0.109	(1.413,1.839)	0.950
3	Clayton(2.00)	4	0.009	0.170	0.181	(1.665,2.373)	0.970	0.001	0.121	0.127	(1.756,2.253)	0.955
4	Clayton(2.44)	4	0.023	0.198	0.203	(2.081,2.876)	0.930	-0.010	0.155	0.145	(2.155,2.724)	0.940
1	Clayton(1.33)	20	0.039	0.070	0.058	(1.245,1.474)	0.850	0.013	0.044	0.037	(1.275,1.420)	0.905
2	Clayton(1.64)	20	-0.005	0.108	0.106	(1.432,1.848)	0.910	-0.014	0.085	0.070	(1.489,1.762)	0.890
3	Clayton(2.00)	20	0.003	0.150	0.147	(1.729,2.303)	0.890	< 0.001	0.122	0.093	(1.820,2.185)	0.900
4	Clayton(2.44)	20	-0.024	0.172	0.181	(2.076,2.786)	0.865	-0.014	0.152	0.123	(2.197,2.680)	0.910
						Maximum Likelihood Estimation						
1	Clayton(1.33)	-	0.021	0.147	0.158	(1.043,1.664)	0.970	-0.004	0.098	0.111	(1.112,1.547)	0.960
2	Clayton(1.64)	-	0.023	0.165	0.176	(1.315,2.003)	0.955	-0.013	0.113	0.123	(1.382,1.864)	0.965
3	Clayton(2.00)	-	0.012	0.191	0.196	(1.628,2.396)	0.965	0.006	0.127	0.139	(1.734,2.277)	0.945
4	Clayton(2.44)	-	0.028	0.216	0.223	(2.035,2.908)	0.955	-0.007	0.155	0.156	(2.131,2.743)	0.950

Table B.3: Simulation results for Setting 3.2

			$n = 200$					$n = 400$				
Cluster	Copula	L	EBias	ESE	ASE	95% Interval	ECP	EBias	ESE	ASE	95% Interval	ECP
						Bayesian Estimation						
1	Clayton(1.33)	4	0.020	0.147	0.150	(1.087,1.677)	0.940	-0.004	0.071	0.081	(1.174,1.490)	0.945
2	Clayton(2.00)	4	0.002	0.169	0.177	(1.665,2.358)	0.965	-0.012	0.115	0.127	(1.744,2.243)	0.960
3	Clayton(3.00)	4	-0.013	0.220	0.233	(2.543,3.456)	0.945	0.003	0.161	0.167	(2.682,3.337)	0.935
4	Clayton(4.67)	4	-0.041	0.346	0.319	(4.018,5.269)	0.945	-0.021	0.249	0.230	(4.203,5.104)	0.940
1	Clayton(1.33)	20	0.054	0.139	0.102	(1.195,1.595)	0.815	0.021	0.052	0.043	(1.265,1.403)	0.895
2	Clayton(2.00)	20	0.038	0.162	0.124	(1.806,2.260)	0.830	0.004	0.094	0.089	(1.834,2.183)	0.910
3	Clayton(3.00)	20	-0.047	0.204	0.178	(2.615,3.313)	0.845	-0.014	0.159	0.142	(2.721,3.279)	0.880
4	Clayton(4.67)	20	-0.071	0.338	0.283	(4.054,5.164)	0.810	-0.033	0.242	0.218	(4.216,5.069)	0.910
						Maximum Likelihood Estimation						
1	Clayton(1.33)	-	0.017	0.148	0.158	(1.040,1.660)	0.965	-0.002	0.099	0.111	(1.113,1.548)	0.965
2	Clayton(2.00)	-	0.009	0.191	0.196	(1.626,2.393)	0.950	-0.014	0.126	0.138	(1.716,2.256)	0.960
3	Clayton(3.00)	-	-0.010	0.232	0.253	(2.494,3.486)	0.950	0.004	0.168	0.179	(2.652,3.355)	0.955
4	Clayton(4.67)	-	-0.036	0.351	0.349	(3.947,5.316)	0.940	-0.019	0.253	0.248	(4.162,5.133)	0.940

Table B.4: Simulation results for Setting 3.3

			$n = 200$					$n = 400$				
Cluster	Copula	L	EBias	ESE	ASE	95% Interval	ECP	EBias	ESE	ASE	95% Interval	ECP
					Bayesian Estimation							
1	Clayton(3)	4	0.004	0.215	0.218	(2.587,3.444)	0.955	0.011	0.141	0.154	(2.705,3.307)	0.945
2	Gumbel(2.5)	4	-0.002	0.140	0.128	(2.253,2.755)	0.920	-0.004	0.075	0.093	(2.321,2.686)	0.915
3	Gaussian(0.81)	4	-0.001	0.018	0.017	(0.772,0.840)	0.955	< 0.001	0.009	0.012	(0.784,0.833)	0.940
4	Frank(7.93)	4	0.011	0.571	0.542	(6.886,9.010)	0.925	0.020	0.406	0.395	(7.148,8.696)	0.930
1	Clayton(3)	4	0.012	0.182	0.219	(2.591,3.449)	0.965	0.001	0.149	0.157	(2.701,3.315)	0.970
2	Gumbel(2.5)	4	-0.010	0.135	0.129	(2.247,2.752)	0.935	-0.002	0.101	0.093	(2.319,2.684)	0.905
3	Gaussian(0.81)	4	-0.001	0.017	0.017	(0.772,0.840)	0.940	< 0.001	0.012	0.012	(0.784,0.833)	0.950
4	Frank(7.93)†	4	0.035	0.570	0.550	(6.931,9.088)	0.955	-0.017	0.440	0.394	(7.143,8.689)	0.925
1	Clayton(3)	20	-0.011	0.155	0.120	(2.761,3.231)	0.850	0.003	0.105	0.079	(2.853,3.161)	0.875
2	Gumbel(2.5)	20	-0.018	0.140	0.088	(2.312,2.657)	0.835	-0.011	0.098	0.064	(2.368,2.618)	0.860
3	Gaussian(0.81)	20	< 0.001	0.016	0.012	(0.788,0.832)	0.845	0.001	0.010	0.008	(0.795,0.826)	0.855
4	Frank(7.93)	20	0.040	0.558	0.341	(7.296,8.633)	0.850	0.015	0.419	0.315	(7.317,8.543)	0.855
1	Clayton(3)	20	-0.010	0.158	0.130	(2.738,3.250)	0.875	-0.008	0.102	0.081	(2.831,3.149)	0.905
2	Gumbel(2.5)	20	-0.014	0.136	0.100	(2.292,2.686)	0.850	-0.010	0.094	0.064	(2.360,2.610)	0.870
3	Gaussian(0.81)	20	0.001	0.015	0.013	(0.784,0.835)	0.845	0.001	0.011	0.008	(0.796,0.828)	0.855
4	Frank(7.93)†	20	0.024	0.549	0.321	(7.325,8.583)	0.820	-0.009	0.404	0.312	(7.318,8.542)	0.860
					Maximum Likelihood Estimation							
1	Clayton(3)	-	0.013	0.253	0.254	(2.515,3.511)	0.955	< 0.001	0.164	0.179	(2.649,3.351)	0.965
2	Gumbel(2.5)	-	-0.004	0.147	0.145	(2.211,2.781)	0.940	-0.003	0.105	0.103	(2.296,2.699)	0.920
3	Gaussian(0.81)	-	-0.001	0.019	0.019	(0.772,0.846)	0.960	< 0.001	0.013	0.013	(0.784,0.836)	0.950
4	Frank(7.93)	-	-0.016	0.643	0.643	(6.653,9.715)	0.960	-0.046	0.456	0.454	(6.995,8.774)	0.930

(†) Using transformation function $g(\theta) = \log\left(\frac{\theta+100}{100-\theta}\right)$

Table B.5: Simulation results for Setting 3.4

			$n = 200$					$n = 400$				
Cluster	Copula	L	EBias	ESE	ASE	95% Interval	ECP	EBias	ESE	ASE	95% Interval	ECP
					Bayesian Estimation							
1	Clayton(3)	4	0.015	0.224	0.229	(2.577,3.477)	0.955	0.012	0.160	0.164	(2.697,3.341)	0.950
2	Gumbel(4)	4	0.018	0.220	0.215	(3.609,4.452)	0.940	-0.021	0.151	0.153	(3.684,4.285)	0.940
3	Gaussian(0.6)	4	-0.006	0.034	0.035	(0.520,0.658)	0.950	-0.004	0.026	0.026	(0.543,0.643)	0.945
4	Frank(13)	4	0.016	0.865	0.804	(11.472,14.623)	0.940	0.087	0.651	0.613	(11.899,14.300)	0.930
1	Clayton(3)	4	0.021	0.225	0.227	(2.589,3.480)	0.955	0.012	0.162	0.163	(2.700,3.339)	0.965
2	Gumbel(4)	4	0.023	0.219	0.212	(3.620,4.450)	0.940	-0.022	0.151	0.152	(3.686,4.280)	0.940
3	Gaussian(0.6)	4	-0.006	0.035	0.035	(0.519,0.658)	0.945	-0.003	0.026	0.025	(0.545,0.643)	0.960
4	Frank(13)†	4	-0.010	0.856	0.786	(11.469,14.551)	0.930	0.090	0.651	0.577	(11.973,14.236)	0.920
1	Clayton(3)	20	0.008	0.189	0.165	(2.697,3.344)	0.905	0.003	0.135	0.116	(2.780,3.234)	0.925
2	Gumbel(4)	20	0.009	0.190	0.157	(3.710,4.327)	0.895	-0.020	0.131	0.117	(3.749,4.208)	0.900
3	Gaussian(0.6)	20	< 0.001	0.028	0.023	(0.552,0.643)	0.835	-0.003	0.021	0.017	(0.562,0.629)	0.895
4	Frank(13)	20	0.009	0.876	0.600	(11.844,14.197)	0.815	0.084	0.639	0.443	(12.220,13.958)	0.850
1	Clayton(3)	20	0.012	0.193	0.168	(2.693,3.350)	0.905	0.002	0.126	0.114	(2.780,3.228)	0.915
2	Gumbel(4)	20	0.005	0.186	0.159	(3.700,4.323)	0.890	-0.015	0.130	0.112	(3.770,4.209)	0.890
3	Gaussian(0.6)	20	-0.002	0.026	0.023	(0.551,0.641)	0.865	-0.002	0.020	0.017	(0.563,0.630)	0.905
4	Frank(13)†	20	-0.021	0.837	0.594	(11.828,14.155)	0.830	0.067	0.645	0.445	(12.204,13.955)	0.845
					Maximum Likelihood Estimation							
1	Clayton(3)	-	0.026	0.242	0.254	(2.524,3.521)	0.960	0.013	0.171	0.180	(2.661,3.365)	0.960
2	Gumbel(4)	-	0.022	0.234	0.236	(3.559,4.486)	0.960	-0.022	0.158	0.165	(3.654,4.302)	0.945
3	Gaussian(0.6)	-	-0.004	0.038	0.039	(0.519,0.673)	0.940	-0.003	0.028	0.028	(0.543,0.651)	0.960
4	Frank(13)	-	-0.076	0.869	0.908	(11.144,14.704)	0.960	0.054	0.658	0.647	(11.785,14.322)	0.935

(†) Using transformation function $g(\theta) = \log\left(\frac{\theta+100}{100-\theta}\right)$

B.3 Additional Results for Data Analysis

B.3.1 Marginal Distribution of Six Features in Three Health Groups

The marginal density of the k-th biomedical feature in the j-th group of people is given by

$$f_{jk}(y_{jik}) = \frac{p_{jk}}{2k_{jk}\sigma_{jk}q_{jk}^{1/p_{jk}}B\left(\frac{1}{p_{jk}}, q_{jk}\right)\left(\frac{|y_{jk}-\mu_{jk}+r_{jk}|^{p_{jk}}}{q_{jk}(s_{jk}\sigma_{jk})^{p_{jk}}(\lambda_{jk}\mathrm{sign}(y_{jik}-\mu_{jk}+r_{jk})+1)^{p_{jk}}}+1\right)^{\frac{1}{p_{jk}}+q_{jk}}},$$

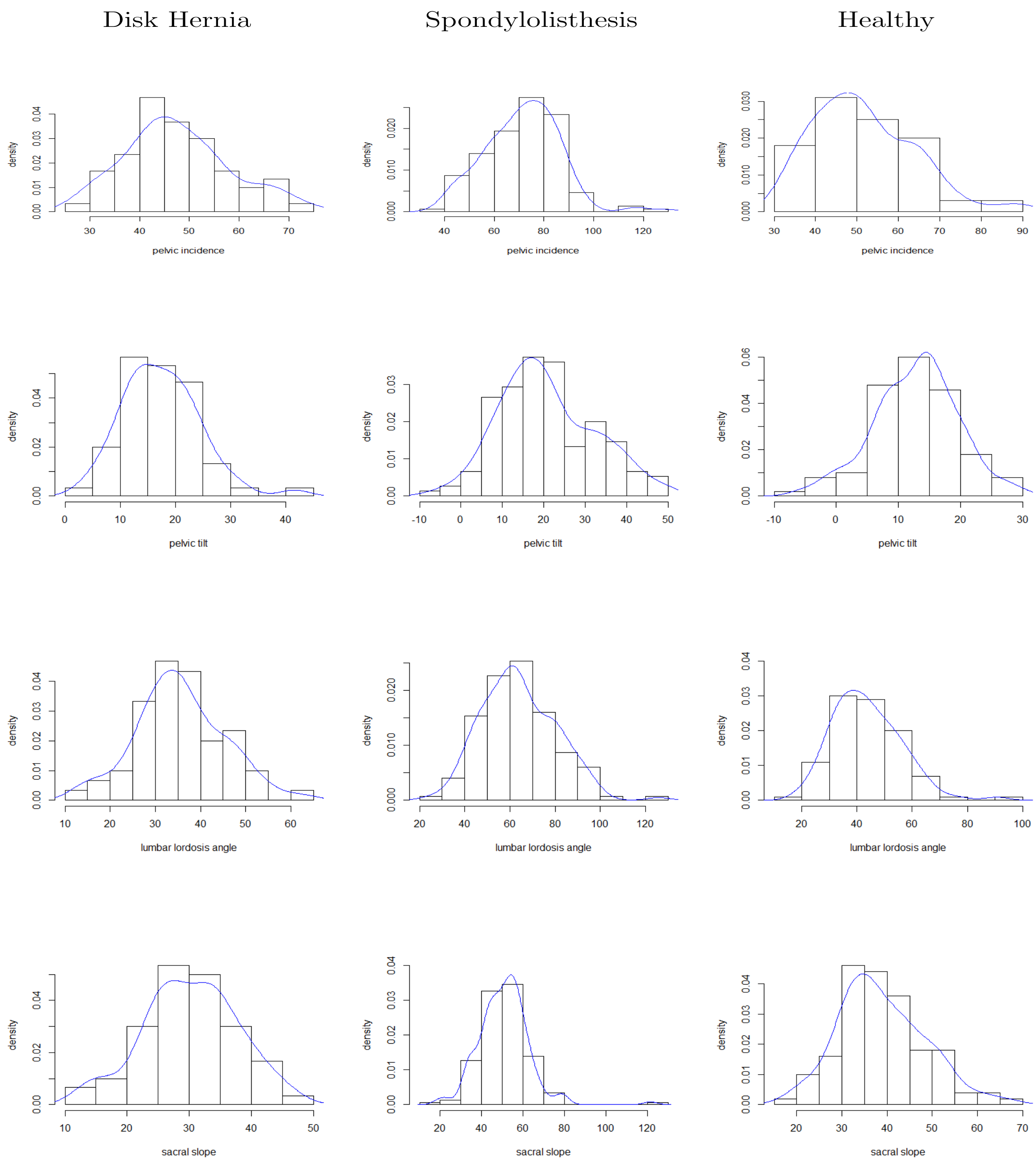
where $B(\cdot)$ is the Beta function, μ is the location parameter, σ is the scale parameter, $\lambda \in (-1, 1)$ is the skewness parameter, p and q are kurtosis parameters, and r_{jk} and s_{jk} are given by

$$r_{jk} = \frac{2v_{jk}\sigma_{jk}\lambda_{jk}q_{jk}^{1/p_{jk}}B(\frac{2}{p_{jk}}, q_{jk}-\frac{1}{p_{jk}})}{B(\frac{1}{p_{jk}}, q_{jk})}$$

$$s_{jk} = \frac{q_{jk}^{1/p_{jk}}}{\sqrt{(3\lambda_{jk}^2+1)\frac{B(\frac{3}{p_{jk}},q_{jk}-\frac{2}{p_{jk}})}{B\left(\frac{1}{p_{jk}},q_{jk}\right)} - 4\lambda_{jk}^2\frac{B(\frac{2}{p_{jk}},q_{jk}-\frac{1}{p_{jk}})^2}{B\left(\frac{1}{p_{jk}},q_{jk}\right)^2}}}.$$

Table B.6: MLE of marginal parameters in the generalized skewed-t distributions

Groups	Features	Skewed t distribution			Normal distribution	
		μ	σ	λ	μ	σ
Disk Hernia	PI	47.711	10.581	0.238	47.638	10.608
	PT	17.431	6.942	0.314	17.398	6.958
	LL	35.522	9.677	0.101	35.464	9.686
	SS	30.261	7.495	-0.095	30.239	7.492
	PR	116.337	9.237	-0.190	116.475	9.277
	DS	2.470	5.483	-0.141	2.480	5.485
Spondylolisthesis	PI	71.538	15.056	0.065	71.514	15.059
	PT	20.821	11.436	0.279	20.748	11.468
	LL	64.100	16.346	0.256	64.110	16.342
	SS	50.993	12.207	0.204	50.766	12.278
	PR	114.599	15.517	0.087	114.519	15.528
	DS	51.897	35.119	0.629	51.897	39.974
Healthy	PI	51.401	12.577	0.635	51.685	12.306
	PT	12.789	6.739	-0.108	12.821	6.745
	LL	43.643	12.239	0.392	43.543	12.299
	SS	38.921	9.551	0.276	38.863	9.576
	PR	123.893	8.969	0.015	123.891	8.969
	DS	2.583	6.043	0.410	2.187	6.276


Disk Hernia
Spondylolisthesis
Healthy
density
pelvic incidence
pelvic tilt
lumbar lordosis angle
sacral slope

Disk Hernia Spondylolisthesis Healthy

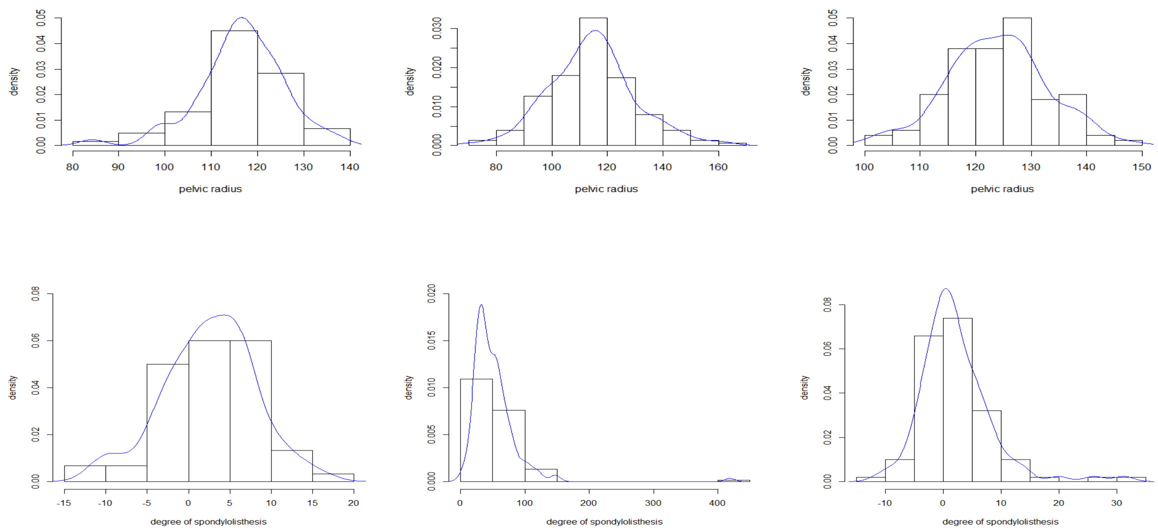
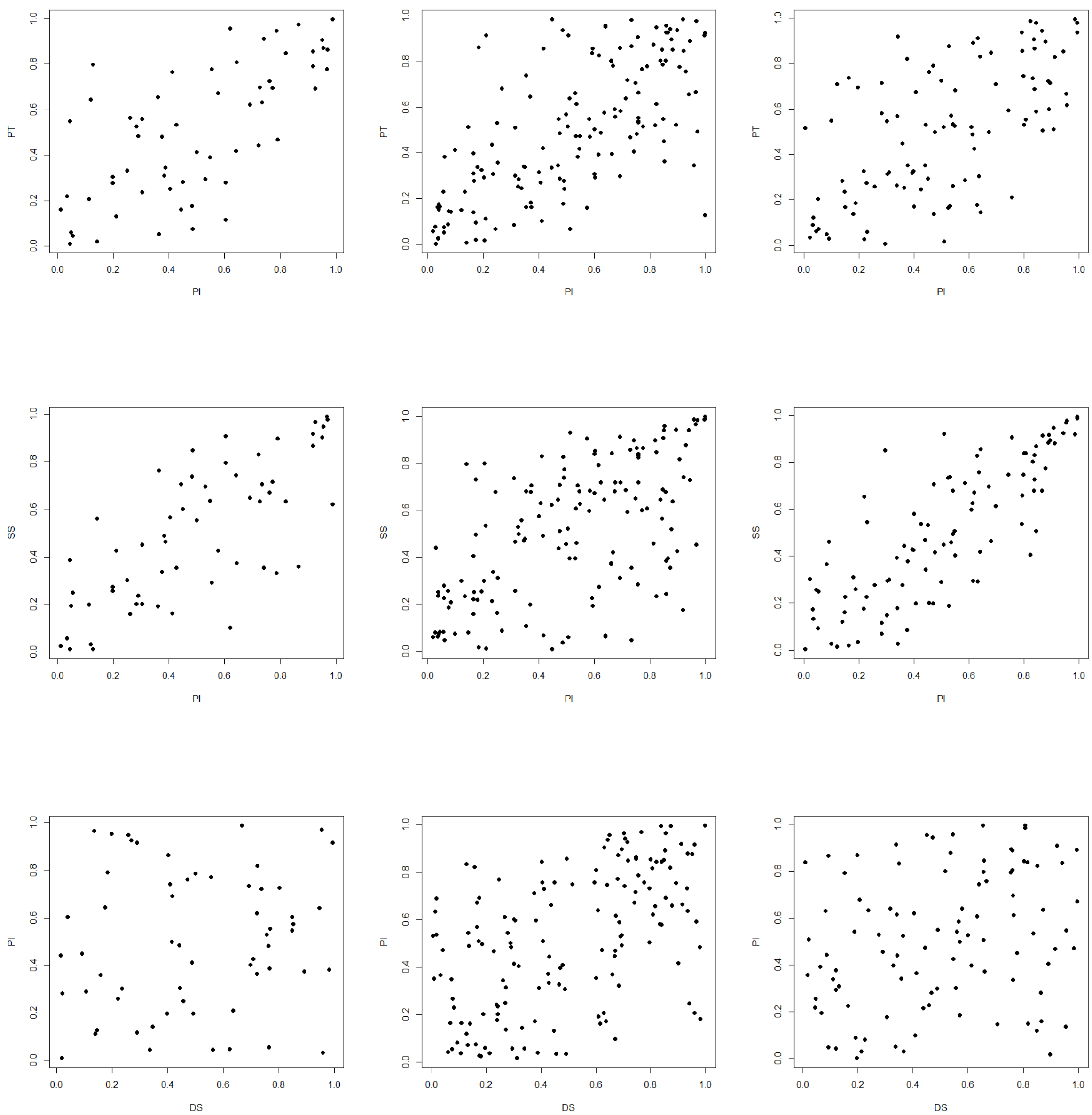


Figure B.2: Histograms of six biomedical features on three groups

B.3.2 Dependence Model

Disk Hernia | Spondylolisthesis | Healthy



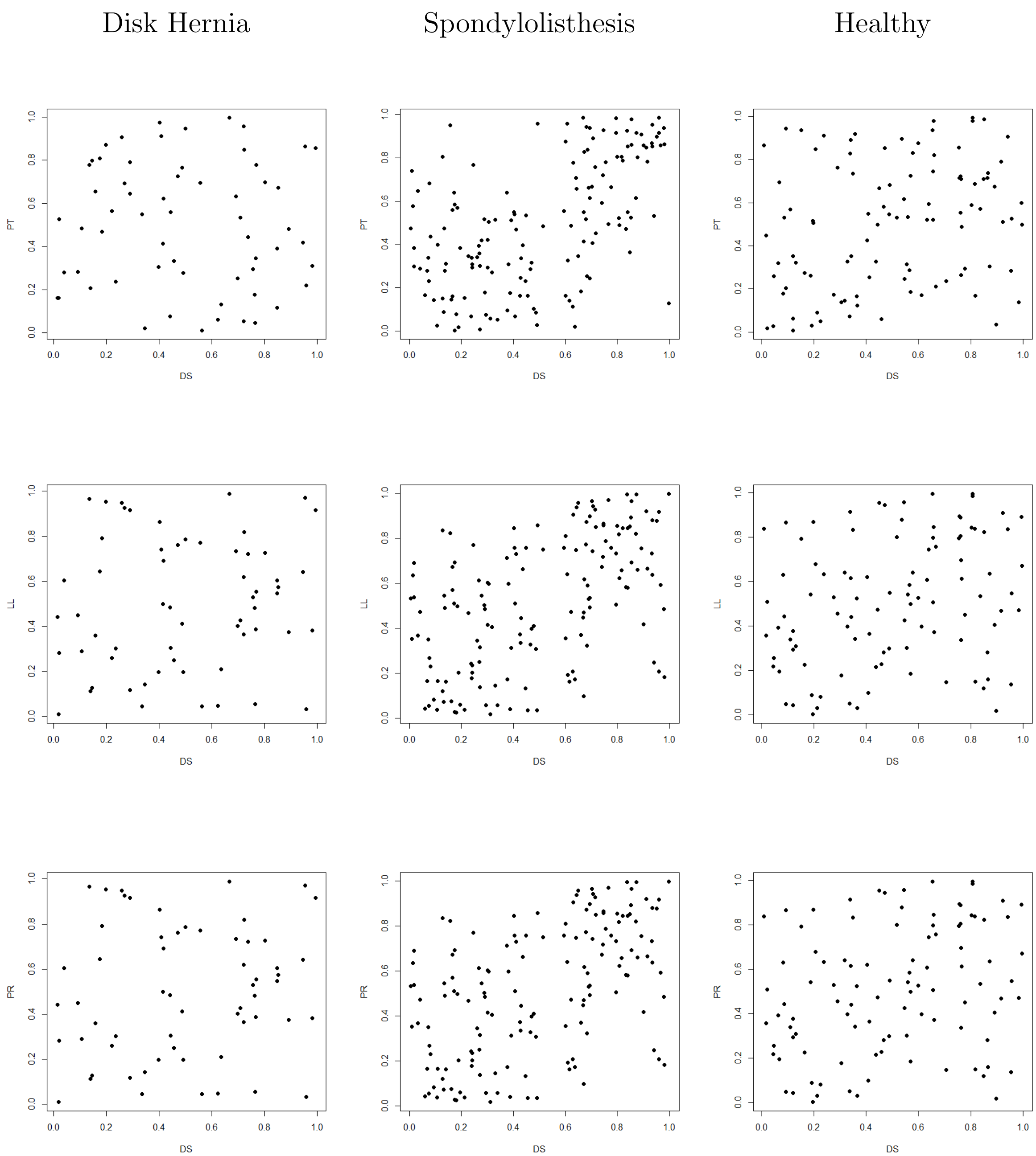


Figure B.4: Scatter plots of six pairs of bivariate dependence in 3 health groups

Appendix C

Appendix for Chapter 4

C.1 Additional Simulation Results

Table C.1: Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for High Signal Setting

M-DPM-CM

	$n = 50$					$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	47.236%	0.047	0.238	0.230	0.924	31.667%	0.026	0.159	0.161	0.966	24.000%	0.005	0.116	0.113	0.930	17.000%	-0.005	0.087	0.081	0.948	11.000%	< 0.001	0.051	0.051	0.949
2	11.055%	-0.009	0.217	0.212	0.950	2.000%	-0.036	0.156	0.150	0.939	0.000%	-0.004	0.106	0.104	0.943	0.000%	-0.001	0.076	0.073	0.927	0.000%	< 0.001	0.050	0.046	0.955
3	46.734%	0.055	0.233	0.231	0.925	31.667%	0.026	0.159	0.161	0.966	24.000%	0.005	0.115	0.113	0.930	17.333%	-0.009	0.079	0.080	0.956	11.000%	-0.002	0.048	0.051	0.949
4	22.613%	-0.006	0.063	0.048	0.877	9.667%	-0.002	0.032	0.032	0.945	0.333%	-0.001	0.023	0.023	0.953	0.000%	0.001	0.017	0.016	0.947	0.000%	< 0.001	0.010	0.010	0.950
5	16.080%	0.055	0.876	0.844	0.946	3.000%	-0.003	0.629	0.576	0.942	1.000%	-0.014	0.393	0.396	0.966	0.000%	-0.017	0.261	0.278	0.963	0.000%	0.013	0.179	0.177	0.950
6	46.734%	0.055	0.233	0.230	0.925	32.000%	0.025	0.160	0.162	0.966	23.667%	0.005	0.116	0.114	0.930	17.333%	-0.009	0.079	0.080	0.956	11.000%	-0.002	0.048	0.051	0.949
7	46.231%	0.046	0.246	0.234	0.925	32.000%	0.024	0.157	0.161	0.971	23.667%	0.005	0.116	0.114	0.930	17.000%	-0.008	0.080	0.080	0.952	11.000%	-0.002	0.048	0.051	0.949
8	12.563%	-0.013	0.222	0.212	0.948	1.667%	-0.028	0.155	0.149	0.942	0.000%	-0.001	0.104	0.104	0.947	0.000%	< 0.001	0.074	0.073	0.933	0.000%	0.004	0.049	0.046	0.955
9	26.633%	-0.012	0.086	0.049	0.878	8.333%	-0.002	0.035	0.032	0.935	1.333%	-0.001	0.022	0.022	0.953	0.000%	< 0.001	0.017	0.016	0.947	0.000%	0.001	0.010	0.010	0.950
10	15.578%	0.018	1.003	0.843	0.917	5.000%	0.018	0.594	0.573	0.944	1.333%	-0.016	0.378	0.395	0.973	0.000%	-0.017	0.261	0.278	0.963	0.000%	0.005	0.175	0.177	0.955
11	46.734%	0.048	0.242	0.232	0.944	32.000%	0.026	0.160	0.161	0.966	23.667%	0.002	0.121	0.114	0.926	17.000%	-0.008	0.080	0.080	0.952	11.000%	-0.002	0.048	0.051	0.949
12	23.116%	-0.021	0.063	0.047	0.900	9.000%	-0.003	0.033	0.032	0.949	1.667%	-0.001	0.022	0.022	0.956	0.667%	< 0.001	0.016	0.016	0.946	0.000%	0.001	0.010	0.010	0.950

AIC

	$n = 50$					$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	53.000%	0.055	0.510	0.507	0.950	46.333%	0.025	0.332	0.357	0.975	36.333%	0.043	0.242	0.255	0.969	31.000%	0.001	0.180	0.179	0.937	22.000%	-0.010	0.107	0.113	0.966
2	39.333%	-0.034	0.273	0.293	0.956	19.333%	-0.044	0.207	0.210	0.950	5.000%	-0.002	0.147	0.146	0.954	0.333%	-0.003	0.105	0.103	0.950	0.000%	0.001	0.068	0.065	0.920
3	55.667%	0.135	0.503	0.521	0.970	39.333%	0.020	0.361	0.357	0.978	44.000%	0.017	0.242	0.254	0.952	42.000%	0.003	0.179	0.179	0.960	29.000%	-0.005	0.115	0.113	0.958
4	64.667%	-0.018	0.075	0.074	0.896	43.667%	-0.001	0.050	0.054	0.953	19.000%	-0.002	0.042	0.038	0.930	5.667%	0.001	0.028	0.028	0.951	0.000%	< 0.001	0.018	0.017	0.953
5	52.000%	0.228	1.300	1.138	0.924	28.333%	-0.045	0.787	0.785	0.940	9.333%	0.006	0.525	0.557	0.960	1.667%	-0.001	0.370	0.393	0.973	0.000%	0.007	0.256	0.249	0.950
6	48.333%	0.031	0.449	0.506	0.968	47.333%	0.025	0.377	0.359	0.949	43.000%	0.025	0.263	0.253	0.936	36.667%	-0.004	0.189	0.178	0.937	30.333%	-0.006	0.108	0.113	0.976
7	46.333%	0.064	0.515	0.508	0.944	48.000%	-0.005	0.330	0.355	0.981	36.333%	-0.011	0.246	0.252	0.963	36.000%	0.001	0.193	0.179	0.917	28.667%	0.013	0.114	0.113	0.963
8	38.333%	-0.049	0.291	0.296	0.957	15.000%	-0.035	0.223	0.208	0.929	3.333%	-0.013	0.149	0.147	0.945	0.000%	-0.002	0.106	0.103	0.930	0.000%	0.003	0.066	0.065	0.963
9	67.000%	-0.027	0.081	0.073	0.848	46.333%	-0.010	0.055	0.053	0.901	23.000%	-0.003	0.037	0.038	0.952	5.000%	-0.002	0.028	0.027	0.951	0.000%	< 0.001	0.017	0.017	0.947
10	50.667%	0.068	1.157	1.124	0.966	29.667%	0.130	0.760	0.796	0.967	11.333%	0.007	0.538	0.557	0.955	1.333%	-0.014	0.366	0.392	0.966	0.000%	0.016	0.239	0.249	0.957
11	56.000%	0.094	0.531	0.513	0.947	44.667%	0.030	0.390	0.359	0.932	45.000%	-0.030	0.259	0.250	0.945	38.667%	0.016	0.189	0.179	0.935	28.000%	0.005	0.105	0.113	0.968
12	66.000%	-0.017	0.078	0.074	0.873	43.000%	-0.002	0.054	0.055	0.953	22.333%	-0.004	0.035	0.038	0.974	5.333%	0.002	0.027	0.028	0.947	0.333%	0.001	0.017	0.017	0.946

Table C.2: Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Low Signal Setting

	M-DPM-CM																								
	$n = 50$					$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	42.500%	0.055	0.248	0.241	0.956	32.000%	0.028	0.155	0.161	0.975	21.667%	0.005	0.115	0.113	0.932	15.667%	-0.006	0.085	0.080	0.953	13.000%	-0.004	0.047	0.051	0.954
2	20.000%	0.038	0.224	0.224	0.944	1.667%	0.023	0.178	0.152	0.932	0.000%	0.009	0.110	0.104	0.930	0.000%	-0.009	0.078	0.073	0.923	0.000%	-0.001	0.046	0.046	0.960
3	42.500%	0.088	0.260	0.245	0.957	32.000%	0.028	0.155	0.161	0.975	21.667%	0.001	0.124	0.114	0.928	15.667%	-0.006	0.085	0.081	0.953	13.000%	-0.004	0.047	0.051	0.954
4	44.000%	-0.011	0.082	0.054	0.830	17.667%	-0.002	0.039	0.032	0.920	4.000%	<0.001	0.023	0.023	0.948	0.667%	0.001	0.018	0.016	0.930	0.000%	<0.001	0.010	0.010	0.980
5	51.000%	-0.016	0.978	0.781	0.905	31.333%	0.039	0.647	0.551	0.923	7.333%	0.001	0.409	0.393	0.946	0.000%	-0.013	0.262	0.279	0.963	0.000%	0.014	0.183	0.177	0.945
6	43.000%	0.076	0.247	0.242	0.956	32.000%	0.028	0.156	0.161	0.975	21.667%	0.005	0.115	0.113	0.932	15.667%	-0.004	0.085	0.080	0.953	13.000%	-0.004	0.047	0.051	0.954
7	44.000%	0.062	0.273	0.249	0.946	32.333%	0.025	0.156	0.162	0.965	21.667%	0.005	0.115	0.113	0.932	15.667%	-0.006	0.080	0.080	0.957	13.000%	-0.004	0.047	0.051	0.954
8	21.000%	0.093	0.272	0.227	0.927	2.667%	0.022	0.166	0.150	0.942	0.000%	0.006	0.106	0.103	0.937	0.000%	-0.005	0.080	0.073	0.920	0.000%	0.002	0.049	0.046	0.950
9	44.000%	0.008	0.063	0.047	0.905	18.667%	0.006	0.038	0.032	0.918	5.000%	-0.001	0.023	0.023	0.958	0.000%	<0.001	0.018	0.016	0.933	0.000%	<0.001	0.010	0.010	0.980
10	50.000%	-0.019	0.943	0.798	0.890	33.000%	0.042	0.631	0.546	0.925	7.000%	0.005	0.393	0.393	0.957	0.667%	-0.016	0.261	0.278	0.963	0.000%	0.004	0.175	0.177	0.955
11	43.000%	0.063	0.253	0.246	0.947	32.000%	0.029	0.162	0.163	0.975	21.667%	0.005	0.115	0.113	0.932	15.667%	-0.009	0.080	0.080	0.957	13.000%	-0.004	0.047	0.051	0.954
12	41.500%	-0.003	0.606	0.500	0.889	21.333%	<0.001	0.039	0.031	0.928	3.333%	-0.001	0.025	0.023	0.952	0.000%	<0.001	0.017	0.016	0.937	0.000%	<0.001	0.010	0.010	0.975

	AIC																								
	$n = 50$					$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	49.000%	0.056	0.550	0.508	0.941	46.333%	0.025	0.332	0.357	0.975	36.333%	0.043	0.242	0.255	0.969	31.000%	0.001	0.180	0.179	0.937	21.000%	-0.016	0.105	0.113	0.968
2	46.500%	0.050	0.267	0.296	0.972	18.667%	0.047	0.244	0.210	0.918	6.000%	0.007	0.147	0.146	0.954	0.000%	-0.012	0.101	0.102	0.943	0.000%	-0.001	0.065	0.065	0.950
3	58.000%	0.166	0.528	0.525	0.952	39.333%	0.020	0.361	0.357	0.978	44.000%	0.017	0.242	0.254	0.952	42.000%	0.003	0.179	0.179	0.960	30.000%	-0.004	0.119	0.113	0.950
4	69.500%	0.023	0.093	0.073	0.820	45.667%	<0.001	0.056	0.055	0.945	21.000%	-0.002	0.038	0.039	0.966	4.667%	0.002	0.029	0.027	0.934	0.000%	<0.001	0.016	0.017	0.970
5	50.000%	0.273	1.274	1.142	0.940	28.333%	-0.045	0.787	0.785	0.940	9.333%	0.006	0.525	0.557	0.960	1.667%	-0.001	0.370	0.393	0.973	0.000%	0.016	0.260	0.249	0.950
6	52.000%	0.050	0.432	0.506	0.979	47.333%	0.025	0.377	0.359	0.949	43.000%	0.025	0.263	0.253	0.936	36.667%	-0.004	0.189	0.178	0.937	30.000%	<0.001	0.115	0.113	0.979
7	48.000%	0.065	0.497	0.509	0.962	48.000%	-0.005	0.330	0.355	0.981	36.333%	-0.011	0.246	0.252	0.963	36.000%	0.001	0.193	0.179	0.917	33.000%	0.020	0.108	0.114	0.970
8	38.000%	0.032	0.271	0.295	0.968	15.667%	0.027	0.200	0.208	0.953	2.667%	0.025	0.151	0.147	0.952	0.667%	0.001	0.107	0.103	0.933	0.000%	0.004	0.064	0.065	0.945
9	62.000%	0.017	0.067	0.074	0.934	42.667%	-0.002	0.053	0.055	0.948	22.667%	0.003	0.038	0.039	0.948	3.000%	-0.001	0.026	0.027	0.962	0.000%	-0.001	0.018	0.017	0.955
10	49.500%	0.042	1.176	1.122	0.960	29.667%	0.130	0.760	0.796	0.967	11.333%	0.007	0.538	0.557	0.955	1.333%	-0.014	0.366	0.392	0.966	0.000%	0.008	0.244	0.249	0.950
11	57.000%	0.054	0.547	0.508	0.919	44.667%	0.030	0.390	0.359	0.934	45.000%	-0.030	0.259	0.250	0.945	38.667%	0.016	0.189	0.179	0.935	29.000%	0.003	0.102	0.113	0.972
12	63.500%	0.020	0.071	0.073	0.890	42.333%	0.005	0.052	0.054	0.913	20.667%	0.002	0.042	0.039	0.920	6.333%	0.001	0.028	0.027	0.932	0.000%	<0.001	0.018	0.017	0.950

Table C.3: Simulation results for copula selection and parameter estimation of M-DPM-CM and AIC methods for Nearly Independent Setting

M-DPM-CM

	$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	73.000%	-0.051	0.068	0.045	0.672	67.500%	-0.029	0.066	0.036	0.752	53.500%	-0.012	0.049	0.028	0.863	22.000%	-0.006	0.024	0.018	0.921
2	89.500%	-0.025	0.076	0.058	0.754	87.500%	0.003	0.050	0.029	0.860	79.000%	0.005	0.032	0.023	0.929	51.000%	0.004	0.018	0.016	0.948
3	71.000%	-0.047	0.076	0.047	0.612	64.500%	-0.021	0.067	0.039	0.721	55.000%	-0.013	0.046	0.027	0.878	23.5%	-0.006	0.025	0.018	0.912
4	83.500%	-0.032	0.478	0.224	0.818	83.000%	-0.098	0.332	0.217	0.857	76.500%	0.014	0.186	0.160	0.936	38.000%	0.003	0.135	0.109	0.939
5	74.500%	0.311	0.880	0.652	0.627	68.000%	0.008	0.470	0.391	0.898	53.500%	0.012	0.330	0.211	0.914	30.500%	0.013	0.139	0.136	0.971
6	75.000%	-0.058	0.082	0.053	0.720	67.000%	-0.034	0.073	0.046	0.715	56.500%	-0.009	0.050	0.028	0.818	24.000%	-0.005	0.026	0.019	0.921
7	74.500%	-0.051	0.080	0.061	0.735	68.500%	-0.034	0.073	0.043	0.705	56.500%	-0.009	0.050	0.028	0.828	23.500%	-0.006	0.025	0.018	0.920
8	91.000%	-0.042	0.052	0.042	0.750	87.000%	0.009	0.061	0.030	0.792	82.000%	0.001	0.029	0.022	0.917	49.000%	0.003	0.017	0.016	0.951
9	83.000%	-0.075	0.412	0.226	0.852	85.500%	0.038	0.285	0.194	0.907	77.500%	-0.031	0.175	0.159	0.933	37.500%	0.009	0.150	0.119	0.924
10	73.500%	0.336	0.872	0.652	0.742	69.500%	-0.005	0.443	0.288	0.875	53.000%	< 0.001	0.334	0.213	0.915	30.500%	0.004	0.133	0.136	0.978
11	72.000%	-0.044	0.081	0.045	0.554	68.500%	-0.033	0.076	0.035	0.726	58.000%	-0.014	0.041	0.027	0.850	23.000%	-0.003	0.026	0.020	0.926
12	81.500%	-0.087	0.528	0.333	0.865	84.500%	0.047	0.408	0.211	0.855	76.000%	0.037	0.224	0.158	0.917	39.500%	-0.001	0.137	0.107	0.939

AIC

	$n = 100$					$n = 200$					$n = 400$					$n = 1000$				
Cluster	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP	MSP	EBias	ESE	ASE	ECP
1	71.000%	0.048	0.106	0.132	1.000	65.000%	0.024	0.090	0.093	0.971	54.667%	0.019	0.066	0.067	0.967	33.500%	-0.003	0.038	0.041	0.962
2	89.500%	0.053	0.073	0.079	1.000	79.500%	0.038	0.041	0.056	1.000	67.000%	0.002	0.034	0.036	0.969	42.000%	0.002	0.022	0.023	0.983
3	74.500%	0.064	0.121	0.137	0.980	61.000%	0.021	0.089	0.091	0.962	56.000%	0.013	0.064	0.065	0.989	37.500%	0.001	0.043	0.041	0.928
4	73.00%	0.104	0.685	0.607	0.889	56.500%	0.036	0.455	0.428	0.954	45.667%	0.075	0.271	0.303	0.963	31.500%	0.008	0.173	0.191	0.978
5	69.000%	-0.242	0.602	0.606	0.952	57.500%	-0.109	0.395	0.429	0.941	49.000%	-0.063	0.307	0.303	0.938	35.000%	0.016	0.192	0.191	0.929
6	72.500%	0.034	0.113	0.129	0.945	61.000%	0.036	0.096	0.095	0.936	51.000%	-0.001	0.065	0.065	0.946	33.500%	0.001	0.045	0.041	0.917
7	69.500%	0.042	0.119	0.130	0.951	63.000%	0.034	0.086	0.092	0.959	52.667%	0.006	0.068	0.066	0.958	31.000%	< 0.001	0.038	0.042	0.971
8	89.500%	0.040	0.054	0.077	1.000	76.000%	0.032	0.052	0.055	0.938	72.000%	0.017	0.040	0.038	0.895	39.000%	0.007	0.021	0.023	0.984
9	70.000%	0.201	0.536	0.610	0.950	61.500%	0.088	0.412	0.427	0.948	45.667%	0.013	0.283	0.302	0.982	26.000%	0.020	0.202	0.191	0.932
10	71.000%	-0.123	0.631	0.607	0.914	59.000%	-0.134	0.385	0.431	0.963	53.333%	-0.054	0.285	0.302	0.977	36.500%	-0.010	0.189	0.191	0.937
11	69.000%	0.049	0.119	0.135	0.968	66.000%	0.018	0.090	0.092	0.956	54.333%	0.020	0.063	0.066	0.930	27.500%	0.002	0.042	0.042	0.959
12	68.500%	0.074	0.565	0.619	0.952	64.500%	0.152	0.494	0.431	0.887	40.667%	0.073	0.278	0.300	0.958	31.000%	0.024	0.184	0.191	0.942

Appendix D

Appendix for Chapter 5

D.1 Proofs of Theorems

D.1.1 Proof of Theorem 5.1

Let $\mathcal{B}^*_{\{0\}} = \mathcal{S}^*$ and $\mathcal{B}^*_{\{j\}} = \mathcal{B}^*_{\varepsilon_1\ldots,\varepsilon_j}$ for $j = 1, 2, \cdots$. Obviously, we have that $\mathcal{B}^*_{\{0\}} \supset \mathcal{B}^*_{\{1\}} \supset \mathcal{B}^*_{\{2\}} \supset \cdots \supset \mathcal{B}^*_{\{M\}}$. As U_i is generated uniformly from $\mathcal{S}^*$, from the property of Monte Carlo integration (Gilks et al., 1995; Brooks et al., 2011), for any $M \geq m \geq 0$,

$$\begin{aligned}\mathcal{F}(\mathcal{B}^*_{\{m\}}) &= \int_{y\in\mathcal{B}^*_{\{m\}}} f(y)dy \\ &= w_{\mathcal{S}^*} \cdot \frac{1}{n^*}\sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\{m\}})f(U_i) + O_p\left(\frac{1}{\sqrt{n^*}}\right) \qquad \text{(D.1)}\end{aligned}$$

The proof of Theorem 5.1 consists of the following three steps. In steps 1 and 2, we show Theorem 5.1 (1) for the two cases with $\mathcal{F}(\mathcal{B}^*_{\{m\}}) > 0$ and $\mathcal{F}(\mathcal{B}^*_{\{m\}}) = 0$, respectively. In step 3, we present the derivations for Theorem 5.1 (2).

Step 1: We first prove Theorem 5.1 (1) for the case with $\mathcal{F}(\mathcal{B}^*_{\{m\}}) > 0$.

If $\mathcal{F}(\mathcal{B}^*_{\{m\}}) > 0$, then

$$E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] = \prod_{j=1}^{m} \frac{\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_j} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_j})f(u^*_i)}{\sum_{l=0}^{1}\left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(u^*_i)\right]}$$

$$
\begin{aligned}
&= \prod_{j=1}^{m} \frac{\alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_j} + \frac{1}{w_{\mathcal{S}^*}} n^* \mathcal{F}(\mathcal{B}^*_{\{j\}}) + O_p(\sqrt{n^*})}{(\sum_{l=0}^{1} \alpha^{\dagger}_{\varepsilon_1\ldots\varepsilon_{j-1}l}) + \frac{1}{w_{\mathcal{S}^*}} n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \\
&= \prod_{j=1}^{m} \frac{\frac{\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j\}}) + O_p\big(\frac{1}{\sqrt{n^*}}\big)}{\frac{2\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p\big(\frac{1}{\sqrt{n^*}}\big)} \\
&= \prod_{j=1}^{m} \frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})\big[\frac{\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} \frac{1}{\mathcal{F}(\mathcal{B}^*_{\{j\}})} + 1 + O_p\big(\frac{1}{\sqrt{n^*}}\big)\big]}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})\big[\frac{2\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} \frac{1}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} + 1 + O_p\big(\frac{1}{\sqrt{n^*}}\big)\big]} \\
&= \prod_{j=1}^{m} \frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})\big[\phi j^2 \cdot w_{\mathcal{S}^*} \frac{1}{\mathcal{F}(\mathcal{B}^*_{\{j\}})} + n^* + O_p\big(\sqrt{n^*}\big)\big]}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})\big[2\phi j^2 \cdot w_{\mathcal{S}^*} \frac{1}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} + n^* + O_p\big(\sqrt{n^*}\big)\big]} \\
&= \prod_{j=1}^{m} \Bigg[\frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} + \frac{\phi j^2 \cdot w_{\mathcal{S}^*} - 2\phi j^2 \cdot w_{\mathcal{S}^*} \frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})}\Bigg], \qquad \text{(D.2)}
\end{aligned}
$$

where the first equality is from (5), and the second equality is the application of (D.1). Here we also use the default choice $\alpha_{\varepsilon_1\ldots\varepsilon_m} = \phi m^2$ as mentioned in Section 2.1 for prior parameter $\alpha_{\varepsilon_1\ldots\varepsilon_m}$; further, ϕ, $w_{\mathcal{S}^*}$ and $\mathcal{F}(\mathcal{B}^*_{\{j\}})$ for $j = 1, \ldots, m$ are constants with order $O(1)$.

It is obvious that $\frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} \leq 1$. For $r = 1, \ldots, 2^m - 1$, let T_r denote any non-empty subset of $\{1, \ldots, m\}$ and let T_r^c denote its compliment. Then (D.2) becomes

$$
\begin{aligned}
&E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] = \prod_{j=1}^{m} \frac{\mathcal{F}(\mathcal{B}^*_{\{j\}})}{\mathcal{F}(\mathcal{B}^*_{\{j-1\}})} + \\
&\sum_{r=1}^{2^m-1} \Bigg[\prod_{h \in T_r^c} \frac{\mathcal{F}(\mathcal{B}^*_{\{h\}})}{\mathcal{F}(\mathcal{B}^*_{\{h-1\}})}\Bigg]\Bigg[\prod_{q \in T_r} \frac{\phi q^2 \cdot w_{\mathcal{S}^*} - 2\phi q^2 \cdot w_{\mathcal{S}^*} \frac{\mathcal{F}(\mathcal{B}^*_{\{q\}})}{\mathcal{F}(\mathcal{B}^*_{\{q-1\}})} + O_p(\sqrt{n^*})}{2\phi q^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{q-1\}}) + O_p(\sqrt{n^*})}\Bigg] \\
\leq\ & \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + \sum_{r=1}^{2^m-1} \Bigg[\prod_{q \in T_r} \frac{\phi q^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi q^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{q-1\}}) + O_p(\sqrt{n^*})}\Bigg] \qquad \text{(D.3)} \\
=\ & \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + \prod_{j=1}^{m} \Bigg[1 + \frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})}\Bigg] - 1 \qquad \text{(D.4)} \\
=\ & \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + \exp\Bigg\{\sum_{j=1}^{m} \log\Bigg[1 + \frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})}\Bigg]\Bigg\} - 1
\end{aligned}
$$

$$
\begin{aligned}
&= \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + \exp\left\{ \sum_{j=1}^{m} \frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \right. \\
&\quad \left. + \sum_{j=1}^{m} O_p\left[\left(\frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \right)^2\right] \right\} - 1 && \text{(D.5)} \\
&= \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + O_p\left(\sum_{j=1}^{m} \frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \right) && \text{(D.6)} \\
&\leq \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + O_p\left(\sum_{j=1}^{m} \frac{\phi j^2 \cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{n^* \mathcal{F}(\mathcal{B}^*_{\{j-1\}})} \right) && \text{(D.7)} \\
&= \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) + \max\left\{ O_p\left(\frac{M}{\sqrt{n^*}}\right), O_p\left(\frac{M^3}{n^*}\right) \right\}, && \text{(D.8)}
\end{aligned}
$$

where inequality(D.3) is obtained through omitting the negative term $-2\phi q^2 \cdot w_{\mathcal{S}^*} \frac{\mathcal{F}(\mathcal{B}^*_{\{q\}})}{\mathcal{F}(\mathcal{B}^*_{\{q-1\}})}$ in the previous step; equation (D.4) is due to the expansion of the product $\prod_{j=1}^m (1+a_j)$ for a series of scalar a_j with $j = 1, \ldots, m$:

$$
\prod_{j=1}^{m} (1+a_j) = \sum_{r=1}^{2^m-1} \left[\prod_{q \in T_r} a_q \right] + 1;
$$

in deriving equation (D.5) and (D.6), we use the Taylor expansions $log(1+a) = a + O(a)$ and $exp(a) = 1 + O(a)$, and inequality (D.7) is obtained through omitting the terms $2\phi j^2 \cdot w_{\mathcal{S}^*}$ and $O_p(\sqrt{n^*})$ in the denominator of previous step.

Step 2: We now prove Theorem 5.1 (1) for the case with $\mathcal{F}(\mathcal{B}^*_{\{m\}}) = 0$.

If $\mathcal{F}(\mathcal{B}^*_{\{m\}}) = 0$, suppose $l_1 = \max\{i | i < m; \mathcal{F}(\mathcal{B}^*_{\{i\}}) > 0\}$, then similarly

$$
\begin{aligned}
E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] &= \prod_{j=1}^{m} \frac{\alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_j} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_j}) f(U_i)}{\sum_{l=0}^{1} \left[\alpha^{\dagger}_{\varepsilon_1 \ldots \varepsilon_{j-1} l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1 \ldots \varepsilon_{j-1} l}) f(U_i) \right]} \\
&= \prod_{j=1}^{l_1+1} \frac{\frac{\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j\}}) + O_p\left(\frac{1}{\sqrt{n^*}}\right)}{\frac{2\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p\left(\frac{1}{\sqrt{n^*}}\right)} \left(\frac{1}{2}\right)^{m-l_1-1} \\
&\leq \mathcal{F}(\mathcal{B}^*_{l_1+1})/(1-\delta) \left(\frac{1}{2}\right)^{m-l_1-1} + \left(\frac{1}{2}\right)^{m-l_1-1} \max\left\{ O_p\left(\frac{M}{\sqrt{n^*}}\right), O_p\left(\frac{M^3}{n^*}\right) \right\}
\end{aligned}
$$

$$= 0 + \max\left\{O_p\left(\frac{M}{\sqrt{n^*}}\right), O_p\left(\frac{M^3}{n^*}\right)\right\}, \tag{D.9}$$

where (D.9) is obtained by $\mathcal{F}(\mathcal{B}^*_{l_1+1}) = 0$.

Step 3: We now prove Var$\left(\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})\right) = O_p\left(\frac{M}{n^*}\right)$ in Theorem 5.1 (2).

First, we present a fact that for independent Z_1 and Z_2,

$$\begin{aligned}
\mathrm{Var}(Z_1 Z_2) &= E(Z_1^2 Z_2^2) - E^2(Z_1 Z_2) \\
&= E(Z_1^2)E(Z_2^2) - E^2(Z_1)E^2(Z_2) \\
&= [E(Z_1^2)E(Z_2^2) - E^2(Z_1)E(Z_2^2)] + [E^2(Z_1)E(Z_2^2) - E^2(Z_1)E^2(Z_2)] \\
&= \mathrm{Var}(Z_1)E(Z_2^2) + E^2(Z_1)\mathrm{Var}(Z_2). \qquad \text{(D.10)}
\end{aligned}$$

Next, write $\mathcal{G}_j = \mathcal{G}_{\varepsilon_1\ldots\varepsilon_j} \in [0,1]$. By the definition of Polya tree, given $\tilde{U}$, the $\mathcal{G}_j$ are independent. Therefore, applying (D.10) gives that

$$\mathrm{Var}\left(\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})\right) = \mathrm{Var}\left(\prod_{j=1}^m \mathcal{G}_j|\tilde{U}\right)$$

$$= \left[\mathrm{Var}(\mathcal{G}_1|\tilde{U})E\left(\prod_{j=2}^m \mathcal{G}_j^2|\tilde{U}\right) + E(\mathcal{G}_1|\tilde{U})^2\mathrm{Var}\left(\prod_{j=2}^m \mathcal{G}_j|\tilde{U}\right)\right] \tag{D.11}$$

$$\leq \left[\mathrm{Var}(\mathcal{G}_1|\tilde{U}) + \mathrm{Var}\left(\prod_{j=2}^m \mathcal{G}_j|\tilde{U}\right)\right] \tag{D.12}$$

$$\leq \sum_{j=1}^m \mathrm{Var}\left(\mathcal{G}_j|\tilde{U}\right) \tag{D.13}$$

$$= \sum_{j=1}^m \frac{\left(\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_j} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_j})f(U_i)\right)\left\{\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}(1-\varepsilon_j)} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}(1-\varepsilon_j)})f(U_i)\right\}}{\left\{\sum_{l=0}^1 \left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(U_i)\right]\right\}^2}$$

$$\times \frac{1}{\left\{\sum_{l=0}^1 \left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(U_i)\right] + 1\right\}} \tag{D.14}$$

$$\leq \sum_{j=1}^m \frac{1}{\sum_{l=0}^1 \left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l})f(U_i)\right] + 1} \tag{D.15}$$

$$= \sum_{j=1}^{m} \frac{1}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^*\mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \leq \frac{M}{n^*\mathcal{F}(\mathcal{B}^*_{\{m-1\}})} = O_p\left(\frac{M}{n^*}\right) \tag{D.16}$$

where inequality (D.12) is due to the fact that $\mathcal{G}_j|\tilde{U}$ is a probability between 0 and 1 for $j = 1, \ldots, m$. Further, (D.13) can be obtained by repeating the procedure of (D.11) and (D.12) for $\mathcal{G}_1|\tilde{U}$ to $\mathcal{G}_j|\tilde{U}$ for $j = 2, \ldots, m$. (D.14) is obtained from the variance of Beta distributions, as $\mathcal{G}_j|\tilde{U}$ follows a Beta distribution. (D.15) is due to the fact that

$$0 \leq \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_j} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_j}) f(U_i) \leq \sum_{l=0}^{1} \left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l}) f(U_i)\right]$$

$$0 \leq \alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}(1-\varepsilon_j)} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}(1-\varepsilon_j)}) f(U_i) \leq \sum_{l=0}^{1} \left[\alpha^\dagger_{\varepsilon_1\ldots\varepsilon_{j-1}l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_{j-1}l}) f(U_i)\right],$$

and the inequality in (D.16) is due to the fact that

$$\begin{aligned}
&\sum_{j=1}^{m} \frac{1}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^*\mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p(\sqrt{n^*})} \leq \sum_{j=1}^{m} \frac{1}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^*\mathcal{F}(\mathcal{B}^*_{\{m-1\}}) + O_p(\sqrt{n^*})} \\
&\leq \sum_{j=1}^{M} \frac{1}{2\phi j^2 \cdot w_{\mathcal{S}^*} + n^*\mathcal{F}(\mathcal{B}^*_{\{m-1\}}) + O_p(\sqrt{n^*})} \\
&\leq \sum_{j=1}^{M} \frac{1}{n^*\mathcal{F}(\mathcal{B}^*_{\{m-1\}})} = \frac{M}{n^*\mathcal{F}(\mathcal{B}^*_{\{m-1\}})}
\end{aligned}$$

where we use the fact that $\phi > 0$.

Finally, for any measurable set $B \in \pi^*_m$ with $m = 1, \ldots, M$, if we consider $n^* = O(M^{3+\eta})$ with $\eta > 0$, as $M \to \infty$, we have that

(1) by (D.8) and (D.9),

$$E\left[\mathcal{G}_{\tilde{U}}(B)\right] - \mathcal{F}(B)/(1-\delta) = \max\left\{O_p\left(\frac{M}{\sqrt{n^*}}\right), O_p\left(\frac{M^3}{n^*}\right)\right\} \xrightarrow{p} 0;$$

(2) by (D.16),

$$\mathrm{Var}\left[\mathcal{G}_{\tilde{U}}(B)\right] = O_p\left(\frac{M}{n^*}\right) \xrightarrow{p} 0;$$

(3) by Chebyshev's Inequality,

$$P\left(\left|\mathcal{G}_{\tilde{U}}(B) - \mathcal{F}(B)/(1-\delta)\right| \geq \epsilon\right) \leq \frac{E\left[\mathcal{G}_{\tilde{U}}(B) - \mathcal{F}(B)/(1-\delta)\right]^2 + \mathrm{Var}\left[\mathcal{G}_{\tilde{U}}(B)\right]}{\epsilon^2}$$
$$= \max\left\{O_p\left(\frac{M^2}{n^*}\right), O_p\left(\frac{M^6}{[n^*]^2}\right)\right\} \xrightarrow{p} 0.$$

D.1.2 Proof of Theorem 5.2

In Theorem 5.1, we have proved the result for any $\mathcal{B}^*_{\{m\}} = \mathcal{B}^*_{\varepsilon_1 \dots, \varepsilon_m}$ with $m \leq M$. Now we consider the case with $m > M$.

We first show the existence of $\gamma(M)$. Since $\mathcal{F}/(1-\delta)$ is an appropriate probability measure with a continuous density function on $\mathcal{S}^*$ and the number of the subset $\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M}$, 2^M, is finite, there exists a subspace $\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M}$ such that $\mathcal{F}(\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M})/(1-\delta) > 0$ and $\gamma(M) = \underset{B \in \mathfrak{V}}{min} \mathcal{F}(B)/(1-\delta)$ exists and is greater than zero, where $\mathfrak{V} = \{\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M} | \mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M} \in \pi^*_M; \mathcal{F}(\mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_M}) > 0\}$ is defined in Theorem 5.2 on the main text. Let $\Omega_M = \{\mathcal{B}^*_{\{m\}} : \mathcal{B}^*_{\{m\}} \in \pi_m; m > M\}$.

The proof of Theorem 5.2 consists of four steps. In steps 1 and 2, we show Theorem 5.2 (1) for two cases with $\mathcal{F}(\mathcal{B}^*_m) > 0$ and $\mathcal{F}(\mathcal{B}^*_m) = 0$, respectively. In step 3, we derive Theorem 5.2 (2), and in step 4, we prove Theorem 5.2 (3).

Step 1: We first prove Theorem 5.2 (1) by finding $\sup_{\mathcal{B}^*_{\{m\}} \in \Omega_M} \left| E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] - \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta) \right|$ **for the case that** $\mathcal{F}(\mathcal{B}^*_{\{m\}}) > 0$.

If $\mathcal{F}(\mathcal{B}^*_{\{m\}}) > 0$, then

$$E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] = \prod_{j=1}^{M} \frac{\alpha^\dagger_{\varepsilon_1 \dots \varepsilon_j} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_j}) f(U_i)}{\sum_{l=0}^{1} \left[\alpha^\dagger_{\varepsilon_1 \dots \varepsilon_{j-1} l} + \sum_{i=1}^{n^*} I(U_i \in \mathcal{B}^*_{\varepsilon_1 \dots \varepsilon_{j-1} l}) f(U_i)\right]} \left[\prod_{j=M+1}^{m} \frac{1}{2}\right]$$
$$= \left(\frac{1}{2}\right)^{m-M} \prod_{j=1}^{M} \frac{\frac{\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j\}}) + O_p\left(\frac{1}{\sqrt{n^*}}\right)}{\frac{2\phi j^2}{n^*} \cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p\left(\frac{1}{\sqrt{n^*}}\right)}$$

$$
\begin{aligned}
&\leq \left(\frac{1}{2}\right)^{m-M}\left[\mathcal{F}(\mathcal{B}^*_{\{M\}})/(1-\delta)+O_p\left(\sum_{j=1}^{M}\frac{\phi j^2\cdot w_{\mathcal{S}^*}+O_p(\sqrt{n^*})}{n^*\mathcal{F}(\mathcal{B}^*_{\{j-1\}})}\right)\right]\\
&\leq \left(\frac{1}{2}\right)^{m-M}\left[\mathcal{F}(\mathcal{B}^*_{\{M\}})/(1-\delta)+\max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right),O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}\right]
\end{aligned} \tag{D.17}
$$

Since $\mathcal{F}$ has an absolute continuous density on $\mathcal{S}^*$, then

$$
\begin{aligned}
&\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left|E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})]-\mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta)\right|\\
\leq\;& \sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left|\left(\frac{1}{2}\right)^{m-M}\mathcal{F}(\mathcal{B}^*_{\{M\}})/(1-\delta)-\mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta)+\max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right),O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}\right|\\
\leq\;& \sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left|\left(\frac{1}{2}\right)^{m-M}\mathcal{F}(\mathcal{B}^*_{\{M\}})/(1-\delta)-\mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta)\right|+\max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right),O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}\\
\leq\;& \left(\frac{1}{2}\right)^{m}\frac{1}{1-\delta}\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\sup_{\substack{y_1\in\mathcal{B}^*_{\{M\}}\\ y_2\in\mathcal{B}^*_{\{m\}}}}\left|f(y_1)-f(y_2)\right|\right\}+\max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right),O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}
\end{aligned} \tag{D.18}
$$

where the first inequality is the application of (D.17) and the second inequality holds by the absolute value inequality, and the last inequality holds due to the fact that

$$
\begin{aligned}
\left(\frac{1}{2}\right)^{M}\inf_{y\in\mathcal{B}^*_{\{M\}}}f(y)\leq\mathcal{F}(\mathcal{B}^*_{\{M\}})\leq\left(\frac{1}{2}\right)^{M}\sup_{y\in\mathcal{B}^*_{\{M\}}}f(y)\\
\left(\frac{1}{2}\right)^{m}\inf_{y\in\mathcal{B}^*_{\{m\}}}f(y)\leq\mathcal{F}(\mathcal{B}^*_{\{m\}})\leq\left(\frac{1}{2}\right)^{m}\sup_{y\in\mathcal{B}^*_{\{m\}}}f(y).
\end{aligned}
$$

Let $\sup_{y\in\mathcal{B}^*_{\{M\}}}|f'(y)|$ represent the supreme value of derivative $f(y)$ in the subset $\mathcal{B}^*_{\{M\}}$, then from the mean value theorem, (D.18) becomes

$$
\begin{aligned}
&\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left|E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})]-\mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta)\right|\\
&\leq\left(\frac{1}{2}\right)^{m}\left(\frac{1}{2}\right)^{M}\frac{1}{1-\delta}\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\sup_{y\in\mathcal{B}^*_{\{M\}}}|f'(y)|\right\}+\max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right),O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}
\end{aligned}
$$

$$= \max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right), O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\}. \tag{D.19}$$

where $\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\sup_{y\in\mathcal{B}^*_{\{M\}}}|f'(y)|\right\}$ is bounded due to the fact that f is absolute continuous on $\mathcal{S}^*$, and $\left(\frac{1}{2}\right)^m\left(\frac{1}{2}\right)^M \leq \left(\frac{1}{2}\right)^{2M} < \frac{1}{M^{3+\eta}} = O_p(\frac{1}{n^*})$. We note that $\mathcal{B}^*_{\{m\}} \subset \mathcal{B}^*_{\{M\}}$.

Step 2: We prove Theorem 5.2 (1) for the case that $\mathcal{F}(\mathcal{B}^*_{\{m\}}) = 0$.

If $\mathcal{F}(\mathcal{B}^*_m) = 0$, suppose $l_1 = \max_{i<m}\{\mathcal{F}(\mathcal{B}^*_i) > 0\}$, then

$$\begin{aligned}
\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M} E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] &= \sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\left(\frac{1}{2}\right)^{m-l_1-1}\prod_{j=1}^{l_1+1}\frac{\frac{\phi j^2}{n^*}\cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_j) + O_p\left(\frac{1}{\sqrt{n^*}}\right)}{\frac{2\phi j^2}{n^*}\cdot w_{\mathcal{S}^*} + \mathcal{F}(\mathcal{B}^*_{\{j-1\}}) + O_p\left(\frac{1}{\sqrt{n^*}}\right)}\right\} \\
&\leq \sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\left(\frac{1}{2}\right)^{m-l_1-1}\left[\mathcal{F}(\mathcal{B}^*_{l_1+1})/(1-\delta) + O\left(\sum_{j=1}^{m}\frac{\phi j^2\cdot w_{\mathcal{S}^*} + O_p(\sqrt{n^*})}{n^*\mathcal{F}(\mathcal{B}^*_{\{j-1\}})}\right)\right]\right\} \\
&= 0 + \max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right), O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\} \qquad \text{(D.20)}
\end{aligned}$$

In Theorem 5.1, we have proved that for any measurable set $B \in \pi^*_m$ with $m = 1, \ldots, M$, $E\left[\mathcal{G}_{\tilde{U}}(B)\right] - \mathcal{F}(B)/(1-\delta) \xrightarrow{p} 0$ as $M \to \infty$. By (D.19) and (D.20), we prove that for $m > M$,

$$\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left|E[\mathcal{G}_{\tilde{U}}(\mathcal{B}^*_{\{m\}})] - \mathcal{F}(\mathcal{B}^*_{\{m\}})/(1-\delta)\right| = \max\left\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right), O_p\left(\frac{M^3}{n^*\gamma(M)}\right)\right\} \xrightarrow{p} 0$$

Further, $\bigcup_{m=1}^M \pi^*_m \bigcup \Omega_M$ contains all measurable subsets of $\mathcal{S}^*$, which is essentially equal to $\mathfrak{S}^*$ defined in Theorem 5.2 in the main text. As a result, we have

$$\sup_{B\in\mathfrak{S}^*}\left|E(\mathcal{G}_{\tilde{U}}) - \mathcal{F}/(1-\delta)\right| \xrightarrow{p} 0.$$

Step 3: We prove Theorem 5.2 (2) by finding the supreme of Var$\left(\mathcal{G}_{\tilde{U}}(B)\right)$.

From the inequality (D.16),

$$\sup_{B\in\mathfrak{S}^*}\text{Var}\Big(\mathcal{G}_{\tilde{U}}(B)\Big)\leq\sup_{\mathcal{B}^*_{\{M\}}\in\pi^*_m}O_p\Big(\frac{M}{n^*\mathcal{F}(\mathcal{B}^*_{\{M\}})}\Big)\leq O_p\Big(\frac{M}{n^*\gamma(M)}\Big).$$

where the last inequality holds due to the definition that $\gamma(M)=\underset{B\in\mathfrak{V}}{min}\mathcal{F}(B)/(1-\delta)$.

Step 4: We prove the results of Theorem 5.2 (3).

Let $I_M^{\Delta}=\{\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M}:\exists y\in\mathcal{B}^*_{\varepsilon_1\ldots\varepsilon_M},f(y)/(1-\delta)<\Delta\}$, and $J_M^{\Delta}=\bigcup_{B\in I_M^{\Delta}}B$. Let $S^*\backslash J_M^{\Delta}$ denote the compliment set of J_M^{Δ}. Therefore, $\inf_{y\in S^*\backslash J_M^{\Delta}}f(y)/(1-\delta)\geq\Delta$. Let $T=w_{\mathcal{S}^*}$. Since $\mathcal{F}$ is differentiable on $\mathcal{S}^*$, $\forall\epsilon>0$, for M large enough, $\forall B\in\pi_M$, and $y_1,y_2\in B$, we have $|\frac{f(y_1)}{1-\delta}-\frac{f(y_2)}{1-\delta}|\leq\frac{\epsilon}{8T}$. Selecting $\Delta=\epsilon/(4T)$, it is obvious that $\epsilon/(4T)>\sup_{y\in J_M^{\epsilon/(8T)}}f(y)/(1-\delta)$. Then

$$\begin{aligned}D\Big(\mathcal{G}_{\tilde{U}},\mathcal{F}/(1-\delta)\Big)&=\int_{\mathcal{S}^*}\Big|g_{\tilde{U}}(y)-f(y)/(1-\delta)\Big|dy\\&=\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}\Big|g_{\tilde{U}}(y)-\frac{f(y)}{1-\delta}\Big|dy+\int_{J_M^{\epsilon/(8T)}}\Big|g_{\tilde{U}}(y)-\frac{f(y)}{1-\delta}\Big|dy\\&\triangleq K_1+K_2.\end{aligned}$$

Further,

$$\begin{aligned}K_2&=\int_{J_M^{\epsilon/(8T)}}\Big|g(y)|\tilde{U}-\frac{f(y)}{1-\delta}\Big|dy\leq\int_{J_M^{\epsilon/(8T)}}g_{\tilde{U}}(y)dy+\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy\\&=1-\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}g_{\tilde{U}}(y)dy+\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy\\&=\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy+\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy-\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}g_{\tilde{U}}(y)dy+\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy\\&\leq\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}\Big|g_{\tilde{U}}(y)-\frac{f(y)}{1-\delta}\Big|dy+2\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy.\end{aligned}$$

Therefore,

$$D\Big(\mathcal{G}_{\tilde{U}},\mathcal{F}/(1-\delta)\Big)\ \leq\ 2\int_{\mathcal{S}^*\backslash J_M^{\epsilon/(8T)}}\Big|g_{\tilde{U}}(y)-\frac{f(y)}{1-\delta}\Big|dy+2\int_{J_M^{\epsilon/(8T)}}\frac{f(y)}{1-\delta}dy$$

$$\leq \quad 2\int_{\mathcal{S}^*\setminus J_M^{\epsilon/(8T)}} \left| g_{\tilde{U}}(y) - \frac{f(y)}{1-\delta} \right| dy + 2 \cdot w_{J_M^{\epsilon/(8T)}} \cdot \epsilon/(4T) \quad \text{(D.21)}$$

$$\leq \quad 2\int_{\mathcal{S}^*\setminus J_M^{\epsilon/(8T)}} \left| g_{\tilde{U}}(y) - \frac{f(y)}{1-\delta} \right| dy + \frac{\epsilon}{2} = 2K_1 + \frac{\epsilon}{2}, \quad \text{(D.22)}$$

where $w_{J_M^{\epsilon/(8T)}}$ denotes the total volume of the subsets in $J_M^{\epsilon/(8T)}$, (D.21) is obtained by the fact that $\epsilon/(4T) > \sup_{y \in J_M^{\epsilon/(8T)}} f(y)/(1-\delta)$ and (D.22) is due to the fact that $w_{J_M^{\epsilon/(8T)}} < w_{\mathcal{S}^*} = T$. For K_1,

$$\begin{aligned} K_1 &= \int_{\mathcal{S}^*\setminus J_M^{\epsilon/(8T)}} \left| g_{\tilde{U}}(y) - \frac{f(y)}{1-\delta} \right| dy \\ &= \int_{\mathcal{S}^*\setminus J_M^{\epsilon/(8T)}} \left| 2^M/T\mathcal{G}_{\tilde{U}}(B_y) - \frac{f(y)}{1-\delta} + \frac{f(b_y)}{1-\delta} - \frac{f(b_y)}{1-\delta} \right| dy \quad \text{(D.23)} \\ &\leq \sum_{B \in \pi_M^* \setminus I_M^{\epsilon/(8T)}} \left\{ \left| \mathcal{G}_{\tilde{U}}(B) - \frac{\mathcal{F}(B)}{1-\delta} \right| + \int_B \left| \frac{f(b_y)}{1-\delta} - \frac{f(y)}{1-\delta} \right| dy \right\}, \quad \text{(D.24)} \end{aligned}$$

where b_y satisfies the fact that $\mathcal{F}(B_y) = f(b_y)w_{B_y}$ with $B_y \in \pi_M^*$ being the subset that the point y belongs to. (D.23) is due to the fact that y is uniformly distributed on $\mathcal{B}_{\{M\}}$ in PTMC for any $\mathcal{B}_{\{M\}} \in \pi_M$ so that $g_{\tilde{U}}(y) = 2^M/T\mathcal{G}_{\tilde{U}}(B_y)$.

Since $\forall B \in \pi_M^*$ and $b_y, y \in B$, we have $|\frac{f(b_y)}{1-\delta} - \frac{f(y)}{1-\delta}| \leq \epsilon/(8T)$. Then

$$\sum_{B \in \pi_M^* \setminus I_M^{\epsilon/(8T)}} \left\{ \int_B \left| \frac{f(b_y)}{1-\delta} - \frac{f(y)}{1-\delta} \right| dy \right\} \leq \epsilon/8. \quad \text{(D.25)}$$

By (D.22), (D.24) and (D.25), we can get

$$\begin{aligned} D\Big(\mathcal{G}_{\tilde{U}}, \mathcal{F}/(1-\delta)\Big) &= 2K_1 + \frac{\epsilon}{2} \\ &\leq 2\sum_{B \in \pi_M^* \setminus I_M^{\epsilon/(8T)}} \left| \mathcal{G}_{\tilde{U}}(B) - \frac{\mathcal{F}(B)}{1-\delta} \right| + \frac{3}{4}\epsilon \end{aligned}$$

Then $P\Big(D\Big(\mathcal{G}_{\tilde{U}}, \mathcal{F}/(1-\delta)\Big) > \epsilon\Big) \leq P\Big(\sum_{B \in \pi_M^* \setminus I_M^{\epsilon/(8T)}} \left| \mathcal{G}_{\tilde{U}}(B) - \frac{\mathcal{F}(B)}{1-\delta} \right| > \frac{\epsilon}{8}\Big)$. Now we consider the second probability,

$$P\Big(\sum_{B \in \pi_M^* \setminus I_M^{\epsilon/(8T)}} \left| \mathcal{G}_{\tilde{U}}(B) - \frac{\mathcal{F}(B)}{1-\delta} \right| > \frac{\epsilon}{8}\Big)$$

$$
\begin{aligned}
&\leq P\Bigg(\max_{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}}\left|\mathcal{G}_{\tilde{U}}(B)-\frac{\mathcal{F}(B)}{1-\delta}\right|\geq\frac{\epsilon}{2^{M+3}}\Bigg) && \text{(D.26)}\\
&\leq P\Bigg(\bigcup_{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}}\left[\left|\mathcal{G}_{\tilde{U}}(B)-\frac{\mathcal{F}(B)}{1-\delta}\right|\geq\frac{\epsilon}{2^{M+3}}\right]\Bigg) && \text{(D.27)}\\
&\leq \sum_{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}} P\Bigg(\left|\mathcal{G}_{\tilde{U}}(B)-\frac{\mathcal{F}(B)}{1-\delta}\right|\geq\frac{\epsilon}{2^{M+3}}\Bigg)\\
&\leq 2^M\left(\frac{2^{M+3}}{\epsilon}\right)^2\Bigg\{\sup_{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}}\left|E[\mathcal{G}_{\tilde{U}}(B)]-\frac{\mathcal{F}(B)}{1-\delta}\right|^2+\sup_{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}}\mathrm{Var}[\mathcal{G}_{\tilde{U}}(B)]\Bigg\} && \text{(D.28)}\\
&= 2^{3M}\max\Bigg\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right)^2, O_p\left(\frac{M^3}{n^*\gamma(M)}\right)^2, O_p\left(\frac{M}{n^*\gamma(M)}\right)\Bigg\}, && \text{(D.29)}
\end{aligned}
$$

where (D.26) and (D.27) are obtained by the fact that for a series of scalars a_j with $j=1,\ldots,m$,

$$
\Big\{\sum_{j=1}^m a_j>\epsilon/4\Big\}\subset\Big\{\max_j a_j\geq\frac{\epsilon}{4m}\Big\}\subset\bigcup_{j=1}^m\Big\{a_j\geq\frac{\epsilon}{4m}\Big\},
$$

(D.28) are derived by applying Chebyshev's inequality, and (D.29) can be obtained by applying the results of step 1-3. In (D.26), since $\pi_M^*\setminus I_M^{\epsilon/(8T)}$ is set with finite element, the maximum value exists.

Since in (D.29), $\gamma(M)=\underset{B\in\pi_M^*\setminus I_M^{\epsilon/(8T)}\bigcap\mathfrak{B}}{min}\mathcal{F}(B)/(1-\delta)$, and by the definition of I_M^Δ and J_M^Δ, $\inf_{y\in S^*\setminus J_M^{\epsilon/(8T)}} f(y)/(1-\delta)\geq\epsilon/(8T)$, we can get $\gamma(M)\geq\epsilon/(8T)*w_{\mathcal{B}^*_{\{M\}}}=\epsilon/(8T)*\frac{T}{2^M}=\epsilon/2^{M+3}$, where $w_{\mathcal{B}^*_{\{M\}}}$ is volume of $\mathcal{B}^*_{\{M\}}$.

Then, with $n^*\propto O_p(2^{5M}M^{3+\eta})$, as $M\to\infty$, we could get

$$
\begin{aligned}
&P\Big(D\big(\mathcal{G}_{\tilde{U}},\mathcal{F}/(1-\delta)\big)\geq\epsilon\Big)\\
&\leq 2^{3M}\max\Bigg\{O_p\left(\frac{M}{\sqrt{n^*}\gamma(M)}\right)^2, O_p\left(\frac{M^3}{n^*\gamma(M)}\right)^2, O_p\left(\frac{M}{n^*\gamma(M)}\right)\Bigg\}\\
&\leq 2^{3M}\max\Bigg\{O_p\left(\frac{M^2 2^{2M+6}}{n^*}\right), O_p\left(\frac{M^6 2^{2M+6}}{[n^*]^2}\right), O_p\left(\frac{M 2^{2M+6}}{n^*}\right)\Bigg\}
\end{aligned}
$$

$$= O_p\left(\frac{1}{M^{\eta}}\right) \xrightarrow{p} 0$$

D.2 Additional Simulation Results

D.2.1 Simulation Results of Setting 5.1

Table D.1: Simulation results for the Dog bowl distribution

Algorithm	n	Dimension	2.5%	Median	97.5%	ESS	CT (in seconds)
Numerical Approximation	-	y_1	-10.630	0.000	10.630	-	-
		y_2	-10.630	0.000	10.630	-	
PTMC	5000	y_1	-10.628	-0.019	10.626	5000	0.507
		y_2	-10.635	0.008	10.637	5000	
PTMC Gibbs Sampler	5000	y_1	-10.638	-0.007	10.634	5000	66.500
		y_2	-10.632	0.007	10.631	5000	
PTMC MH	5000	y_1	-10.628	0.011	10.636	1666	33.663
		y_2	-10.633	-0.017	10.628	1667	
MCMC (big stepsize)	5000	y_1	-10.054	0.345	10.304	9	0.046
		y_2	-10.229	0.129	10.228	9	
MCMC (small stepsize)	5000	y_1	-2.880	3.390	8.847	9	0.042
		y_2	-2.517	3.702	9.038	9	
LMC (adaptive stepsize)	5000	y_1	-3.765	0.000	3.592	7	0.072
		y_2	-3.989	-0.269	3.390	7	
LMC (cyclical stepsize)	5000	y_1	-8.436	-0.028	8.341	15	0.124
		y_2	-8.531	0.156	8.504	16	

Table D.2: Simulation results for 25-normal mixture distribution

Algorithm	n	Dimension	2.5%	Median	97.5%	ESS	CT (in seconds)
Numerical Approximation	-	y_1	-4.200	0	4.200	-	-
		y_2	-4.200	0	4.200	-	
PTMC	5000	y_1	-4.198	0.001	4.200	5000	0.998
		y_2	-4.199	-0.004	4.199	5000	
PTMC Gibbs Sampler	5000	y_1	-4.200	-0.001	4.199	5000	493.080
		y_2	-4.199	0.000	4.199	5000	
PTMC MH	5000	y_1	-4.200	0.002	4.199	890	214.777
		y_2	-4.199	-0.002	4.199	890	
MCMC (big stepsize)	5000	y_1	-3.735	0.051	3.841	6	0.582
		y_2	-3.650	0.173	3.823	7	
MCMC (small stepsize)	5000	y_1	1.658	2.000	2.342	579	0.571
		y_2	1.654	1.995	2.337	576	
LMC (adaptive stepsize)	5000	y_1	-0.338	0.001	0.339	44	1.417
		y_2	-0.341	-0.001	0.342	44	
LMC (cyclical stepsize)	5000	y_1	-2.430	0.067	2.606	25	3.100
		y_2	-2.558	0.153	2.434	25	

Table D.3: Simulation results for 5-normal mixture distribution

Algorithm	n	Dimension	2.5%	Median	97.5%	ESS	CT (in seconds)
Numerical Approximation	-	y_1	-4.200	0.000	4.200	-	-
		y_2	-4.200	0.000	4.200	-	
PTMC	5000	y_1	-4.188	-0.030	4.189	5000	0.662
		y_2	-4.188	-0.027	4.189	5000	
PTMC Gibbs Sampler	5000	y_1	-0.342	0.000	0.341	5000	122.443
		y_2	-0.342	0.000	0.342	5000	
PTMC MH	5000	y_1	-4.199	0.000	4.198	97	62.409
		y_2	-4.198	-0.001	4.199	97	
MCMC (big stepsize)	5000	y_1	0.912	1.905	2.883	137	0.141
		y_2	0.910	1.904	2.883	135	
MCMC (small stepsize)	5000	y_1	1.662	2.000	2.339	580	0.143
		y_2	1.661	2.000	2.338	580	
LMC (adaptive stepsize)	5000	y_1	-0.361	0.001	0.363	36	0.326
		y_2	-0.365	-0.001	0.365	36	
LMC (cyclical stepsize)	5000	y_1	-2.399	-0.007	2.348	9	0.660
		y_2	-2.396	-0.008	2.347	8	

D.2.2 Simulation Results of Setting 5.2

Simulation Results of Setting 5.2.1

Table D.4: Simulation results for one-dimensional distribution

	n		PTMC Algorithm 5.1 (n^*=500)							PTMC Algorithm 5.1 (n^*=1000)							MCMC MH					
			EBias*[1]	ESE	ASE	ECP	ESS	CT[2]	RCT[3]	EBias*[1]	ESE	ASE	ECP	ESS	CT[2]	RCT[3]	EBias*[1]	ESE	ASE	ECP	ESS	CT[2]
Geometric distribution	5000	β_1	-0.978	0.017	0.018	0.964	5000	0.015	10.867	-0.184	0.016	0.018	0.964	5000	0.027	6.037	-0.611	0.017	0.018	0.962	280	0.163
	8000	β_1	-0.885	0.017	0.018	0.964	7988	0.015	16.133	-0.123	0.016	0.018	0.964	8000	0.028	8.571	-1.073	0.017	0.018	0.960	447	0.242
Poisson distribution	5000	β_1	-0.098	0.086	0.086	0.954	5000	0.020	10.550	-0.099	0.085	0.087	0.958	5000	0.036	5.861	-0.102	0.084	0.086	0.966	288	0.213
	8000	β_1	-0.090	0.086	0.086	0.952	8000	0.020	15.900	-0.099	0.085	0.087	0.962	8000	0.037	8.540	-0.148	0.085	0.086	0.964	459	0.318
Gaussian copula	5000	β_1	-0.394	0.035	0.034	0.952	5000	0.125	11.960	-0.375	0.035	0.034	0.960	5000	0.246	6.077	-0.375	0.035	0.034	0.958	921	1.495
	8000	β_1	-0.394	0.035	0.034	0.954	8000	0.126	18.127	-0.378	0.035	0.034	0.960	8000	0.245	9.322	-0.371	0.035	0.034	0.958	1476	2.284
Clayton copula	5000	β_1	0.014	0.182	0.179	0.942	5000	0.150	10.987	0.014	0.181	0.179	0.938	5000	0.248	6.645	0.013	0.182	0.179	0.936	219	1.648
	8000	β_1	0.014	0.183	0.179	0.944	8000	0.150	18.127	0.014	0.181	0.179	0.938	8000	0.242	10.393	0.014	0.182	0.179	0.938	344	2.515

[1] For Geometric distribution, EBias*=EBias$\times 10^4$. For Poisson distribution and Gaussian copula, EBias*=EBias$\times 10^2$, and for Clayton copula, EBias*=EBias;

[2] CTs are in minutes;

[3] RCT=$\frac{\text{CT of MCMC MH}}{\text{CT of PTMC Algorithm 5.1 with } n^*}$.

Table D.5: Simulation results for two-dimensional distribution

	n		PTMC Algorithm 5.1 (n^*=1000)							PTMC Algorithm 5.1 (n^*=2000)							MCMC MH					
			EBias	ESE	ASE	ECP	ESS	CT[1]	RCT[2]	EBias	ESE	ASE	ECP	ESS	CT[1]	RCT[2]	EBias	ESE	ASE	ECP	ESS	CT[1]
Beta distribution	5000	β_1	0.058	0.205	0.204	0.940	5000	0.276	4.388	0.058	0.202	0.207	0.946	4994	0.421	2.876	0.053	0.207	0.202	0.946	189	1.211
		β_2	0.078	0.277	0.279	0.948	5000			0.078	0.272	0.281	0.960	4987			0.072	0.268	0.274	0.954	170	
	8000	β_1	0.058	0.205	0.204	0.940	8000	0.270	6.744	0.058	0.202	0.207	0.948	8000	0.422	4.251	0.055	0.201	0.203	0.948	293	1.821
		β_2	0.078	0.277	0.279	0.948	8000			0.078	0.271	0.281	0.960	8000			0.074	0.270	0.276	0.958	268	
Gamma distribution	5000	β_1	0.043	0.221	0.196	0.900	5000	0.771	0.678	0.043	0.212	0.202	0.934	5000	0.870	0.601	0.040	0.211	0.199	0.942	143	0.523
		β_2	0.058	0.317	0.285	0.894	5000			0.060	0.304	0.292	0.932	5000			0.055	0.299	0.288	0.944	131	
	8000	β_1	0.043	0.221	0.196	0.896	8000	0.783	0.983	0.043	0.212	0.202	0.936	7999	0.896	0.859	0.041	0.211	0.199	0.930	223	0.770
		β_2	0.059	0.317	0.285	0.894	7995			0.060	0.304	0.292	0.934	8000			0.055	0.301	0.288	0.944	208	
Joe-Gumbel copula	5000	β_1	0.109	0.576	0.619	0.938	5000	1.618	2.279	0.113	0.574	0.621	0.940	5000	1.668	2.210	0.118	0.580	0.621	0.930	88	3.687
		β_2	0.090	0.644	0.666	0.936	5000			0.085	0.643	0.667	0.936	5000			0.083	0.646	0.669	0.926	87	
	8000	β_1	0.109	0.576	0.619	0.938	8000	1.609	3.403	0.113	0.574	0.621	0.938	7995	1.683	3.253	0.117	0.578	0.620	0.928	138	5.463
		β_2	0.090	0.644	0.666	0.934	8000			0.085	0.643	0.667	0.936	8000			0.082	0.642	0.668	0.938	138	
Clayton-Gumbel copula	5000	β_1	0.060	0.394	0.408	0.940	5000	0.546	4.625	0.059	0.392	0.410	0.950	5000	0.655	3.855	0.060	0.396	0.407	0.940	124	2.525
		β_2	0.019	0.369	0.370	0.938	5000			0.018	0.367	0.371	0.940	5000			0.018	0.368	0.371	0.940	129	
	8000	β_1	0.060	0.394	0.407	0.944	7999	0.512	7.398	0.059	0.392	0.410	0.944	8000	0.632	5.994	0.060	0.393	0.409	0.946	195	3.788
		β_2	0.019	0.369	0.370	0.942	7993			0.018	0.367	0.372	0.942	8000			0.017	0.365	0.372	0.942	203	
Joe-Clayton copula	5000	β_1	0.031	0.198	0.205	0.952	5000	0.492	5.055	0.030	0.196	0.207	0.952	5000	0.582	4.273	0.030	0.197	0.206	0.952	193	2.487
		β_2	0.037	0.346	0.329	0.928	5000			0.034	0.341	0.330	0.930	5000			0.030	0.337	0.329	0.932	289	
	8000	β_1	0.031	0.198	0.205	0.954	8000	0.471	7.915	0.030	0.196	0.207	0.948	7990	0.551	6.766	0.030	0.197	0.206	0.946	308	3.728
		β_2	0.037	0.346	0.329	0.932	8000			0.035	0.341	0.330	0.930	8000			0.031	0.337	0.329	0.932	461	
Tawn Type I copula	5000	β_1	0.040	0.331	0.302	0.930	5000	0.546	15.577	0.036	0.328	0.303	0.938	5000	0.892	9.535	0.035	0.323	0.304	0.946	280	8.505
		β_2	< 0.001	0.022	0.022	0.946	5000			< 0.001	0.022	0.022	0.946	5000			< 0.001	0.022	0.022	0.950	486	
	8000	β_1	0.040	0.331	0.302	0.930	7997	0.661	19.345	0.037	0.328	0.303	0.938	7981	0.904	14.145	0.034	0.323	0.302	0.954	450	12.787
		β_2	< 0.001	0.022	0.022	0.948	8000			< 0.001	0.022	0.022	0.946	8000			< 0.001	0.021	0.022	0.950	774	
Tawn Type II copula	5000	β_1	0.066	0.319	0.299	0.924	4992	1.883	4.554	0.070	0.315	0.302	0.928	5000	2.096	4.083	0.070	0.309	0.299	0.940	278	8.557
		β_2	< 0.001	0.023	0.021	0.928	4995			0.000	0.023	0.021	0.930	5000			< 0.001	0.022	0.021	0.946	489	
	8000	β_1	0.066	0.319	0.299	0.924	7988	1.854	6.933	0.070	0.315	0.302	0.930	8000	2.037	6.310	0.068	0.319	0.304	0.938	442	12.854
		β_2	0.000	0.023	0.021	0.930	8000			0.000	0.023	0.021	0.934	8000			0.000	0.023	0.021	0.944	774	

[1] CTs are in minutes;
[2] RCT=$\frac{\text{CT of MCMC MH}}{\text{CT of PTMC Algorithm 5.1 with } n^*}$.

Simulation Results for Setting 5.2.2

Table D.6: Simulation results for Gamma-normal mixture distribution

	PTMC Algorithm 5.1 ($n^* = 5,000,000$)											
	EBias	ESE	ASE	ECP	ESS	CT						
β_1	-0.008	0.026	0.028	0.950	5000							
β_2	0.238	0.563	0.486	0.930	4998							
β_3	0.121	0.271	0.253	0.940	5000	1.553						
β_4	0.091	0.296	0.279	0.960	5000							
β_5	0.096	0.238	0.221	0.930	5000							
	PTMC Gibbs Sampler						PTMC MH					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	-0.008	0.026	0.029	0.950	2975		-0.007	0.026	0.028	0.950	878	
β_2	0.236	0.570	0.513	0.930	258		0.233	0.568	0.482	0.930	29	
β_3	0.121	0.275	0.268	0.950	257	16.862	0.119	0.274	0.251	0.930	32	2.887
β_4	0.088	0.294	0.285	0.950	2459		0.083	0.297	0.278	0.940	143	
β_5	0.098	0.234	0.224	0.920	2287		0.094	0.233	0.219	0.920	177	
	MCMC(0)						MCMC(500)					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	-0.009	0.026	0.031	0.960	570		-0.007	0.026	0.028	0.950	3137	
β_2	0.334	0.582	0.741	0.960	42		0.238	0.579	0.487	0.920	259	
β_3	0.173	0.285	0.391	0.960	42	0.076	0.121	0.279	0.253	0.930	258	17.345
β_4	0.106	0.298	0.312	0.940	478		0.085	0.293	0.279	0.940	2516	
β_5	0.111	0.234	0.242	0.940	449		0.095	0.233	0.220	0.920	2355	

Table D.7: Simulation results for D-Vine

	PTMC Algorithm 5.2 ($n^* = 1,500,000$)						PTMC Algorithm 5.2 ($n^* = 2,500,000$)					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	0.027	0.207	0.156	0.810	5000	19.302	0.021	0.188	0.153	0.860	4992	27.978
β_2	0.004	0.087	0.065	0.810	5000		0.005	0.077	0.066	0.890	5000	
β_3	0.004	0.144	0.124	0.860	5000		0.006	0.130	0.124	0.890	5000	
β_4	-0.005	0.024	0.021	0.850	4995		-0.004	0.022	0.022	0.880	4996	
β_5	0.021	0.159	0.146	0.850	5000		0.016	0.161	0.148	0.880	5000	
β_6	-0.007	0.038	0.031	0.820	4995		-0.003	0.037	0.031	0.880	4983	

	PTMC Algorithm 5.2 ($n^* = 5,000,000$)						PTMC Algorithm 5.2 ($n^* = 12,500,000$)					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	0.019	0.190	0.152	0.860	5000	36.082	0.017	0.171	0.158	0.930	4987	75.263
β_2	0.006	0.076	0.065	0.920	5000		0.004	0.072	0.067	0.940	4982	
β_3	0.003	0.122	0.122	0.950	4984		-0.005	0.109	0.124	0.950	5000	
β_4	-0.003	0.022	0.022	0.910	5000		-0.003	0.022	0.022	0.960	5000	
β_5	0.003	0.155	0.137	0.920	5000		0.008	0.148	0.142	0.940	5000	
β_6	-0.003	0.034	0.030	0.910	5000		-0.002	0.032	0.031	0.950	5000	

	PTMC Gibbs Sampler						PTMC MH					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	0.016	0.169	0.161	0.940	2044	377.480	0.015	0.173	0.159	0.920	119	39.356
β_2	0.004	0.070	0.069	0.950	1253		0.003	0.070	0.068	0.950	138	
β_3	-0.005	0.107	0.125	0.980	1769		-0.006	0.109	0.123	0.990	141	
β_4	-0.003	0.021	0.023	0.960	2398		-0.003	0.022	0.023	0.970	480	
β_5	0.005	0.142	0.147	0.950	2525		0.006	0.142	0.147	0.950	137	
β_6	-0.002	0.032	0.032	0.960	3284		-0.002	0.032	0.032	0.960	353	

	MCMC(0)						MCMC(500)					
	EBias	ESE	ASE	ECP	ESS	CT	EBias	ESE	ASE	ECP	ESS	CT
β_1	0.016	0.169	0.160	0.920	450	8.270	0.015	0.169	0.159	0.930	2051	2103.512
β_2	0.004	0.071	0.068	0.950	188		0.003	0.070	0.068	0.950	1271	
β_3	-0.006	0.107	0.123	0.990	354		-0.006	0.107	0.123	0.980	1776	
β_4	-0.003	0.021	0.022	0.960	528		-0.003	0.021	0.022	0.960	2418	
β_5	0.006	0.143	0.145	0.960	591		0.006	0.142	0.145	0.950	2506	
β_6	-0.002	0.032	0.031	0.950	718		-0.002	0.032	0.031	0.950	3293	

D.3 Additional Results of Data Analysis

Table D.8: Data analysis results for the Fishery Data

1000000 iterations		Mode 1			Mode 2			Mode 3			Mode 4			Mode 5			Mode 6			
		Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Sum of ESS
PTMC Gibbs	μ_1	7.206	0.287	11348	7.210	0.290	27651	5.170	0.08	52584	5.170	0.081	73838	3.225	0.089	88342	3.225	0.089	124039	377802
	μ_2	5.170	0.080	52170	3.225	0.090	71268	7.206	0.287	11252	3.225	0.09	67764	7.208	0.288	32724	5.170	0.081	132660	367838
	μ_3	3.225	0.090	88723	5.170	0.081	78067	3.226	0.089	89877	7.208	0.29	26610	5.170	0.081	94091	7.208	0.289	50532	427900
	σ_1	1.836	0.146	33599	1.834	0.146	54565	0.505	0.085	15640	0.506	0.086	34672	0.285	0.086	79002	0.285	0.082	112277	329755
	σ_2	0.505	0.085	15583	0.285	0.084	64637	1.836	0.146	35753	0.285	0.089	58920	1.836	0.146	65881	0.505	0.086	63607	304381
	σ_3	0.286	0.086	74263	0.506	0.086	38397	0.285	0.082	76164	1.835	0.146	52231	0.506	0.086	45608	1.836	0.146	101141	387804
PTMC MH	μ_1	7.239	0.284	242	7.226	0.280	219	5.169	0.079	7871	5.172	0.079	626	3.225	0.088	11179	3.225	0.088	6355	26492
	μ_2	5.171	0.077	1214	3.231	0.094	732	7.211	0.292	1625	3.225	0.085	683	7.202	0.287	3086	5.171	0.079	7949	15289
	μ_3	3.222	0.094	602	5.172	0.079	1125	3.224	0.086	7680	7.219	0.296	180	5.169	0.079	14700	7.209	0.286	1652	25939
	σ_1	1.827	0.150	529	1.827	0.149	641	0.506	0.086	3461	0.508	0.094	347	0.283	0.078	10219	0.286	0.113	1654	16851
	σ_2	0.512	0.085	381	0.287	0.085	596	1.830	0.145	3816	0.288	0.088	801	1.836	0.145	6916	0.505	0.085	3796	16306
	σ_3	0.284	0.084	545	0.510	0.086	552	0.283	0.075	7347	1.844	0.145	343	0.505	0.085	6876	1.835	0.144	4146	19809
MCMC (BIG)	μ_1	7.202	0.283	459				5.169	0.078	3532	5.169	0.079	16547	3.226	0.088	1695	3.224	0.086	20270	42503
	μ_2	5.165	0.076	1509				7.203	0.283	1122	3.224	0.086	16133	7.203	0.289	1547	5.171	0.078	20575	40886
	μ_3	3.223	0.084	1805				3.224	0.089	3704	7.205	0.287	14179	5.170	0.076	1860	7.209	0.286	17321	38869
	σ_1	1.834	0.142	1568				0.504	0.083	1743	0.504	0.085	20088	0.283	0.075	2498	0.283	0.079	28259	54156
	σ_2	0.502	0.082	801				1.836	0.144	4047	0.283	0.088	16106	1.836	0.144	2890	0.506	0.085	23234	47078
	σ_3	0.283	0.088	2585				0.283	0.083	4023	1.836	0.145	25871	0.503	0.084	2183	1.836	0.145	31466	66128
MCMC (SMALL)	μ_1	7.214	0.288	2378													3.223	0.084	32916	35294
	μ_2	5.170	0.078	9382													5.169	0.078	39713	49095
	μ_3	3.223	0.083	10351													7.208	0.285	13626	23977
	σ_1	1.831	0.147	2043													0.282	0.073	13586	15629
	σ_2	0.507	0.086	2634													0.505	0.084	12452	15086
	σ_3	0.281	0.071	4900													1.833	0.144	6634	11534
LMC (ADAPTIVE)	μ_1	7.201	0.299	309																
	μ_2	5.167	0.078	3917																
	μ_3	3.222	0.082	3487																
	σ_1	1.837	0.147	1260																
	σ_2	0.505	0.085	2665																
	σ_3	0.281	0.072	3723																

Table D.9: Data analysis results for the Hidalgo Stamp Data

1000000 iterations		Mode 1			Mode 2			Mode 3			Mode 4			Mode 5			Mode 6			
		Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Estimate	SE	ESS	Sum of ESS
PTMC Gibbs	μ_1	9.900	0.139	11391	9.898	0.138	8047	7.891	0.052	25108	7.890	0.052	15106	7.162	0.060	17631	7.161	0.059	9484	86767
	μ_2	7.891	0.052	2600	7.163	0.061	1365	9.901	0.137	117430	7.163	0.060	12060	9.899	0.137	104353	7.889	0.051	11736	249544
	μ_3	7.165	0.061	1962	7.890	0.052	1630	7.164	0.061	20473	9.900	0.138	81012	7.889	0.051	21208	9.899	0.136	65887	192172
	σ_1	1.398	0.089	9676	1.399	0.089	7035	0.219	0.040	21426	0.219	0.039	12819	0.176	0.055	17253	0.176	0.055	8959	77168
	σ_2	0.218	0.040	2205	0.177	0.055	1224	1.398	0.089	105101	0.177	0.057	12278	1.398	0.089	83372	0.220	0.039	10027	214207
	σ_3	0.179	0.059	2093	0.220	0.040	1484	0.178	0.056	19979	1.398	0.089	64209	0.220	0.039	19084	1.399	0.088	55078	161927
PTMC MH	μ_1	9.893	0.140	1438	9.900	0.142	1342	7.882	0.052	1142	7.880	0.051	971	7.160	0.067	537	7.156	0.062	550	5980
	μ_2	7.878	0.051	813	7.156	0.058	627	9.904	0.144	1691	7.155	0.059	868	9.905	0.150	1353	7.881	0.052	683	6035
	μ_3	7.151	0.055	736	7.882	0.050	756	7.158	0.064	704	9.893	0.151	1855	7.882	0.054	804	9.903	0.143	1560	6415
	σ_1	1.403	0.096	1652	1.400	0.095	1551	0.224	0.051	1050	0.227	0.053	967	0.171	0.076	722	0.171	0.095	1641	7583
	σ_2	0.229	0.052	1028	0.169	0.072	691	1.400	0.095	2558	0.169	0.087	827	1.400	0.100	1831	0.226	0.052	705	7640
	σ_3	0.168	0.109	1139	0.225	0.048	815	0.171	0.084	1417	1.402	0.095	2380	0.225	0.061	811	1.399	0.095	1835	8397
MCMC (BIG)	μ_1										7.882	0.039	3164							3164
	μ_2										7.150	0.045	2286							2286
	μ_3										9.895	0.130	75956							75956
	σ_1										0.225	0.033	3391							3391
	σ_2										0.162	0.043	2344							2344
	σ_3										1.403	0.085	76114							76114
MCMC (SMALL)	μ_1				9.894	0.130	86050													86050
	μ_2				7.149	0.044	3675													3675
	μ_3				7.881	0.038	4610													4610
	σ_1				1.404	0.085	66625													66625
	σ_2				0.161	0.043	3858													3858
	σ_3				0.225	0.033	5041													5041
LMC (ADAPTIVE)	μ_1	9.,898	0.127	991																
	μ_2	7.880	0.038	3500																
	μ_3	7.148	0.044	2378																
	σ_1	1.400	0.083	2373																
	σ_2	0.226	0.033	3482																
	σ_3	0.161	0.042	2143																

Appendix E

Appendix for Chapter 6

E.1 Proof of Theorems

E.1.1 Proof of Theorem 6.1

Let $N_x = \sum_{i=1}^{n} \prod_{j=1}^{p} I(x_j - h_j \leq X_{ij} \leq x_j + h_j)$ denote the number of data points in the nearest neighbor of x. By the Law of Large Numbers, as $n \to \infty$,

$$
\begin{aligned}
\frac{N_x}{n} =& \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{p} I(x_j - h_j \leq X_{ij} \leq x_j + h_j) \\
\xrightarrow{p}& E\Big[\prod_{j=1}^{p} I(x_j - h_j \leq X_j \leq x_j + h_j) \Big] \\
=& P(X \in \mathcal{S}_{x,h}) \\
=& f_X(\omega) \cdot 2^p \prod_{j=1}^{p} h_j = O(n^{-\eta})
\end{aligned} \tag{E.1}
$$

where ω is a certain value in $\mathcal{S}_{x,h}$, f_X is the density function of X and the last step is obtained by applying the mean value theorem and $h_j = O(n^{-\eta/p})$. As a result, $N_x = O_p(n^{1-\eta})$, which goes to infinity as n goes to infinity.

The proof of Theorem 6.1 consists of the following steps. In the first step, we show that $\frac{1}{w_x} N_{\varepsilon_1 \ldots \varepsilon_m, x}(\tilde{Z})$ satisfies the Lyapunov condition of the Central Limit Theorem for independent but not identically distributed random variables. Based on the results of step

1, we evaluate the order of $\sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z})-F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|$ and $\mathrm{var}(\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}))$ in step 2. In step 3, we prove the result of Theorem 6.1 using the results of step 2.

Step 1: we show that $\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z})$ satisfies the Lyapunov condition.

For the random vector $Z_i = (Y_i, X_i^{\mathrm{T}})^{\mathrm{T}}$ with $X_i \in \mathcal{S}_{x,h}$, we re-index the random vector as $Z_{\{k\}} = (Y_{\{k\}}, X_{\{k\}}^{\mathrm{T}})^{\mathrm{T}}$ for $k = 1,\ldots,N_x$ and $x_{\{k\}} = (x_{\{k1\}},\ldots,x_{(\{kp\})})^{\mathrm{T}}$. Let $\tilde{Z}_{\{x\}} = \{Z_{\{k\}} : k = 1,\ldots,N_x\}$ denote the collection of data in $\mathcal{S}_{x,h}$, and let $w(X_{\{k\}}) = \prod_{j=1}^{p} w(X_{\{kj\}})$ for notation simplicity. Then we have that $w_x = \sum_{k=1}^{N_x} w(X_{\{k\}}) = O_p(N_x) = O_p(n^{1-\eta})$, as $w(\cdot)$ is a positive bounded function on $\mathcal{S}_{x,h}$. Consequently,

$$\begin{aligned}\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) &= \frac{1}{w_x}\sum_{i=1}^{n}\prod_{j=1}^{p} w(X_{ij})I\big(Y_i\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\big)\prod_{j=1}^{p} I\left(X_{ij}\in[x_j-h_j,x_j+h_j]\right)\\ &= \frac{1}{w_x}\sum_{k=1}^{N_x} I(Y_{\{k\}}\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})w(X_{\{k\}}),\end{aligned}$$

and

$$\begin{aligned}I(Y_{\{k\}}\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) &= \begin{cases}1 & F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\\ 0 & 1-F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m});\end{cases}\\ E[I(Y_{\{k\}}\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})] &= F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m});\\ \mathrm{var}[I(Y_{\{k\}}\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})] &= F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})[1-F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})].\end{aligned}$$

Let $W_k = I(Y_{\{k\}}\in\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})w(X_{\{k\}}) - F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})$, and $T_{N_x} = \sum_{k=1}^{N_x} W_k$. Then we have

$$\begin{aligned}s_{N_x}^2 = \mathrm{var}(T_{N_x}) &= \sum_{k=1}^{N_x}\mathrm{var}(W_k)\\ &= \sum_{k=1}^{N_x} w(X_{\{k\}})^2 F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})[1-F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})].\end{aligned}$$

For notation simplicity, let $F_{\{k\}} = F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})$. Now we only discuss the case that $F_{\{k\}} > 0$ for some k, as the case with $F_{\{k\}} = 0$ for all $k = 1,\ldots,N_x$ is trivial:

$$\left|\frac{1}{s_{N_x}^3}\sum_{k=1}^{N_x} E(|W_k|^3)\right| = \left|\frac{1}{s_{N_x}^3}\sum_{k=1}^{N_x} w(X_{\{k\}})^3\{[1-F_{\{k\}}]^3F_{\{k\}} + F_{\{k\}}^3[1-F_{\{k\}}]\}\right|$$

$$
\begin{aligned}
&= \left| \frac{1}{\left(\sum_{k=1}^{N_x} w(X_{\{k\}})^2 F_{\{k\}}[1-F_{\{k\}}] \right)^{\frac{3}{2}}} \sum_{k=1}^{N_x} w(X_{\{k\}})^3 F_{\{k\}}[1-F_{\{k\}}]\{[1-F_{\{k\}}]^2 - F_{\{k\}}^2\} \right| \\
&\leq \left| \frac{w_{\max}^3}{w_{\min}^3 \left(\sum_{k=1}^{N_x} F_{\{k\}}[1-F_{\{k\}}] \right)^{\frac{3}{2}}} \sum_{k=1}^{N_x} F_{\{k\}}[1-F_{\{k\}}]\{[1-F_{\{k\}}]^2 - F_{\{k\}}^2\} \right| \qquad \text{(E.2)} \\
&\leq \frac{w_{\max}^3}{w_{\min}^3} \frac{1}{\left(\sum_{k=1}^{N_x} F_{\{k\}}[1-F_{\{k\}}] \right)^{\frac{3}{2}}} \sum_{k=1}^{N_x} F_{\{k\}}[1-F_{\{k\}}] \qquad \text{(E.3)} \\
&\to 0 \qquad \text{as } N_x \to \infty,
\end{aligned}
$$

where (E.2) is obtained by the fact that $w(X_{\{k\}}) \in [w_{\min}, w_{\max}]$, and (E.3) can be obtained by applying the absolute value inequality. Therefore, we prove that T_{N_x} satisfies the Lyapunov condition.

Step 2: we evaluate the order of $\sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m} \left| \frac{1}{w_x} N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) \right|$ and $\mathrm{var}(\frac{1}{w_x} N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}))$.

First, we find the order of the variance of $\frac{1}{w_x} N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z})$,

$$
\begin{aligned}
&\mathrm{var}\left(\frac{1}{w_x} N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) \right) = \mathrm{var}\left(\frac{1}{w_x} T_{N_x} \right) \\
=\ & \mathrm{var}\left(\frac{1}{w_x} \sum_{k=1}^{N_x} I(Y_{\{k\}} \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) w(X_{\{k\}}) \right) \\
=\ & \frac{1}{w_x^2} \sum_{k=1}^{N_x} w(X_{\{k\}})^2 F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})[1 - F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})] \\
\leq\ & \sum_{k=1}^{N_x} \frac{w(x_{(k)})^2}{[\sum_{k=1}^{N_x} w(x_{(k)})]^2} = \sum_{k=1}^{N_x} O_p(\frac{1}{N_x^2}) \qquad \text{(E.4)} \\
=\ & O_p\left(\frac{1}{N_x} \right) = O_p\left(\frac{1}{n^{1-\eta}} \right) \qquad \text{(E.5)}
\end{aligned}
$$

where (E.4) is due to the fact that $F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) \in [0,1]$. By the Central Limit Theorem for independent and non-identically distributed variables, we obtain that as $n \to \infty$,

$$\frac{T_{N_x}}{s_{N_x}} \xrightarrow{d} N(0,1),$$

implying that
$$\frac{1}{w_x}\sum_{k=1}^{N_x} I(Y_{\{k\}} \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})w(X_{\{k\}}) = \frac{1}{w_x}\sum_{k=1}^{N_x} F_{\{k\}} + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) \tag{E.6}$$

by the fact that $\frac{1}{w_x}s_{N_x} = O_p(\frac{1}{\sqrt{N_x}})$ from (E.5).

Step 3: we prove the result of Theorem 6.1 by evaluating the order of the upper bound of $\sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|$.

We first find an upper bound of the supreme of the absolute difference $\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|$ for $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m} \in \pi_m$:

$$\begin{aligned}
&\sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m} \left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| \\
= \quad & \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}\sum_{k=1}^{N_x} I(Y_{\{k\}} \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})w(X_{\{k\}}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| \\
= \quad & \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}\sum_{k=1}^{N_x} F_{\{k\}} + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| \qquad \text{(E.7)} \\
\leq \quad & \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}\sum_{k=1}^{N_x} w(x_{\{k\}})F_{\{k\}} - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) \qquad \text{(E.8)}
\end{aligned}$$

where (E.7) is the direct application of (E.6), and (E.8) is due to the absolute value inequality.

Since $\min_k F_{\{k\}} < \sum_{k=1}^{N_x} \frac{w(x_{\{k\}})}{w_x}F_{\{k\}} < \max_k F_{\{k\}}$ for $k = 1, \ldots, N_x$, then as $n \to \infty$, (E.8) becomes

$$\sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m} \left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|$$

$$
\begin{aligned}
&\leq \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}\sum_{k=1}^{N_x} w(x_{\{k\}})F_{\{k\}} - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) \\
&\leq \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left[\sup_k\left|F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|\right] + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) \\
&\leq \sup_{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}\in\pi_m}\left[2^p\prod_{j=1}^{p} h_j\cdot\sup_{x\in\mathcal{S}_{x,h}}|g'_{\varepsilon_1\ldots\varepsilon_m}(x)|\right] + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) \qquad \text{(E.9)} \\
&= 2^pO(n^{-\eta}) + O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right) = \max\left\{O\left(\frac{1}{n^\eta}\right), O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right)\right\} \qquad \text{(E.10)} \\
&= \max\left\{O_p\left(\frac{1}{n^\eta}\right), O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right)\right\} \longrightarrow 0. \qquad \text{(E.11)}
\end{aligned}
$$

where (E.9) is due to the following fact based on the Mean Value Theorem:

$$
\begin{aligned}
F_{x_{\{k\}}}(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m}) &= g_{\varepsilon_1\ldots\varepsilon_m}(x_{\{k\}}) - g_{\varepsilon_1\ldots\varepsilon_m}(x) \\
&= 2^p\prod_{j=1}^{p} h_j g'_{\varepsilon_1\ldots\varepsilon_m}(b) \qquad \text{for } b\in\mathcal{S}_{x,h} \\
&\leq 2^p\prod_{j=1}^{p} h_j \sup_{x\in\mathcal{S}_{x,h}}|g'_{\varepsilon_1\ldots\varepsilon_m}(x)|,
\end{aligned}
$$

and (E.10) is due to (E.1) and the assumption that $g_{\varepsilon_1\ldots\varepsilon_m}(x)$ is smooth so that $\sup_{x\in\mathcal{S}_{x,h}}|g'_{\varepsilon_1\ldots\varepsilon_m}(x)|$ is bounded.

By Chebyshev's Inequality, for any $x\in\mathcal{S}_x$, with $n\to\infty$, we have

$$
\begin{aligned}
&P\left(\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right| > \epsilon\right) \\
&\leq \frac{E^2\left(\left|\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z}) - F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_m})\right|\right) + \mathrm{var}\left(\frac{1}{w_x}N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z})\right)}{\epsilon^2} \\
&\leq \frac{\left[\max\left\{O_p\left(\frac{1}{n^\eta}\right), O_p\left(\frac{1}{n^{\frac{1-\eta}{2}}}\right)\right\}\right]^2 + O_p\left(\frac{1}{n^{1-\eta}}\right)}{\epsilon^2} \\
&\longrightarrow 0
\end{aligned}
$$

□

E.1.2 Proof of Theorem 6.2

From Theorem 6.1, we prove that

$$\sup_{\mathcal{B}_{\varepsilon_1\dots\varepsilon_m}\in\pi_m}\left|\frac{1}{w_x}N_{\varepsilon_1\dots\varepsilon_m,x}(\tilde{Z})-F_x(\mathcal{B}_{\varepsilon_1\dots\varepsilon_m})\right|\le\max\left\{O_p\left(\frac{1}{n^\eta}\right),O_p\left(\frac{1}{\sqrt{N_x}}\right)\right\}.$$

Therefore,

$$\begin{aligned}N_{\varepsilon_1\dots\varepsilon_m,x}(\tilde{Z})&=w_xF_x(\mathcal{B}_{\varepsilon_1\dots\varepsilon_m})+\max\left\{O_p\left(\frac{N_x}{n^\eta}\right),O_p\left(\sqrt{N_x}\right)\right\}\\&=w_xF_x(\mathcal{B}_{\varepsilon_1\dots\varepsilon_m})+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\},\end{aligned}\tag{E.12}$$

and $w_x=O_p(N_x)=O_p(n^{1-\eta})$. Let $\mathcal{B}_{\{0\}}=\mathcal{S}$ and $\mathcal{B}_{\{j\}}=\mathcal{B}_{\varepsilon_1\dots,\varepsilon_j}$ for $j=1,2,\cdots$. Obviously, we have that $\mathcal{B}_{\{0\}}\supset\mathcal{B}_{\{1\}}\supset\mathcal{B}_{\{2\}}\supset\dots$.

The proof of Theorem 6.2 consists of the following three steps. In the first step, we show Theorem 6.2 (1) for $B\in\pi_m$ with $m=1,\dots,M$. In the second step, we show Theorem 6.2 (1) for the case that $B\in\pi_m$ with $m>M$. In the final step, we prove Theorem 6.2 (2).

Step 1: we first prove Theorem 6.2 (1) for $\mathcal{B}_{\{m\}}$ when $m\le M$.

If $F_x(\mathcal{B}_{\{m\}})>0$, then

$$\begin{aligned}E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})]&=\prod_{j=1}^{m}\frac{\alpha_{\varepsilon_1\dots\varepsilon_j}+N_{\varepsilon_1\dots\varepsilon_m,x}(\tilde{Z})}{\sum_{l=0}^{1}\left[\alpha_{\varepsilon_1\dots\varepsilon_{j-1}l}+N_{\varepsilon_1\dots\varepsilon_{j-1}l,x}(\tilde{Z})\right]}\\
&=\prod_{j=1}^{m}\frac{\alpha_{\varepsilon_1\dots\varepsilon_j}+w_xF_x(\mathcal{B}_{\{j\}})+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}{(\sum_{l=0}^{1}\alpha_{\varepsilon_1\dots\varepsilon_{j-1}l})+w_xF_x(\mathcal{B}_{\{j-1\}})+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}\\
&=\prod_{j=1}^{m}\frac{\frac{\phi j^2}{w_x}+F_x(\mathcal{B}_{\{j-1\}})+\max\left\{O_p\left(n^{-\eta}\right),O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}}{\frac{2\phi j^2}{w_x}+F_x(\mathcal{B}_{\{j-1\}})+\max\left\{O_p\left(n^{-\eta}\right),O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}}\\
&=\prod_{j=1}^{m}\left[\frac{F_x(\mathcal{B}_{\{j\}})}{F_x(\mathcal{B}_{\{j-1\}})}+\frac{\phi j^2-2\phi j^2\frac{F_x(\mathcal{B}_{\{j\}})}{F_x(\mathcal{B}_{\{j-1\}})}+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}{2\phi j^2+w_xF_x(\mathcal{B}_{\{j-1\}})+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}\right]\end{aligned}\tag{E.13}$$

where the first equality is from (6), and the second equality is the application of (E.12). Here we also use the default choice $\alpha_{\varepsilon_1 \ldots \varepsilon_m} = \phi m^2$ as mentioned in Section 2.2.

It is obvious that $\frac{F_x(\mathcal{B}_{\{j\}})}{F_x(\mathcal{B}_{\{j-1\}})} \leq 1$. For $r = 1, \ldots, 2^m - 1$, let T_r denote any non-empty subset of $\{1, \ldots, m\}$ and let T_r^c denote its compliment. Then (E.13) becomes

$$
\begin{aligned}
& E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] = \prod_{j=1}^{m} \frac{F_x(\mathcal{B}_{\{j\}})}{F_x(\mathcal{B}_{\{j-1\}})} + \\
& \sum_{r=1}^{2^m-1} \Bigg[\prod_{g \in T_r^c} \frac{F_x(\mathcal{B}_{\{g\}})}{F_x(\mathcal{B}_{\{g-1\}})} \Bigg] \Bigg[\prod_{q \in T_r} \frac{\phi q^2 - 2\phi q^2 \frac{F_x(\mathcal{B}_{\{q\}})}{F_x(\mathcal{B}_{\{q-1\}})} + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi q^2 + w_x F(\mathcal{B}_{q-1}|x) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \Bigg] \\
\leq \;& F_x(\mathcal{B}_{\{m\}}) + \sum_{r=1}^{2^m-1} \Bigg[\prod_{q \in T_r} \frac{\phi q^2 + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi q^2 + w_x F_x(\mathcal{B}_{\{q-1\}}) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \Bigg] \qquad \text{(E.14)} \\
= \;& F_x(\mathcal{B}_{\{m\}}) + \prod_{j=1}^{m} \Bigg[1 + \frac{\phi j^2 + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \Bigg] - 1 \qquad \text{(E.15)} \\
= \;& F_x(\mathcal{B}_{\{m\}}) + \exp\Bigg\{ \sum_{j=1}^{m} \log \Bigg[1 + \frac{\phi j^2 + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \Bigg] \Bigg\} - 1 \\
= \;& F_x(\mathcal{B}_{\{m\}}) + \exp\Bigg\{ \sum_{j=1}^{m} \frac{\phi j^2 + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \\
+ \;& \sum_{j=1}^{m} O_p \Bigg[\Bigg(\frac{\phi j^2 + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{ O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big) \Big\}} \Bigg)^2 \Bigg] \Bigg\} - 1 \qquad \text{(E.16)}
\end{aligned}
$$

$$= \quad F_x(\mathcal{B}_{\{m\}}) + O_p\Bigg(\sum_{j=1}^{m} \frac{\phi j^2 + \max\Big\{O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big)\Big\}}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big)\Big\}}\Bigg) \tag{E.17}$$

$$\leq \quad F_x(\mathcal{B}_{\{m\}}) + O_p\Bigg(\sum_{j=1}^{m} \frac{\phi j^2 + \max\Big\{O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big)\Big\}}{w_x F_x(\mathcal{B}_{\{j-1\}})}\Bigg) \tag{E.18}$$

$$= \quad F_x(\mathcal{B}_{\{m\}}) + \max\Big\{O_p\Big(\frac{M}{n^{\eta}}\Big), O_p\Big(\frac{M}{n^{\frac{1-\eta}{2}}}\Big), O_p\Big(\frac{M^3}{n^{1-\eta}}\Big)\Big\} \tag{E.19}$$

where the inequality (E.14) is obtained by omitting the negative term $-2\phi q^2 \frac{F_x(\mathcal{B}_{\{q\}})}{F_X(\mathcal{B}_{\{q-1\}})}$ in the previous step; equation (E.15) is due to the expansion of the product $\prod_{j=1}^{m}(1+a_j)$ for a series of scalar a_j with $j=1,\ldots,m$ that $\prod_{j=1}^{m}(1+a_j) = \sum_{r=1}^{2^m-1}\Big[\prod_{q\in T_r} a_q\Big] + 1$; in deriving equation (E.16) and (E.17), we use the Taylor expansions $\log(1+a) = a + O(a)$ and $\exp(a) = 1 + O(a)$, and inequality (E.18) is obtained by omitting the terms $2\phi j^2$ and $\max\Big\{O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big)\Big\}$ in the denominator of previous step.

If $F_x(\mathcal{B}_{\{m\}}) = 0$, suppose $l_1 = \max\{i | i < m; F_x(\mathcal{B}_{\{i\}}) > 0\}$, then

$$\begin{aligned}
E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] &= \prod_{j=1}^{m} \frac{\alpha_{\varepsilon_1\ldots\varepsilon_j} + N_{\varepsilon_1\ldots\varepsilon_m,x}(\tilde{Z})}{\sum_{l=0}^{1}\big[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l,x}(\tilde{Z})\big]} \\
&= \prod_{j=1}^{l_1+1} \frac{\frac{\phi j^2}{w_x} + F_x(\mathcal{B}_{\{j\}}) + \max\Big\{O_p\Big(n^{-\eta}\Big), O_p\Big(n^{-\frac{1-\eta}{2}}\Big)\Big\}}{\frac{2\phi j^2}{w_x} + F_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{O_p\Big(n^{-\eta}\Big), O_p\Big(n^{-\frac{1-\eta}{2}}\Big)\Big\}} \Big(\frac{1}{2}\Big)^{m-l_1-1} \\
&\leq F_x(\mathcal{B}_{\{l_1+1\}})\Big(\frac{1}{2}\Big)^{m-l_1-1} + \Big(\frac{1}{2}\Big)^{m-l_1-1} \max\Big\{O_p\Big(\frac{M}{n^{\eta}}\Big), O_p\Big(\frac{M}{n^{\frac{1-\eta}{2}}}\Big), O_p(\frac{M^3}{n^{1-\eta}})\Big\} \\
&= 0 + \max\Big\{O_p\Big(\frac{M}{n^{\eta}}\Big), O_p\Big(\frac{M}{n^{\frac{1-\eta}{2}}}\Big), O_p(\frac{M^3}{n^{1-\eta}})\Big\}, \qquad \text{(E.20)}
\end{aligned}$$

where (E.20) is obtained by $F(\mathcal{B}^*_{\{l_1+1\}}) = 0$.

Now we prove the order $\mathrm{var}\Big(G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})\Big)$ for $m < M$. First, we present a fact that for independent Z_1 and Z_2,

$$\begin{aligned}
\mathrm{var}(Z_1Z_2) &= E(Z_1^2Z_2^2) - E^2(Z_1Z_2) \\
&= E(Z_1^2)E(Z_2^2) - E^2(Z_1)E^2(Z_2) \\
&= [E(Z_1^2)E(Z_2^2) - E^2(Z_1)E(Z_2^2)] + [E^2(Z_1)E(Z_2^2) - E^2(Z_1)E^2(Z_2)] \\
&= \mathrm{var}(Z_1)E(Z_2^2) + E^2(Z_1)\mathrm{var}(Z_2). \qquad \text{(E.21)}
\end{aligned}$$

Next, write $G_{j,x} = G_{\varepsilon_1\ldots\varepsilon_j,x}(\tilde{Z}) \in [0,1]$. By the definition of Polya tree, the $G_{j,x}$ are independent. Therefore, applying (E.21) to $\mathrm{var}\Big(G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})\Big)$ gives that

$$\begin{aligned}
&\mathrm{var}\Big(G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})\Big) = \mathrm{var}\Big(\prod_{j=1}^{m} G_{j,x}\Big) \\
= \ & \left[\mathrm{var}(G_{1,x})E\Big(\prod_{j=2}^{m} G_{j,x}\Big) + E(G_{1,x})^2\mathrm{var}\Big(\prod_{j=2}^{m} G_{j,x}\Big)\right] \qquad \text{(E.22)} \\
\leq \ & \left[\mathrm{var}(G_{1,x}) + \mathrm{var}\Big(\prod_{j=2}^{m} G_{j,x}\Big)\right] \qquad \text{(E.23)} \\
\leq \ & \sum_{j=1}^{m} \mathrm{var}(G_{j,x}) \qquad \text{(E.24)} \\
= \ & \sum_{j=1}^{m} \frac{\Big(\alpha_{\varepsilon_1\ldots\varepsilon_j} + N_{\varepsilon_1\ldots\varepsilon_j,x}(\tilde{Z})\Big)\Big\{\sum_{i\neq\varepsilon_j}\big[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}i} + N_{\varepsilon_1\ldots\varepsilon_{j-1}i,x}(\tilde{Z})\big]\Big\}}{\Big\{\sum_{l=0}^{1}\big[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l,x}(\tilde{Z})\big]\Big\}^2\Big\{\sum_{l=0}^{1}\big[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l,x}(\tilde{Z})\big] + 1\Big\}} \qquad \text{(E.25)} \\
\leq \ & \sum_{j=1}^{m} \frac{1}{\sum_{l=0}^{1}\big[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l,x}(\tilde{Z})\big] + 1} \qquad \text{(E.26)} \\
= \ & \sum_{j=1}^{m} \frac{1}{2\phi j^2 + w_xF_x(\mathcal{B}_{\{j-1\}}) + \max\Big\{O_p\Big(n^{1-2\eta}\Big), O_p\Big(n^{\frac{1-\eta}{2}}\Big)\Big\}} \\
\leq \ & \frac{M}{w_xF_x(\mathcal{B}_{\{m-1\}})} = O_p\Big(\frac{M}{n^{1-\eta}}\Big), \qquad \text{(E.27)}
\end{aligned}$$

where the inequality (E.23) is due to the fact that $G_{j,x}$ is a probability between 0 and 1 for $j = 1, \ldots, m$; (E.24) is obtained by repeating the procedure of (E.22) and (E.23) for

$G_{1,x}$ to $G_{j,x}$ for $j = 2, \dots, m$; (E.25) is obtained from the variance of Beta distributions, as $G_{j,x}$ follows a Beta distribution; (E.26) is due to the fact that

$$0 \leq \alpha_{\varepsilon_1 \dots \varepsilon_j} + N_{\varepsilon_1 \dots \varepsilon_j, x}(\tilde{Z}) \leq \sum_{l=0}^{1} \left[\alpha_{\varepsilon_1 \dots \varepsilon_{j-1} l} + N_{\varepsilon_1 \dots \varepsilon_{j-1} l, x}(\tilde{Z}) \right]$$

$$0 \leq \sum_{i \neq \varepsilon_j} \left[\alpha_{\varepsilon_1 \dots \varepsilon_{j-1} i} + N_{\varepsilon_1 \dots \varepsilon_{j-1} i, x}(\tilde{Z}) \right] \leq \sum_{l=0}^{1} \left[\alpha_{\varepsilon_1 \dots \varepsilon_{j-1} l} + N_{\varepsilon_1 \dots \varepsilon_{j-1} l, x}(\tilde{Z}) \right];$$

and the inequality in (E.27) is due to the fact that

$$\begin{aligned}
&\sum_{j=1}^{m} \frac{1}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{j-1\}}) + \max\left\{ O_p\left(n^{1-2\eta}\right), O_p\left(n^{\frac{1-\eta}{2}}\right) \right\}} \\
&\leq \sum_{j=1}^{m} \frac{1}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{m-1\}}) + \max\left\{ O_p\left(n^{1-2\eta}\right), O_p\left(n^{\frac{1-\eta}{2}}\right) \right\}} \\
&\leq \sum_{j=1}^{M} \frac{1}{2\phi j^2 + w_x F_x(\mathcal{B}_{\{m-1\}}) + \max\left\{ O_p\left(n^{1-2\eta}\right), O_p\left(n^{\frac{1-\eta}{2}}\right) \right\}} \\
&\leq \sum_{j=1}^{M} \frac{1}{w_x F_x(\mathcal{B}_{\{m-1\}})} = \frac{M}{w_x F_x(\mathcal{B}_{\{m-1\}})}
\end{aligned}$$

together with the fact that $\phi > 0$.

Therefore for any measurable set $B \subset \pi_m$ with $m = 1, \dots, M$, if we consider $n = \max\left\{ O(M^{\frac{3}{1-\eta}+\xi}), O(M^{1/\eta+\xi}) \right\}$ with $\xi > 0$, then we have that as $M \to \infty$,

- by (E.19) and (E.20),

$$E\left[G_{x|\tilde{Z}}(B)\right] - F_x(B) = \max\left\{ O_p\left(\frac{M}{n^\eta}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}}\right), O_p\left(\frac{M^3}{n^{1-\eta}}\right) \right\} \xrightarrow{p} 0;$$

- by (E.27),

$$\operatorname{var}\left[G_{x|\tilde{Z}}(B)\right] = O_p\left(\frac{M}{n^{1-\eta}}\right) \xrightarrow{p} 0;$$

- by Chebyshev's Inequality,

$$P\left(\left|G_{x|\tilde{Z}}(B) - F_x(B)\right| \geq \epsilon\right) \leq \frac{E\left[G_{x|\tilde{Z}}(B) - F_x(B)\right]^2 + \text{var}\left[G_{x|\tilde{Z}}(B)\right]}{\epsilon^2}$$
$$= \max\left\{O_p\left(\frac{M^2}{n^{2\eta}}\right), O_p\left(\frac{M^2}{n^{1-\eta}}\right), O_p\left(\frac{M^6}{n^{2(1-\eta)}}\right)\right\} \xrightarrow{p} 0.$$

Step 2: next we prove Theorem 6.2 (1) for $\mathcal{B}_{\{m\}}$ when $m > M$.

Let $\mathfrak{V} = \{\mathcal{B}_{\varepsilon_1\ldots,\varepsilon_m} | \mathcal{B}_{\varepsilon_1\ldots,\varepsilon_m} \in \pi_m; F(\mathcal{B}_{\varepsilon_1\ldots,\varepsilon_m}) > 0\}$ and $\gamma(M) = \min\limits_{B\in\mathfrak{V}} F_x(B)$. We first show the existence of $\gamma(M)$. Since F_x is an appropriate probability measure with a continuous density function on $\mathcal{S}$ and the number of the subset $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}$, 2^M, is finite, there exists a subspace $\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}$ such that $F_x(\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}) > 0$, and $\gamma(M) = \underset{B\in\mathfrak{V}}{min} F_x(B)$ exists and is greater than zero. Let $\Omega_M = \{\mathcal{B}_{\{m\}} : \mathcal{B}_{\{m\}} \in \pi_m; m > M\}$.

Now we consider the case that $m > M$. If $F_x(\mathcal{B}_{\{m\}}) > 0$, then

$$E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] = \left\{\prod_{j=1}^{M} \frac{\alpha_{\varepsilon_1\ldots\varepsilon_j} + N_{\varepsilon_1\ldots\varepsilon_j,x}(\tilde{Z})}{\sum_{l=0}^{1}\left[\alpha_{\varepsilon_1\ldots\varepsilon_{j-1}l} + N_{\varepsilon_1\ldots\varepsilon_{j-1}l,x}(\tilde{Z})\right]}\right\}\left\{\prod_{j=M+1}^{m}\frac{1}{2}\right\}$$

$$\begin{aligned}
&= \left(\frac{1}{2}\right)^{m-M}\prod_{j=1}^{M}\frac{\frac{\phi j^2}{w_x} + F_x(\mathcal{B}_{\{j\}}) + \max\left\{O_p\left(n^{-\eta}\right), O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}}{\frac{2\phi j^2}{w_x} + F_x(\mathcal{B}_{\{j-1\}}) + \max\left\{O_p\left(n^{-\eta}\right), O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}} \\
&\leq \left(\frac{1}{2}\right)^{m-M}\left[F_x(\mathcal{B}_{\{M\}}) + O_p\left(\sum_{j=1}^{M}\frac{\phi j^2 + \max\left\{O_p\left(n^{1-2\eta}\right), O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}{w_x F_x(\mathcal{B}_{\{j-1\}})}\right)\right] \qquad \text{(E.28)} \\
&\leq \left(\frac{1}{2}\right)^{m-M}\left[F_x(\mathcal{B}_{\{M\}}) + \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\}\right],
\end{aligned}$$

where (E.28) is from (E.18).

Since we assume the joint density $f(y, x)$ is smooth, the conditional probability measure F_x is differentiable on $\mathcal{S}$, then

$$\sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left|E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] - F_x(\mathcal{B}_{\{m\}})\right|$$

$$
\begin{aligned}
&\leq \sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\Bigg|\left(\frac{1}{2}\right)^{m-M} F_x(\mathcal{B}_{\{M\}}) - F_x(\mathcal{B}_{\{m\}}) \\
&\quad + \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\}\Bigg| \\
&\leq \sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left|\left(\frac{1}{2}\right)^{m-M} F_x(\mathcal{B}_{\{M\}}) - F_x(\mathcal{B}_{\{m\}})\right| \\
&\quad + \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\} \\
&\leq \left(\frac{1}{2}\right)^{m} \sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left\{\sup_{\substack{y_1\in\mathcal{B}_{\{M\}}\\ y_2\in\mathcal{B}_{\{m\}}}}\Big|f(y_1|x) - f(y_2|x)\Big|\right\} \\
&\quad + \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\}
\end{aligned} \tag{E.29}
$$

where the second inequality holds by the absolute value inequality, and the last inequality holds due to the fact that

$$
\begin{aligned}
\left(\frac{1}{2}\right)^{M} \inf_{y\in\mathcal{B}_{\{M\}}} f(y|x) \leq F_x(\mathcal{B}_{\{M\}}) \leq \left(\frac{1}{2}\right)^{M} \sup_{y\in\mathcal{B}_{\{M\}}} f(y|x) \\
\left(\frac{1}{2}\right)^{m} \inf_{y\in\mathcal{B}_{\{m\}}} f(y|x) \leq F_x(\mathcal{B}_{\{m\}}) \leq \left(\frac{1}{2}\right)^{m} \sup_{y\in\mathcal{B}_{\{m\}}} f(y|x).
\end{aligned}
$$

Let $\sup_{y\in\mathcal{B}_{\{M\}}} |f'(y|x)|$ represent the supreme value of derivative $f'(y|x)$ in the subset $\mathcal{B}_{\{M\}}$, then from the Mean Value Theorem, (E.29) becomes

$$
\begin{aligned}
&\sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left|E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] - F_x(\mathcal{B}_{\{m\}})\right| \\
&\leq \left(\frac{1}{2}\right)^{m}\left(\frac{1}{2}\right)^{M} \sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left\{\sup_{y\in\mathcal{B}^*_{\{M\}}} |f'(y|x)|\right\} \\
&+ \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\} \\
&= \max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right), O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right), O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\}.
\end{aligned} \tag{E.30}
$$

where $\sup_{\mathcal{B}^*_{\{m\}}\in\Omega_M}\left\{\sup_{y\in\mathcal{B}_{\{M\}}}|f'(y|x)|\right\}$ is bounded due to the fact that F_x is differentiable on $\mathcal{S}$.

If $F_x(B_{\{m\}})=0$, suppose $l_1=\max\{i|i<m; F_x(\mathcal{B}_{\{i\}})>0\}$, then

$$
\begin{aligned}
&\sup_{\mathcal{B}_{\{m\}}\in\Omega_M} E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})] \\
&=\sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left\{\left(\frac{1}{2}\right)^{m-l_1-1}\prod_{j=1}^{l_1+1}\frac{\frac{\phi j^2}{w_x}+F(\mathcal{B}_j|x)+\max\left\{O_p\left(n^{-\eta}\right),O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}}{\frac{2\phi j^2}{w_x}+F(\mathcal{B}_{j-1}|x)+\max\left\{O_p\left(n^{-\eta}\right),O_p\left(n^{-\frac{1-\eta}{2}}\right)\right\}}\right\} \\
&\leq\sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left\{\left(\frac{1}{2}\right)^{m-l_1-1}\left[F(\mathcal{B}_{l_1+1})+O_p\left(\sum_{j=1}^{m}\frac{\phi j^2+\max\left\{O_p\left(n^{1-2\eta}\right),O_p\left(n^{\frac{1-\eta}{2}}\right)\right\}}{w_xF(\mathcal{B}_{j-1}|x)}\right)\right]\right\} \\
&=0+\max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right),O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right),O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\}. \qquad \text{(E.31)}
\end{aligned}
$$

In step 1, we have proved that for any measurable set $B\in\pi_m$ with $m=1,\dots,M$, $E[G_{x|\tilde{Z}}(B)]-F_x(B)\xrightarrow{p}0$ as $M\to\infty$. By (E.30) and (E.31), we prove that for $m>M$, as $M\to\infty$

$$
\begin{aligned}
&\sup_{\mathcal{B}_{\{m\}}\in\Omega_M}\left|E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})]-F_x(\mathcal{B}_{\{m\}})\right|] \\
&=\max\left\{O_p\left(\frac{M}{n^{\eta}\gamma(M)}\right),O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right),O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)\right\} \\
&\xrightarrow{p}0
\end{aligned}
$$

Further, $\bigcup_{j=1}^{M}\pi_m\bigcup\Omega_M$ contains all measurable subsets of $\mathcal{S}$, which is essentially equal to $\mathfrak{S}$ defined in Theorem 6.2 in the main text. As a result, we have

$$\sup_{B\in\mathfrak{S}}\left|E[G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})]-F_x(\mathcal{B}_{\{m\}})\right|\xrightarrow{p}0.$$

From the inequality (E.27) in step 1,

$$\sup_{B\in\mathfrak{S}}\mathrm{var}\left(G_{x|\tilde{Z}}(\mathcal{B}_{\{m\}})\right)\leq\sup_{\mathcal{B}_{\{M\}}\in\pi_m}O_p\left(\frac{M}{w_xF_x(\mathcal{B}_{\{M\}})}\right)\leq O_p\left(\frac{M}{n^{1-\eta}\gamma(M)}\right).$$

where the last inequality holds due to the definition that $\gamma(M) = \underset{B\in\mathfrak{V}}{min} F_x(B)$.

Step 3: finally we prove Theorem 6.2 (2).

Let $I_M^\Delta = \{\mathcal{B}_{\varepsilon_1\ldots\varepsilon_M} : \exists y \in \mathcal{B}_{\varepsilon_1\ldots\varepsilon_M}, f(y|x) < \Delta\}$, and $J_M^\Delta = \bigcup_{B\in I_M^\Delta} B$. Let $\mathcal{S}\backslash J_M^\Delta$ denote the compliment set of J_M^Δ. Therefore, $\inf_{y\in\mathcal{S}\backslash J_M^\Delta} f(y|x) \geq \Delta$. Since $f(y|x)$ is smooth on $\mathcal{S}$, $\forall\epsilon > 0$, for M large enough, $\forall B \in \pi_M$, and $y_1, y_2 \in B$, we have $|f(y_1|x) - f(y_2|x)| \leq \frac{\epsilon}{8T}$, with T to be the volume of $\mathcal{S}$. Selecting $\Delta = \epsilon/(4T)$, it is obvious that $\epsilon/(4T) > \sup_{y\in J_M^{\epsilon/(8T)}} f(y|x)$.

Then

$$
\begin{aligned}
D\Big(G_{x|\tilde{Z}}, F_x\Big) &= \int_{\mathcal{S}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy \\
&= \int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy + \int_{J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy \\
&\triangleq K_1 + K_2.
\end{aligned}
$$

Further,

$$
\begin{aligned}
K_2 = \int_{J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy &\leq \int_{J_M^{\epsilon/(8T)}} g_{x|\tilde{Z}}(y)dy + \int_{J_M^{\epsilon/(8T)}} f(y|x)dy \\
\leq \int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy &+ 2\int_{J_M^{\epsilon/(8T)}} f(y|x)dy,
\end{aligned}
$$

where the two inequalities are due to the application of absolute value inequality. Therefore,

$$
\begin{aligned}
D\Big(G_{x|\tilde{Z}}, F_x\Big) &\leq 2\int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy + 2\int_{J_M^{\epsilon/(8T)}} f(y|x)dy \\
&\leq 2\int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy + 2\cdot v_{J_M^{\epsilon/(8T)}} \frac{\epsilon}{4T} \qquad \text{(E.32)} \\
&\leq 2\int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy + \frac{\epsilon}{2} \qquad \text{(E.33)}
\end{aligned}
$$

where $v_{J_M^{\epsilon/(8T)}}$ denotes the total volume of the subsets in $J_M^{\epsilon/(8T)}$, (E.32) is obtained by the fact that $\epsilon/(4T) > \sup_{y\in J_M^{\epsilon/(8T)}} f(y|x)$ and (E.33) is due to the fact that $v_{J_M^{\epsilon/(8T)}} < T$. For K_1,

$$
K_1 = 2\int_{\mathcal{S}\backslash J_M^{\epsilon/(8T)}} \Big|g_{x|\tilde{Z}}(y) - f(y|x)\Big|dy
$$

$$= \int_{\mathcal{S}\setminus J_M^{\epsilon/(8T)}} \left|(2^M/T)G_{x|\tilde{Z}}(B_y) - f(y|x) + f(b_y|x) - f(b_y|x)\right| dy \tag{E.34}$$

$$\le \sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left\{\left|G_{x|\tilde{Z}}(B) - F_x(B)\right| + \int_B |f(b_y|x) - f(y|x)|dy\right\}, \tag{E.35}$$

where b_y satisfies the fact that $F_x(B_y) = f(b_y|x)v_{B_y}$ with $B_y \in \pi_M^*$ being the subset that the point y belongs to and v_{B_y} being the volume of B_y. (E.34) is due to the fact that y is uniformly distributed on $\mathcal{B}_{\{M\}}$ in NNPT for any $\mathcal{B}_{\{M\}} \in \pi_M$ so that $g_{x|\tilde{Z}}(y) = (2^M/T)G_{x|\tilde{Z}}(B_y)$. Since $\forall B \in \pi_M$ and $b_y, y \in B$, we have $|f(b_y|x) - f(y|x)| \le \epsilon/(8T)$. Then

$$\sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left\{\int_B \left|f(b_y|x) - f(y|x)\right|dy\right\} \le \epsilon/8. \tag{E.36}$$

By (E.33), (E.35) and (E.36), we can get

$$\begin{aligned} D\Big(G_{x|\tilde{Z}}, F_x\Big) &= 2K_1 + \frac{\epsilon}{2} \\ &\le 2\sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left|G_{x|\tilde{Z}}(B) - F_x(B)\right| + \frac{3}{4}\epsilon \end{aligned}$$

Then $P\Big(D\Big(G_{x|\tilde{Z}}, F_x\Big) > \epsilon\Big) \le P\Big(\sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left|G_{x|\tilde{Z}}(B) - F_x(B)\right| > \frac{\epsilon}{8}\Big)$.

Now we consider the second probability,

$$P\Big(\sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left|G_{x|\tilde{Z}}(B) - F_x(B)\right| > \frac{\epsilon}{8}\Big)$$

$$\le P\Big(\max_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \left|G_{x|\tilde{Z}}(B) - F_x(B)\right| \ge \frac{\epsilon}{2^{M+3}}\Big) \tag{E.37}$$

$$\le P\Big(\bigcup_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} \Big[\left|G_{x|\tilde{Z}}(B) - F_x(B)\right| \ge \frac{\epsilon}{2^{M+3}}\Big]\Big) \tag{E.38}$$

$$\le \sum_{B\in\pi_M\setminus I_M^{\epsilon/(8T)}} P\Big(\left|G_{x|\tilde{Z}}(B) - F_x(B)\right| \ge \frac{\epsilon}{2^{M+3}}\Big)$$

$$
\begin{aligned}
&\leq 2^M\left(\frac{2^{M+3}}{\epsilon}\right)^2\left\{\sup_{B\in\pi_M\backslash I_M^{\epsilon/(8T)}}\left|E[G_{x|\tilde{Z}}(B)]-F_x(B)\right|^2+\sup_{B\in\pi_M\backslash I_M^{\epsilon/(8T)}}\mathrm{var}[G_{x|\tilde{Z}}(B)]\right\}\\
&= 2^{3M}\max\left\{O_p\left(\frac{M}{n^\eta\gamma(M)}\right)^2, O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right)^2, O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)^2, O_p\left(\frac{M}{n^{1-\eta}\gamma(M)}\right)\right\}\\
&= 2^{3M}\max\left\{O_p\left(\frac{M}{n^\eta\gamma(M)}\right)^2, O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right)^2, O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)}\right)^2\right\}
\end{aligned}
$$

where (E.37) and (E.38) are obtained by the fact that for a series of scalars a_j with $j=1,\ldots,m$,

$$
\left\{\sum_{j=1}^m a_j > \epsilon/4\right\}\subset\left\{\max_j a_j \geq \frac{\epsilon}{4m}\right\}\subset\bigcup_{j=1}^m\left\{a_j\geq\frac{\epsilon}{4m}\right\},
$$

In (E.37), since $\pi_M\backslash I_M^{\epsilon/(8T)}$ is set with finite element, the maximum value exists.

Now we find a lower bound for $\gamma(M)$ in the set $\pi_M\backslash I_M^{\epsilon/(8T)}$. By the definition of I_M^Δ and J_M^Δ, $\inf_{y\in\mathcal{S}\backslash J_M^{\epsilon/(8T)}} f(y|x)\geq\epsilon/(8T)$, we can get $\gamma(M)\geq\{\epsilon/(8T)\}\times v_{\mathcal{B}_{\{M\}}}=\{\epsilon/(8T)\}\times\frac{T}{2^M}=\epsilon/2^{M+3}$, where $v_{\mathcal{B}_{\{M\}}}$ is volume of $\mathcal{B}_{\{M\}}$.

Then, with $n=O(2^{\frac{5M}{\eta^*}}M^{\frac{3}{\eta^*}})$ for $\eta^*=\min\{\eta,1-\eta\}$, as $M\to\infty$, we could get

$$
\begin{aligned}
&P\left(D\left(G_{x|\tilde{Z}},F_x\right)>\epsilon\right)\\
&\leq 2^{3M}\max\left\{O_p\left(\frac{M}{n^\eta\gamma(M)}\right)^2, O_p\left(\frac{M}{n^{\frac{1-\eta}{2}}\gamma(M)}\right)^2, O_p\left(\frac{M^3}{n^{1-\eta}\gamma(M)^2}\right)^2\Bigg)\right\}\\
&\leq 2^{3M}\max\left\{O_p\left(\frac{M^2 2^{2M+6}}{n^{2\eta}}\right), O_p\left(\frac{M^2 2^{2M+6}}{n^{1-\eta}}\right), O_p\left(\frac{M^6 2^{2M+6}}{n^{2(1-\eta)}}\right)\right\}\\
&\xrightarrow{p} 0
\end{aligned}
$$

E.2 Additional Simulation Results

E.2.1 Setting 6.1: Monte Carlo-Based Results

Table E.1: Setting 6.1: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.1.1	0.189 (0.24)	0.131 (0.16)	0.103 (0.12)	0.084 (0.10)	0.062 (0.08)	0.209 (0.24)	0.136 (0.16)	0.101 (0.12)	0.082 (0.10)	0.041 (0.08)	0.245 (0.24)	0.162 (0.16)	0.119 (0.12)	0.082 (0.10)	0.059 (0.08)
6.1.2	0.245 (0.26)	0.179 (0.18)	0.142 (0.14)	0.113 (0.12)	0.084 (0.12)	0.250 (0.26)	0.168 (0.18)	0.121 (0.14)	0.085 (0.10)	0.052 (0.08)	0.279 (0.26)	0.189 (0.18)	0.138 (0.14)	0.100 (0.10)	0.064 (0.08)
6.1.3	0.678 (0.50)	0.580 (0.40)	0.520 (0.34)	0.471 (0.32)	0.425 (0.28)	0.591 (0.46)	0.438 (0.32)	0.338 (0.26)	0.258 (0.20)	0.181 (0.16)	0.513 (0.40)	0.336 (0.26)	0.232 (0.20)	0.157 (0.14)	0.094 (0.10)
6.1.4	0.660 (0.50)	0.570 (0.42)	0.512 (0.38)	0.466 (0.36)	0.422 (0.32)	0.576 (0.46)	0.433 (0.39)	0.340 (0.30)	0.264 (0.24)	0.190 (0.20)	0.502 (0.42)	0.335 (0.30)	0.236 (0.22)	0.162 (0.16)	0.099 (0.12)
6.1.5	0.601 (0.38)	0.514 (0.35)	0.478 (0.35)	0.474 (0.35)	0.472 (0.38)	0.725 (0.46)	0.654 (0.43)	0.577 (0.41)	0.496 (0.35)	0.396 (0.32)	0.705 (0.43)	0.543 (0.35)	0.417 (0.30)	0.306 (0.24)	0.190 (0.22)
6.1.6	0.598 (0.43)	0.510 (0.41)	0.493 (0.38)	0.489 (0.35)	0.485 (0.41)	0.705 (0.51)	0.651(0.46)	0.572 (0.43)	0.494 (0.38)	0.414 (0.35)	0.687 (0.46)	0.545 (0.38)	0.408 (0.32)	0.317 (0.27)	0.194 (0.22)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.1.1	0.304 (0.26)	0.214 (0.20)	0.153 (0.16)	0.118 (0.12)	0.088 (0.08)	0.385 (0.28)	0.292 (0.22)	0.235 (0.18)	0.181 (0.16)	0.141 (0.12)	0.161 (0.22)	0.115 (0.15)	0.093 (0.12)	0.076 (0.10)	0.057 (0.09)
6.1.2	0.333 (0.28)	0.241 (0.20)	0.187 (0.16)	0.143 (0.12)	0.100 (0.10)	0.410 (0.30)	0.318 (0.40)	0.261 (0.20)	0.212 (0.16)	0.160 (0.14)	0.214 (0.25)	0.155 (0.17)	0.122 (0.14)	0.096 (0.12)	0.071 (0.11)
6.1.3	0.498 (0.38)	0.335 (0.24)	0.244 (0.18)	0.178 (0.14)	0.120 (0.10)	0.550 (0.38)	0.412 (0.26)	0.329 (0.22)	0.262 (0.18)	0.195 (0.14)	0.607 (0.48)	0.514 (0.35)	0.460 (0.30)	0.416 (0.29)	0.374 (0.26)
6.1.4	0.486 (0.40)	0.330 (0.28)	0.240 (0.20)	0.175 (0.14)	0.118 (0.10)	0.534 (0.40)	0.399 (0.28)	0.319 (0.24)	0.253 (0.18)	0.189 (0.14)	0.593 (0.48)	0.507 (0.39)	0.456 (0.34)	0.413 (0.33)	0.372 (0.29)
6.1.5	0.674 (0.43)	0.488 (0.32)	0.363 (0.27)	0.261 (0.22)	0.164 (0.16)	0.703 (0.43)	0.537 (0.32)	0.430 (0.27)	0.343 (0.22)	0.249 (0.19)	0.574 (0.39)	0.503 (0.36)	0.473 (0.35)	0.471 (0.34)	0.490 (0.37)
6.1.6	0.653 (0.46)	0.472 (0.35)	0.351 (0.30)	0.276 (0.22)	0.161 (0.16)	0.710 (0.46)	0.545 (0.35)	0.434 (0.30)	0.353 (0.24)	0.241 (0.22)	0.576 (0.44)	0.507 (0.39)	0.478 (0.37)	0.472 (0.36)	0.485 (0.40)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.1.1	0.175 (0.23)	0.115 (0.14)	0.085 (0.11)	0.063 (0.09)	0.042 (0.07)	0.203 (0.24)	0.135 (0.16)	0.100 (0.12)	0.075 (0.09)	0.052 (0.07)	0.253 (0.26)	0.178 (0.18)	0.136 (0.15)	0.105 (0.11)	0.076 (0.09)
6.1.2	0.214 (0.25)	0.142 (0.16)	0.102 (0.13)	0.072 (0.10)	0.044 (0.07)	0.237 (0.25)	0.159 (0.17)	0.116 (0.13)	0.084 (0.10)	0.054 (0.08)	0.283 (0.27)	0.204 (0.19)	0.157 (0.15)	0.120 (0.12)	0.084 (0.09)
6.1.3	0.512 (0.43)	0.371 (0.29)	0.284 (0.23)	0.217 (0.19)	0.151 (0.15)	0.435 (0.39)	0.277 (0.24)	0.189 (0.17)	0.128 (0.13)	0.078 (0.09)	0.420 (0.37)	0.276 (0.23)	0.199 (0.16)	0.146 (0.13)	0.099 (0.10)
6.1.4	0.502 (0.43)	0.370 (0.32)	0.289 (0.26)	0.224 (0.22)	0.161 (0.18)	0.427 (0.39)	0.279 (0.27)	0.194 (0.20)	0.134 (0.15)	0.082 (0.11)	0.411 (0.38)	0.272 (0.25)	0.197 (0.18)	0.144 (0.13)	0.097 (0.10)
6.1.5	0.676 (0.45)	0.592 (0.41)	0.513 (0.37)	0.436 (0.32)	0.346 (0.29)	0.634 (0.41	0.472 (0.34)	0.355 (0.28)	0.256 (0.22)	0.157 (0.17)	0.592 (0.40)	0.414 (0.30)	0.300 (0.24)	0.212 (0.18)	0.132 (0.14)
6.1.6	0.663 (0.48)	0.582 (0.43)	0.507 (0.39)	0.434 (0.34)	0.348 (0.32)	0.623 (0.44)	0.469 (0.36)	0.358 (0.31)	0.263 (0.24)	0.165 (0.20)	0.583 (0.42)	0.413 (0.32)	0.302 (0.26)	0.216 (0.19)	0.135 (0.15)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.1.1	0.322 (0.27)	0.243 (0.21)	0.195 (0.17)	0.157 (0.14)	0.118 (0.12)	0.181 (0.56)	0.162 (0.46)	0.158 (0.44)	0.149 (0.40)	0.140 (0.42)	0.362 (0.28)	0.263 (0.22)	0.204 (0.18)	0.161 (0.14)	0.105 (0.10)
6.1.2	0.350 (0.29)	0.270 (0.22)	0.221 (0.19)	0.179 (0.15)	0.135 (0.12)	0.221 (0.64)	0.185 (0.50)	0.179 (0.44)	0.166 (0.40)	0.159 (0.42)	0.404 (0.30)	0.302 (0.24)	0.243 (0.20)	0.181 (0.16)	0.125 (0.12)
6.1.3	0.470 (0.37)	0.346 (0.25)	0.273 (0.20)	0.216 (0.16)	0.160 (0.13)	0.302 (0.54)	0.243 (0.44)	0.224 (0.44)	0.205 (0.42)	0.186 (0.44)	0.638 (0.42)	0.441 (0.30)	0.322 (0.22)	0.223 (0.16)	0.144 (0.12)
6.1.4	0.456 (0.39)	0.336 (0.26)	0.265 (0.22)	0.210 (0.16)	0.156 (0.14)	0.319 (0.58)	0.259 (0.44)	0.238 (0.44)	0.222 (0.42)	0.201 (0.42)	0.622 (0.44)	0.421 (0.32)	0.323 (0.24)	0.222 (0.18)	0.146 (0.14)
6.1.5	0.613 (0.40)	0.455 (0.31)	0.358 (0.25)	0.281 (0.21)	0.202 (0.17)	0.465 (0.57)	0.410 (0.51)	0.351 (0.46)	0.320 (0.46)	0.298 (0.46)	0.785 (0.46)	0.592 (0.38)	0.464 (0.32)	0.348 (0.24)	0.217 (0.19)
6.1.6	0.603 (0.43)	0.451 (0.32)	0.355 (0.27)	0.279 (0.21)	0.201 (0.18)	0.495 (0.65)	0.408 (0.54)	0.383 (0.49)	0.352 (0.46)	0.301 (0.46)	0.766 (0.49)	0.587 (0.38)	0.466 (0.35)	0.349 (0.27)	0.223 (0.22)

Table E.2: Setting 6.1: Square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.1.1	0.200 (0.15)	0.164 (0.10)	0.152 (0.08)	0.132 (0.06)	0.121 (0.04)	0.207 (0.13)	0.162 (0.11)	0.129 (0.08)	0.107 (0.07)	0.082 (0.05)	0.222 (0.12)	0.176 (0.10)	0.138 (0.08)	0.123 (0.07)	0.093 (0.04)
6.1.2	0.227 (0.14)	0.192 (0.10)	0.172 (0.08)	0.158 (0.08)	0.142 (0.06)	0.227 (0.13)	0.180 (0.11)	0.149 (0.09)	0.124 (0.08)	0.097 (0.05)	0.238 (0.13)	0.190 (0.11)	0.156 (0.09)	0.127 (0.08)	0.098 (0.05)
6.1.3	0.356 (0.10)	0.346 (0.08)	0.338 (0.08)	0.329 (0.07)	0.317 (0.07)	0.332 (0.09)	0.298 (0.08)	0.269 (0.08)	0.240 (0.06)	0.204 (0.06)	0.304 (0.09)	0.252 (0.08)	0.209 (0.08)	0.169 (0.07)	0.125 (0.05)
6.1.4	0.360 (0.11)	0.351 (0.10)	0.344 (0.10)	0.335 (0.10)	0.325 (0.09)	0.338 (0.11)	0.307 (0.10)	0.282 (0.10)	0.256 (0.09)	0.224 (0.08)	0.312 (0.11)	0.265 (0.11)	0.225 (0.10)	0.185 (0.08)	0.141 (0.06)
6.1.5	0.338 (0.08)	0.323 (0.08)	0.320 (0.08)	0.326 (0.08)	0.342 (0.08)	0.366 (0.10)	0.358 (0.09)	0.346 (0.09)	0.329 (0.08)	0.301 (0.08)	0.356 (0.09)	0.322 (0.08)	0.289 (0.08)	0.252 (0.07)	0.199 (0.06)
6.1.6	0.348 (0.10)	0.332 (0.10)	0.329 (0.10)	0.327 (0.09)	0.330 (0.10)	0.368 (0.11)	0.357 (0.10)	0.346 (0.10)	0.338 (0.10)	0.305 (0.10)	0.31 (0.11)	0.332 (0.10)	0.304 (0.10)	0.269 (0.09)	0.209 (0.09)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.1.1	0.245 (0.10)	0.205 (0.09)	0.168 (0.08)	0.147 (0.07)	0.120 (0.05)	0.273 (0.09)	0.239 (0.08)	0.203 (0.07)	0.192 (0.06)	0.161 (0.05)	0.185 (0.16)	0.154 (0.11)	0.138 (0.08)	0.127 (0.06)	0.114 (0.04)
6.1.2	0.259 (0.11)	0.217 (0.10)	0.187 (0.09)	0.158 (0.07)	0.126 (0.05)	0.285 (0.10)	0.252 (0.09)	0.226 (0.08)	0.201 (0.07)	0.170 (0.06)	0.210 (0.15)	0.179 (0.11)	0.162 (0.08)	0.149 (0.08)	0.134 (0.06)
6.1.3	0.295 (0.09)	0.241 (0.08)	0.201 (0.08)	0.166 (0.06)	0.130 (0.05)	0.306 (0.09)	0.263 (0.08)	0.232 (0.07)	0.204 (0.06)	0.171 (0.05)	0.345 (0.10)	0.332 (0.08)	0.322 (0.08)	0.311 (0.07)	0.298 (0.07)
6.1.4	0.303 (0.11)	0.251 (0.10)	0.210 (0.09)	0.174 (0.08)	0.135 (0.05)	0.312 (0.11)	0.269 (0.10)	0.237 (0.09)	0.208 (0.07)	0.174 (0.06)	0.349 (0.11)	0.338 (0.10)	0.329 (0.10)	0.319 (0.10)	0.307 (0.09)
6.1.5	0.346 (0.09)	0.297 (0.08)	0.254 (0.07)	0.211 (0.07)	0.159 (0.05)	0.350 (0.09)	0.304 (0.08)	0.268 (0.07)	0.234 (0.07)	0.193 (0.06)	0.336 (0.09)	0.325 (0.08)	0.323 (0.08)	0.328 (0.08)	0.339 (0.08)
6.1.6	0.351 (0.11)	0.307 (0.10)	0.265 (0.09)	0.221 (0.08)	0.168 (0.07)	0.353 (0.11)	0.309 (0.10)	0.278 (0.09)	0.244 (0.08)	0.196 (0.07)	0.345 (0.10)	0.335 (0.10)	0.332 (0.10)	0.336 (0.10)	0.344 (0.10)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.1.1	0.189 (0.15)	0.148 (0.12)	0.124 (0.09)	0.105 (0.08)	0.086 (0.05)	0.202 (0.14)	0.160 (0.11)	0.133 (0.09)	0.112 (0.07)	0.091 (0.05)	0.225 (0.12)	0.187 (0.10)	0.160 (0.09)	0.138 (0.07)	0.113 (0.05)
6.1.2	0.208 (0.15)	0.165 (0.12)	0.138 (0.09)	0.117 (0.08)	0.094 (0.05)	0.217 (0.14)	0.172 (0.12)	0.143 (0.09)	0.119 (0.08)	0.094 (0.05)	0.238 (0.13)	0.198 (0.11)	0.169 (0.09)	0.145 (0.07)	0.117 (0.05)
6.1.3	0.314 (0.10)	0.278 (0.09)	0.249 (0.08)	0.221 (0.06)	0.187 (0.06)	0.284 (0.11)	0.231 (0.09)	0.190 (0.08)	0.154 (0.07)	0.117 (0.05)	0.273 (0.11)	0.219 (0.09)	0.181 (0.08)	0.151 (0.06)	0.120 (0.05)
6.1.4	0.322 (0.12)	0.290 (0.11)	0.263 (0.10)	0.238 (0.09)	0.207 (0.08)	0.284 (0.11)	0.231 (0.09)	0.190 (0.08)	0.154 (0.07)	0.117 (0.05)	0.281 (0.12)	0.229 (0.11)	0.190 (0.09)	0.157 (0.07)	0.125 (0.05)
6.1.5	0.358 (0.09)	0.347 (0.09)	0.332 (0.09)	0.312 (0.08)	0.283 (0.07)	0.342 (0.09)	0.304 (0.08)	0.269 (0.08)	0.232 (0.07)	0.183 (0.06)	0.327 (0.09)	0.275 (0.08)	0.231 (0.08)	0.190 (0.07)	0.143 (0.05)
6.1.6	0.363 (0.11)	0.352 (0.10)	0.337 (0.10)	0.319 (0.10)	0.292 (0.10)	0.348 (0.10)	0.312 (0.10)	0.279 (0.10)	0.244 (0.09)	0.197 (0.09)	0.333 (0.11)	0.284 (0.10)	0.241 (0.09)	0.200 (0.08)	0.151 (0.07)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.1.1	0.252 (0.10)	0.220 (0.08)	0.196 (0.08)	0.174 (0.07)	0.148 (0.05)	0.212 (0.23)	0.206 (0.18)	0.202 (0.14)	0.196 (0.13)	0.193 (0.11)	0.292 (0.07)	0.251 (0.07)	0.223 (0.07)	0.194 (0.06)	0.137 (0.05)
6.1.2	0.264 (0.11)	0.231 (0.10)	0.207 (0.09)	0.183 (0.07)	0.154 (0.06)	0.216 (0.22)	0.208 (0.18)	0.204 (0.15)	0.201 (0.13)	0.197 (0.12)	0.321 (0.08)	0.272 (0.08)	0.233 (0.08)	0.204 (0.07)	0.151 (0.05)
6.1.3	0.283 (0.10)	0.240 (0.09)	0.210 (0.08)	0.184 (0.06)	0.155 (0.05)	0.252 (0.16)	0.231 (0.13)	0.223 (0.12)	0.217 (0.10)	0.209 (0.10)	0.343 (0.08)	0.292 (0.08)	0.252 (0.08)	0.205 (0.06)	0.151 (0.05)
6.1.4	0.289 (0.12)	0.246 (0.10)	0.215 (0.09)	0.188 (0.07)	0.158 (0.06)	0.263 (0.17)	0.249 (0.14)	0.238 (0.13)	0.229 (0.12)	0.220 (0.11)	0.339 (0.10)	0.301 (0.10)	0.248 (0.09)	0.209 (0.08)	0.162 (0.06)
6.1.5	0.327 (0.10)	0.279 (0.08)	0.243 (0.08)	0.210 (0.07)	0.172 (0.05)	0.318 (0.12)	0.301 (0.11)	0.288 (0.11)	0.271 (0.10)	0.250 (0.09)	0.369 (0.09)	0.331 (0.08)	0.287 (0.07)	0.246 (0.07)	0.192 (0.06)
6.1.6	0.334 (0.11)	0.286 (0.10)	0.249 (0.09)	0.215 (0.08)	0.176 (0.07)	0.322 (0.15)	0.309 (0.14)	0.292 (0.12)	0.280 (0.12)	0.257 (0.11)	0.372 (0.10)	0.334 (0.09)	0.292 (0.09)	0.251 (0.08)	0.201 (0.08)

E.2.2 Setting 6.1: Grid-Based Results

Table E.3: Setting 6.1: Grid-based K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.1.1	0.109 (0.25)	0.089 (0.15)	0.076 (0.11)	0.063 (0.09)	0.048 (0.10)	0.107	0.080 (0.16)	0.064 (0.10)	0.048 (0.07)	0.031 (0.07)	0.117 (0.30)	0.089 (0.18)	0.072 (0.10)	0.055 (0.06)	0.038 (0.05)
6.1.2	0.149 (0.30)	0.121 (0.15)	0.100 (0.12)	0.083 (0.11)	0.066 (0.12)	0.141 (0.32)	0.104 (0.16)	0.078 (0.12)	0.057 (0.09)	0.037 (0.08)	0.152 (0.36)	0.115 (0.18)	0.088 (0.13)	0.067 (0.08)	0.045 (0.06)
6.1.3	0.498 (0.29)	0.462 (0.14)	0.431 (0.11)	0.400 (0.10)	0.362 (0.09)	0.384 (0.29)	0.305 (0.15)	0.247 (0.12)	0.195 (0.09)	0.138 (0.08)	0.279 (0.29)	0.196 (0.15)	0.143 (0.11)	0.100 (0.08)	0.061 (0.06)
6.1.4	0.495 (0.32)	0.460 (0.17)	0.429 (0.13)	0.399 (0.11)	0.361 (0.11)	0.390 (0.32)	0.314 (0.17)	0.257 (0.13)	0.207 (0.10)	0.149 (0.09)	0.293 (0.33)	0.210 (0.18)	0.156 (0.12)	0.112 (0.09)	0.069 (0.07)
6.1.5	0.459 (0.25)	0.443 (0.14)	0.440 (0.10)	0.452 (0.09)	0.482 (0.09)	0.557 (0.29)	0.523 (0.15)	0.470 (0.10)	0.411 (0.07)	0.334 (0.07)	0.474 (0.29)	0.363 (0.16)	0.281 (0.11)	0.208 (0.08)	0.131 (0.06)
6.1.6	0.469 (0.28)	0.452 (0.15)	0.445 (0.11)	0.452 (0.10)	0.473 (0.09)	0.554 (0.32)	0.520 (0.17)	0.467 (0.11)	0.409 (0.08)	0.334 (0.07)	0.480 (0.33)	0.374 (0.18)	0.292 (0.12)	0.220 (0.09)	0.141 (0.06)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					LDTFP1 (linear predictor)				
6.1.1	0.143 (0.35)	0.116 (0.22)	0.097 (0.12)	0.078 (0.07)	0.057 (0.04)	0.183 (0.42)	0.160 (0.27)	0.140 (0.16)	0.118 (0.10)	0.091 (0.06)	0.054 (0.12)	0.047 (0.05)	0.045 (0.03)	0.044 (0.02)	0.043 (0.01)
6.1.2	0.184 (0.43)	0.151 (0.23)	0.125 (0.16)	0.100 (0.10)	0.072 (0.06)	0.232 (0.52)	0.209 (0.30)	0.185 (0.22)	0.158 (0.15)	0.124 (0.09)	0.098 (0.20)	0.085 (0.09)	0.073 (0.06)	0.062 (0.03)	0.048 (0.02)
6.1.3	0.234 (0.31)	0.172 (0.17)	0.135 (0.12)	0.104 (0.08)	0.073 (0.05)	0.242 (0.35)	0.203 (0.21)	0.175 (0.14)	0.149 (0.10)	0.117 (0.06)	0.434 (0.16)	0.429 (0.05)	0.425 (0.03)	0.423 (0.02)	0.420 (0.01)
6.1.4	0.250 (0.35)	0.185 (0.20)	0.145 (0.13)	0.113 (0.10)	0.079 (0.06)	0.258 (0.40)	0.216 (0.24)	0.186 (0.16)	0.158 (0.12)	0.125 (0.07)	0.436 (0.23)	0.437 (0.09)	0.429 (0.11)	0.392 (0.03)	0.344 (0.02)
6.1.5	0.360 (0.32)	0.251 (0.18)	0.184 (0.12)	0.132 (0.08)	0.085 (0.05)	0.303 (0.34)	0.229 (0.21)	0.187 (0.15)	0.154 (0.10)	0.119 (0.06)	0.436 (0.12)	0.430 (0.06)	0.425 (0.03)	0.424 (0.02)	0.420 (0.01)
6.1.6	0.377 (0.37)	0.271 (0.21)	0.200 (0.14)	0.145 (0.09)	0.093 (0.06)	0.328 (0.42)	0.254 (0.25)	0.207 (0.18)	0.170 (0.12)	0.130 (0.07)	0.441 (0.15)	0.437 (0.08)	0.432 (0.05)	0.430 (0.03)	0.423 (0.02)
	LDTFP2 (quadratic predictor)					kernel density estimation					PT density estimation				
6.1.1	0.017 (0.11)	0.009 (0.06)	0.005 (0.02)	0.004 (0.02)	0.003 (0.01)	0.137 (0.46)	0.123 (0.37)	0.117 (0.36)	0.113 (0.34)	0.111 (0.34)	0.188 (0.43)	0.140 (0.26)	0.109 (0.14)	0.083 (0.07)	0.056 (0.03)
6.1.2	0.073 (0.19)	0.058 (0.10)	0.043 (0.06)	0.031 (0.04)	0.018 (0.02)	0.143 (0.46)	0.130 (0.37)	0.123 (0.35)	0.117 (0.33)	0.114 (0.33)	0.250 (0.58)	0.196 (0.30)	0.152 (0.20)	0.113 (0.13)	0.073 (0.06)
6.1.3	0.017 (0.11)	0.009 (0.06)	0.005 (0.02)	0.004 (0.02)	0.003 (0.01)	0.248 (0.42)	0.204 (0.34)	0.180 (0.32)	0.159 (0.31)	0.143 (0.33)	0.362 (0.38)	0.272 (0.19)	0.210 (0.12)	0.156 (0.07)	0.099 (0.04)
6.1.4	0.083 (0.19)	0.062 (0.10)	0.046 (0.06)	0.032 (0.04)	0.018 (0.02)	0.261 (0.43)	0.219 (0.34)	0.194 (0.32)	0.172 (0.30)	0.153 (0.32)	0.377 (0.44)	0.289 (0.25)	0.224 (0.16)	0.169 (0.11)	0.111 (0.07)
6.1.5	0.177 (0.24)	0.234 (2.71)	0.162 (0.88)	0.179 (1.65)	0.300 (3.00)	0.417 (0.43)	0.363 (0.35)	0.327 (0.32)	0.296 (0.32)	0.259 (0.31)	0.493 (0.35)	0.404 (0.16)	0.327 (0.09)	0.253 (0.06)	0.168 (0.04)
6.1.6	0.251 (0.54)	0.342 (1.32)	0.220 (0.65)	0.243 (0.73)	0.310 (1.27)	0.419 (0.43)	0.370 (0.34)	0.335 (0.32)	0.304 (0.31)	0.267 (0.31)	0.507 (0.43)	0.413 (0.23)	0.329 (0.16)	0.251 (0.13)	0.164 (0.09)

Table E.4: Setting 6.1: Grid-based square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.1.1	0.175 (0.19)	0.154 (0.11)	0.141 (0.08)	0.130 (0.06)	0.117 (0.04)	0.171 (0.19)	0.142 (0.13)	0.122 (0.10)	0.108 (0.08)	0.090 (0.05)	0.175 (0.19)	0.145 (0.13)	0.126 (0.10)	0.111 (0.07)	0.094 (0.05)
6.1.2	0.199 (0.15)	0.176 (0.11)	0.161 (0.08)	0.148 (0.06)	0.134 (0.05)	0.188 (0.17)	0.156 (0.13)	0.134 (0.10)	0.116 (0.08)	0.095 (0.05)	0.188 (0.17)	0.156 (0.13)	0.135 (0.10)	0.116 (0.08)	0.096 (0.05)
6.1.3	0.343 (0.07)	0.331 (0.05)	0.321 (0.04)	0.310 (0.03)	0.296 (0.02)	0.299 (0.10)	0.266 (0.07)	0.241 (0.06)	0.216 (0.04)	0.184 (0.03)	0.251 (0.12)	0.206 (0.08)	0.175 (0.07)	0.147 (0.06)	0.115 (0.04)
6.1.4	0.348 (0.07)	0.336 (0.05)	0.327 (0.04)	0.318 (0.03)	0.305 (0.02)	0.310 (0.09)	0.281 (0.07)	0.258 (0.06)	0.235 (0.04)	0.205 (0.03)	0.269 (0.12)	0.226 (0.09)	0.194 (0.07)	0.164 (0.06)	0.128 (0.04)
6.1.5	0.329 (0.07)	0.321 (0.04)	0.322 (0.03)	0.328 (0.02)	0.338 (0.01)	0.361 (0.07)	0.349 (0.04)	0.332 (0.03)	0.311 (0.02)	0.281 (0.02)	0.330 (0.07)	0.286 (0.06)	0.253 (0.05)	0.218 (0.04)	0.174 (0.03)
6.1.6	0.339 (0.06)	0.332 (0.04)	0.331 (0.02)	0.335 (0.02)	0.343 (0.01)	0.365 (0.06)	0.354 (0.04)	0.338 (0.03)	0.318 (0.02)	0.291 (0.02)	0.338 (0.07)	0.298 (0.06)	0.265 (0.05)	0.232 (0.04)	0.189 (0.03)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					LDTFP1 (linear predictor)				
6.1.1	0.192 (0.18)	0.164 (0.12)	0.144 (0.09)	0.129 (0.07)	0.111 (0.04)	0.216 (0.18)	0.192 (0.12)	0.173 (0.09)	0.157 (0.07)	0.138 (0.04)	0.123 (0.12)	0.117 (0.06)	0.114 (0.04)	0.113 (0.03)	0.111 (0.02)
6.1.2	0.203 (0.18)	0.175 (0.13)	0.156 (0.10)	0.137 (0.08)	0.116 (0.05)	0.227 (0.19)	0.205 (0.15)	0.188 (0.13)	0.170 (0.10)	0.149 (0.07)	0.162 (0.14)	0.142 (0.10)	0.129 (0.08)	0.121 (0.07)	0.112 (0.04)
6.1.3	0.224 (0.13)	0.184 (0.10)	0.159 (0.08)	0.138 (0.07)	0.116 (0.04)	0.223 (0.14)	0.194 (0.09)	0.176 (0.07)	0.159 (0.06)	0.140 (0.04)	0.321 (0.04)	0.320 (0.02)	0.319 (0.02)	0.318 (0.01)	0.317 (0.01)
6.1.4	0.243 (0.13)	0.200 (0.10)	0.171 (0.08)	0.147 (0.07)	0.120 (0.09)	0.240 (0.15)	0.208 (0.11)	0.187 (0.08)	0.169 (0.07)	0.146 (0.04)	0.328 (0.07)	0.327 (0.04)	0.323 (0.05)	0.311 (0.10)	0.299 (0.09)
6.1.5	0.284 (0.11)	0.232 (0.07)	0.195 (0.06)	0.162 (0.05)	0.128 (0.04)	0.257 (0.12)	0.212 (0.08)	0.186 (0.07)	0.164 (0.06)	0.141 (0.04)	0.323 (0.04)	0.320 (0.02)	0.319 (0.02)	0.318 (0.01)	0.317 (0.01)
6.1.6	0.298 (0.11)	0.247 (0.08)	0.209 (0.06)	0.174 (0.05)	0.135 (0.04)	0.273 (0.12)	0.228 (0.09)	0.199 (0.08)	0.175 (0.06)	0.148 (0.04)	0.332 (0.04)	0.329 (0.03)	0.328 (0.02)	0.327 (0.01)	0.325 (0.01)
	LDTFP2 (quadratic predictor)					kernel density estimation					PT density estimation				
6.1.1	0.067 (0.20)	0.053 (0.14)	0.044 (0.09)	0.042 (0.07)	0.034 (0.06)	0.207 (0.23)	0.200 (0.17)	0.197 (0.13)	0.197 (0.11)	0.197 (0.09)	0.216 (0.18)	0.179 (0.12)	0.152 (0.10)	0.132 (0.07)	0.109 (0.04)
6.1.2	0.135 (0.14)	0.106 (0.12)	0.084 (0.10)	0.070 (0.07)	0.056 (0.07)	0.210 (0.22)	0.202 (0.16)	0.199 (0.13)	0.197 (0.10)	0.197 (0.08)	0.238 (0.23)	0.204 (0.17)	0.178 (0.13)	0.154 (0.09)	0.125 (0.05)
6.1.3	0.066 (0.20)	0.053 (0.14)	0.044 (0.09)	0.042 (0.07)	0.034 (0.06)	0.246 (0.14)	0.230 (0.11)	0.220 (0.09)	0.213 (0.08)	0.207 (0.06)	0.274 (0.12)	0.228 (0.07)	0.197 (0.06)	0.168 (0.05)	0.135 (0.04)
6.1.4	0.136 (0.15)	0.106 (0.11)	0.085 (0.10)	0.070 (0.07)	0.057 (0.07)	0.258 (0.13)	0.242 (0.10)	0.231 (0.09)	0.223 (0.07)	0.215 (0.06)	0.289 (0.13)	0.247 (0.11)	0.213 (0.09)	0.182 (0.07)	0.145 (0.04)
6.1.5	0.214 (0.13)	0.218 (0.52)	0.203 (0.20)	0.203 (0.33)	0.233 (0.70)	0.316 (0.10)	0.297 (0.08)	0.283 (0.07)	0.269 (0.05)	0.253 (0.04)	0.327 (0.08)	0.286 (0.06)	0.251 (0.05)	0.216 (0.04)	0.172 (0.03)
6.1.6	0.280 (0.25)	0.287 (0.65)	0.265 (0.33)	0.290 (0.25)	0.221 (0.65)	0.323 (0.10)	0.304 (0.08)	0.290 (0.06)	0.277 (0.05)	0.260 (0.04)	0.335 (0.08)	0.294 (0.06)	0.259 (0.05)	0.223 (0.05)	0.177 (0.04)

E.2.3 Setting 6.2: Monte Carlo-Based Results

Table E.5: Setting 6.2: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.2.1	0.159 (0.24)	0.103 (0.13)	0.080 (0.11)	0.064 (0.09)	0.048 (0.08)	0.178 (0.25)	0.113 (0.14)	0.084 (0.11)	0.061 (0.08)	0.041 (0.06)	0.213 (0.25)	0.141 (0.15)	0.105 (0.12)	0.077 (0.09)	0.054 (0.07)
6.2.2	0.201 (0.26)	0.144 (0.15)	0.116 (0.13)	0.092 (0.11)	0.069 (0.10)	0.207 (0.26)	0.139 (0.15)	0.103 (0.12)	0.073 (0.09)	0.046 (0.07)	0.411 (0.31)	0.282 (0.24)	0.204 (0.19)	0.143 (0.14)	0.087 (0.10)
6.2.3	0.462 (0.36)	0.398 (0.33)	0.360 (0.31)	0.328 (0.29)	0.297 (0.28)	0.437 (0.33)	0.330 (0.28)	0.258 (0.24)	0.196 (0.19)	0.133 (0.16)	0.411 (0.31)	0.282 (0.24)	0.204 (0.19)	0.143 (0.14)	0.087 (0.10)
6.2.4	0.467 (0.38)	0.404 (0.35)	0.368 (0.34)	0.336 (0.32)	0.306 (0.31)	0.440 (0.35)	0.337 (0.29)	0.267 (0.26)	0.205 (0.22)	0.143 (0.19)	0.414 (0.33)	0.288 (0.26)	0.211 (0.21)	0.149 (0.16)	0.092 (0.11)
6.2.5	0.552 (0.37)	0.496 (0.35)	0.473 (0.32)	0.464 (0.32)	0.458 (0.31)	0.631 (0.41)	0.582 (0.38)	0.526 (0.34)	0.458 (0.32)	0.370 (0.29)	0.599 (0.39)	0.466 (0.32)	0.366 (0.27)	0.271 (0.23)	0.172 (0.17)
6.2.6	0.555 (0.38)	0.500 (0.35)	0.476 (0.33)	0.465 (0.34)	0.471 (0.34)	0.620 (0.42)	0.570 (0.38)	0.518 (0.35)	0.455 (0.34)	0.371 (0.31)	0.589 (0.40)	0.461 (0.32)	0.364 (0.28)	0.273 (0.24)	0.175 (0.19)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.2.1	0.271 (0.27)	0.193 (0.18)	0.149 (0.14)	0.115 (0.12)	0.082 (0.09)	0.350 (0.29)	0.267 (0.22)	0.218 (0.18)	0.175 (0.15)	0.132 (0.12)	0.134 (0.22)	0.090 (0.12)	0.072 (0.10)	0.057 (0.09)	0.043 (0.08)
6.2.2	0.288 (0.28)	0.211 (0.20)	0.167 (0.15)	0.129 (0.12)	0.093 (0.10)	0.362 (0.31)	0.284 (0.23)	0.235 (0.19)	0.192 (0.16)	0.147 (0.13)	0.175 (0.24)	0.126 (0.14)	0.101 (0.13)	0.080 (0.11)	0.060 (0.10)
6.2.3	0.440 (0.33)	0.315 (0.23)	0.242 (0.19)	0.183 (0.16)	0.125 (0.11)	0.521 (0.36)	0.404 (0.26)	0.332 (0.23)	0.271 (0.20)	0.205 (0.16)	0.421 (0.35)	0.358 (0.32)	0.321 (0.29)	0.293 (0.27)	0.264 (0.26)
6.2.4	0.441 (0.34)	0.318 (0.24)	0.245 (0.21)	0.186 (0.17)	0.128 (0.12)	0.519 (0.38)	0.404 (0.27)	0.333 (0.24)	0.272 (0.21)	0.173 (0.07)	0.426 (0.38)	0.366 (0.33)	0.330 (0.32)	0.301 (0.30)	0.272 (0.29)
6.2.5	0.557 (0.38)	0.398 (0.29)	0.299 (0.23)	0.218 (0.18)	0.143 (0.13)	0.573 (0.37)	0.429 (0.30)	0.345 (0.24)	0.276 (0.19)	0.207 (0.16)	0.534 (0.38)	0.484 (0.35)	0.462 (0.32)	0.452 (0.32)	0.454 (0.32)
6.2.6	0.548 (0.38)	0.394 (0.29)	0.297 (0.23)	0.217 (0.19)	0.142 (0.14)	0.562 (0.38)	0.422 (0.29)	0.339 (0.24)	0.271 (0.21)	0.203 (0.17)	0.536 (0.39)	0.487 (0.35)	0.464 (0.32)	0.452 (0.34)	0.452 (0.33)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.2.1	0.149 (0.24)	0.097 (0.13)	0.073 (0.11)	0.054 (0.08)	0.036 (0.06)	0.178 (0.25)	0.119 (0.15)	0.090 (0.11)	0.067 (0.09)	0.048 (0.07)	0.226 (0.26)	0.161 (0.18)	0.126 (0.13)	0.097 (0.11)	0.071 (0.09)
6.2.2	0.178 (0.24)	0.119 (0.15)	0.088 (0.12)	0.063 (0.09)	0.040 (0.07)	0.200 (0.25)	0.138 (0.16)	0.103 (0.12)	0.076 (0.10)	0.051 (0.07)	0.245 (0.27)	0.180 (0.18)	0.142 (0.14)	0.110 (0.12)	0.080 (0.09)
6.2.3	0.386 (0.32)	0.284 (0.27)	0.219 (0.22)	0.165 (0.18)	0.112 (0.15)	0.354 (0.31)	0.237 (0.23)	0.169 (0.18)	0.117 (0.13)	0.072 (0.09)	0.375 (0.31)	0.264 (0.22)	0.201 (0.18)	0.151 (0.14)	0.103 (0.11)
6.2.4	0.390 (0.34)	0.292 (0.28)	0.229 (0.25)	0.174 (0.20)	0.121 (0.18)	0.359 (0.32)	0.244 (0.24)	0.176 (0.19)	0.123 (0.15)	0.076 (0.11)	0.378 (0.32)	0.268 (0.23)	0.204 (0.19)	0.154 (0.15)	0.105 (0.11)
6.2.5	0.594 (0.41)	0.532 (0.37)	0.472 (0.32)	0.406 (0.30)	0.324 (0.27)	0.545 (0.38)	0.412 (0.31)	0.317 (0.26)	0.231 (0.21)	0.144 (0.16)	0.492 (0.36)	0.343 (0.27)	0.253 (0.22)	0.182 (0.16)	0.118 (0.12)
6.2.6	0.583 (0.41)	0.524 (0.36)	0.468 (0.33)	0.405 (0.32)	0.327 (0.28)	0.536 (0.38)	0.408 (0.30)	0.317 (0.26)	0.233 (0.22)	0.147 (0.17)	0.484 (0.36)	0.340 (0.27)	0.252 (0.22)	0.181 (0.18)	0.117 (0.13)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.2.1	0.294 (0.28)	0.223 (0.21)	0.182 (0.17)	0.146 (0.14)	0.111 (0.12)	0.087 (0.48)	0.064 (0.39)	0.053 (0.39)	0.047 (0.38)	0.040 (0.38)	0.352 (0.29)	0.248 (0.21)	0.192 (0.17)	0.143 (0.14)	0.095 (0.10)
6.2.2	0.309 (0.30)	0.241 (0.22)	0.200 (0.17)	0.163 (0.14)	0.125 (0.12)	0.109 (0.56)	0.076 (0.40)	0.061 (0.40)	0.054 (0.39)	0.042 (0.39)	0.373 (0.31)	0.274 (0.23)	0.215 (0.18)	0.162 (0.15)	0.107 (0.11)
6.2.3	0.446 (0.34)	0.342 (0.24)	0.278 (0.21)	0.226 (0.18)	0.169 (0.14)	0.138 (0.56)	0.095 (0.46)	0.076 (0.43)	0.069 (0.38)	0.059 (0.36)	0.568 (0.37)	0.401 (0.27)	0.303 (0.23)	0.227 (0.20)	0.150 (0.15)
6.2.4	0.445 (0.35)	0.343 (0.25)	0.280 (0.22)	0.227 (0.19)	0.171 (0.15)	0.161 (0.62)	0.111 (0.46)	0.089 (0.41)	0.077 (0.39)	0.063 (0.37)	0.570 (0.39)	0.406 (0.28)	0.310 (0.25)	0.233 (0.21)	0.154 (0.16)
6.2.5	0.498 (0.35)	0.366 (0.28)	0.292 (0.22)	0.231 (0.18)	0.172 (0.15)	0.321 (0.50)	0.253 (0.42)	0.215 (0.40)	0.185 (0.40)	0.150 (0.40)	0.609 (0.38)	0.460 (0.31)	0.364 (0.26)	0.279 (0.22)	0.188 (0.17)
6.2.6	0.489 (0.36)	0.360 (0.27)	0.287 (0.22)	0.228 (0.19)	0.169 (0.15)	0.335 (0.51)	0.264 (0.43)	0.224 (0.40)	0.193 (0.41)	0.155 (0.40)	0.600 (0.39)	0.454 (0.31)	0.359 (0.26)	0.276 (0.23)	0.187 (0.18)

Table E.6: Setting 6.2: Square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.2.1	0.178 (0.16)	0.140 (0.10)	0.123 (0.09)	0.110 (0.06)	0.100 (0.05)	0.188 (0.15)	0.146 (0.10)	0.121 (0.10)	0.100 (0.06)	0.080 (0.05)	0.205 (0.13)	0.163 (0.10)	0.136 (0.09)	0.112 (0.06)	0.089 (0.05)
6.2.2	0.200 (0.15)	0.166 (0.10)	0.149 (0.09)	0.136 (0.06)	0.124 (0.05)	0.203 (0.15)	0.161 (0.11)	0.135 (0.10)	0.112 (0.07)	0.089 (0.06)	0.215 (0.14)	0.173 (0.11)	0.145 (0.10)	0.119 (0.07)	0.094 (0.05)
6.2.3	0.291 (0.10)	0.282 (0.09)	0.277 (0.09)	0.273 (0.09)	0.267 (0.08)	0.203 (0.15)	0.161 (0.11)	0.135 (0.10)	0.112 (0.07)	0.089 (0.06)	0.263 (0.10)	0.219 (0.08)	0.185 (0.07)	0.153 (0.06)	0.115 (0.05)
6.2.4	0.298 (0.11)	0.289 (0.10)	0.284 (0.11)	0.280 (0.10)	0.274 (0.10)	0.287 (0.11)	0.261 (0.10)	0.240 (0.10)	0.218 (0.09)	0.189 (0.09)	0.271 (0.12)	0.228 (0.10)	0.195 (0.09)	0.163 (0.08)	0.124 (0.07)
6.2.5	0.329 (0.09)	0.320 (0.08)	0.319 (0.07)	0.322 (0.08)	0.317 (0.06)	0.351 (0.09)	0.344 (0.08)	0.333 (0.08)	0.316 (0.08)	0.291 (0.07)	0.337 (0.09)	0.301 (0.08)	0.269 (0.07)	0.234 (0.07)	0.187 (0.06)
6.2.6	0.336 (0.09)	0.328 (0.09)	0.326 (0.08)	0.328 (0.09)	0.334 (0.08)	0.354 (0.10)	0.347 (0.09)	0.336 (0.08)	0.321 (0.09)	0.296 (0.09)	0.341 (0.10)	0.306 (0.09)	0.275 (0.08)	0.240 (0.08)	0.193 (0.07)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.2.1	0.230 (0.12)	0.193 (0.09)	0.168 (0.08)	0.143 (0.06)	0.117 (0.05)	0.260 (0.10)	0.229 (0.08)	0.206 (0.07)	0.184 (0.06)	0.156 (0.05)	0.164 (0.17)	0.133 (0.11)	0.120 (0.09)	0.109 (0.06)	0.098 (0.05)
6.2.2	0.239 (0.13)	0.202 (0.10)	0.175 (0.09)	0.150 (0.07)	0.122 (0.05)	0.267 (0.11)	0.236 (0.09)	0.214 (0.08)	0.191 (0.07)	0.163 (0.06)	0.186 (0.16)	0.157 (0.10)	0.143 (0.09)	0.131 (0.07)	0.118 (0.06)
6.2.3	0.267 (0.11)	0.221 (0.08)	0.189 (0.07)	0.161 (0.06)	0.129 (0.05)	0.290 (0.11)	0.251 (0.08)	0.223 (0.08)	0.198 (0.06)	0.169 (0.05)	0.283 (0.10)	0.274 (0.09)	0.268 (0.09)	0.262 (0.09)	0.254 (0.08)
6.2.4	0.273 (0.12)	0.228 (0.09)	0.196 (0.09)	0.166 (0.07)	0.134 (0.06)	0.296 (0.12)	0.256 (0.10)	0.229 (0.09)	0.203 (0.07)	0.172 (0.07)	0.290 (0.12)	0.281 (0.10)	0.275 (0.11)	0.270 (0.10)	0.262 (0.10)
6.2.5	0.318 (0.10)	0.266 (0.08)	0.225 (0.07)	0.187 (0.06)	0.145 (0.05)	0.315 (0.10)	0.267 (0.09)	0.233 (0.07)	0.203 (0.06)	0.170 (0.05)	0.328 (0.09)	0.321 (0.08)	0.319 (0.07)	0.320 (0.08)	0.325 (0.08)
6.2.6	0.322 (0.11)	0.270 (0.09)	0.230 (0.08)	0.191 (0.08)	0.148 (0.06)	0.319 (0.11)	0.270 (0.10)	0.237 (0.09)	0.207 (0.08)	0.173 (0.06)	0.335 (0.10)	0.327 (0.09)	0.325 (0.08)	0.325 (0.09)	0.329 (0.08)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.2.1	0.172 (0.16)	0.135 (0.11)	0.115 (0.10)	0.098 (0.07)	0.080 (0.05)	0.188 (0.15)	0.150 (0.11)	0.127 (0.10)	0.107 (0.07)	0.088 (0.05)	0.212 (0.13)	0.177 (0.10)	0.154 (0.09)	0.133 (0.06)	0.111 (0.05)
6.2.2	0.187 (0.16)	0.149 (0.12)	0.127 (0.10)	0.107 (0.08)	0.087 (0.06)	0.198 (0.15)	0.159 (0.12)	0.134 (0.10)	0.112 (0.07)	0.091 (0.05)	0.220 (0.14)	0.185 (0.11)	0.161 (0.09)	0.138 (0.07)	0.115 (0.05)
6.2.3	0.267 (0.09)	0.240 (0.09)	0.217 (0.08)	0.193 (0.08)	0.162 (0.07)	0.246 (0.11)	0.203 (0.09)	0.171 (0.08)	0.141 (0.07)	0.108 (0.06)	0.245 (0.12)	0.202 (0.09)	0.173 (0.08)	0.147 (0.06)	0.120 (0.05)
6.2.4	0.275 (0.11)	0.249 (0.10)	0.227 (0.10)	0.205 (0.09)	0.176 (0.09)	0.254 (0.12)	0.213 (0.10)	0.181 (0.09)	0.150 (0.08)	0.116 (0.07)	0.252 (0.13)	0.209 (0.10)	0.179 (0.09)	0.152 (0.07)	0.124 (0.06)
6.2.5	0.343 (0.09)	0.332 (0.08)	0.318 (0.07)	0.300 (0.07)	0.273 (0.07)	0.323 (0.09)	0.285 (0.08)	0.253 (0.07)	0.218 (0.07)	0.173 (0.06)	0.299 (0.10)	0.246 (0.09)	0.207 (0.08)	0.172 (0.06)	0.133 (0.05)
6.2.6	0.335 (0.10)	0.327 (0.09)	0.325 (0.08)	0.325 (0.09)	0.329 (0.08)	0.327 (0.10)	0.290 (0.09)	0.258 (0.08)	0.224 (0.08)	0.179 (0.07)	0.303 (0.11)	0.251 (0.09)	0.212 (0.09)	0.176 (0.08)	0.137 (0.06)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.2.1	0.240 (0.11)	0.210 (0.09)	0.189 (0.08)	0.168 (0.06)	0.144 (0.05)	0.145 (0.26)	0.128 (0.17)	0.121 (0.15)	0.117 (0.12)	0.116 (0.09)	0.281 (0.08)	0.244 (0.07)	0.214 (0.07)	0.181 (0.06)	0.140 (0.05)
6.2.2	0.248 (0.12)	0.218 (0.10)	0.196 (0.09)	0.175 (0.07)	0.150 (0.06)	0.150 (0.26)	0.131 (0.18)	0.124 (0.16)	0.119 (0.13)	0.117 (0.10)	0.289 (0.10)	0.254 (0.08)	0.223 (0.08)	0.189 (0.07)	0.148 (0.06)
6.2.3	0.266 (0.12)	0.228 (0.09)	0.203 (0.08)	0.180 (0.06)	0.154 (0.05)	0.176 (0.19)	0.153 (0.15)	0.142 (0.13)	0.133 (0.11)	0.126 (0.09)	0.315 (0.10)	0.260 (0.08)	0.224 (0.07)	0.193 (0.06)	0.157 (0.05)
6.2.4	0.271 (0.13)	0.234 (0.10)	0.208 (0.09)	0.184 (0.07)	0.158 (0.06)	0.187 (0.19)	0.163 (0.16)	0.150 (0.41)	0.141 (0.12)	0.132 (0.11)	0.320 (0.11)	0.266 (0.09)	0.231 (0.08)	0.199 (0.07)	0.162 (0.07)
6.2.5	0.293 (0.10)	0.244 (0.09)	0.212 (0.08)	0.185 (0.06)	0.156 (0.05)	0.266 (0.13)	0.241 (0.10)	0.223 (0.08)	0.208 (0.08)	0.188 (0.07)	0.332 (0.09)	0.284 (0.08)	0.248 (0.07)	0.215 (0.06)	0.174 (0.05)
6.2.6	0.297 (0.11)	0.248 (0.10)	0.215 (0.09)	0.188 (0.08)	0.158 (0.06)	0.273 (0.13)	0.248 (0.11)	0.230 (0.09)	0.214 (0.09)	0.194 (0.08)	0.336 (0.10)	0.288 (0.09)	0.252 (0.08)	0.218 (0.08)	0.177 (0.07)

E.2.4 Setting 6.3: Monte Carlo-Based Results

Table E.7: Setting 6.3: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.3.1	0.310 (0.28)	0.188 (0.18)	0.128 (0.13)	0.090 (0.09)	0.061 (0.07)	0.352 (0.28)	0.223 (0.18)	0.148 (0.13)	0.096 (0.09)	0.054 (0.06)	0.407 (0.29)	0.283 (0.20)	0.203 (0.15)	0.141 (0.11)	0.084 (0.08)
6.3.2	0.324 (0.29)	0.209 (0.19)	0.150 (0.14)	0.112 (0.10)	0.080 (0.08)	0.358 (0.29)	0.232 (0.20)	0.157 (0.14)	0.105 (0.10)	0.061 (0.07)	0.407 (0.30)	0.286 (0.21)	0.207 (0.16)	0.145 (0.11)	0.088 (0.08)
6.3.3	0.584 (0.39)	0.482 (0.31)	0.416 (0.27)	0.370 (0.24)	0.327 (0.21)	0.535 (0.36)	0.400 (0.28)	0.310 (0.22)	0.240 (0.17)	0.168 (0.14)	0.492 (0.35)	0.353 (0.25)	0.263 (0.19)	0.191 (0.14)	0.120 (0.10)
6.3.4	0.581 (0.39)	0.483 (0.34)	0.420 (0.29)	0.377 (0.25)	0.336 (0.24)	0.537 (0.37)	0.407 (0.30)	0.320 (0.24)	0.253 (0.19)	0.182 (0.16)	0.496 (0.36)	0.361 (0.26)	0.272 (0.20)	0.201 (0.16)	0.128 (0.11)
6.3.5	0.515 (0.31)	0.422 (0.26)	0.380 (0.23)	0.367 (0.23)	0.379 (0.23)	0.633 (0.37)	0.553 (0.34)	0.472 (0.29)	0.397 (0.25)	0.317 (0.20)	0.623 (0.35)	0.477 (0.30)	0.371 (0.24)	0.280 (0.19)	0.184 (0.14)
6.3.6	0.524 (0.32)	0.436 (0.28)	0.394 (0.25)	0.378 (0.25)	0.386 (0.25)	0.628 (0.38)	0.551 (0.35)	0.473 (0.31)	0.401 (0.26)	0.326 (0.22)	0.622 (0.36)	0.483 (0.31)	0.379 (0.26)	0.289 (0.20)	0.193 (0.16)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.3.1	0.473 (0.29)	0.366 (0.22)	0.289 (0.18)	0.222 (0.14)	0.148 (0.10)	0.543 (0.30)	0.459 (0.24)	0.395 (0.21)	0.332 (0.17)	0.252 (0.14)	0.220 (0.26)	0.126 (0.16)	0.083 (0.10)	0.058 (0.07)	0.040 (0.05)
6.3.2	0.470 (0.30)	0.365 (0.23)	0.290 (0.19)	0.224 (0.14)	0.151 (0.11)	0.536 (0.31)	0.456 (0.25)	0.392 (0.22)	0.331 (0.18)	0.253 (0.15)	0.235 (0.27)	0.145 (0.16)	0.103 (0.11)	0.077 (0.08)	0.056 (0.06)
6.3.3	0.496 (0.33)	0.379 (0.24)	0.301 (0.19)	0.236 (0.16)	0.165 (0.12)	0.542 (0.33)	0.448 (0.26)	0.383 (0.22)	0.325 (0.19)	0.254 (0.16)	0.524 (0.38)	0.425 (0.29)	0.367 (0.25)	0.328 (0.22)	0.290 (0.20)
6.3.4	0.500 (0.34)	0.384 (0.25)	0.306 (0.21)	0.241 (0.16)	0.169 (0.13)	0.544 (0.35)	0.450 (0.27)	0.385 (0.24)	0.327 (0.20)	0.257 (0.17)	0.523 (0.39)	0.429 (0.31)	0.373 (0.27)	0.337 (0.23)	0.300 (0.22)
6.3.5	0.578 (0.33)	0.437 (0.28)	0.342 (0.21)	0.262 (0.17)	0.179 (0.13)	0.581 (0.32)	0.467 (0.28)	0.395 (0.23)	0.331 (0.20)	0.257 (0.16)	0.489 (0.32)	0.407 (0.26)	0.373 (0.23)	0.362 (0.23)	0.369 (0.23)
6.3.6	0.583 (0.36)	0.444 (0.29)	0.349 (0.22)	0.268 (0.18)	0.183 (0.14)	0.586 (0.35)	0.474 (0.29)	0.401 (0.24)	0.335 (0.20)	0.261 (0.17)	0.497 (0.33)	0.419 (0.27)	0.385 (0.25)	0.371 (0.24)	0.375 (0.25)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.3.1	0.256 (0.27)	0.156 (0.17)	0.105 (0.07)	0.067 (0.08)	0.039 (0.05)	0.312 (0.29)	0.210 (0.20)	0.148 (0.13)	0.101 (0.09)	0.062 (0.06)	0.384 (0.30)	0.288 (0.22)	0.222 (0.15)	0.166 (0.12)	0.111 (0.09)
6.3.2	0.264 (0.28)	0.164 (0.17)	0.110 (0.12)	0.074 (0.08)	0.045 (0.06)	0.315 (0.29)	0.213 (0.19)	0.151 (0.13)	0.105 (0.09)	0.064 (0.07)	0.383 (0.30)	0.288 (0.22)	0.223 (0.16)	0.168 (0.12)	0.113 (0.09)
6.3.3	0.473 (0.37)	0.344 (0.26)	0.264 (0.20)	0.204 (0.16)	0.142 (0.13)	0.435 (0.35)	0.303 (0.23)	0.221 (0.17)	0.159 (0.13)	0.099 (0.09)	0.444 (0.33)	0.331 (0.23)	0.258 (0.18)	0.199 (0.15)	0.137 (0.11)
6.3.4	0.477 (0.37)	0.353 (0.28)	0.275 (0.22)	0.217 (0.17)	0.156 (0.15)	0.442 (0.36)	0.312 (0.25)	0.231 (0.19)	0.168 (0.14)	0.106 (0.10)	0.450 (0.34)	0.337 (0.24)	0.263 (0.20)	0.204 (0.15)	0.141 (0.11)
6.3.5	0.587 (0.37)	0.494 (0.32)	0.416 (0.26)	0.350 (0.23)	0.280 (0.19)	0.566 (0.36)	0.419 (0.28)	0.320 (0.22)	0.239 (0.17)	0.155 (0.13)	0.525 (0.35)	0.385 (0.27)	0.295 (0.20)	0.223 (0.16)	0.149 (0.12)
6.3.6	0.584 (0.38)	0.495 (0.32)	0.420 (0.28)	0.356 (0.24)	0.289 (0.20)	0.567 (0.37)	0.426 (0.29)	0.329 (0.23)	0.248 (0.18)	0.164 (0.14)	0.531 (0.37)	0.393 (0.27)	0.303 (0.21)	0.228 (0.16)	0.152 (0.12)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.3.1	0.462 (0.30)	0.381 (0.24)	0.320 (0.19)	0.262 (0.16)	0.195 (0.12)	0.300 (0.50)	0.251 (0.04)	0.224 (0.33)	0.202 (0.29)	0.181 (0.27)	0.569 (0.29)	0.445 (0.24)	0.341 (0.19)	0.245 (0.15)	0.142 (0.11)
6.3.2	0.459 (0.31)	0.380 (0.24)	0.319 (0.20)	0.262 (0.16)	0.196 (0.13)	0.321 (0.51)	0.272 (0.37)	0.244 (0.33)	0.222 (0.29)	0.202 (0.27)	0.563 (0.31)	0.443 (0.25)	0.344 (0.20)	0.251 (0.16)	0.150 (0.12)
6.3.3	0.493 (0.33)	0.400 (0.25)	0.337 (0.21)	0.281 (0.18)	0.216 (0.15)	0.379 (0.49)	0.320 (0.41)	0.285 (0.38)	0.252 (0.33)	0.219 (0.33)	0.590 (0.35)	0.460 (0.28)	0.364 (0.22)	0.280 (0.18)	0.184 (0.14)
6.3.4	0.497 (0.35)	0.403 (0.26)	0.339 (0.23)	0.284 (0.18)	0.219 (0.16)	0.397 (0.51)	0.339 (0.42)	0.304 (0.39)	0.269 (0.35)	0.235 (0.33)	0.593 (0.38)	0.464 (0.29)	0.369 (0.24)	0.286 (0.19)	0.189 (0.15)
6.3.5	0.531 (0.34)	0.418 (0.27)	0.348 (0.22)	0.287 (0.18)	0.218 (0.15)	0.456 (0.50)	0.403 (0.40)	0.371 (0.38)	0.336 (0.33)	0.297 (0.33)	0.659 (0.35)	0.537 (0.32)	0.435 (0.27)	0.334 (0.21)	0.218 (0.15)
6.3.6	0.537 (0.36)	0.425 (0.28)	0.354 (0.23)	0.291 (0.19)	0.221 (0.16)	0.473 (0.52)	0.420 (0.40)	0.387 (0.38)	0.352 (0.34)	0.311 (0.33)	0.660 (0.37)	0.541 (0.33)	0.439 (0.28)	0.338 (0.22)	0.223 (0.16)

Table E.8: Setting 6.3: Square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.3.1	0.183 (0.09)	0.145 (0.08)	0.122 (0.06)	0.104 (0.04)	0.089 (0.03)	0.193 (0.08)	0.154 (0.07)	0.124(0.06)	0.099 (0.05)	0.074 (0.03)	0.205 (0.07)	0.171 (0.06)	0.144 (0.06)	0.118 (0.04)	0.089 (0.03)
6.3.2	0.196 (0.09)	0.161 (0.08)	0.138 (0.06)	0.121 (0.05)	0.105 (0.03)	0.202 (0.08)	0.164 (0.07)	0.134 (0.07)	0.108 (0.05)	0.081(0.04)	0.212 (0.07)	0.178 (0.07)	0.150 (0.06)	0.123 (0.05)	0.093 (0.04)
6.3.3	0.241 (0.06)	0.232 (0.05)	0.225 (0.04)	0.218 (0.04)	0.211 (0.04)	0.232 (0.06)	0.211 (0.06)	0.192 (0.05)	0.173 (0.04)	0.148 (0.04)	0.221 (0.06)	0.192 (0.06)	0.166 (0.05)	0.140 (0.04)	0.108 (0.03)
6.3.4	0.247 (0.06)	0.238 (0.06)	0.231 (0.06)	0.225 (0.06)	0.219 (0.05)	0.238 (0.06)	0.219 (0.07)	0.201 (0.06)	0.184 (0.06)	0.161 (0.05)	0.228 (0.07)	0.201 (0.07)	0.175 (0.06)	0.150 (0.05)	0.118 (0.04)
6.3.5	0.232 (0.06)	0.220 (0.05)	0.215 (0.04)	0.217 (0.04)	0.225 (0.04)	0.249 (0.05)	0.242 (0.05)	0.233 (0.04)	0.221 (0.04)	0.204 (0.04)	0.244 (0.05)	0.225 (0.05)	0.205 (0.05)	0.181 (0.04)	0.147 (0.04)
6.3.6	0.240 (0.06)	0.229 (0.06)	0.225 (0.05)	0.225 (0.05)	0.232 (0.05)	0.254 (0.06)	0.247 (0.06)	0.238 (0.06)	0.227 (0.05)	0.212 (0.05)	0.250 (0.06)	0.232 (0.06)	0.213 (0.06)	0.190 (0.05)	0.157 (0.05)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.1$)				
6.3.1	0.220 (0.06)	0.194 (0.05)	0.172 (0.05)	0.150 (0.04)	0.121 (0.03)	0.234 (0.05)	0.216 (0.04)	0.201 (0.04)	0.184 (0.04)	0.160 (0.03)	0.153 (0.012)	0.119 (0.09)	0.099 (0.07)	0.085 (0.05)	0.074 (0.03)
6.3.2	0.225 (0.06)	0.200 (0.06)	0.178 (0.05)	0.154 (0.05)	0.125 (0.04)	0.239 (0.06)	0.222 (0.05)	0.206 (0.05)	0.189 (0.04)	0.165 (0.04)	0.164 (0.11)	0.132 (0.08)	0.114 (0.06)	0.100 (0.04)	0.089 (0.03)
6.3.3	0.218 (0.06)	0.191 (0.05)	0.169 (0.05)	0.148 (0.04)	0.121 (0.03)	0.225 (0.05)	0.205 (0.05)	0.189 (0.04)	0.174 (0.04)	0.152 (0.03)	0.235 (0.06)	0.224 (0.05)	0.215 (0.04)	0.208 (0.04)	0.200 (0.04)
6.3.4	0.225 (0.07)	0.199 (0.06)	0.177 (0.05)	0.155 (0.05)	0.126 (0.04)	0.231 (0.07)	0.211 (0.06)	0.196 (0.05)	0.179 (0.05)	0.157 (0.04)	0.241 (0.07)	0.230 (0.06)	0.222 (0.06)	0.216 (0.05)	0.208 (0.05)
6.3.5	0.236 (0.05)	0.210 (0.05)	0.187 (0.05)	0.162 (0.04)	0.130 (0.03)	0.234 (0.05)	0.212 (0.05)	0.194 (0.04)	0.177 (0.04)	0.154 (0.03)	0.231 (0.06)	0.220 (0.05)	0.217 (0.04)	0.218 (0.04)	0.224 (0.04)
6.3.6	0.243 (0.06)	0.218 (0.06)	0.195 (0.05)	0.170 (0.05)	0.137 (0.04)	0.241 (0.06)	0.219 (0.05)	0.201 (0.05)	0.183 (0.05)	0.160 (0.04)	0.239 (0.07)	0.230 (0.06)	0.226 (0.05)	0.226 (0.05)	0.230 (0.05)
	PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)					PTNN (Gaussian weight, $\eta = 0.4$)				
6.3.1	0.164 (0.10)	0.129 (0.09)	0.105 (0.07)	0.085 (0.05)	0.067 (0.04)	0.180 (0.09)	0.148 (0.08)	0.124 (0.06)	0.102 (0.05)	0.080 (0.04)	0.198 (0.07)	0.172 (0.07)	0.151 (0.05)	0.131 (0.04)	0.107 (0.03)
6.3.2	0.171 (0.10)	0.136 (0.08)	0.112 (0.06)	0.091 (0.05)	0.072 (0.04)	0.184 (0.09)	0.152 (0.08)	0.128 (0.06)	0.106 (0.05)	0.083 (0.04)	0.201 (0.08)	0.176 (0.07)	0.155 (0.05)	0.134 (0.05)	0.110 (0.04)
6.3.3	0.223 (0.07)	0.200 (0.06)	0.180 (0.05)	0.161 (0.04)	0.137 (0.03)	0.211 (0.07)	0.180 (0.07)	0.153 (0.05)	0.128 (0.05)	0.099 (0.03)	0.208 (0.07)	0.180 (0.06)	0.157 (0.05)	0.136 (0.04)	0.111 (0.04)
6.3.4	0.230 (0.07)	0.208 (0.07)	0.190 (0.06)	0.172 (0.06)	0.150 (0.05)	0.219 (0.08)	0.189 (0.07)	0.163 (0.06)	0.139 (0.05)	0.109 (0.04)	0.215 (0.07)	0.187 (0.06)	0.165 (0.06)	0.143 (0.05)	0.115 (0.04)
6.3.5	0.245 (0.06)	0.235 (0.05)	0.224 (0.04)	0.211 (0.04)	0.193 (0.04)	0.237 (0.05)	0.215 (0.05)	0.193 (0.05)	0.169 (0.04)	0.136 (0.04)	0.228 (0.06)	0.200 (0.05)	0.175 (0.05)	0.150 (0.04)	0.119 (0.03)
6.3.6	0.250 (0.06)	0.241 (0.06)	0.230 (0.06)	0.218 (0.05)	0.202 (0.105)	0.244 (0.06)	0.223 (0.06)	0.202 (0.06)	0.178 (0.05)	0.146 (0.04)	0.235 (0.06)	0.208 (0.06)	0.183 (0.06)	0.158 (0.05)	0.126 (0.04)
	PTNN (uniform weight, $\eta = 0.5$)					Kernel density estimation					PT density estimation				
6.3.1	0.216 (0.06)	0.197 (0.06)	0.181 (0.04)	0.164 (0.04)	0.142 (0.03)	0.189 (0.11)	0.172 (0.08)	0.162 (0.07)	0.155 (0.06)	0.148 (0.05)	0.246 (0.05)	0.221 (0.04)	0.197 (0.04)	0.168 (0.04)	0.129 (0.03)
6.3.2	0.219 (0.07)	0.200 (0.06)	0.184 (0.05)	0.167 (0.04)	0.145 (0.04)	0.197 (0.12)	0.180 (0.08)	0.171 (0.08)	0.163 (0.06)	0.157 (0.06)	0.250 (0.05)	0.227 (0.05)	0.203 (0.05)	0.175 (0.04)	0.136 (0.04)
6.3.3	0.215 (0.06)	0.194 (0.05)	0.178 (0.05)	0.162 (0.04)	0.140 (0.04)	0.212 (0.09)	0.194 (0.07)	0.184 (0.06)	0.173 (0.06)	0.162 (0.05)	0.237 (0.05)	0.215 (0.05)	0.194 (0.04)	0.171 (0.04)	0.138 (0.03)
6.3.4	0.222 (0.07)	0.201 (0.06)	0.184 (0.05)	0.167 (0.05)	0.145 (0.04)	0.220 (0.10)	0.204 (0.08)	0.193 (0.07)	0.183 (0.07)	0.172 (0.06)	0.243 (0.06)	0.222 (0.06)	0.202 (0.05)	0.179 (0.05)	0.145 (0.04)
6.3.5	0.226 (0.05)	0.201 (0.05)	0.183 (0.05)	0.165 (0.04)	0.142 (0.03)	0.234 (0.08)	0.222 (0.06)	0.214 (0.06)	0.204 (0.05)	0.193 (0.05)	0.245 (0.05)	0.227 (0.05)	0.210 (0.05)	0.187 (0.04)	0.152 (0.04)
6.3.6	0.233 (0.06)	0.209 (0.06)	0.190 (0.05)	0.171 (0.05)	0.147 (0.04)	0.241 (0.09)	0.230 (0.07)	0.221 (0.06)	0.212 (0.06)	0.200 (0.06)	0.251 (0.06)	0.234 (0.06)	0.217 (0.06)	0.194 (0.05)	0.160 (0.04)

E.2.5 Setting 6.3: Grid-Based Results

Table E.9: Setting 6.3: Grid-based K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.3.1	0.137 (0.28)	0.104 (0.18)	0.086 (0.10)	0.068 (0.09)	0.048 (0.09)	0.147 (0.30)	0.109 (0.19)	0.086 (0.10)	0.063 (0.08)	0.040 (0.06)	0.167 (0.32)	0.130 (0.20)	0.106 (0.12)	0.083 (0.08)	0.056 (0.05)
6.3.2	0.167 (0.31)	0.132 (0.17)	0.108 (0.12)	0.086 (0.10)	0.065 (0.10)	0.170 (0.32)	0.129 (0.18)	0.098 (0.12)	0.072 (0.09)	0.046 (0.07)	0.188 (0.35)	0.149 (0.19)	0.118 (0.13)	0.090 (0.09)	0.060 (0.06)
6.3.3	0.370 (0.27)	0.349 (0.15)	0.329 (0.10)	0.307 (0.09)	0.280 (0.08)	0.314 (0.28)	0.263 (0.17)	0.218 (0.11)	0.176 (0.08)	0.127 (0.07)	0.267 (0.31)	0.201 (0.19)	0.152 (0.11)	0.111 (0.08)	0.070 (0.05)
6.3.4	0.377 (0.30)	0.358 (0.15)	0.338 (0.10)	0.319 (0.09)	0.292 (0.09)	0.330 (0.31)	0.280 (0.18)	0.236 (0.11)	0.195 (0.08)	0.144 (0.07)	0.287 (0.36)	0.224 (0.21)	0.173 (0.13)	0.129 (0.09)	0.082 (0.06)
6.3.5	0.354 (0.28)	0.338 (0.14)	0.334 (0.10)	0.340 (0.09)	0.358 (0.08)	0.409 (0.31)	0.389 (0.16)	0.356 (0.09)	0.317 (0.08)	0.265 (0.06)	0.371 (0.31)	0.308 (0.18)	0.251 (0.11)	0.195 (0.08)	0.129 (0.05)
6.3.6	0.372 (0.31)	0.359 (0.15)	0.352 (0.10)	0.354 (0.09)	0.366 (0.09)	0.419 (0.34)	0.400 (0.17)	0.368 (0.10)	0.330 (0.08)	0.278 (0.07)	0.388 (0.36)	0.329 (0.19)	0.272 (0.13)	0.215 (0.10)	0.145 (0.06)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					LDTFP1 (linear predictor)				
6.3.1	0.196 (0.34)	0.167 (0.21)	0.144 (0.13)	0.120 (0.09)	0.090 (0.06)	0.232 (0.37)	0.213 (0.23)	0.194 (0.14)	0.171 (0.11)	0.141 (0.07)	0.239 (0.23)	0.113 (0.16)	0.077 (0.10)	0.065 (0.08)	0.066 (0.06)
6.3.2	0.221 (0.39)	0.188 (0.21)	0.157 (0.14)	0.128 (0.09)	0.093 (0.06)	0.263 (0.45)	0.241 (0.25)	0.214 (0.17)	0.184 (0.11)	0.146 (0.07)	0.239 (0.21)	0.118 (0.16)	0.085 (0.11)	0.074 (0.11)	0.069 (0.08)
6.3.3	0.255 (0.35)	0.198 (0.21)	0.157 (0.13)	0.124 (0.09)	0.088 (0.05)	0.314 (0.28)	0.238 (0.25)	0.206 (0.16)	0.176 (0.12)	0.139 (0.07)	0.385 (0.15)	0.356 (0.09)	0.332 (0.10)	0.316 (0.07)	0.308 (0.08)
6.3.4	0.278 (0.42)	0.223 (0.27)	0.180 (0.17)	0.144 (0.11)	0.102 (0.07)	0.304 (0.50)	0.269 (0.33)	0.236 (0.22)	0.205 (0.16)	0.163 (0.10)	0.388 (0.15)	0.346 (0.25)	0.286 (0.31)	0.258 (0.27)	0.248 (0.30)
6.3.5	0.327 (0.35)	0.254 (0.21)	0.198 (0.13)	0.149 (0.09)	0.100 (0.05)	0.316 (0.40)	0.259 (0.25)	0.219 (0.16)	0.182 (0.12)	0.142 (0.07)	0.387 (0.20)	0.361 (0.08)	0.343 (0.05)	0.332 (0.04)	0.321 (0.02)
6.3.6	0.350 (0.41)	0.282 (0.25)	0.224 (0.16)	0.173 (0.12)	0.117 (0.07)	0.343 (0.48)	0.293 (0.32)	0.250 (0.22)	0.212 (0.17)	0.166 (0.11)	0.392 (0.20)	0.370 (0.09)	0.354 (0.06)	0.344 (0.04)	0.337 (0.02)
	LDTFP2 (quadratic predictor)					kernel density estimation					PT density estimation				
6.3.1	0.264 (0.19)	0.126 (0.20)	0.060 (0.10)	0.039 (0.09)	0.038 (0.05)	0.267 (0.39)	0.214 (0.29)	0.187 (0.24)	0.164 (0.21)	0.143 (0.20)	0.248 (0.37)	0.218 (0.22)	0.183 (0.14)	0.145 (0.11)	0.103 (0.06)
6.3.2	0.270 (0.18)	0.138 (0.20)	0.072 (0.10)	0.051 (0.06)	0.037 (0.04)	0.281 (0.39)	0.231 (0.29)	0.203(0.24)	0.179 (0.21)	0.156 (0.19)	0.281 (0.46)	0.243 (0.23)	0.199 (0.15)	0.157 (0.10)	0.110 (0.07)
6.3.3	0.260 (0.18)	0.125 (0.20)	0.059 (0.10)	0.040 (0.09)	0.037 (0.05)	0.338 (0.36)	0.281 (0.27)	0.247 (0.22)	0.215 (0.20)	0.182 (0.18)	0.310 (0.43)	0.259 (0.25)	0.213 (0.15)	0.168 (0.09)	0.115 (0.05)
6.3.4	0.271 (0.18)	0.139 (0.19)	0.072 (0.10)	0.050 (0.05)	0.036 (0.04)	0.352 (0.37)	0.297 (0.27)	0.263 (0.22)	0.231 (0.20)	0.197 (0.18)	0.341 (0.55)	0.293 (0.33)	0.242 (0.20)	0.193 (0.13)	0.133 (0.08)
6.3.5	0.352 (0.35)	0.375 (0.57)	0.327 (0.29)	0.405 (0.64)	0.349 (0.33)	0.415 (0.39)	0.366 (0.28)	0.332 (0.24)	0.300 (0.21)	0.262(0.21)	0.387 (0.38)	0.345 (0.21)	0.295 (0.12)	0.238 (0.09)	0.166 (0.05)
6.3.6	0.379 (0.37)	0.364 (0.35)	0.419 (0.69)	0.398 (0.45)	0.405 (0.57)	0.424 (0.38)	0.378 (0.28)	0.346 (0.24	0.314 (0.21)	0.275 (0.21)	0.409 (0.45)	0.367 (0.25)	0.314 (0.15)	0.255 (0.11)	0.180 (0.06)

Table E.10: Setting 6.3: Grid-based square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4$ and 0.5, kernel density estimation, PT density estimation, LDTFP1 and LDTFP2 when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.3.1	0.145 (0.12)	0.121 (0.08)	0.109 (0.06)	0.097 (0.04)	0.085 (0.03)	0.149 (0.13)	0.121 (0.09)	0.104 (0.07)	0.088 (0.05)	0.072 (0.04)	0.159 (0.13)	0.132 (0.08)	0.115 (0.06)	0.098 (0.04)	0.081 (0.03)
6.3.2	0.159 (0.11)	0.136 (0.07)	0.121 (0.06)	0.110 (0.04)	0.097 (0.03)	0.158 (0.12)	0.130 (0.08)	0.110 (0.07)	0.094 (0.05)	0.076 (0.04)	0.166 (0.12)	0.138 (0.08)	0.118 (0.06)	0.100 (0.05)	0.081 (0.03)
6.3.3	0.229 (0.06)	0.221 (0.04)	0.214 (0.03)	0.208 (0.02)	0.199 (0.01)	0.211 (0.07)	0.191 (0.05)	0.174 (0.04)	0.157 (0.03)	0.134 (0.02)	0.193 (0.09)	0.164 (0.07)	0.141 (0.05)	0.119 (0.04)	0.094 (0.03)
6.3.4	0.235 (0.06)	0.226 (0.03)	0.220 (0.02)	0.215 (0.02)	0.207 (0.01)	0.220 (0.07)	0.201 (0.05)	0.185 (0.04)	0.169 (0.03)	0.147 (0.02)	0.204 (0.08)	0.177 (0.06)	0.154 (0.05)	0.131 (0.03)	0.103 (0.03)
6.3.5	0.224 (0.06)	0.216 (0.03)	0.215 (0.02)	0.217 (0.01)	0.223 (0.01)	0.240 (0.06)	0.231 (0.03)	0.221 (0.02)	0.209 (0.02)	0.192 (0.01)	0.228 (0.06)	0.205 (0.05)	0.185 (0.04)	0.163 (0.03)	0.132 (0.02)
6.3.6	0.233 (0.06)	0.226 (0.03)	0.224 (0.02)	0.225 (0.01)	0.229 (0.01)	0.245 (0.06)	0.236 (0.03)	0.227 (0.02)	0.216 (0.01)	0.200 (0.01)	0.235 (0.06)	0.213 (0.04)	0.194 (0.03)	0.173 (0.03)	0.143 (0.02)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					LDTFP1 (linear predictor)				
6.3.1	0.173 (0.12)	0.151 (0.07)	0.134 (0.06)	0.118 (0.04)	0.100 (0.03)	0.189 (0.11)	0.173 (0.07)	0.159 (0.05)	0.144 (0.04)	0.125 (0.02)	0.189 (0.12)	0.122 (0.12)	0.103 (0.09)	0.093 (0.05)	0.089 (0.04)
6.3.2	0.179 (0.12)	0.155 (0.08)	0.137 (0.06)	0.120 (0.04)	0.100 (0.03)	0.196 (0.11)	0.178 (0.07)	0.161 (0.06)	0.146 (0.05)	0.127 (0.03)	0.191 (0.11)	0.122 (0.12)	0.104 (0.10)	0.095 (0.09)	0.092 (0.08)
6.3.3	0.187 (0.10)	0.159 (0.07)	0.139 (0.05)	0.121 (0.04)	0.100 (0.03)	0.211 (0.07)	0.173 (0.08)	0.159 (0.05)	0.144 (0.04)	0.125 (0.03)	0.228 (0.05)	0.218 (0.04)	0.210 (0.05)	0.206 (0.04)	0.204 (0.04)
6.3.4	0.198 (0.10)	0.171 (0.07)	0.151 (0.06)	0.131 (0.04)	0.107 (0.03)	0.204 (0.11)	0.185 (0.08)	0.170 (0.06)	0.154 (0.05)	0.135 (0.03)	0.233 (0.05)	0.216 (0.12)	0.195 (0.13)	0.190 (0.10)	0.189 (0.10)
6.3.5	0.213 (0.08)	0.184 (0.06)	0.160 (0.05)	0.136 (0.04)	0.109 (0.03)	0.208 (0.09)	0.183 (0.07)	0.164 (0.05)	0.147 (0.04)	0.127 (0.03)	0.229 (0.05)	0.221 (0.04)	0.216 (0.02)	0.213 (0.01)	0.211 (0.01)
6.3.6	0.223 (0.08)	0.195 (0.05)	0.171 (0.05)	0.147 (0.04)	0.117 (0.03)	0.218 (0.09)	0.194 (0.07)	0.175 (0.06)	0.158 (0.05)	0.136 (0.04)	0.235 (0.05)	0.228 (0.03)	0.224 (0.02)	0.221 (0.01)	0.219 (0.01)
	LDTFP2 (quadratic predictor)					kernel density estimation					PT density estimation				
6.3.1	0.202 (0.09)	0.129 (0.14)	0.087 (0.10)	0.070 (0.08)	0.058 (0.06)	0.185 (0.11)	0.169 (0.08)	0.159 (0.07)	0.151 (0.05)	0.145 (0.04)	0.197 (0.10)	0.174 (0.06)	0.152 (0.05)	0.130 (0.03)	0.105 (0.02)
6.3.2	0.208 (0.08)	0.135 (0.14)	0.092 (0.10)	0.074 (0.07)	0.062 (0.05)	0.196 (0.11)	0.180 (0.08)	0.170 (0.07)	0.161 (0.05)	0.153 (0.04)	0.205 (0.11)	0.180 (0.08)	0.157 (0.07)	0.136 (0.05)	0.113 (0.04)
6.3.3	0.202 (0.09)	0.129 (0.15)	0.087 (0.10)	0.069 (0.08)	0.058 (0.05)	0.209 (0.08)	0.192 (0.06)	0.181 (0.05)	0.170 (0.04)	0.159 (0.03)	0.204 (0.10)	0.182 (0.07)	0.163 (0.05)	0.143 (0.04)	0.117 (0.02)
6.3.4	0.208 (0.08)	0.136 (0.14)	0.092 (0.10)	0.073 (0.07)	0.062 (0.05)	0.217 (0.08)	0.202 (0.06)	0.191 (0.05)	0.180 (0.04)	0.169 (0.03)	0.216 (0.11)	0.195 (0.08)	0.176 (0.06)	0.155 (0.04)	0.127 (0.03)
6.3.5	0.287 (0.18)	0.293 (0.20)	0.281 (0.17)	0.295 (0.25)	0.289 (0.21)	0.232 (0.07)	0.221 (0.05)	0.212 (0.04)	0.203 (0.03)	0.191 (0.02)	0.228 (0.06)	0.209 (0.05)	0.190 (0.03)	0.168 (0.03)	0.138 (0.02)
6.3.6	0.294 (0.19)	0.291 (0.20)	0.306 (0.18)	0.289 (0.17)	0.295 (0.19)	0.239 (0.07)	0.228 (0.05)	0.220 (0.04)	0.210 (0.03)	0.198 (0.02)	0.236 (0.06)	0.216 (0.05)	0.197 (0.04)	0.176 (0.04)	0.146 (0.03)

E.2.6 Setting 6.4: Monte Carlo-Based Results

Table E.11: Setting 6.4: K-L divergence (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.4.1	0.310 (0.28)	0.211 (0.19)	0.162 (0.14)	0.129 (0.13)	0.096 (0.11)	0.357 (0.29)	0.242 (0.20)	0.179 (0.15)	0.135 (0.13)	0.093 (0.10)	0.424 (0.31)	0.297 (0.22)	0.223 (0.17)	0.167 (0.14)	0.114 (0.10)
6.4.2	0.443 (0.39)	0.320 (0.30)	0.250 (0.27)	0.198 (0.23)	0.149 (0.19)	0.495 (0.41)	0.351 (0.30)	0.263 (0.26)	0.200 (0.21)	0.135 (0.16)	0.575 (0.45)	0.414 (0.32)	0.316 (0.28)	0.240 (0.22)	0.162 (0.16)
6.4.3	0.797 (0.54)	0.711 (0.47)	0.647 (0.45)	0.594 (0.43)	0.535 (0.38)	0.778 (0.51)	0.651 (0.43)	0.556 (0.40)	0.473 (0.37)	0.381 (0.29)	0.791 (0.51)	0.633 (0.41)	0.517 (0.37)	0.414 (0.33)	0.299 (0.24)
6.4.4	0.958 (0.75)	0.872 (0.64)	0.811 (0.67)	0.754 (0.63)	0.695 (0.57)	0.938 (0.72)	0.805 (0.60)	0.709 (0.61)	0.619 (0.55)	0.521 (0.47)	0.951 (0.72)	0.783 (0.57)	0.658 (0.57)	0.542 (0.49)	0.413 (0.38)
6.4.5	0.930 (0.61)	0.908 (0.60)	0.892 (0.61)	0.877 (0.60)	0.842 (0.56)	0.942 (0.61)	0.883 (0.57)	0.834 (0.55)	0.782 (0.53)	0.699 (0.47)	0.943 (0.60)	0.838 (0.53)	0.753 (0.50)	0.667 (0.46)	0.547 (0.38)
6.4.6	1.100 (0.81)	1.075 (0.78)	1.063 (0.83)	1.052 (0.81)	1.020 (0.76)	1.112 (0.81)	1.048 (0.75)	1.002 (0.77)	0.952 (0.73)	0.870 (0.66)	1.114 (0.79)	1.002 (0.71)	0.918 (0.71)	0.832 (0.66)	0.707 (0.56)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (uniform weight, $\eta = 0.6$)				
6.4.1	0.512 (0.34)	0.377 (0.25)	0.295 (0.20)	0.229 (0.17)	0.162 (0.13)	0.621 (0.37)	0.486 (0.29)	0.399 (0.24)	0.326 (0.22)	0.247 (0.17)	0.749 (0.41)	0.625 (0.34)	0.539 (0.29)	0.463 (0.27)	0.376 (0.23)
6.4.2	0.681 (0.50)	0.512 (0.37)	0.405 (0.32)	0.319 (0.26)	0.229 (0.20)	0.812 (0.55)	0.647 (0.43)	0.536 (0.38)	0.443 (0.32)	0.340 (0.26)	0.963 (0.62)	0.816 (0.51)	0.710 (0.46)	0.617 (0.42)	0.508 (0.36)
6.4.3	0.842 (0.52)	0.671 (0.41)	0.545 (0.37)	0.433 (0.32)	0.308 (0.23)	0.930 (0.56)	0.765 (0.44)	0.640 (0.40)	0.529 (0.35)	0.403 (0.26)	1.049 (0.60)	0.905 (0.50)	0.793 (0.46)	0.692 (0.42)	0.571 (0.33)
6.4.4	1.004 (0.73)	0.818 (0.57)	0.679 (0.56)	0.547 (0.47)	0.401 (0.35)	1.096 (0.77)	0.914 (0.61)	0.774 (0.59)	0.643 (0.50)	0.493 (0.37)	1.219 (0.83)	1.062 (0.67)	0.837 (0.66)	0.818 (0.59)	0.679 (0.46)
6.4.5	0.968 (0.59)	0.827 (0.51)	0.715 (0.46)	0.606 (0.42)	0.463 (0.33)	1.024 (0.61)	0.868 (0.51)	0.745 (0.46)	0.628 (0.40)	0.481 (0.32)	1.113 (0.64)	0.965 (0.54)	0.849 (0.49)	0.740 (0.44)	0.606 (0.37)
6.4.6	1.140 (0.80)	0.992 (0.68)	0.877 (0.67)	0.763 (0.61)	0.606 (0.49)	1.200 (0.82)	1.035 (0.69)	0.906 (0.66)	0.777 (0.59)	0.609 (0.48)	1.294 (0.87)	1.135 (0.73)	1.012 (0.70)	0.890 (0.63)	0.733 (0.53)
	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.4.1	0.234 (0.27)	0.167 (0.18)	0.131 (0.14)	0.106 (0.13)	0.078 (0.10)	0.263 (0.28)	0.182 (0.18)	0.138 (0.14)	0.106 (0.12)	0.073 (0.10)	0.309 (0.29)	0.216 (0.19)	0.165 (0.15)	0.125 (0.13)	0.087 (0.10)
6.4.2	0.343 (0.35)	0.248 (0.27)	0.192 (0.24)	0.153 (0.20)	0.118 (0.17)	0.373 (0.36)	0.263 (0.27)	0.196 (0.23)	0.146 (0.18)	0.099 (0.14)	0.426 (0.38)	0.303 (0.28)	0.229 (0.23)	0.170 (0.19)	0.114 (0.14)
6.4.3	0.705 (0.47)	0.592 (0.38)	0.536 (0.33)	0.497 (0.31)	0.456 (0.30)	0.670 (0.45)	0.518 (0.35)	0.434 (0.29)	0.369 (0.26)	0.303 (0.24)	0.663 (0.45)	0.482 (0.32)	0.377 (0.26)	0.294 (0.22)	0.211 (0.19)
6.4.4	0.871 (0.74)	0.784 (0.61)	0.722 (0.62)	0.669 (0.58)	0.618 (0.52)	0.825 (0.70)	0.699 (0.56)	0.607 (0.56)	0.527 (0.50)	0.444 (0.42)	0.808 (0.67)	0.651 (0.52)	0.537 (0.51)	0.435 (0.43)	0.328 (0.33)
6.4.5	0.900 (0.61)	0.873 (0.60)	0.842 (0.59)	0.813 (0.57)	0.771 (0.51)	0.889 (0.59)	0.827 (0.56)	0.765 (0.53)	0.703 (0.50)	0.620 (0.42)	0.860 (0.57)	0.757 (0.51)	0.666 (0.47)	0.577 (0.43)	0.464 (0.34)
6.4.6	1.066 (0.81)	1.039 (0.78)	1.015 (0.81)	0.990 (0.78)	0.948 (0.72)	1.054 (0.79)	0.990 (0.74)	0.933 (0.74)	0.873 (0.69)	0.788 (0.61)	1.024 (0.76)	0.917 (0.69)	0.828 (0.67)	0.738 (0.61)	0.618 (0.51)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.6$)				
6.4.1	0.375 (0.31)	0.273 (0.22)	0.213 (0.18)	0.166 (0.15)	0.120 (0.11)	0.462 (0.34)	0.354 (0.25)	0.289 (0.21)	0.234 (0.18)	0.177 (0.14)	0.574 (0.09)	0.465 (0.29)	0.395 (0.25)	0.336 (0.23)	0.270 (0.19)
6.4.2	0.505 (0.42)	0.373 (0.31)	0.291 (0.26)	0.226 (0.21)	0.160 (0.16)	0.612 (0.47)	0.475 (0.36)	0.387 (0.31)	0.316 (0.26)	0.240 (0.21)	0.748 (0.54)	0.614 (0.43)	0.524 (0.38)	0.449 (0.33)	0.364 (0.28)
6.4.3	0.692 (0.46)	0.498 (0.33)	0.383 (0.25)	0.291 (0.21)	0.199 (0.17)	0.759 (0.48)	0.571 (0.35)	0.456 (0.28)	0.362 (0.23)	0.262 (0.19)	0.858 (0.52)	0.694 (0.40)	0.589 (0.34)	0.499 (0.29)	0.395 (0.24)
6.4.4	0.831 (0.66)	0.655 (0.51)	0.529 (0.49)	0.414 (0.40)	0.294 (0.28)	0.898 (0.69)	0.720 (0.52)	0.591 (0.50)	0.474 (0.41)	0.351 (0.30)	1.007 (0.74)	0.844 (0.57)	0.724 (0.55)	0.614 (0.48)	0.493 (0.36)
6.4.5	0.851 (0.55)	0.714 (0.48)	0.604 (0.43)	0.498 (0.37)	0.368 (0.29)	0.875 (0.56)	0.721 (0.47)	0.602 (0.41)	0.491 (0.35)	0.360 (0.27)	0.941 (0.58)	0.788 (0.48)	0.674 (0.43)	0.569 (0.37)	0.449 (0.31)
6.4.6	1.015 (0.75)	0.872 (0.65)	0.758 (0.62)	0.644 (0.55)	0.498 (0.44)	1.041 (0.75)	0.877 (0.64)	0.749 (0.59)	0.623 (0.52)	0.469 (0.41)	1.110 (0.78)	0.943 (0.66)	0.817 (0.61)	0.696 (0.53)	0.551 (0.44)
	kernel density estimation					PT density estimation									
6.4.1	0.365 (0.82)	0.312 (0.62)	0.307 (0.55)	0.293 (0.55)	0.287 (0.55)	0.811 (0.46)	0.626 (0.36)	0.513 (0.29)	0.423 (0.29)	0.331 (0.24)					
6.4.2	0.462 (0.82)	0.377 (0.74)	0.362 (0.62)	0.341 (0.65)	0.322 (0.62)	1.010 (0.65)	0.812 (0.55)	0.683 (0.46)	0.576 (0.43)	0.455 (0.36)					
6.4.3	0.620 (0.89)	0.509 (0.68)	0.492 (0.62)	0.437 (0.59)	0.401 (0.57)	1.039 (0.62)	0.875 (0.51)	0.748 (0.49)	0.634 (0.43)	0.494 (0.35)					
6.4.4	0.761 (1.03)	0.652 (0.84)	0.593 (0.76)	0.540 (0.70)	0.488 (0.70)	1.226 (0.84)	1.047 (0.26)	0.911 (0.70)	0.780 (0.23)	0.622 (0.49)					
6.4.5	0.855 (0.95)	0.776 (0.76)	0.731 (0.65)	0.682 (0.65)	0.624 (0.59)	1.068 (0.65)	0.927 (0.54)	0.831 (0.54)	0.750 (0.51)	0.636 (0.43)					
6.4.6	1.046 (1.08)	0.949 (0.92)	0.894 (0.84)	0.836 (0.86)	0.772 (0.76)	1.252 (0.86)	1.110 (0.73)	0.994 (0.76)	0.905 (0.70)	0.776 (0.59)					

Table E.12: Setting 6.4: Square root of MISE (standard error×10) of PTNN with $\eta = 0.1, 0.2, 0.3, 0.4, 0.5$ and 0.6, kernel density estimation and PT density estimation when sample size $n = 100, 250, 500, 1000$ and 2500

	Sample Size					Sample Size					Sample Size				
	100	250	500	1000	2500	100	250	500	1000	2500	100	250	500	1000	2500
Setting	PTNN (uniform weight, $\eta = 0.1$)					PTNN (uniform weight, $\eta = 0.2$)					PTNN (uniform weight, $\eta = 0.3$)				
6.4.1	0.247 (0.11)	0.200 (0.09)	0.172 (0.07)	0.152 (0.06)	0.135 (0.05)	0.262 (0.10)	0.213 (0.09)	0.179 (0.08)	0.150 (0.06)	0.121 (0.05)	0.282 (0.09)	0.236 (0.08)	0.201 (0.07)	0.169 (0.06)	0.133 (0.05)
6.4.2	0.334 (0.15)	0.288 (0.13)	0.259 (0.13)	0.237 (0.12)	0.215 (0.10)	0.349 (0.15)	0.296 (0.13)	0.258 (0.12)	0.225 (0.11)	0.187 (0.09)	0.372 (0.15)	0.318 (0.13)	0.277 (0.12)	0.238 (0.11)	0.191 (0.09)
6.4.3	0.375 (0.09)	0.362 (0.08)	0.353 (0.08)	0.345 (0.08)	0.336 (0.08)	0.368 (0.09)	0.344 (0.08)	0.324 (0.08)	0.306 (0.08)	0.283 (0.07)	0.368 (0.10)	0.334 (0.08)	0.306 (0.08)	0.277 (0.08)	0.240 (0.07)
6.4.4	0.466 (0.16)	0.456 (0.14)	0.447 (0.15)	0.440 (0.15)	0.433 (0.15)	0.459 (0.16)	0.438 (0.14)	0.420 (0.15)	0.403 (0.14)	0.382 (0.14)	0.458 (0.16)	0.427 (0.14)	0.400 (0.15)	0.372 (0.14)	0.335 (0.14)
6.4.5	0.404 (0.10)	0.403 (0.09)	0.403 (0.09)	0.404 (0.09)	0.403 (0.09)	0.404 (0.10)	0.396 (0.09)	0.389 (0.09)	0.383 (0.09)	0.371 (0.08)	0.400 (0.10)	0.383 (0.09)	0.369 (0.08)	0.353 (0.08)	0.329 (0.08)
6.4.6	0.494 (0.16)	0.494 (0.14)	0.495 (0.15)	0.496 (0.16)	0.495 (0.15)	0.494 (0.16)	0.487 (0.15)	0.482 (0.15)	0.476 (0.15)	0.465 (0.15)	0.490 (0.16)	0.475 (0.15)	0.462 (0.15)	0.448 (0.15)	0.426 (0.14)
	PTNN (uniform weight, $\eta = 0.4$)					PTNN (uniform weight, $\eta = 0.5$)					PTNN (uniform weight, $\eta = 0.6$)				
6.4.1	0.307 (0.08)	0.265 (0.08)	0.234 (0.07)	0.203 (0.06)	0.166 (0.05)	0.333 (0.08)	0.298 (0.07)	0.271 (0.07)	0.245 (0.06)	0.212 (0.05)	0.361 (0.08)	0.334 (0.07)	0.312 (0.07)	0.291 (0.06)	0.263 (0.06)
6.4.2	0.400 (0.15)	0.351 (0.13)	0.313 (0.12)	0.275 (0.11)	0.226 (0.09)	0.430 (0.15)	0.391 (0.13)	0.358 (0.13)	0.326 (0.11)	0.283 (0.10)	0.459 (0.15)	0.430 (0.13)	0.407 (0.13)	0.382 (0.12)	0.349 (0.12)
6.4.3	0.375 (0.10)	0.339 (0.08)	0.307 (0.08)	0.274 (0.07)	0.229 (0.06)	0.390 (0.10)	0.357 (0.09)	0.329 (0.08)	0.298 (0.08)	0.257 (0.07)	0.408 (0.10)	0.384 (0.09)	0.362 (0.09)	0.339 (0.08)	0.308 (0.07)
6.4.4	0.465 (0.17)	0.430 (0.15)	0.398 (0.16)	0.363 (0.14)	0.315 (0.13)	0.478 (0.17)	0.446 (0.15)	0.416 (0.16)	0.383 (0.14)	0.336 (0.13)	0.494 (0.17)	0.471 (0.15)	0.448 (0.16)	0.423 (0.15)	0.389 (0.14)
6.4.5	0.401 (0.10)	0.377 (0.09)	0.356 (0.08)	0.332 (0.08)	0.295 (0.07)	0.408 (0.10)	0.382 (0.09)	0.358 (0.09)	0.331 (0.08)	0.290 (0.07)	0.419 (0.10)	0.396 (0.09)	0.376 (0.09)	0.353 (0.08)	0.320 (0.08)
6.4.6	0.490 (0.17)	0.468 (0.15)	0.449 (0.15)	0.427 (0.15)	0.392 (0.49)	0.496 (0.17)	0.472 (0.15)	0.449 (0.16)	0.424 (0.16)	0.382 (0.14)	0.506 (0.17)	0.484 (0.16)	0.465 (0.16)	0.442 (0.16)	0.407 (0.15)
	PTNN (Gaussian weight, $\eta = 0.1$)					PTNN (Gaussian weight, $\eta = 0.2$)					PTNN (Gaussian weight, $\eta = 0.3$)				
6.4.1	0.215 (0.15)	0.178 (0.10)	0.159 (0.08)	0.145 (0.06)	0.131 (0.05)	0.227 (0.14)	0.184 (0.11)	0.158 (0.08)	0.138 (0.06)	0.118 (0.05)	0.244 (0.12)	0.201 (0.10)	0.172 (0.08)	0.148 (0.06)	0.123 (0.05)
6.4.2	0.298 (0.15)	0.259 (0.13)	0.235 (0.13)	0.217 (0.11)	0.199 (0.10)	0.306 (0.15)	0.258 (0.13)	0.226 (0.13)	0.198 (0.11)	0.168 (0.09)	0.323 (0.15)	0.272 (0.13)	0.235 (0.12)	0.202 (0.10)	0.165 (0.08)
6.4.3	0.364 (0.10)	0.346 (0.09)	0.335 (0.07)	0.327 (0.07)	0.318 (0.07)	0.350 (0.10)	0.319 (0.08)	0.299 (0.07)	0.281 (0.06)	0.260 (0.06)	0.342 (0.09)	0.301 (0.08)	0.271 (0.07)	0.243 (0.06)	0.210 (0.06)
6.4.4	0.453 (0.16)	0.442 (0.14)	0.433 (0.15)	0.425 (0.14)	0.417 (0.14)	0.440 (0.16)	0.418 (0.14)	0.400 (0.15)	0.383 (0.14)	0.363 (0.14)	0.432 (0.16)	0.400 (0.14)	0.372 (0.15)	0.344 (0.14)	0.309 (0.13)
6.4.5	0.401 (0.09)	0.400 (0.09)	0.398 (0.09)	0.397 (0.09)	0.394 (0.09)	0.397 (0.09)	0.389 (0.09)	0.380 (0.09)	0.371 (0.09)	0.357 (0.08)	0.388 (0.09)	0.370 (0.09)	0.354 (0.08)	0.337 (0.08)	0.311 (0.08)
6.4.6	0.492 (0.16)	0.491 (0.14)	0.490 (0.15)	0.489 (0.15)	0.486 (0.15)	0.488 (0.16)	0.480 (0.14)	0.473 (0.15)	0.465 (0.15)	0.453 (0.14)	0.479 (0.16)	0.462 (0.14)	0.448 (0.15)	0.433 (0.15)	0.409 (0.14)
	PTNN (Gaussian weight, $\eta = 0.4$)					PTNN (Gaussian weight, $\eta = 0.5$)					PTNN (Gaussian weight, $\eta = 0.6$)				
6.4.1	0.266 (0.10)	0.227 (0.09)	0.198 (0.08)	0.173 (0.06)	0.144 (0.05)	0.292 (0.09)	0.258 (0.08)	0.232 (0.07)	0.208 (0.06)	0.180 (0.05)	0.321 (0.09)	0.292 (0.08)	0.271 (0.07)	0.251 (0.06)	0.225 (0.05)
6.4.2	0.349 (0.15)	0.300 (0.13)	0.263 (0.12)	0.229 (0.10)	0.188 (0.08)	0.380 (0.15)	0.337 (0.13)	0.304 (0.12)	0.273 (0.10)	0.233 (0.09)	0.414 (0.15)	0.380 (0.13)	0.354 (0.13)	0.327 (0.11)	0.293 (0.10)
6.4.3	0.343 (0.09)	0.297 (0.08)	0.262 (0.07)	0.229 (0.06)	0.189 (0.05)	0.352 (0.09)	0.309 (0.08)	0.277 (0.07)	0.245 (0.07)	0.206 (0.05)	0.369 (0.09)	0.334 (0.08)	0.308 (0.07)	0.282 (0.07)	0.249 (0.06)
6.4.4	0.433 (0.16)	0.394 (0.14)	0.359 (0.15)	0.323 (0.13)	0.277 (0.12)	0.443 (0.16)	0.404 (0.14)	0.369 (0.15)	0.333 (0.13)	0.285 (0.12)	0.460 (0.17)	0.429 (0.15)	0.400 (0.15)	0.370 (0.14)	0.331 (0.12)
6.4.5	0.382 (0.10)	0.357 (0.09)	0.333 (0.08)	0.308 (0.08)	0.270 (0.07)	0.383 (0.10)	0.353 (0.09)	0.326 (0.08)	0.297 (0.07)	0.254 (0.07)	0.392 (0.10)	0.363 (0.09)	0.338 (0.09)	0.312 (0.08)	0.275 (0.07)
6.4.6	0.473 (0.16)	0.449 (0.15)	0.427 (0.15)	0.403 (0.15)	0.366 (0.13)	0.473 (0.16)	0.445 (0.15)	0.418 (0.15)	0.388 (0.15)	0.342 (0.13)	0.481 (0.17)	0.453 (0.15)	0.428 (0.15)	0.399 (0.15)	0.357 (0.13)
	kernel density estimation					PT density estimation									
6.4.1	0.293 (0.25)	0.290 (0.21)	0.284 (0.18)	0.281 (0.16)	0.279 (0.14)	0.382 (0.08)	0.347 (0.07)	0.324 (0.07)	0.300 (0.06)	0.269 (0.05)					
6.4.2	0.342 (0.28)	0.337 (0.24)	0.333 (0.22)	0.330 (0.20)	0.327 (0.19)	0.479 (0.15)	0.448 (0.14)	0.425 (0.13)	0.400 (0.12)	0.362 (0.10)					
6.4.3	0.345 (0.16)	0.321 (0.14)	0.312 (0.13)	0.306 (0.13)	0.299 (0.11)	0.414 (0.10)	0.387 (0.09)	0.364 (0.08)	0.339 (0.08)	0.304 (0.07)					
6.4.4	0.423 (0.19)	0.410 (0.19)	0.389 (0.18)	0.378 (0.17)	0.369 (0.17)	0.502 (0.17)	0.478 (0.15)	0.456 (0.16)	0.431 (0.15)	0.395 (0.14)					
6.4.5	0.401 (0.14)	0.392 (0.12)	0.382 (0.11)	0.367 (0.11)	0.356 (0.10)	0.418 (0.10)	0.395 (0.09)	0.378 (0.08)	0.360 (0.08)	0.333 (0.07)					
6.4.6	0.487 (0.17)	0.474 (0.17)	0.465 (0.16)	0.456 (0.17)	0.442 (0.16)	0.506 (0.17)	0.485 (0.15)	0.469 (0.16)	0.452 (0.15)	0.425 (0.14)					