Beyond Gaussian : copula graphical models for non-Gaussian data

Yu, Hang

2015

Yu, H. (2015). Beyond Gaussian : copula graphical models for non-Gaussian data. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65404>

<https://doi.org/10.32657/10356/65404>

**Beyond Gaussian:
Copula Graphical Models for non-Gaussian Data**

YU HANG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2014

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

.....

Date

.....

YU HANG

Acknowledgements

I leave no trace of wings in the air, but I am glad I have had my flight.

—Rabindranath Tagore

Above all, I am immensely indebted to my supervisor, Prof. Justin Dauwels, for his invaluable guidance and endless support through all these four years. At the very beginning of my graduate study, Justin directed me to work on the exciting project about copula-based graphical models. There is so much to be explored in this field that I was always motivated and inspired to do research. Justin's expectations were high, yet was available for advice whenever I was lost. I have great respect for his wisdom, dedication and enthusiasm. This dissertation would never have been possible without his brilliant ideas and extraordinary drive for research. I would also like to thank him for giving me the opportunities to meet and interact with world-class researchers through international conferences, summer schools, and the exchange program in Kyoto University.

I also wish to express my deep gratitude to Prof. Shigeru Shinomoto. He was my host during my three-month exchange in the nonlinear dynamics group (NDG) in Kyoto University. As an advisor, Shinomoto-san was always supportive and encouraging of my work. During the weekly discussions, I have benefited a lot from the amazing combination of intuition and rigor in his thinking. The change point detection model came from the discussion with him. I am also grateful to Hideaki Kim, Yasuhiro Mochizuki, and Toshiaki Shintani from the NDG, who helped me a lot when I was in Japan. I really enjoyed our numerous discussions, both academic and otherwise.

Thanks are also due to our collaborators in Shell Research and MIT. Many thanks to Dr. Philip Jonathan for offering those insightful ideas during our Skype meetings at the initial stage of this project, and for sharing the data of extreme wave heights in the Gulf of Mexico. I also would like to thank Prof. Venkat Chandrasekaran, Dr. Myung Jin Choi, and Dr. Matthew Johnson for answering my questions and sharing code.

Special thanks go to my former FYP students as well as the co-authors of my papers: Wang Xueou, Choo Zheng, Zhou Qiao, Xu Shiyan, Zhang Xu, Wayne Uy, Zhang Liaofan, Cheng Jingjing, Li Chenyang, and Zhu Junting. I will always remember the days and nights of hard work in the lab. I did learn how to do research while working with them and guiding them to finish their final year projects. In particular, I am very grateful to Xueou, Choo Zheng, and Wayne for listening to my half-baked ideas all the time and further discussing with me, which is interesting and inspiring.

I also benefited a lot from helpful interactions with my former and current lab mates – Zhong Jingshan, Srinivasan Kannan, Manuel Vázquez, Zhao Wei, Jing Jin, Muhammad Tayyab Asif, Wang Mengmeng, Nikola Mitrovic, Yasir Tahir, Qi Yongbin, Du Bo, Jiang Chaoyang, Guan Zheming, Wei Zhe, Liu Tianchi, Zheng Wenjie, Dhanya Mohan, Nishant Sinha, Umer Rasheed, Esmail Fathi, Maryam Khani, Dhruv Aggarwal, and Yang Zai. I wish to thank them for all the pleasant time we spent together, both inside and outside the lab. In addition, thanks Yang Zai for sharing with me his templates of QE report and PhD thesis.

Staff at NTU also deserve to be mentioned here. In particular, I appreciate Ms. Yang Xin for helping me set up 8 desktops to run simulations during my second year. I owe the 6 conference papers published that year to her. I also thank Mr. Tan Chai Soon for all the assistance after we moved to the new lab.

Living in Singapore in a completely new environment was tough at the beginning, and I would not have been able to go through it without the help of my teachers from the English corner: Esther Soo, Ross Harries, Serena Harries, David Elliot, Donna Elliot, and Sheena Sutherson. They not only helped me to improve my oral

English skills, but made me feel at home for the first time after I came to Singapore. I am especially thankful to Serena for baking the tasty orange cake for my 23rd birthday.

In addition, I would like to take this opportunity to thank all my very good friends back in China: Chen Lingjie, He Qing, Hu liwen, Lei Qinyuan, Lei Zhen, Li Wei, Li Yunting, Lian Dingfu, Liu Longtao, Liu Yalan, Luo Han, Wang Jia, Wang Yuxia, Qian Cong, Qin Peng, Qiu Shuai, Zhao Sizhe and Zhang Bingzi. Thanks for being the support and companion when I dealt with stresses of completing the doctoral program. Thanks for refreshing me every time I went back to Nanchang and Beijing.

Finally, I wish to express my sincere thanks to my parents for their constant love, support and encouragement. They are the source of my determination and perseverance in pursuing all my endeavors. This thesis is dedicated to them.

Abstract

Graphical models, which can be viewed as a marriage of graph theory and probability theory, provide a powerful formalism for multivariate statistical modeling of complex systems. Graphical models harness the complexity of large-scale systems by representing the statistical relations among a large number of variables in a compact manner. This compact structure can in turn be leveraged to derive highly efficient techniques for data analysis. However, research on graphical models for continuous variables so far mostly focuses on Gaussian statistics. Unfortunately, this limitation severely handicaps the utility of graphical models in real-world applications that are often associated with non-Gaussian variables. Physics and earth sciences, for instance, are often characterized by all positive quantities (e.g., amplitude, energy and magnitude), and thus cannot be described accurately by Gaussian distributions. In addition, the behavior of extreme events, such as hurricanes and floods, are theoretically governed by extreme-value distributions with fat tails instead of by Gaussian distributions. In this thesis, we move beyond Gaussian graphical models, and propose a portfolio of novel graphical models for non-Gaussian data. Such graphical models are powerful tools to solve real-life inference problems, while avoiding restrictive assumptions of Gaussian statistics, hence yielding more reliable solutions.

The first part of the thesis copes with “nominal” (non-extremal) data. This type of data follows neither Gaussian nor fat-tailed distributions. Gaussian copulas are employed here to tie any kind of marginal distributions (Gaussian, non-Gaussian and even non-parametric) together to form a joint distribution. Through the language of graphical models, we further impose constraints of sparse dependence structure on

the resulting non-Gaussian distribution, leading to sparse copula Gaussian graphical models (CGGM). Such models have the same mathematical convenience of Gaussian graphical models, yet are applicable to marginally non-Gaussian data. Along this line, we proceed to construct hidden variable copula Gaussian graphical models (HVCGGM) and discrete copula Gaussian graphical models (DCGGM). The two models are applicable to different practical scenarios. Specifically, the HVCGGM yields sparse graphical models when data is unavailable for some relevant variables; the DCGGM extends CGGM to discrete data in a straightforward manner. Since real data are often non-stationary and statistical models designed for stationary data may not yield accurate results, we further consider learning graphical models for piecewise-stationary data. In other words, we first detect change points in the time series, and then infer graphical models within each stationary segment.

Besides modeling nominal data, we also build graphical models to handle extreme events. Extreme events are often modeled in two stages: first the extreme-value marginal distributions are estimated, and then the joint distribution of extreme values is constructed based on the marginals. The second part of the thesis aims to exploit graphical models to estimate the marginal distributions accurately. The main difficulty of fitting extreme-value marginal distributions to measurements is the lack of data, as extreme events are by definition rare. To improve the accuracy of the estimated parameter values, we utilize graphical models to couple together the marginal parameters with similar values of covariates. For instance, extreme wave heights vary smoothly across an ocean, and such prior knowledge can be expressed by graphical models; in this example, the spatial coordinates are covariates. Additionally, we also propose to utilize graphical model to capture the spatio-temporal dependence among marginal parameters, and further forecast the trend of extreme events in the future.

The last part of the thesis is devoted to the second stage of extreme events modeling, that is, coupling all the marginals to form a joint distribution. We first investigate the performance of CGGMs on extreme-value data and propose an efficient interpolation algorithm to predict the extreme events at unmonitored locations

in a spatial domain. The CGGM, however, does not possess tail dependence, and it is therefore theoretically unfavorable for quantifying extreme events. This motivates us to build a graphical model on based extreme-value copulas. As multivariate extreme-value copulas are intractable, we aim to construct pairwise copula graphical models. In particular, the ensemble-of-trees of pairwise copulas (ETPC) is introduced. Concretely, extreme-value marginal distributions are stitched together by means of pairwise copulas, which in turn are the building blocks of the ensemble of trees. We further prove that, under mild conditions, the ETPC model exhibits the favorable property of tail-dependence between an arbitrary pair of sites (variables); consequently, the model is able to reliably capture statistical dependence between extreme values at different sites.

The proposed models have been successfully applied in such diverse areas as computational biology and neuroscience, geophysics and earth science, and sociology, indicating that the portfolio of graphical models provides a powerful toolbox to tackle real-world problems with non-Gaussian variables.

Contents

Acknowledgements	iv
Abstract	viii
List of Contents	xv
List of Figures	xxi
List of Tables	xxiv
1 Introduction	1
1.1 Motivation and Related Works	1
1.1.1 Graphical Models for Nominal Data	3
1.1.2 Graphical Models for Marginal Analysis of Extremal Data	7
1.1.3 Graphical Models for Joint Analysis of Extremal Data	13
1.2 Thesis Organization and Overview of Contributions	15
2 Preliminaries	21
2.1 Graphical Models	21
2.1.1 Basic Graph Theory	22

2.1.2	Probability Factorization and Graphical Models	23
2.1.3	Gaussian Graphical Models	26
2.1.4	Learning Gaussian Graphical Models	26
2.1.5	Inference in Gaussian Graphical Models	28
2.2	Univariate Extreme Value Theory	33
2.2.1	Generalized Extreme Value Distribution	33
2.2.2	Generalized Pareto Distribution	36
2.3	Copulas	38
2.3.1	Elliptical Copulas	41
2.3.2	Archimedean Copulas	43
2.3.3	Extreme-Value Copulas	46
I	Graphical Models for Nominal Data	51
3	Copula Gaussian Graphical Models with Hidden Variables	53
3.1	Standard Copula Gaussian Graphical Models	54
3.2	Hidden Variable Copula Gaussian Graphical Models	55
3.2.1	Hidden Variable Gaussian Graphical Model	56
3.2.2	Copula Gaussian Graphical Models with Hidden Variables . .	58
3.3	Learning and Inference	59
3.3.1	Structure Learning: Stability Selection	60
3.3.2	Parameter Learning	61
3.4	Numerical Results	61
3.4.1	Synthetic Data	61
3.4.2	Cell Signaling Data	65
3.4.3	Electroencephalogram (EEG) Data	66
3.5	Summary	68

4	Copula Gaussian Graphical Models for Discrete Data	69
4.1	Discrete Copula Gaussian Graphical Models	70
4.2	Monte-Carlo Expectation Maximization Algorithm	71
4.3	Regularization Selection	74
4.3.1	Structure Learning	74
4.3.2	Parameter Learning	77
4.4	Numerical Results	77
4.4.1	Synthetic Data	77
4.4.2	Real Data	79
4.5	Summary	83
5	Network Inference and Change Point Detection for Piecewise-Stationary Time Series	85
5.1	Change Point Detection	86
5.2	Regularization Selection	88
5.3	Numerical Results	90
5.3.1	Synthetic Data	91
5.3.2	Real Data	93
5.4	Summary	95
II	Graphical Models for Marginal Analysis of Extreme-Value Data	97
6	Extreme-Value Graphical Models with Multiple Covariates	99
6.1	Thin-membrane models	101

6.2	Modeling Multiple Covariates	103
6.2.1	Local estimates of GEV parameters	104
6.2.2	Prior distribution	106
6.2.3	Posterior distribution	109
6.3	Learning and Inference	111
6.3.1	Inferring smoothed GEV Parameters	111
6.3.2	Estimating Covariance Matrices R_z	114
6.3.3	Learning Smoothness Parameters	114
6.3.4	Bootstrapping the uncertainty of GEV estimates	118
6.4	Numerical Results	118
6.4.1	Synthetic Data	118
6.4.2	Real Data	123
6.5	Summary	126
7	Modeling Spatio-Temporal Extreme Events using Graphical Models	127
7.1	Thin-Plate Models	129
7.2	Spatio-Temporal Models for Extreme Events	132
7.2.1	Likelihood: Generalized Extreme Value Distributions	132
7.2.2	Prior: Thin-plate Models	133
7.2.3	Spatio-Temporal Graphical Models for Extreme Values	136
7.3	Learning and Inference	136
7.3.1	Learning GEV and Smoothness Parameters	137
7.3.2	Prediction of Future GEV Parameters	147

7.4	Numerical Results	147
7.4.1	Synthetic Data	149
7.4.2	Real Data	155
7.5	Summary	158
 III Graphical Models for Joint Analysis of Extreme-Value Data		159
8	Modeling Spatial Extremes by Copula Gaussian Graphical Models	161
8.1	Thin-Membrane Models	162
8.2	GEV Marginals	163
8.3	Copula Gaussian Graphical Models	164
8.4	Interpolation	166
8.5	Numerical Results	168
8.5.1	Synthetic Data	168
8.5.2	Real Data	173
8.6	Summary	176
9	Modeling Spatial Extremes via Ensemble-of-Trees of Pairwise Copulas	177
9.1	Ensemble-of-Trees of Pairwise Copulas	178
9.2	Inference of Missing Data	183
9.2.1	Smoothing-based Optimization	184
9.2.2	Stochastic Smoothing-based Optimization	185
9.2.3	SSO for Imputation in the ETPC Model	187

9.3	Theoretical Results	188
9.4	Experimental Results	191
9.4.1	Model Fitting	192
9.4.2	Imputation of Missing Data	193
9.5	Summary	196
10	Conclusions and Outlook	197
10.1	Summary of Contributions	197
10.1.1	Graphical Models for Nominal Data	197
10.1.2	Graphical Models for Marginal Analysis of Extremal Data	199
10.1.3	Graphical Models for Joint Analysis of Extremal Data	200
10.2	Recommendations for Future Works	201
10.2.1	Cyclic Pairwise Copula Graphical Models	201
10.2.2	Multiscale Pairwise Copula Graphical Models	201
A	Proofs and Derivations of Results in Chapter 6	203
A.1	Derivation of the first step in the FTM solver	203
A.2	Proof of Theorem 6.1	204
A.3	Proof of Theorem 6.2	205
A.4	Derivation of the Q -function	206
B	Proofs and Derivations of Results in Chapter 7	207
B.1	Derivation of the Gradients	207
B.2	Gradients of the Logarithm of GEV densities	208
B.3	Proof of Proposition 7.1	209
B.4	Gibbs Sampling Method to Learn the STM	210

C Proofs of Results in Chapter 9	212
C.1 Proof of Theorem 9.2	212
C.2 Proof of Proposition 9.1	213
C.3 Proof of Proposition 9.2	216
C.4 Proof of Proposition 9.3	219
 Author's Publications	 221
 Bibliography	 242

List of Figures

1.1	A graphical model with hidden variables (yellow nodes): (a) the joint graphical model; (b) the marginal graphical model of observed variables (green nodes); (c) the conditional graphical model of observed variables.	4
2.1	$\mathcal{S}$ separates the nodes sets $\mathcal{A}$ and $\mathcal{B}$. The Markov property in the undirected graph model indicates that $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ are independent conditioned on $\mathbf{x}_{\mathcal{S}}$	23
2.2	Message from node i to its neighbor j . The subtree in the blue circle is the one rooted at node i away from j	30
2.3	PDF of the Gumbel (blue), the Fréchet (red) and the Weibull (black) distribution.	35
2.4	PDFs of the Generalized Pareto Distributions with the shape parameter $\xi > 0$, $\xi = 0$ and $\xi < 0$	38
2.5	Densities of elliptical copulas. Note that the density of the t copula has been cut at a level of 15.	42
2.6	Densities of the Clayton (a) and the Frank (b) copula. Note that the density of the Clayton copula has been cut at a level of 15.	45
2.7	Densities of extreme-value copulas. Note that all copulas been cut at a level of 15.	48

3.1	A graphical model with hidden variables (yellow nodes): (a) the joint graphical model; (b) the marginal graphical model of observed variables (green nodes); (c) the conditional graphical model of observed variables.	56
3.2	A copula Gaussian graphical model with hidden variables: blue nodes denote observed non-Gaussian variables, green nodes denote latent Gaussian variables and yellow nodes denote hidden variables in the latent layer.	58
3.3	Results of different methods on the 20-dimensional synthetic dataset.	64
3.4	Results of different methods on the 60-dimensional synthetic dataset.	64
3.5	Histogram of the cell signaling data	65
3.6	Results of different methods on the cell signaling data.	66
3.7	Conditional precision matrix K_0 (HVCGGM) of first three AD patients and control subjects.	67
3.8	Boxplots for the sparsity measure of HVCGGM and HVGGM (The corresponding p -value are 0.0121, 0.9048, 0.9894 and 0.1561)	68
4.1	The distribution of selection frequencies based on a simulated dataset (a) all the edges; (b) null edges and true edges separately.	75
4.2	Results of different methods on the GSS data: (a) Hoff's method [28]; (b) the GGM; (c) the CGGM; (d) DCGGM.	80
4.3	Results of different methods on the Rochdale data: (a) Whittaker's method [116]; (b) the GGM; (c) the CGGM; (d) the DCGGM.	81
4.4	Network of US senators: Democratic, Republican, and Independent senators are respectively represented by red, blue, and yellow nodes. .	83
5.1	Adaptive regularization selection.	89

5.2	Results of functional networks resulting from graphical models without hidden variables (a)-(g) and with hidden variables (h)-(n) (with the vertical lines denoting the onset and end of one example seizure) .	94
6.1	Thin-membrane models: (a) chain graph; (b) cycle graph; (c) lattice; (d) spatio-directional model.	107
6.2	Estimates of scale parameter σ across all sites in one directional sector.	119
6.3	(a) Estimates of the shape parameters (solid lines) with 95% confidence intervals (dashed lines) across different directions at a randomly selected site; (b) magnified image corresponding to the part in the black square.	120
6.4	Mean square error (MSE) of the GEV estimates as a function of sample size (averaged over 100 data sets).	121
6.5	MSE for varying proportion of observed sites (averaged over 100 trials). The MSE decreases with increasing number of observed sites, as expected.	122
6.6	Heterogeneity in GOMOS data. The wave heights clearly depend on the location in the Gulf of Mexico and the wind direction.	123
6.7	Scatter plots of wave height at pairs of sites.	124
6.8	AIC score as a function of the selected number of directional sectors.	125
7.1	Neighborhood structure of the graphical model for spatial components.	134
7.2	(a) cycle graph to describe periodicity; (b) chain graph to describe trend; (c) neighborhood structure of the graphical model for temporal components.	135
7.3	Estimates of the spatial component of a scale parameter <i>w.r.t.</i> number of iterations using a proportion of all the observations.	149

7.4	Comparison the results from the MCMC method and the SVI method: the distribution of the spatial component of a shape (a), a scale (b), and a location parameter (c) and the distribution of the temporal component of a shape (d), a scale (e), and a location parameter (f).	151
7.5	Estimates of GEV parameters across time: (a) estimates of scale parameter σ across time; (b) estimates of location parameter μ across time.	152
7.6	Estimates of scale parameter σ across all sites at one time point.	153
7.7	Estimates of location parameter μ across all sites at one time point.	153
7.8	MSE for varying proportion of missing data (averaged over 100 trials). The MSE increases with growing proportion of missing data, as expected.	154
7.9	Non-stationarity in extreme rainfall data: (a) Distribution of monthly maximum rainfall of all months in 26 years at 12 randomly selected sites; (b) Distribution of monthly maximum rainfall from January to December of all 26 years at a random site. The distribution of extreme rainfall clearly depends on the location and the month.	156
7.10	Scatter plots of monthly maximum rainfall amount at pairs of sites: (a) two contiguous sites; (b) two distant sites.	156
7.11	Estimated shape parameters in different months.	157
8.1	Generalized thin-membrane model: (a) Regular grid; (b) Irregular grid.	163
8.2	Illustration of the modified interpolation method	167
8.3	True GEV parameters for synthetic data: (a) Location parameter surface; (b) Scale parameter surface.	169
8.4	Coordinates of observed and unobserved sites for Case Study 1.	170
8.5	Coordinates of observed and unobserved sites for Case Study 2.	172

8.6	The maximum wave height corresponding to one storm for 256 sites located on 16x16 lattice.	173
8.7	Interpolation of the maximum wave heights caused by a storm (in small regular grid). (a) True graph; (b) Observed sites (black) and unobserved sites (red) on the grid; Interpolation by (c) the copula MRF-GEV model; (d) the MRF-GEV model; (e) the copula GEV model; (f) the GGM.	174
8.8	Interpolation of the maximum wave heights caused by a storm (irregular grid covering the Gulf of Mexico). (a) True graph; (b) Observed sites (black) and unobserved sites (red) on the grid; Interpolation by (c) the copula MRF-GEV model; (d) the MRF-GEV model; (e) the copula GEV model; (f) the GGM.	175
9.1	ET model: the lattice (a) and several decomposed spanning trees (b) - (e).	180
9.2	Normalized histogram (blue bars) and estimated GEV distribution (red lines) for one randomly selected site in each of the four regions. .	191
9.3	Scatter plot of transformed extreme precipitation values for a randomly selected pair of sites at the Gaussian layer for each region. . .	192
9.4	Color plots of the imputed 6×6 area in the first region for the 99 th quantile.	195
10.1	A graphical models with multiscale structure: (a) a monoscale graphical model; (b) a latent tree (i.e., a quadtree); (c) a multiscale graphical model.	202
C.1	An example of a tree graphical model with pairwise upper tail dependence.	216

List of Tables

3.1	Quantitative comparison of different methods	63
4.1	Monte-Carlo EM algorithm for learning the discrete copula Gaussian graphical model.	73
4.2	Quantitative comparison of Gaussian graphical models (GGM), copula Gaussian graphical models (CGGM), and discrete copula Gaussian graphical models (DCGGM).	78
5.1	Comparison of different methods for accuracy of change point (cp) detection	92
5.2	Results for seizure data	93
6.1	The learning and inference algorithm for modeling multiple covariates.	117
6.2	Quantitative comparison of different models	121
7.1	Stochastic variational inference of the spatio-temporal model.	146
7.2	Performance of the SVI algorithm when the size of the mini-batch changes.	150
7.3	Quantitative comparison of different models for spatio-temporal extreme-value data	154
8.1	Mean square estimation error for GEV parameters in Case Study 1 .	170

8.2	Comparison for Case Study 1	171
8.3	Mean square estimation error for GEV parameters in Case Study 2 .	172
8.4	Comparison for Case Study 2	173
8.5	Comparison for regular grid in the Gulf of Mexico	174
8.6	Comparison for irregular grid in the Gulf of Mexico	176
9.1	The learning algorithm of the ETPC model.	183
9.2	The imputation algorithm of the ETPC model.	188
9.3	Comparison of goodness of fit of three models.	193
9.4	Summary of the MAE (mm) of imputed precipitation values with respect to the length of the square region of missing values for 5 quantiles.	194

Chapter 1

Introduction

1.1 Motivation and Related Works

Graphical models [1], also known in different scenarios as Bayesian networks, Markov random fields, and factor graphs, provide a refined language to describe complicated systems with large numbers of random variables. Despite the great quantities of variables involved, graphical models allow efficient representation of such systems with few parameters by learning or imposing a sparse dependency structure among these variables. Furthermore, such structure can in turn be utilized to derive highly efficient computational algorithms for making inference about likely values of some variables given observations of others. As a result, graphical models have found applications and permeated the literature in a wide variety of domains, such as signal processing [2, 3], image processing and computer vision [4, 5, 6], genetics and computational biology [7, 8, 9, 10], geophysics and remote sensing [12, 11, 13], and communication and coding theory [14].

However, despite their benefits and utility, most techniques of graphical models for continuous variables are only applicable to Gaussian variables. In other words, continuous graphical models have yet to be explored in the realm of non-Gaussian statistics. On the other hand, many real-world problems exhibit non-Gaussian behaviors. For example, this is the case of all positive quantities (amplitude, energy,

magnitude), which are commonplace in physics and earth sciences. In addition, extreme events such as hurricanes and floods, which are of special interest in natural hazards and risk analysis, cannot be described accurately by means of Gaussian distributions. Although statistical models have been developed for such data, they are often limited to tens of variables, and therefore are typically infeasible for practical large-scale problems. To bridge the gap, we propose a portfolio of new graphical models for non-Gaussian data in this thesis. Such graphical models enable us to solve large-scale real-life inference problems efficiently, while relaxing the assumption of Gaussianity, and allowing us to better quantify uncertainties and trade-offs in model parameters.

To move toward this goal, we apply copula theory (cf. [15]) when devising graphical models. Copulas are powerful tools which aid in flexible construction of the joint probability distribution by “gluing” together arbitrary families of marginal distributions (Gaussian, non-Gaussian and even non-parametric). To integrate both graphical models and copulas, we construct copula-based joint distributions whose dependency structures are specified through the language of graphical models. The resulting copula graphical models can capture the interactions between a multitude of non-Gaussian random variables in an accurate and compact manner. We then develop a set of algorithms to learn the model structure and parameters, and also to complete a variety of inference tasks, including i) imputing missing data; ii) interpolation; and iii) forecasting future values. More concretely, we broadly divide non-Gaussian data into two classes: nominal (non-Gaussian and non-extremal) and extremal data, and then employ different families of copulas as the building blocks of the corresponding graphical models. In the remainder of this section, we review related works for both types of data, point out their disadvantages, and further motivate our research.

1.1.1 Graphical Models for Nominal Data

For nominal data, the building blocks are Gaussian copulas. The resulting graphical models are referred to as copula Gaussian graphical models (CGGM). Such models enjoy the same advantages of Gaussian graphical models (GGM), while being applicable to non-Gaussian data. Here, we focus on learning the CGGM, in particular, the dependency structure, from the non-Gaussian data. As CGGMs can be built upon Gaussian graphical models (GGM), we first review GGMs without and with hidden variables, and further consider the corresponding CGGMs. We will develop models with constant parameters first, and next extend them to include abruptly time-changing parameters.

A GGM is directly defined by its precision matrix (inverse covariance matrix): a zero element corresponds to the absence of an edge in the graphical model or the conditional independence between two variables. When learning a graphical model, sparsity is a priority since a sparse graph can display the most significant dependence and provide a clear direction for follow-up analysis. For instance, inferring the gene regulatory networks (i.e., which genes may have correlation and which are expressed independently) is currently en vogue, since it is an effective strategy to detect novel disease mechanisms [16]. Now suppose one has a collection of one million genes and a graphical model is used to represent the network where the nodes denote the genes and there exists an edge between two nodes if the corresponding two genes are conditionally dependent. Under this setting, we only need to detect the most significant correlated pairs among the genes, which can then be characterized by the limited number of edges in a sparse graphical model. Such a graphical model will greatly reduce the time and effort for further analysis. A dense graphical model, on the contrary, buries the important connections by massive false positives, and thus hinders the interpretation of the gene network.

As such, for a standard GGM, we aim to estimate a sparse precision matrix. One common method of finding the sparse precision matrix given a certain number of Gaussian variables is to maximize the Gaussian log-likelihood with an ℓ_1 penalty

Figure 1.1: A graphical model with hidden variables (yellow nodes): (a) the joint graphical model; (b) the marginal graphical model of observed variables (green nodes); (c) the conditional graphical model of observed variables.

on the elements of the precision matrix [17]. The convex optimization problem can be solved by a simple and fast algorithm, namely the graphical lasso or glasso algorithm [18].

However, glasso can produce an interpretable network only when all the relevant variables are available, i.e., none of them are missing. Unfortunately, there may exist some hidden variables. In the example of gene networks, one typically measures the expression of a limited subset of genes (green nodes in Fig. 1.1(a)), chosen among the large number of genes of an organism; the observed genes may be strongly affected by genes that have not been measured. The latter may then be treated as hidden variables in a statistical model (yellow nodes in Fig. 1.1(a)). In this case, glasso would yield a dense gene network (see Fig. 1.1(b)) since the observed genes are related to the common set of hidden variables and therefore integrating out the hidden variables introduces a large number of interactions between the observed variables. Instead, considering the hidden variables can lead to a simple explanation for the statistical relations between the observed genes (see Fig. 1.1(c)). Therefore, sparse graphical models with hidden variables are quite powerful models for a large variety of real-life data sets. When the observed variables $\mathbf{z}_O$ and hidden variables $\mathbf{z}_H$ are jointly Gaussian distributed, the structure of the Gaussian graphical model can be defined by the precision matrix of the observed and hidden variables. Recently, Chandrasekaran *et al.* [19] decomposed the (marginal) precision matrix of $\mathbf{z}_O$ into a sparse matrix K_{OO} (conditional precision matrix) and a low-rank matrix L , which

describes the coupling between the observed and hidden variables. The conditional precision matrix K_{OO} and the number of hidden variables, measured by the rank of L , are inferred by solving a convex regularized maximum likelihood problem. The conditional precision matrix K_{OO} corresponds to Fig. 1.1(c). This graph visualizes the dependence among the observed variables conditioned on the hidden variables. In other words, such graph shows the interactions between observed variables that cannot be explained by hidden variables alone.

From the above discussion, there seems to be a proper model for all kinds of large-scale Gaussian distributed data. However, all these Gaussian graphical models are limited on the assumption of Gaussianity. On the other hand, existing models for non-Gaussian nominal data are restricted to low dimensions. The climate system, for example, can be defined as the coupled system of ocean, atmosphere, cryosphere, biosphere, lithosphere and anthroposphere. This overall system is described by physical quantities such as precipitation, wind speed, cloud cover, and humidity, which can be regarded as non-Gaussian random variables [20]. So far, standard copulas have been used in meteorological and climate research, as they can deal with those non-Gaussian physical variables [21]. However, standard copula models are limited to about 50 variables, and hence are of limited use in climate modeling where a huge number of variables are involved. To alleviate this issue, we develop a set of CGGMs as a non-Gaussian counterpart of the aforementioned set of GGMs. Specifically, non-Gaussian observed variables is transformed to Gaussian latent variables according to the definition of Gaussian copulas, and next a sparse graphical model is learned from the Gaussian data. Moreover, as shown in [22], conditional independence between non-Gaussian variables is characterized by the precision matrix of the latent Gaussian variables. Consequently, such models can handle non-Gaussian variables, and are applicable to large-scale systems, therefore, making them highly suitable for practical applications.

We now turn our attention to discrete data. Existing structure learning algorithm for discrete graphical models can only tackle binary variables, such as the Ising model [23, 24], and may be extended to the case where all the variables take the

same set of ordinal values [25, 26]. When variables are heterogeneous (i.e., take different sets of values), we can resort to Gaussian copula based approaches. However, when mapping the discrete observed variables to the corresponding Gaussian latent variables, the situation is more convoluted; discrete data cannot be transformed directly into Gaussian data, since the mapping is one-to-many. A common approach is to apply Markov chain Monte Carlo method (MCMC) to simulate both the latent Gaussian variables and the posterior distribution of the precision matrix [27, 28, 29]. Different priors have been selected for the precision matrix (or the covariance matrix), including covariance selection prior [27], the inverse-Wishart distribution [28] and the G-Wishart distribution [29]. On the other hand, the framework of factor model is also used to learn the dependence structure with the generalized double Pareto prior [30]. All these models based on MCMC methods suffer from the issue of a significant amount of computational cost. In this thesis, we develop a novel learning method of discrete copula Gaussian graphical models (DCGGM). The proposed method is reliable yet much more efficient than the full MCMC approach, as it avoids the costly Monte Carlo simulations for the posterior distribution of the precision matrix.

So far, we only consider stationary *i.i.d.* (independent and identically distributed) data. However, real data are often non-stationary, and statistical models designed for stationary data may not yield accurate results. For example, during epileptic seizures, functional brain networks are shown to evolve through a sequence of distinct topologies [32]. Inferring such evolving networks in the framework of graphical models has received little attention until now. A reasonable approach is to detect change points, and then infer graphical models in the stationary segments between the change points. Xuan *et al* [33] employed the Bayesian change point detection approaches: they adopt a geometric prior on the time segment lengths, and then iterate between maximum a posteriori (MAP) segmentation and graphical model inference. The main restriction, however, is that the graph for all segments must be decomposable. On the other hand, a greedy binary segmentation scheme is proposed in [34]. A change point is inserted such that the Bayesian Information

Criterion (BIC) score of the two graphical models of the data before and after the change point is minimized; this procedure is repeated until no further splits reduce the BIC score. Unfortunately, besides the high computational complexity, the binary segmentation can be misleading and overestimate the number of change points, as pointed out in [35]. To address this concern, dynamic programming is applied in [36], resulting in joint estimation of all the change points. Unfortunately, the method has computational complexity of order $\mathcal{O}(N^3)$ in the length of a time series N , which is impractical for most real-life time series with tens or hundreds of change points. In order to reduce the computational complexity, we disentangle the process of change point detection and graphical model inference in our approach. Specifically, we first detect the change points by minimizing a cost function defined on covariance matrix using low-complexity Pruned Exact Linear Time (PELT) method [37], and next learn the graphical model based on the covariance of each stationary time segment using methods we have developed before. Another limitation of the aforementioned methods is the assumption of Gaussian distributed data, which is not always fulfilled in practice. To overcome the limitation, we learn CGGMs instead of GGMs from each stationary time segment.

1.1.2 Graphical Models for Marginal Analysis of Extremal Data

The second type of data we analyze in this thesis is extreme-value data. Different from nominal data, extreme-value data usually follows fat-tailed distributions. In practice, extreme events such as heat waves, cold snaps, tropical cyclones, hurricanes, heavy precipitation and floods, droughts and wild fires, have possibly tremendous impact on people's lives and properties. There are plenty of such examples. To name a few, China experienced massive flooding of parts of the Yangtze River in the summer of 1998, resulting in about 4,000 dead, 15 million homeless and 26 billion USD in economic loss [38]; Hurricane Katrina in 2005 was the costliest and deadliest Atlantic hurricane, with a death toll of at least 1,833 and a total loss of 108 billion

USD [39]. To make matters worse, both observational data and computer climate models suggest that the occurrence and sizes of such catastrophes will increase in the future [40]. It is therefore imperative to model such events, assess the risk, and further take precaution measurements.

Literature on extreme events can be broadly divided into two classes: marginal and joint analysis. The objective of marginal analysis is to accurately estimate the marginal extreme-value distributions. Extreme-value theory provides closed-form distribution functions for the extremes of single variables (marginals). Specifically, regardless of the underlying distribution, peaks over a sufficiently high threshold can be described by a Generalized Pareto (GP) distribution, while the block maxima (e.g., annual maxima) can be characterized by a Generalized Extreme Value (GEV) distribution [41]. The main challenge in fitting such distributions to measurements is the lack of data, as extreme events are by definition very rare. The problem can be alleviated by assuming that all the collected data (e.g., extreme wave heights at different measuring sites [42]) are stationary and follow the same distribution. After combining all the data, the resulting sample size may be sufficiently large to yield reliable estimates. However, there usually exists clear heterogeneity in the extreme-value data caused by the underlying mechanisms that drive the weather events. Extreme temperature, for example, is greatly influenced by the altitude of the measuring site. The latter can be regarded as a covariate. Accommodating heterogeneity in the model is essential since the estimated model will be unreliable otherwise [43]. In order to handle both heterogeneity as well as the problem of small sample size, the interactions among extreme events with different covariate values are often exploited. For instance, extreme temperatures at similar altitudes behave similarly, implying that the parameters of the corresponding extreme-value distributions vary smoothly with the covariate (i.e., altitude). Such prior knowledge may help to improve the fitting of extreme-value distributions.

Studies on modeling covariates can be further divided into two categories: models with single [44]-[48] and multiple covariates [49]-[52]. Approaches of the first group usually treat the parameters of the marginal extreme-value distributions as a func-

tion of the covariate. In [44]-[46], directional and seasonal effects are considered when describing the marginal behavior of extreme wave heights. The dependence of the parameters on the single covariate is captured by a Fourier expansion. Spatial effects are investigated in [47]: the parameters are assumed to be Legendre polynomials of the location; extreme-value threshold is determined through quantile regression. Although these parametric models offer a simple framework to capture the covariate effect, they are prone to model misspecification. A more appealing approach is to incorporate the covariate in a non-parametric manner. The work in [48] employs a Gaussian process to induce spatial dependence among the parameters of extreme-value distributions in a spatial domain. This also enables the use of the MCMC algorithm to learn the model. However, such procedures are computationally complex and can be prohibitive for large-scale systems.

On the other hand, few attempts have been made to model multiple covariates. A standard method is to predefine the distribution parameters as a function of all the covariates [41]. Unfortunately, the function can be quite complicated as the number of covariates increases, whereas only linear or log-linear model are used in [41] for simplicity. The resulting estimates may be biased due to the misspecification of the functional form. As an alternative, Eastoe *et al.* [49] proposed to remove the heterogeneity of the entire data set, both extreme and non-extreme through preprocessing, and then model the extremal part of the preprocessed data using the above mentioned standard approach. They found that the preprocessing technique can indeed remove almost all the heterogeneity, and consequently, simple linear or log-linear models are capable of expressing the residual heterogeneity. Motivated by this success, Jonathan *et al.* [50] proposed to process two different covariates individually. They removed the effect of the first covariate by whitening the data using a linear location-scale model and then employed the methods for a single covariate to accommodate the second one. A setback of their method, however, is that the dependence on the first covariate may be nonlinear and therefore the whitening step cannot completely remove the effect of the first covariate. Furthermore, capturing the two covariates independently fails to consider the possible correlation between

them. To accommodate all the covariates at the same time, a spline-based generalized additive model is introduced in [51], where the spline smoothers for each covariate are added to the original likelihood function and then the penalized likelihood is maximized. Similarly, Randell *et al.* [52] addressed the problem by means of penalized tensor products of B-splines so as to obtain a smooth dependence of the distribution parameters *w.r.t.* all the covariates. The spline-based methods have the virtue of extending the methods for single covariates to the case of multiple covariates. Unfortunately, directly maximizing the complex penalized likelihood has several problems. First, the algorithm can be time-consuming. Typically, Newton's method is first employed and the iterative back fitting is then used to solve each Newton step. Thus, the algorithm has at least two loops, and the inner loop, i.e., the iterative back fitting, usually has a slow rate of convergence. Moreover, good initial points are essential for the algorithm to find the global optimum. Finally, choosing the smoothness parameters can be problematic as pointed out in [53]. The aforementioned shortcomings spark our interest in exploiting graphical models to incorporate multiple covariates in the extreme-value model. The interdependencies between extreme values with different covariates are often highly structured, and such structure renders efficient inference possible.

Next, let us turn our attention to the marginal analysis of multiple extreme-value time series. As an example, extreme precipitation can characterize climate change [40] and cause flood or flash-flood related hazards [54]. Therefore, assessing the temporal pattern of such events and making reliable predictions of future trends is crucial for risk management and disaster prevention. For stationary data, one of the most common approaches for describing the extreme events of stationary data is the block maximum approach, which models the maxima of a set of contiguous blocks of observations using the GEV distribution [41]. It has been shown that block maxima of stationary time series that are sufficiently separated (e.g., annual maxima) are almost independent [41]. However, increasing the block size to well separate the block maxima often leads to a small sample size, and thus, the resulting estimates of GEV parameters are unreliable. The problem can be alleviated by considering

the dependence between multiple GEV distributed variables (e.g., annual maximum precipitation at multiple locations) [47, 48]. As an alternative, instead of using one single sample (i.e., the maximum) in a large block, we can shrink the size of the blocks and introduce temporal dependence to the extreme-value samples so as to utilize the data more effectively. The resulting analysis often copes with GEV distributions whose parameters varying smoothly across both space and time. Apart from stationary time series, non-stationarity in the underlying process such as seasonality, trend, regime changes and dependence on external factors are often the rule rather than the exception. Hence, when modeling block maxima of this process, such covariate effects also need to be taken into account, and this again leads to a spatio-temporal model. Unfortunately, under both settings, only one sample is available at each measuring site and time point, whereas the corresponding GEV distribution has three parameters to be estimated.

Due to the abovementioned challenge, there are only a handful of spatio-temporal models for extreme events at present. In the following, we review the literature on spatio-temporal extreme events. A clear account of temporal dependence is presented in [41], which defines the temporal changes of GEV parameters through deterministic functions, such as linear, log-linear and quadratic functions. However, the restrictive functional forms pose a serious limitation in practice. A more satisfactory approach is to replace the deterministic function with a linear combination of suitable basis functions, such as splines, and add a penalty to guard against overfitting [55]. The smoothness parameters (i.e., penalty parameters) are chosen through cross validation or AIC. The tuning process is usually computationally burdensome since numerous candidate values of the smoothness parameters have to be tested before the proper amount of smoothness is determined. Moreover, the computational complexity increases exponentially with the number of smoothness parameters. To overcome this deficiency, a Bayesian approach is advanced in [56], in which the smoothness parameters are regarded as random variables and Gamma priors are imposed on them. Nevertheless, such Bayesian models are often inferred by the Monte Carlo Markov chain (MCMC) algorithm, and the latter can be unac-

ceptably slow for large-scale problems. On the other hand, Neville *et al.* [57] apply the mean field variational Bayes method to learn the model. Similarly as in [58], the GEV distributions are approximated by Gaussian mixtures for a set of shape parameters, in order to avoid the complex functional form of the GEV distributions. Closed-form update rules for the parameters of the variational distributions are derived. As a consequence, the algorithm needs to be run once for each possible value of the shape parameter in the predefined set before the one associated with the largest likelihood is selected.

Another line of research investigates the application of dynamic linear models (DLM) to extreme-value time series, cf. [59]-[63]. DLMs relate the present GEV parameters to the historical estimates while embedding the spatial dependence implicitly in the evolution matrix. Such models are often estimated via MCMC methods [59]-[61, 63] or generalized expectation maximization [62]. Unfortunately, the computational cost of learning a DLM is $\mathcal{O}(NP^3)$, where P is the number of measuring stations and N is the number of block maxima observed at each station (i.e., length of time series). As a result, such models are prohibitive for the cases where P is large. Furthermore, DLMs use directed acyclic graphs to represent the dependence from past to present, and hence, the estimates of GEV parameters at time t only depend on extreme-value samples up to t . Obviously, it is more tempting to take full advantage of all the observed samples, both before and after t , to yield more reliable estimates of GEV parameters. This indicates that GEV parameters at different time points depend on each other, thus constituting an undirected cyclic graph.

Apart from the above mentioned issues, the previous works [55]-[62] often restrict the shape (and the scale) parameters of the GEV distributions to be constant across the spatio-temporal domain. This assumption stems from the analysis on spatial extremes in [64], which shows that varying the shape parameter across the space only slightly improves the model fitting, but the resulting score of the deviance information criterion (DIC) [65] is larger than that with a constant shape parameter. Moreover, it also simplifies the corresponding learning algorithms. However, as pointed out in [63], treating the shape (and the scale) parameter as a constant is

inappropriate for modeling monthly maxima.

To break through the limitations of those previous works, we propose to exploit undirected graphical models (i.e., Markov random fields) to capture the highly structured spatio-temporal dependencies among all the three GEV parameters (i.e., shape, scale and location). We aim to estimate the temporal pattern of extreme events, such as the trend or seasonality of the data in time. Furthermore, we intend to predict the distribution of extreme events in the future based on the current trend. In the example of extreme rainfall, forecasting whether the size of extremes will increase in the future is crucial for flood warning and strategic planning.

1.1.3 Graphical Models for Joint Analysis of Extremal Data

Let us focus on joint analysis of extreme events now. The objective of joint analysis is to build a joint distribution of extreme events. It is often concerned with spatial extremes, since the extreme values (rather than the parameters of extreme-value distributions) at different locations are related to each other. The work of Cooley *et al.* [66], which surveys the different approaches that have been taken in incorporating spatial dependence when modeling extreme events, finds that recent literature have dealt with a two-stage approach. The first stage involves accurate marginal estimation, which has the same objective as marginal analysis, and thus the methods of marginal analysis can be directly applied. In the second stage, one couples the extreme-value marginals together to form a joint distribution via tools such as copulas and max-stable processes. Sang *et al.* [67] performed a two-stage analysis where a Gaussian copula connects the marginal distributions to form a joint distribution in the second stage. However, the resulting Gaussian covariance matrix is dense which renders extreme-value inference computationally complex, especially for domains with numerous sites. To address this concern, we propose to use a CGGM with a sparse precision matrix, and thus, that model can represent large-scale extreme-value data in a compact manner. The sparse graphical model

structure also greatly aids in the interpolation of extreme values on an unobserved location given data at other locations.

Although Gaussian copulas are computationally attractive, they are asymptotically tail independent, and therefore, they are not able to capture the statistical dependence between extremes (e.g., extreme wind gusts at different locations). This shortcoming has sparked interest in the statistics community in models that are asymptotically tail dependent, including max-stable processes and extreme-value (max-stable) copulas [68, 69]. Unfortunately, such models are defined by distribution functions with complicated functional forms. The corresponding likelihood function involves differentiation with respect to all the variables, resulting in a combinatorial explosion. As an example, the likelihood of a 10-dimensional extreme-value copula would include a sum of over 100,000 different terms [69]. Consequently, when fitting max-stable copulas or processes to extreme-value data, composite likelihoods are often employed [70]. This approach replaces the original likelihood by a pseudo-likelihood constructed from pairwise likelihoods. The main drawback, however, is that only one single class of copulas can be used in this framework, thus seriously limiting the modeling power. Vine copula models [71] address this shortcoming by using various types of pairwise copulas as building blocks to construct large multivariate models. More specifically, in such models, one decomposes the joint density as a product of conditional densities and further approximates the latter by products of pairwise copulas. It has been proven in [72] that vine copulas can capture tail dependence by properly selecting the base pairwise copulas. However, the biggest hurdle in applying vine copulas to large-scale data arises due to the quadratically increasing number of parameters with respect to the number of variables. Furthermore, both composite likelihood models and vine copulas suffer from the limitation that there is no principled approach to impose interpretable structure on these models. Consequently, different configurations have to be tested before the best structure can be determined. The problem is exacerbated in the setting of high-dimensional data, since the number of possible configurations increases factorially with the dimension. Again, as opposed to the aforementioned models, graphical

models are highly structured statistical models in which conditional independence can be easily defined given the location of the measuring stations. We therefore construct a graphical model by using pairwise copulas as building blocks, and prove that the tail dependence can be preserved under mild conditions.

1.2 Thesis Organization and Overview of Contributions

The thesis is organized as follows. Chapter 2 provides an overview of the background knowledge that is essential to the following chapters. Specifically, we begin this chapter with an introduction of graphical models. We review the corresponding learning and inference algorithms with a special focus on Gaussian graphical models. Next, we introduce extreme value theory, which involves two fundamental extreme-value families of distributions, i.e., the Generalized Pareto distribution and the Generalized Extreme Value distributions. We outline the relevant theorems and properties. Lastly, we present the copula theory, and introduce different families of copulas and their properties.

The remainder of the thesis, presenting our main contributions, can be divided into three parts: Part I (Chapter 3-5) is concerned with copula Gaussian graphical models for nominal non-Gaussian data, Part II (Chapter 6-7) deals with graphical models for marginal analysis of extreme events, and Part III (Chapter 8-9) focuses on graphical models for joint analysis of extreme events. In the sequel, we outline the content of each chapter.

Part I: Graphical Models for Nominal Data

In Chapter 3, we first give a brief introduction to the standard copula Gaussian graphical model and its learning algorithm. We also prove that the conditional independence between observed non-Gaussian variables can be characterized by the precision matrix of latent Gaussian variables. We then consider the case where there

exist hidden variables with no observations available, and propose hidden variable copula Gaussian graphical models. Such models capture dependencies between observed variables by introducing a suitable number of hidden variables. We also present an algorithm for automatically choosing the proper amount of regularization when learning those graphical models. The proposed model is then applied to both synthetic data and cell signaling data.

In Chapter 4, we extend copula Gaussian graphical models to deal with discrete variables. The proposed model can handle an arbitrary combination of heterogeneous discrete variables. We develop a low-complexity learning algorithm. More precisely, instead of sampling the precision matrix from its posterior distribution based on MCMC algorithm, we propose to generate MAP estimates using Monte-Carlo expectation maximization. In the E-step, an efficient Gibbs sampler is applied. In the M-step, the sparse graphical model is inferred by solving a penalized maximum likelihood problem. The regularization parameter is determined through the BINCO method. Numerical results for both synthetic and real social survey data are presented at the end of this chapter.

In Chapter 5, we study the case where the data are non-stationary. Techniques are proposed to infer graphical models from piecewise stationary time series; first change points are detected in the time series, and then graphical models are inferred for each stationary segment. Specifically, a low-complexity algorithm based on Pruned Exact Linear Time method is proposed to identify change points. Copula Gaussian graphical models (with and without hidden variables) are then generated for each stationary segment. The key advantage of the proposed approach is that it determines the number and location of the change points as well as the graphical models in a fully automated manner. We then validate the proposed algorithms on synthetic data. Finally, the model is employed to infer the time varying functional brain networks based on scalp electroencephalograms of epileptic seizure patients.

Part II: Graphical Models for Marginal Analysis of Extremal Data

We turn our attention to marginal analysis of extreme events in Chapter 6. Extreme

events often display heterogeneity; their statistics varying continuously with a number of covariates. Previous studies suggest that models considering covariate effects lead to reliable estimates of extreme value distributions. As such, we develop a novel model to incorporate the effects of multiple covariates. Concretely, we analyze as an example the extreme sea states in the Gulf of Mexico, where the distribution of extreme wave heights changes systematically with location and wind direction. The block maxima at each location and sector of wind direction are assumed to follow the Generalized Extreme Value (GEV) distribution. The GEV parameters are coupled across the spatio-directional domain through a graphical model, particularly, a multidimensional thin-membrane model. Efficient learning and inference algorithms are then developed based on the special characteristics of the thin-membrane model. We further show how to extend the model to incorporate an arbitrary number of covariates in a straightforward manner. Numerical results for both synthetic and real data indicate that the proposed model can accurately describe marginal behaviors of extreme events.

In Chapter 7, we study the spatio-temporal variation of marginal extremes. The model can be used, for instance, to estimate extreme-value temporal pattern such as seasonality and trend, and further to predict the distribution of extreme events in the future. Such model usually involves thousands or even millions of variables in the spatio-temporal domain, whereas only one single observation is available for each location and time point. To address this challenge, previous works usually employ learning and inference methods that are computationally burdensome, and therefore are prohibitive for large-scale data. Moreover, they assume that the shape (and the scale) parameters of the extreme-value distributions are constant across the spatio-temporal domain, which is often too restrictive in practice. In this chapter, we break through the limitations by exploring graphical models to capture the highly structured dependencies among the parameters of extreme-value distributions. Furthermore, we develop an efficient stochastic variational inference (SVI) algorithm to learn the parameters of the resulting non-Gaussian graphical model. The computational complexity of the SVI algorithm is sublinear in the number of

variables, thus enabling the proposed model to tackle large-scale spatio-temporal data in real-life applications. Results of both synthetic and real data demonstrate the effectiveness of the proposed approach.

Part III: Graphical Models for Joint Analysis of Extremal Data

We proceed to the second stage of extreme events modeling in Chapter 8. In this Chapter, we develop copula Gaussian graphical models to model extreme-value data. In particular, after smoothing the GEV parameters across the space using thin-membrane models, we further couple the extreme values through a copula Gaussian graphical model with a lattice structure. We then derive learning and interpolation algorithms for the proposed model. As a result, this model can be used to estimate extreme values at unmonitored sites. In addition, we further extend the model to the case of irregular grids. The approach is validated on synthetic data as well as real data related to hurricanes in the Gulf of Mexico. Numerical results indicate that it can accurately describe extreme events in spatial domain, and can reliably interpolate extreme values at arbitrary sites.

In Chapter 9, we address the tail independent problem of copula Gaussian graphical models by using tail dependent copulas as the building blocks of graphical models. In particular, we introduce an ensemble-of-trees of pairwise copulas (ETPC). In the proposed graphical model, extreme-value marginal distributions are stitched together by means of pairwise copulas, which in turn are the building blocks of the ensemble of trees. In addition, novel stochastic smoothing-based optimization (SSO) algorithms are proposed for inferring missing data in the ETPC model, which are applicable to large-scale statistical problems involving extreme values. We further prove that, under mild conditions, the ETPC model exhibits the favorable property of tail dependence between an arbitrary pair of sites (variables); consequently, the model is able to reliably capture statistical dependence between extreme values at different sites. Experimental results for extreme precipitation data of Japan demonstrate the advantages of the ETPC model.

Chapter 10 concludes the thesis by summarizing our contributions and suggesting

possible directions for future works.

Chapter 2

Preliminaries

In this chapter, we present the background knowledge necessary for the subsequent chapters. We begin Section 2.1 by formalizing the concept of graphical models, and then focus on Gaussian graphical models and their learning and inference methods. In Section 2.2, we outline the basics of univariate extreme value theory, and discuss two fundamental extreme-value distributions and their properties. Finally, in Section 2.3, we present copula theory along with different families of copulas and their properties.

2.1 Graphical Models

A graphical model consists of two parts: (1) a graph (including nodes and edges) whose connectivity characterizes the dependency relations in an underlying probability model, and (2) functions defined on local neighborhoods of nodes on the graph that parameterize local components of the overall joint probability model [3, 73]. We therefore introduce graphical models by first reviewing the fundamentals of graph theory, and then illustrating how to decompose the joint probability according to the graph.

2.1.1 Basic Graph Theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with a collection of nodes (or vertices) $\mathcal{V}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ (i.e., $\mathcal{E}$ is a subset of all pairs of nodes). Two nodes i and j are said to be *neighbors* if there is an edge $(i, j) \in \mathcal{E}$ connecting them. As a result, the *neighborhood* of a node i can be defined as the set $N(i) = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$, and the *degree* of the node i is the number of its neighbors $|N(i)|$. A *subgraph* $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ of $\mathcal{G}$ is defined by a subset of nodes $\mathcal{V}_s \subset \mathcal{V}$ and edges $\mathcal{E}_s \subset \mathcal{E}$ such that $\mathcal{E}_s \subset \mathcal{V}_s \times \mathcal{V}_s$. A *clique* C is a fully connected subgraph, that is, a subgraph in which each node is a neighbor of every other node. A *path* is a sequence of nodes and edges such that no node or edge is repeated and each pair of consecutive nodes i and $i + 1$ is connected by an edge $(i, i + 1) \in \mathcal{E}$. A graph is said to be *connected* if there is a path between two arbitrary nodes in the graph.

We now define some basic graphs. A *chain* is a connected graph where two of the nodes have one neighbor each, and all rest vertices have two neighbors. A *cycle* is a connected graph, where every node has exactly two neighbors. A *tree* is a connected graph which contains no cycles as subgraphs. In a tree, nodes with degree 1 are called *leaf nodes*. The *distance* between two arbitrary nodes is the number of edges in the path between them. Particularly, in an undirected tree, we can choose any node as the *root node*, and further define the parent-child relationships with respect to the root: for a pair of neighboring nodes i and j , if i is closer to the root (with a smaller distance), then i is said to be the parent of j , and j is called the child of i . One interesting property for a tree graph is that the root node does not have any parent, and there is only one parent for all other nodes in the tree. On the other hand, for a connected undirected graph $\mathcal{G}$, a *spanning tree* of the graph $\mathcal{G}$ is the tree that includes all the nodes.

Finally, we introduce *graph separation*, which is closely related to the definition of graphical models. Suppose that $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{S}$ are subsets of $\mathcal{V}$. Then, $\mathcal{S}$ separates $\mathcal{A}$ and $\mathcal{B}$ if there is no path connecting $\mathcal{A}$ and $\mathcal{B}$ that does not pass through $\mathcal{S}$, as shown in Fig. 2.1. Also, $\mathcal{S}$ is said to be a *separator* of the graph $\mathcal{G}$ if removing these

Figure 2.1: $\mathcal{S}$ separates the nodes sets $\mathcal{A}$ and $\mathcal{B}$. The Markov property in the undirected graph model indicates that $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ are independent conditioned on $\mathbf{x}_{\mathcal{S}}$.

nodes (and all edges that contain any of these nodes) disconnects some part of the graph such that the number of connected components increases.

2.1.2 Probability Factorization and Graphical Models

Graphical models are a family of multivariate distributions that factorizes according to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. More explicitly, the joint density of a collection of random variables can be decomposed as a product of several factors, each depending only on a small subset of the variables. Such factorizations can be represented by graphs in various ways; Bayesian networks [74, 75], Markov random fields (MRF) [76, 77], and factor graphs [3, 78] employ directed, undirected and hyper graphs respectively. In this thesis, we restrict our attention to Markov random fields (i.e., undirected graphical models).

Now suppose that we have a random vector $\mathbf{x} = [x_1, x_2, \dots, x_P]^T$ and each random variable x_i corresponds to a node i in the graph $\mathcal{G}$. A probability distribution $p(\mathbf{x})$ is said to be Markov with respect to the undirected graph $\mathcal{G}$ if it satisfies the global Markov property, that is, for any subsets $\mathcal{A}, \mathcal{B} \subset \mathcal{V}$ that are separated by a certain

$\mathcal{S} \subset \mathcal{V}$, $\mathbf{x}_{\mathcal{A}}$ and $\mathbf{x}_{\mathcal{B}}$ are independent conditioned on $\mathbf{x}_{\mathcal{S}}$:

$$p(\mathbf{x}_{\mathcal{A}}, \mathbf{x}_{\mathcal{B}} | \mathbf{x}_{\mathcal{S}}) = p(\mathbf{x}_{\mathcal{A}} | \mathbf{x}_{\mathcal{S}}) p(\mathbf{x}_{\mathcal{B}} | \mathbf{x}_{\mathcal{S}}). \quad (2.1)$$

Note that the above conditional independence is often denoted by $\mathbf{x}_{\mathcal{A}} \perp\!\!\!\perp \mathbf{x}_{\mathcal{B}} | \mathbf{x}_{\mathcal{S}}$ in the literature of graphical models. We can then obtain the local Markov property from the global one, that is, a random variable x_i is independent of all other variables conditioned on its neighbors:

$$p(x_i | \mathbf{x}_{\mathcal{V}|i}) = p(x_i | \mathbf{x}_{N(i)}), \quad (2.2)$$

where $\mathbf{x}_{\mathcal{V}|i}$ denotes all the variables except x_i . In addition, we can also derive the pairwise Markov property, i.e., an edge (i, j) is absent in the graph $\mathcal{G}$ if the corresponding two variables x_i and x_j are conditionally independent given the remaining variables:

$$p(x_i, x_j | \mathbf{x}_{\mathcal{V}|i,j}) = p(x_i | \mathbf{x}_{\mathcal{V}|i,j}) p(x_j | \mathbf{x}_{\mathcal{V}|i,j}). \quad (2.3)$$

In the subsequent chapters, unless otherwise specified, we will say that x_i and x_j are conditionally independent when $x_i \perp\!\!\!\perp x_j | \mathbf{x}_{\mathcal{V}|i,j}$.

Hammersley and Clifford further summarize the relation between the Markov properties implied by $\mathcal{G}$ and the factorization of the probability distribution $p(\mathbf{x})$ in the following theorem [79]:

Theorem 2.1. (Hammersley-Clifford Theorem) *If $p(\mathbf{x}) > 0$ for all $\mathbf{x}$, and $p(\mathbf{x})$ is Markov with respect to the graph $\mathcal{G}$, then $p(\mathbf{x})$ can be expressed as a product of factors corresponding to cliques of $\mathcal{G}$:*

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(\mathbf{x}_C), \quad (2.4)$$

where $\psi_C(\mathbf{x}_C)$ is a compatibility function defined on a clique C , $\mathcal{C}$ is the set of all cliques in $\mathcal{G}$, and Z is a normalization term called the partition function. Conversely,

if $p(\mathbf{x})$ can be factorized as in (2.4) with $\psi_C(\mathbf{x}_C) \geq 0$, then $p(\mathbf{x})$ is Markov with respect to the corresponding graph $\mathcal{G}$.

In this thesis, we restrict our attention to MRFs with pairwise interactions. In other words, the cliques are chosen as the nodes and edges in the graph, resulting in a pairwise MRF with the probability density function (PDF):

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \quad (2.5)$$

where $\psi_i(x_i)$ is a node potential and $\psi_{ij}(x_i, x_j)$ is an edge potential.

As mentioned in Section 2.1.1, if an undirected graph does not include any loops, it is called a tree (i.e., an acyclic graph). The corresponding joint distribution factorizes as:

$$p(\mathbf{x}) = \prod_{i \in \mathcal{V}} p_i(x_i) \prod_{(i,j) \in \mathcal{E}} \frac{p_{ij}(x_i, x_j)}{p_i(x_i)p_j(x_j)}. \quad (2.6)$$

Note that the above equation can be easily derived by choosing an arbitrary node as the root and expressing the joint distribution as the product the conditional distributions based on the parent-child relationships in the tree. For example, for a three-node Markov chain, we can select the first node as the root and the resulting joint distribution is factorized as:

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2) \quad (2.7)$$

$$= p(x_1) \left[\frac{p(x_1, x_2)}{p(x_1)} \right] \left[\frac{p(x_2, x_3)}{p(x_2)} \right] \quad (2.8)$$

$$= p(x_1)p(x_2)p(x_3) \left[\frac{p(x_1, x_2)}{p(x_1)p(x_2)} \right] \left[\frac{p(x_2, x_3)}{p(x_2)p(x_3)} \right] \quad (2.9)$$

Additionally, we can tell from Eq. (2.6) that the node marginals $p_i(x_i)$ and the pairwise joint distributions on edges $p_{ij}(x_i, x_j)$ fully describe a tree graphical model. In this case, the node potential $\psi_i(x_i) = p_i(x_i)$, the edge potential $\psi_{ij}(x_i, x_j) = p_{ij}(x_i, x_j)/p_i(x_i)p_j(x_j)$, and the partition function $Z = 1$.

For general cyclic graphs, computing the partition function Z is known as an NP-hard problem [80]. One exception is the Gaussian graphical model (GGM) in which all the variables are Gaussian distributed, which will be the subject of Section 2.1.3.

2.1.3 Gaussian Graphical Models

If the random variables $\mathbf{x}$ corresponding to the nodes in $\mathcal{G}$ are jointly Gaussian, then the undirected graphical model is called a Gaussian graphical model (GGM) (or Gauss-Markov random field (GMRF)). Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with mean vector $\boldsymbol{\mu}$ and positive-definite covariance matrix Σ . The GGM can be written equivalently as $\mathcal{N}(K^{-1}\mathbf{h}, K^{-1})$ with a precision matrix $K = \Sigma^{-1}$ and a potential vector $\mathbf{h} = K\boldsymbol{\mu}$. The resulting PDF can be expressed as:

$$p(\mathbf{x}) = (2\pi)^{-\frac{P}{2}} |K|^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T K \mathbf{x} + \mathbf{h}^T \mathbf{x}\right\}. \quad (2.10)$$

where $(2\pi)^{-\frac{P}{2}} |K|^{\frac{1}{2}}$ is the closed-form partition function. The corresponding node and edge potentials are:

$$\psi_i(x_i) = \exp\left\{-\frac{1}{2}K_{ii}x_i^2 + h_i x_i\right\}, \quad (2.11)$$

$$\psi_{ij}(x_i, x_j) = \exp\{-x_i K_{ij} x_j\}. \quad (2.12)$$

Obviously, $K_{ij} = 0$ implies that x_i and x_j are conditionally independent in a GGM. We emphasize that $K_{ij} = 0$ does not mean x_i and x_j are uncorrelated. Instead, Σ_{ij} usually is nonzero, since the inverse of a sparse precision matrix is often a full covariance matrix.

2.1.4 Learning Gaussian Graphical Models

2.1.4.1 Structure Learning

Since the structure of a GGM is well characterized by the precision matrix, we only need to infer the precision matrix when learning the GGM. Moreover, as mentioned

in Section 1.1.1, a sparse graphical model is usually preferable in practice. We therefore aim to learn a sparse precision matrix given the N multivariate Gaussian distributed observations $\mathbf{x}^{(1:N)}$. The resulting optimization problem can be formulated as [17]:

$$\hat{K} := \underset{K \succ 0}{\operatorname{argmin}} \operatorname{tr}(SK) - \log \det K + \lambda \|K\|_1, \quad (2.13)$$

where S is the empirical covariance matrix defined as:

$$S = \frac{1}{N} \sum_{i=1}^N \mathbf{x}^{(i)} (\mathbf{x}^{(i)})^T, \quad (2.14)$$

$\|\cdot\|_1$ is the ℓ_1 norm (i.e., the sum of the absolute value of all the elements in the matrix), and λ is the regularization or penalty parameter. Note that the term $\operatorname{tr}(SK) - \log \det K$ can be interpreted as the divergence between the estimated distribution and the empirical distribution or the negative log-likelihood of the observed data $\mathbf{x}^{(1:N)}$. As such, the problem minimizes the divergence between the estimated and original distribution with an ℓ_1 -norm penalty on the elements of the precision matrix K , resulting in a sparse GGM approximation. It is a tradeoff between the data fidelity and the model sparsity, which is determined by the regularization parameter λ . As shown in [18], if λ is small, then $\hat{K}$ tends to be dense; otherwise, $\hat{K}$ tends to be sparse. Actually, when λ is sufficiently large, all the off-diagonal elements of $\hat{K}$ are forced to be zero. To recover the correct matrices K , the parameter λ needs to be chosen appropriately, which is a critical issue that will be addressed in the following chapters. After selecting a proper λ , (2.13) can be solved efficiently using interior-point methods, block coordinate descent methods, or the graphical lasso (glasso) algorithm [18]. It has been proven in [81] that solving (2.13) can consistently yield the true graphical model structure (i.e., the sparsity pattern of the precision matrix) under certain conditions. However, it is not guaranteed to provide a consistent estimator of the nonzero elements in the precision matrix.

2.1.4.2 Parameter Learning

When the structure of the GGM is given, the parameters can be learned reliably by iterative proportional fitting (IPF) [82] (also known as iterative scaling [83]). In principle, IPF can be applied to general graphical models. It is a simple procedure that iteratively cycles over each clique in the graphical model, and updates the corresponding potential function such that the marginal distribution of the variables in the clique $p(\mathbf{x}_C)$ equals the corresponding empirical distribution. For GGMs with zero mean vector, IPF replaces the covariance of $\mathbf{x}_C$ by the empirical one in each iteration. Specifically, we can represent the current precision matrix $K^{(\kappa)}$ after permutation as:

$$K^{(\kappa)} = \begin{bmatrix} K_{CC}^{(\kappa)} & K_{C,\mathcal{V}|C}^{(\kappa)} \\ K_{\mathcal{V}|C,C}^{(\kappa)} & K_{\mathcal{V}|C,\mathcal{V}|C}^{(\kappa)} \end{bmatrix}, \quad (2.15)$$

where C is a clique. The covariance matching procedure then updates the precision matrix as [84]:

$$K^{(\kappa+1)} = \begin{bmatrix} S_{CC}^{-1} + K_{CC}^{(\kappa)} - [(K^{(\kappa)})^{-1}]_{CC}^{-1} & K_{C,\mathcal{V}|C}^{(\kappa)} \\ K_{\mathcal{V}|C,C}^{(\kappa)} & K_{\mathcal{V}|C,\mathcal{V}|C}^{(\kappa)} \end{bmatrix}, \quad (2.16)$$

where S_{CC} is the empirical covariance of $\mathbf{x}_C$, and $[(K^{(\kappa)})^{-1}]_{CC}$ is the covariance of $\mathbf{x}_C$ in iteration κ . When the algorithm converges, the resulting $\hat{K}$ is the maximum likelihood estimation of K with the given structure as proven in [1]. For an improved IPF with low complexity, readers may consult [85] and references therein.

2.1.5 Inference in Gaussian Graphical Models

Probabilistic inference in graphical models is concerned with inferring the unobserved variables $\mathbf{x}_U$ given observed ones $\mathbf{x}_O$. More concretely, one often would like

to obtain the posterior marginals of each unobserved variable:

$$p(x_i|\mathbf{x}_O) = \sum_{\mathbf{x}_{U|i}} p(\mathbf{x}_U|\mathbf{x}_O), \quad (2.17)$$

or compute the maximum a posteriori (MAP) estimate:

$$\hat{\mathbf{x}}_U = \operatorname{argmax}_{\mathbf{x}_U} p(\mathbf{x}_U|\mathbf{x}_O). \quad (2.18)$$

In the case of GGMs, we only need to compute the mean and variance of each unobserved variable.

The most popular tool used in this area is the belief propagation (BP) algorithm [86]. It provides an efficient linear-complexity algorithm for exact inference in tree graphs. However, cyclic graphs compare favorably with tree graphs in practice due to their richer modeling power. To deal with the complex estimation problem in cyclic graphs, various algorithms have been and are still being proposed. A straightforward method is to extend BP to cyclic graphical models, resulting in loopy belief propagation (LBP) [87, 88]. Unfortunately, the LBP is not guaranteed to converge or give accurate results. Alternatively, the embedded subgraphs (ES) algorithm [89, 90, 91] employs the idea of performing inference iteratively on tractable subgraphs. In the following, we review the BP and the ES algorithm, which will be used in the subsequent chapters.

2.1.5.1 Belief Propagation

Here, we introduce the BP algorithm sequentially for tree graphical models, cyclic graphical models, and Gaussian graphical models.

Recall from Section 2.1.2 that the partition function of a tree equals one and hence the corresponding PDF can be expressed as the product of node and edge potentials:

$$p(\mathbf{x}) = \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j). \quad (2.19)$$

Figure 2.2: Message from node i to its neighbor j . The subtree in the blue circle is the one rooted at node i away from j .

We can further define the posterior distribution of $\mathbf{x}_U$ conditioned on $\mathbf{x}_O$ as:

$$p(\mathbf{x}_U | \mathbf{x}_O) \propto \prod_{i \in \mathcal{V}} \tilde{\psi}_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \quad (2.20)$$

where $\tilde{\psi}_i(x_i) = \psi_i(x_i)$ for $i \in U$, while $\tilde{\psi}_i(x_i) = \psi_i(x_i)\delta(x_i, c_i)$ for $i \in O$. In the latter expression, c_i is the observed value of x_i and $\delta(x_i, c_i) = 1$ if $x_i = c_i$ and 0 otherwise. The marginal distribution of each unobserved variable can be computed by integrating over all other variables. Such marginalization can be performed efficiently via message passing between neighboring nodes in a tree [86]. Concretely, the message from node i to its neighbor j is given by:

$$m_{i \rightarrow j}(x_j) = \int_{x_i} \psi_{ij}(x_i, x_j) \tilde{\psi}_i(x_i) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i) dx_i. \quad (2.21)$$

This message $m_{i \rightarrow j}(x_j)$ captures the effect of eliminating (i.e., marginalizing) all variables in the subtree rooted at node i away from j (cf. Fig. 2.2). The belief propagation algorithm can then be proceeded in a tree as follows. We select an arbitrary node as the root and specify the parent-child relationships as described in Section 2.1.1. Next, we begin the upward pass by passing messages from each of the leaf nodes to its parent. The latter collects messages from all its children and further sends a message to its parent. Finally, when we reach the root, a downward pass starts by passing the messages in the opposite direction from the root to the leaves. After this one upward-downward pass, every node receives messages from all

its neighbors and the resulting marginal distribution can be computed as:

$$p(x_j|\mathbf{x}_O) \propto \tilde{\psi}_j(x_j) \prod_{i \in N(j)} m_{i \rightarrow j}(x_j). \quad (2.22)$$

Alternatively, instead of passing messages sequentially (upward followed by downward), we can also pass messages in parallel from all nodes to their neighbors by setting all the initial messages to 1. Both of the two message passing methods have the linear complexity. Note that the LBP algorithm adopts the parallel message passing scheme regardless of the graph structure, thus extending the BP to cyclic graphs in a straightforward manner. Similarly, the MAP estimates can be obtained by replacing the integrations with max operation.

For GGMs with node and edge potentials defined in (2.11) and (2.12), the message can be computed in a closed form as:

$$m_{i \rightarrow j}(x_j) = \exp\left\{\frac{1}{2}\Delta K_{i \rightarrow j}x_j^2 + \Delta h_{i \rightarrow j}x_j\right\}. \quad (2.23)$$

In practice, it is the parameters $(\Delta K_{i \rightarrow j}, \Delta h_{i \rightarrow j})$ that serve as the actual messages in Gaussian belief propagation. We initialize these parameters to 0, corresponding to uninformative initial messages $u_{i \rightarrow j}(x_j) = 1$. To calculate $\Delta K_{i \rightarrow j}$ and $\Delta h_{i \rightarrow j}$, we first combine messages gathered at node i with the node potential $\tilde{\psi}_i(x_i)$. In terms of parameters, this becomes:

$$\hat{K}_{i|j} = K_{ii} + \sum_{k \in N(i)|j} \Delta K_{k \rightarrow i}, \quad (2.24)$$

$$\hat{h}_{i|j} = h_i + \sum_{k \in N(i)|j} \Delta h_{k \rightarrow i}. \quad (2.25)$$

Then, by multiplying with the edge potential $\psi_{ij}(x_i, x_j)$ and integrating out x_i , we have:

$$\Delta K_{i \rightarrow j} = -K_{ji}\hat{K}_{i|j}^{-1}K_{ij}, \quad (2.26)$$

$$\Delta h_{i \rightarrow j} = -K_{ji}\hat{K}_{i|j}^{-1}\hat{h}_{i|j}. \quad (2.27)$$

After every node receives messages from its neighbors, the corresponding marginal parameters can be obtained by combining the node potential with the messages from all neighbors:

$$\hat{K}_{ii} = K_{ii} + \sum_{k \in N(i)} \Delta K_{k \rightarrow i}, \quad (2.28)$$

$$\hat{h}_i = h_i + \sum_{k \in N(i)} \Delta h_{k \rightarrow i}. \quad (2.29)$$

The resulting marginal mean and variance are given by:

$$\hat{\mu}_i = \hat{K}_{ii}^{-1} \hat{h}_i, \quad (2.30)$$

$$\hat{\Sigma}_{ii} = \hat{K}_{ii}^{-1}. \quad (2.31)$$

2.1.5.2 Iterative Methods and Embedded Subgraphs

If we partition the vector of all random variables as $\mathbf{x} = [\mathbf{x}_U^T, \mathbf{x}_O^T]^T$, the resulting joint PDF can be written as:

$$p(\mathbf{x}) \propto \exp \left\{ -\frac{1}{2} \begin{bmatrix} \mathbf{x}_U^T & \mathbf{x}_O^T \end{bmatrix} \begin{bmatrix} K_{UU} & K_{UO} \\ K_{OU} & K_{OO} \end{bmatrix} \begin{bmatrix} \mathbf{x}_U \\ \mathbf{x}_O \end{bmatrix} + \begin{bmatrix} \mathbf{h}_U^T & \mathbf{h}_O^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_U \\ \mathbf{x}_O \end{bmatrix} \right\}. \quad (2.32)$$

It follows that the posterior distribution of $\mathbf{x}_U$ conditioned on $\mathbf{x}_O$ is:

$$p(\mathbf{x}_U | \mathbf{x}_O) \propto \exp \left\{ -\frac{1}{2} \mathbf{x}_U^T K_{UU} \mathbf{x}_U + (\mathbf{h}_U - K_{UO} \mathbf{x}_O)^T \mathbf{x}_U \right\}. \quad (2.33)$$

Therefore, the MAP estimates as well as the mean of the posterior distribution can be computed as:

$$\hat{\mathbf{x}}_U = \underset{\mathbf{x}_U}{\operatorname{argmax}} p(\mathbf{x}_U | \mathbf{x}_O) = E[\mathbf{x}_U | \mathbf{x}_O] = K_{UU}^{-1} (\mathbf{h}_U - K_{UO} \mathbf{x}_O). \quad (2.34)$$

In other words, inference in GGMs is equivalent to solving a linear system. If K_{UU} does not correspond to a tree graphical model, directly solving the system is

computationally prohibitive due to the $\mathcal{O}(P^3)$ complexity. Instead, we can solve the linear system iteratively following the idea of matrix splitting [89, 90, 91]. These methods often decompose K_{UU} into two parts, that is, $K_{UU} = K_1 + K_2$. Furthermore, there is a fast algorithm to solve $K_1 \mathbf{x}_U = \mathbf{b}$ for $\mathbf{x}_U$ given $\mathbf{b} = \mathbf{h}_U - K_{UO} \mathbf{x}_O$. Then, starting from an initial guess $\hat{\mathbf{x}}_U^{(0)}$, the algorithm proceeds by updating the estimate of $\mathbf{x}_U$ as:

$$\hat{\mathbf{x}}_U^{(\kappa+1)} = K_1^{-1}(\mathbf{b} - K_2 \hat{\mathbf{x}}_U^{(\kappa)}). \quad (2.35)$$

Note that (2.35) corresponds to preconditioned Richardson iterations [92]. To guarantee that each iteration can be performed efficiently, K_1 can be chosen to correspond to a spanning tree [89], or a set of disconnected small subgraphs [90].

2.2 Univariate Extreme Value Theory

In this section, we briefly introduce univariate extreme value theory. The foundations of univariate extreme value theory are shaped by the two distributions: Generalized Extreme Value distributions and Generalized Pareto distributions. They are normally employed as marginal distributions in extreme events modeling. We will closely follow [41] when discussing the two distributions.

2.2.1 Generalized Extreme Value Distribution

Oftentimes, applications of extreme events modeling are centered on the distribution of the maximum of a set of data over a time period (i.e., block maxima). More explicitly, if $X^{(1)}, \dots, X^{(n)}$ is a sequence of n *i.i.d.* random variables with distribution G , the focus is on the distribution of the block maxima denoted as $M_n = \max(X^{(1)}, \dots, X^{(n)})$, which can be expressed explicitly as $P(M_n \leq x) = \{G(x)\}^n$. If we define $x_F = \sup\{x : G(x) < 1\}$, we can find that the distribution of the block maxima $P(M_n \leq x) = \{G(x)\}^n \rightarrow 0$ as $n \rightarrow \infty$ for all $x < x_F$. In other words,

$P(M_n)$ converges to a degenerate distribution. Instead, we consider the distribution of M_n after shift and scale transformations, since it can be non-degenerate in the same spirit as the Gaussian distribution, which is the limit of sums of random variables under a suitable renormalization. It turns out that the only possible non-degenerate limit laws of renormalized maxima are max-stable distributions; such distributions $F(x)$ must satisfy

$$F((x - \mu)/\sigma)^n = F(x), \quad (2.36)$$

for any $n \in \mathbb{N}$. The interpretation is that max-stability is satisfied by distributions for which the operation of taking sample maxima results in the same distribution, apart from the shift and scale transformations. It can be proven that the class of all possible non-degenerate limit laws for (properly renormalized) maxima of *i.i.d.* random variables coincides with the class of max-stable distributions [93]. Specifically, the class of max-stable distributions can only have three parametric functional forms as stated by the following theorem:

Theorem 2.2. (Extremal Types Theorem or Fisher-Tippett-Gnedenko Theorem) *Suppose that $\exists \{\sigma > 0\} \subset \mathbb{R}$ and $\{\mu\} \subset \mathbb{R}$ such that $P((M_n - \mu)/\sigma \leq x) \rightarrow F(x)$ when $n \rightarrow \infty$ with F being a nondegenerate distribution function. Then F belongs to either one of the families specified below:*

Type I (Gumbel):

$$F(x) = \exp \left\{ - \exp \left[- \frac{x - \mu}{\sigma} \right] \right\}, \quad x \in \mathbb{R}, \quad (2.37)$$

Type II (Fréchet):

$$F(x) = \begin{cases} 0, & x \leq \mu \\ \exp \left\{ - \left(\frac{x - \mu}{\sigma} \right)^{-\alpha} \right\}, & x > \mu \end{cases} \quad (2.38)$$

Figure 2.3: PDF of the Gumbel (blue), the Fréchet (red) and the Weibull (black) distribution.

Type III (Reversed Weibull):

$$F(x) = \begin{cases} \exp \left\{ - \left[- \left(\frac{x - \mu}{\sigma} \right)^\alpha \right] \right\}, & x < \mu \\ 1, & x \geq \mu \end{cases}, \quad (2.39)$$

where $\sigma > 0$, $\mu \in \mathbb{R}$, and $\alpha > 0$

The abovementioned three families can be unified into the so-called Generalized Extreme Value (GEV) distribution which has the following cumulative distribution function (CDF):

$$F(x) = \begin{cases} \exp \left\{ - \left[1 + \frac{\xi}{\sigma}(x - \mu) \right]^{-\frac{1}{\xi}} \right\}, & \xi \neq 0 \\ \exp \left\{ - \exp \left[-\frac{1}{\sigma}(x - \mu) \right] \right\}, & \xi = 0. \end{cases} \quad (2.40)$$

The support of the above distribution function is $\{x : 1 + \xi(x - \mu)/\sigma > 0\}$ if $\xi \neq 0$ and is $\mathbb{R}$ if $\xi = 0$. $\mu \in \mathbb{R}$ is the location parameter, $\sigma > 0$ is the scale parameter, and $\gamma \in \mathbb{R}$ is the shape parameter. The key parameter is the shape parameter,

determining the type of the limit distribution. Specifically, the Weibull, Gumbel, and Fréchet distributions correspond respectively to $\xi < 0$, $\xi = 0$, and $\xi > 0$. The three classes of distributions have distinct tail behaviors. Concretely, the upper-end point of the Weibull density function is finite, whereas that of the Gumbel and Fréchet distributions is infinite. Moreover, the density function decays exponentially for the Gumbel distribution and polynomially for the Fréchet distribution. In other words, the Weibull, Gumbel and Fréchet distributions have bounded, light and heavy tails respectively. Any of the three families mentioned above can be recovered from the GEV distribution.

Figure 2.3 illustrates the Gumbel, the Fréchet and the Weibull density. Note that the domain of the Gumbel, Fréchet, and Weibull distribution is infinite, lower bounded, and upper bounded respectively.

2.2.2 Generalized Pareto Distribution

An alternative approach is to consider extreme events above a certain threshold [41].

Theorem 2.3. (Pickands-Balkema-de Haan Theorem) *Let $X^{(1)}, \dots, X^{(n)}$ be i.i.d samples of distribution G , and let F_u be the conditional excess distribution function, that is,*

$$F_u(x) = P(X > x | X > u) = \frac{G(x) - G(u)}{1 - G(u)}, \quad x > u. \quad (2.41)$$

If the distribution of the block maxima $M_n = \max(X^{(1)}, \dots, X^{(n)})$ as $n \rightarrow \infty$ after shift and scale transformations is non-degenerate, then for $u \gg 1$, the conditional excess distribution F_u is well approximated by the generalized Pareto (GP) distribution:

$$F_u(x) \rightarrow F(x; u, \sigma, \gamma), \quad \text{as } u \rightarrow \infty, \quad (2.42)$$

where $F(x; u, \sigma, \gamma)$ is the GP distribution:

$$F(x) = \begin{cases} 1 - \left\{ 1 + \frac{\xi}{\sigma}(x - u) \right\}^{-\frac{1}{\xi}}, & \xi \neq 0 \\ 1 - \exp\left(-\frac{x - u}{\sigma}\right), & \xi = 0 \end{cases} \quad (2.43)$$

which is valid for $x \geq u$ and $1 + \xi/\sigma(x - u) > 0$. Here, u is denoted as the threshold while ξ and σ are the shape and scale parameters, respectively.

As an example, we may assume G to be a GEV distribution. By substituting the CDF of the GEV distribution into (2.41), we obtain the GP distribution. On the other hand, let $M_n = \max(X_1, \dots, X_n)$, where X_i are *i.i.d* GP distributed samples and n has a Poisson distribution. Then M_n essentially follows a GEV distribution [94]. Furthermore, the shape parameter of the GEV distribution is equal to that of the GP distribution. The above discussion implies a close connection between GP and GEV distributions.

A notable feature of the GP distribution is that if X follows the GP distribution for a fixed threshold u and a scale parameter σ , then the conditional distribution of $X - t$ given $X \geq t$, corresponding to a higher threshold $t > u$, is also GP distributed with the same shape parameter [94]. The scale parameter σ_t , on the other hand, is a linear function of u , i.e., $\sigma_t = \sigma + \xi(t - u)$. This property, called the “threshold stability” property, is often used to test the goodness-of-fit of the GP distribution.

Figure 2.4 illustrates the PDF of the GP distribution with different values of shape parameters.

In practice, instead of using GEV distributions, another popular method to describe block maxima is the peak (i.e., block maxima) over threshold approach [95]. It consists of three components:

1. Determination of the threshold value u .
2. Number of occurrences of exceedances over the threshold u over a given period of time which is assumed to be governed by a Poisson process.

Figure 2.4: PDFs of the Generalized Pareto Distributions with the shape parameter $\xi > 0$, $\xi = 0$ and $\xi < 0$.

3. The excess values (i.e., the amount by which the threshold is exceeded) following a GP distribution.

2.3 Copulas

In the following, we give a brief introduction to copulas. A copula $C(u_1, \dots, u_P)$ can be defined as a distribution function mapping from the unit P -cube $[0, 1]^P$ to the unit interval $[0, 1]$ (i.e., a distribution with uniform marginals), satisfying the following conditions [96]:

1. $C(1, \dots, 1, a_i, 1, \dots, 1) = a_i$ for every $i \leq P$ and all a_i in $[0, 1]$;
2. $C(a_1, \dots, a_P) = 0$ if $a_i = 0$ for any $i \leq P$;
3. C is P -increasing.

The term copula was first introduced by Sklar [97]. Thus, we next present the Sklar's Theorem:

Theorem 2.4. (Sklar’s Theorem) *Suppose that $x_1, \dots, x_P$ are random variables with marginal CDFs $F_1, \dots, F_P$ and joint CDF F , then there exists a copula C such that for all $\mathbf{x} = [x_1, \dots, x_P]^T$:*

$$F(x_1, \dots, x_P) = C(F_1(x_1), \dots, F_P(x_P)). \quad (2.44)$$

Furthermore, if the marginals $F_1, \dots, F_P$ are continuous, then the copula C can be uniquely determined. Conversely, given any marginals $F_1, \dots, F_P$ and an arbitrary copula C , F defined through (2.44) is a P -dimensional distribution function with marginals $F_1, \dots, F_P$.

Note that $u_i = F_i(x_i)$ follows a uniform distribution in $[0, 1]$. According to Sklar’s Theorem, the joint distribution can be specified via its marginals and a copula that “glues” the marginals together. As such, copulas allow us to construct joint distributions when only marginals are known with certainty. As explained in Section 2.2, the extreme value theory provides closed-form marginals for extreme events. However, such closed-form expressions are not available for multivariate extreme events. As a result, copulas can be powerful tools which aid in constructing the joint probability distribution for extreme events.

It is worthy noticing that copulas separate the marginal specification from dependence modeling. On the other hand, the intricate dependencies between numerous variables can be captured effectively and efficiently by graphical models (cf. Section 2.1). Therefore, as demonstrated in this thesis, copula-based graphical models can tackle data with various families of marginal distributions in a compact and flexible fashion.

Assuming that the partial derivatives of C exist, the PDF of the joint distribution (2.44) can be written as [96]:

$$f(x_1, \dots, x_p) = \frac{\partial^p F(x_1, \dots, x_p)}{\partial x_1 \cdots \partial x_p} \quad (2.45)$$

$$= c(F_1(x_1), \dots, F_p(x_p)) \prod_{i=1}^p f_i(x_i), \quad (2.46)$$

where c is the copula density function. In particular, for the bivariate case, we have:

$$c(F_i(x_i), F_j(x_j)) = \frac{f(x_i, x_j)}{f_i(x_i)f_j(x_j)}. \quad (2.47)$$

Also, from the fact that the marginals of a copula distribution are uniformly distributed we obtain:

$$\int_0^1 c(u_i, u_j) du_j = 1. \quad (2.48)$$

This important identity will be used to prove the propositions in Chapter 9.

Because copulas are primarily applied to analyze the dependence between random variables, several measures of dependence within the copula framework have been formulated, including the likes of Kendall's tau and Spearman's rho [15]. We are primarily concerned with tail dependence in this study as it is the most relevant for the analysis of extremes. Upper tail dependence coefficient, denoted by λ_U , quantifies the probability that one random variable takes on large values given that another random variable assumes large values. Mathematically, this can be expressed as [96]:

$$\lambda_U = \lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u), \quad (2.49)$$

where $U_1 = F_1(X_1)$ and $U_2 = F_2(X_2)$ are uniformly distributed. If we couple U_1 and U_2 by a copula C , Eq. (2.49) can be rewritten as follows:

$$\lambda_U = \lim_{u \rightarrow 1^-} \frac{1 - P(U_1 \leq u) - P(U_2 \leq u) + P(U_1 \leq u, U_2 \leq u)}{1 - P(U_2 \leq u)} \quad (2.50)$$

$$= \lim_{u \rightarrow 1^-} \frac{1 - 2u + C(u, u)}{1 - u}. \quad (2.51)$$

Lower tail dependence coefficient, on the other hand, is defined similarly [96]:

$$\lambda_L = \lim_{u \rightarrow 0^+} P(U_1 < u | U_2 < u) = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u}. \quad (2.52)$$

After providing an outline of copulas and some of their properties, we give a survey of specific examples of copulas considered in this thesis. The copulas listed below occur commonly in the extreme-value literature and other statistical fields, cf. [69] and [98]. They can be categorized into three major types: elliptical, Archimedean, and extreme-value copulas.

2.3.1 Elliptical Copulas

Elliptical copulas are aptly named because they embody the copulas of elliptical distributions. This type of copulas is derived from multivariate distributions. Provided that the inverse of the marginal functions F_i^{-1} exists, (2.44) can be written in the following form:

$$C(u_1, \dots, u_P) = F(F_1^{-1}(u_1), \dots, F_P^{-1}(u_P)). \quad (2.53)$$

In addition, the radial symmetry of elliptical distributions implies that upper and lower tail dependence of elliptical copulas are equal [99]. The two most common elliptical copulas are the Gaussian copula and the t copula.

2.3.1.1 Gaussian Copula

Gaussian copulas have the form:

$$C(u_1, \dots, u_P) = \Phi \left(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_P); \Sigma \right), \quad (2.54)$$

where $\Phi(\cdot, \Sigma)$ represents a zero-mean multivariate Gaussian distribution with covariance matrix Σ (with normalized diagonal), while Φ is the standard normal distribution. The Gaussian copula can also be constructed by introducing a vector of latent Gaussian variables $\mathbf{z} = [z_1, \dots, z_P]^T$ that are related to the observed variables

Figure 2.5: Densities of elliptical copulas. Note that the density of the t copula has been cut at a level of 15.

$\mathbf{x} = [x_1, \dots, x_P]^T$. The two-layer model can be formulated as:

$$\mathbf{z} \sim N(\mathbf{0}, \Sigma) \quad (2.55)$$

$$x_i = F_i^{-1}(\Phi(z_i)). \quad (2.56)$$

The correlation matrix Σ is the dependence parameter for Gaussian copulas; it can easily be determined once the latent variables $\mathbf{z}$ are inferred. The flexibility and analytical tractability of Gaussian copulas make them a handy tool in many applications. Their popularity is due to the fact that they describe dependence between variables in much the same way that Gaussian distributions do.

However, the Gaussian copula is less suitable for extreme-value analysis due to the asymptotic tail independence property. Particularly, for pairwise Gaussian copulas, the tail dependence coefficients $\lambda_U = \lambda_L = 0$ unless the correlation coefficient of the latent Gaussian variables equals 1.

2.3.1.2 t Copula

To remedy the limitations of the Gaussian copula, studies in the field of financial risk often resort to the t copula. The t copula has fatter tails than the Gaussian

copula, and consequently, it is capable of modeling tail dependence. The t copula is characterized by

$$C(u_1, \dots, u_P) = T_\nu(T_\nu^{-1}(u_1), \dots, T_\nu^{-1}(u_P); \Sigma), \quad (2.57)$$

with $T_\nu(\cdot; \Sigma)$ and T_ν being the joint and marginal t distributions, respectively, where ν is the parameter for the degrees of freedom and Σ is the correlation matrix [69]. As was mentioned regarding elliptical copulas, the t copula exhibit equivalent upper and lower tail dependence, cf. Fig. 2.5(b). We further show the bivariate Gaussian copula with the same correlation coefficient in Fig. 2.5(a) as a comparison. We can observe that the behavior of the two types of copulas in the center are quite similar, however, they are different at the four corners, especially in the upper right and lower left corners. The more pronounced peaks at $(0, 0)$ and $(1, 1)$ indicate that the t copula is able to model extreme cases where the values of both variables are very large or small. Therefore, the upper and lower tail dependence coefficients are larger than zero. In particular, for the bivariate t copula, if ρ is the correlation between the two variables, the upper and lower tail dependence coefficients can be computed as [100]:

$$\lambda_U = \lambda_L = 2t_{\nu+1} \left(-\frac{\sqrt{\nu+1}\sqrt{1-\rho}}{\sqrt{1+\rho}} \right). \quad (2.58)$$

We note however that the t copula is not an extreme-value copula, meaning that the t copula is not derived from block maxima of multivariate distributions, cf. Section 2.3.3. In [100], the authors further discuss other copulas which can be derived from the t copula including the t -EV copula, an extreme-value variant of the t copula.

2.3.2 Archimedean Copulas

The second type of copulas considered in this thesis is the Archimedean copula; it has a closed-form expression with typically only one parameter. The general form

of Archimedean copulas is:

$$C(u_1, \dots, u_P) = \psi^{-1}(\psi(u_1) + \dots + \psi(u_P)), \quad (2.59)$$

where $\psi(t)$ is called the generator of the copula that maps $[0, 1]$ into $[0, \infty]$.

Theorem 2.5. [15, Ch. 4] *Consider a continuous and strictly decreasing function $\psi : [0, 1] \rightarrow [0, \infty]$ with $\psi(1) = 0$. Then*

$$C(u_1, u_2) = \begin{cases} \psi^{-1}(\psi(u_1) + \psi(u_2)) & \text{if } \psi(u_1) + \psi(u_2) \leq \psi(0), \\ 0 & \text{otherwise} \end{cases} \quad (2.60)$$

is a copula, if and only if ψ is convex.

If $\psi(0) = \infty$, the generator is said to be strict and $C(u_1, u_2) = \psi^{-1}(\psi(u_1) + \psi(u_2))$. Note that Theorem 2.5 can be easily generated to multivariate cases [15]. Unlike the elliptical family, this family of copulas can describe asymmetric tail behaviors, and thus, it may be of benefit to various applications [99].

2.3.2.1 Clayton Copula

The generator for Clayton copulas is given by

$$\psi(u) = u^{-\theta} - 1, \quad (2.61)$$

where $\theta \in (0, \infty)$. As shown in Fig. 2.6(a), this type of copulas has different behavior at the left lower and right upper corners. Specifically, it shows an extremely uprising peak at $(0, 0)$ but a much less pronounced one at $(1, 1)$. Correspondingly, $\lambda_U = 0$ while $\lambda_L = 2^{-1/\theta}$, where θ is the parameter of the distribution [15]. As such, Clayton copulas can only accommodate lower tail dependence.

Figure 2.6: Densities of the Clayton (a) and the Frank (b) copula. Note that the density of the Clayton copula has been cut at a level of 15.

2.3.2.2 Frank Copula

The generator for Frank copulas is defined by:

$$\psi(u) = -\ln \left[\frac{\exp(-\theta u) - 1}{\exp(-\theta) - 1} \right], \quad \theta \in (-\infty, \infty). \quad (2.62)$$

Frank copulas are similar to Gaussian copulas. First, unlike other Archimedean copulas, Frank copulas can capture both positive and negative dependence, corresponding respectively to $\theta > 0$ and $\theta < 0$. Second, as shown in Fig. 2.6(b), dependence is symmetric in both tails. Lastly, they do not possess upper nor lower tail dependence. However, dependence in the tails of Frank copulas tends to be relatively weak compared to Gaussian copulas. This type of copulas are often utilized in financial risk applications [98].

2.3.2.3 Gumbel Copula

Finally, the last type of Archimedean copula considered in this thesis is the Gumbel copula. The generator of this copula can be expressed as:

$$\psi(u) = -\ln(u)^\theta, \quad \theta \in [1, \infty). \quad (2.63)$$

As shown in Fig. 2.7(a), in contrast to the Clayton copula, the Gumbel copula exhibits upper tail dependence but no lower tail dependence. Concretely, $\lambda_U = 2 - 2^{1/\theta}$ and $\lambda_L = 0$, where θ is the underlying parameter[15].

An advantage of the Gumbel copula is that it is also an extreme-value copula. Such type of copulas can be categorized to the class of Archimax copulas, which containing both the Archimedean and the extreme-value copulas as a special case [101], and are useful for extreme value analysis.

2.3.3 Extreme-Value Copulas

Let $\mathbf{X} = [X_1, \dots, X_P]^T$ be a P -dimensional random vector with marginals $G_1, \dots, G_P$, and let $\mathbf{X}^{(1:n)}$ be n *i.i.d* random vectors distributed as X . We further use a copula C_F to couple the marginals together to construct the joint distribution G of $\mathbf{X}$. Now let us focus on the vector of componentwise maxima:

$$\mathbf{M}_n = [M_{n1}, \dots, M_{nP}]^T, \quad (2.64)$$

where $M_{nj} = \max(X_j^{(1:n)})$. Since the joint distribution functions of $\mathbf{M}_n$ can be expressed as $G(\mathbf{X})^n$ while the marginal distribution of each component M_{nj} is given by $G_j(X_j)^n$, it follows from (2.53) that the copula for $\mathbf{M}_n$ is [102]:

$$C_n(u_1, \dots, u_P) = C_F \left(u_1^{\frac{1}{n}}, \dots, u_P^{\frac{1}{n}} \right)^n. \quad (2.65)$$

The limiting distributions of these copulas C_n as $n \rightarrow \infty$ define the family of extreme-value copulas. Specifically, a copula C is categorized as an extreme-value copula if there exists a copula C_F such that [102]:

$$C_F \left(u_1^{\frac{1}{n}}, \dots, u_P^{\frac{1}{n}} \right)^n \rightarrow C(u_1, \dots, u_P), \quad (2.66)$$

as $n \rightarrow \infty$, $\forall (u_1, \dots, u_P) \in [0, 1]^P$. C_F is said to be in the *domain of attraction* of C , while C is said to be *attracted by* C_F . This definition offers an effective approach

to construct an extreme-value copula C from an existing copula C_F .

On the other hand, it is straightforward to extend the results of univariate extreme value theory to the multivariate case. It turns out that the limiting joint distribution F of the renormalized componentwise maxima $(\mathbf{M}_n - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}$ must be max-stable, i.e., $F((\mathbf{M}_n - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma})^n = F(\mathbf{M}_n)$ [68], where $\oslash$ denotes componentwise division. This leads to the family of max-stable copulas that satisfy:

$$C(u_1, \dots, u_P) = C\left(u_1^{\frac{1}{n}}, \dots, u_P^{\frac{1}{n}}\right)^n, \quad \forall n \in \mathbb{N}. \quad (2.67)$$

As proven in [102, 103], there is a one-to-one correspondence between max-stable copulas and extreme-value copulas. From the above discuss, we can see that extreme-value copulas arise as a natural tool for capturing dependence between multiple extreme-value variables.

The general form of extreme-value copulas can be written as [102]:

$$C(u_1, \dots, u_P) = \exp \{-V(-\log u_1, \dots, \log u_P)\}, \quad (2.68)$$

where the exponent measure V is defined as [102]:

$$V(z_1, \dots, z_P) = \int_{S_P} \max(w_1 z_1, \dots, w_P z_P) dH(w_1, \dots, w_P), \quad (2.69)$$

for $(z_1, \dots, z_P) \in [0, \infty)^P$, and $dH(w_1, \dots, w_P)$ is a measure on the unit simplex S_P in $\mathbb{R}^P$, satisfying the mean constraints:

$$\int_{S_P} w_j dH(w_1, \dots, w_P) = \frac{1}{D}, \quad j = 1, \dots, P. \quad (2.70)$$

More often, extreme-value copulas are equivalently formulated as:

$$C(u_1, \dots, u_P) = \exp \left\{ \left(\sum_{i=1}^P \log u_i \right) A \left(\frac{\log u_1}{\sum_{i=1}^P \log u_i}, \dots, \frac{\log u_P}{\sum_{i=1}^P \log u_i} \right) \right\}, \quad (2.71)$$

Figure 2.7: Densities of extreme-value copulas. Note that all copulas been cut at a level of 15.

where the Pickands dependence function A is related to exponent measure V via

$$V(x_1, \dots, x_P) = \left(\sum_{i=1}^P x_i \right) A \left(\frac{x_1}{\sum_{i=1}^P x_i}, \dots, \frac{x_P}{\sum_{i=1}^P x_i} \right). \quad (2.72)$$

Fig. 2.7 shows the densities of several extreme-value copulas. We can observe that there exists a high peak at $(1, 1)$, implying that all extreme-value copulas have upper tail dependence. In particular, bivariate copulas under this family share a common expression of upper tail dependence coefficient, i.e.,

$$\lambda_u = \lim_{t \rightarrow 0^+} \frac{2t - 1 + C(1-t, 1-t)}{t} = 2 - V(1, 1) = 2 \left(1 - A \left(\frac{1}{2} \right) \right). \quad (2.73)$$

The above identity can be easily obtained by substituting (2.72) into the definition of upper tail dependence coefficient (2.51).

As can be observed from (2.71), the complexity of the expression for the max-stable copula stems from the Pickands dependence function which usually takes on a complicated form. As mentioned in Section 1.1.3, this poses a challenge when applying multivariate extreme-value copulas. Instead, pairwise copulas are used more frequently in practice. Consequently, we only present the bivariate Pickands dependence function of commonly used extreme-value copulas below based on [100] and [69].

As mentioned in the last section, the Gumbel copula belongs to the family of the extreme-value copulas. The Pickands dependence function of a bivariate Gumbel copula is given by:

$$A(w) = (w^\theta + (1 - w)^\theta)^{1/\theta}, \quad \theta \in [1, \infty). \quad (2.74)$$

We next briefly discuss the other copulas under this family that are considered in this thesis.

2.3.3.1 Galambos Copula

Let $C(u_1, u_2)$ be a bivariate Clayton copula. Its survival copula $\bar{C}$ can be defined as [96]:

$$\bar{C}(u_1, u_2) = u_1 + u_2 - 1 + C(1 - u_1, 1 - u_2). \quad (2.75)$$

Then, the Galambos copula can be constructed by specifying C_F in (2.66) as the survival Clayton copula. The resulting Pickands dependence function is represented by:

$$A(w) = 1 - (w^{-\theta} + (1 - w)^{-\theta})^{-1/\theta}, \quad (2.76)$$

where $\theta \in [0, \infty)$.

2.3.3.2 t -EV Copula

By definition, the t -EV copula is attracted by the t copula [100]. The Pickands dependence function is formulated as:

$$A(w) = wT_{\nu+1} \left[\frac{w/(1-w)^{1/\nu} - \rho}{(1-\rho^2)/(\nu+1)^{1/2}} \right] + (1-w)T_{\nu+1} \left[\frac{1-w/w^{1/\nu} - \rho}{(1-\rho^2)/(\nu+1)^{1/2}} \right], \quad (2.77)$$

where $\rho \in (-1, 1)$ is the correlation parameter, and ν is the degree of freedom. As can be observed, due to the complicated functional form, the t -EV copula might be difficult to implement in practice. Instead, we may apply Gumbel and Galambos copulas, which are proven to be good approximations of the t -EV copula in [100]. On the other hand, since the t -EV copula has two parameters, it is a more flexible model than other extreme-value copulas [69].

2.3.3.3 Hüsler-Reiss Copula

Finally, the Pickands dependence function of the Hüsler-Reiss Copula is as follows:

$$A(w) = (1-w)\Phi \left\{ \frac{\theta}{2} + \theta^{-1} \log \left(\frac{1-w}{w} \right) \right\} + w\Phi \left\{ \frac{\theta}{2} + \theta^{-1} \log \left(\frac{w}{1-w} \right) \right\}, \quad (2.78)$$

where $\theta \in (0, \infty)$. The Hüsler-Reiss copula can be derived in two ways. First, it is attracted by a bivariate Gaussian copula C_{ρ_n} whose correlation coefficient ρ_n increases with n such that $\rho_n \rightarrow 1$ as $n \rightarrow \infty$. More precisely, given that $(1 - \rho_n) \log n \rightarrow \lambda \in [0, \infty]$ as $n \rightarrow \infty$, one can show that $C_{\rho_n}(u_1^{1/n}, u_2^{1/n})^n$ converges to the Hüsler-Reiss copula [102]. The other approach rests on the fact that the t distribution converges to the normal distribution as the degrees of freedom $\nu \rightarrow \infty$. Consequently, we can obtain the Hüsler-Reiss copula from the expression of the t -EV copula. As demonstrated in [69], given $\theta > 0$ and setting $\rho = \exp(-\theta^2/2\nu) \sim 1 - \theta^2/2\nu$, the limit as $\nu \rightarrow \infty$ of the t -EV copula yields the Hüsler-Reiss copula.

Part I

Graphical Models for Nominal Data

Chapter 3

Copula Gaussian Graphical Models with Hidden Variables

As mentioned in the introduction, the first part of this thesis focuses on copula Gaussian graphical models for nominal data. The three chapters in this part deal with continuous data, discrete data, and piecewise-stationary data respectively.

In this chapter, we present a novel type of copula Gaussian graphical models for continuous data. In particular, we consider copula Gaussian graphical models with hidden variables. The resulting learning algorithm can infer a sparse graphical model given non-Gaussian data when some variables are unobserved. Specifically, in Section 3.1, we first introduce standard copula Gaussian graphical models (CGGM) and show how they characterize the conditional independence between variables. In order to cope with hidden variables, we propose hidden variable copula Gaussian graphical models (HVCGGM) in Section 3.2. We then explain how we learn the structure and parameters of the proposed model in Section 3.3. In Section 3.4, we assess the proposed model and benchmark it with other models, by means of synthetic and real cell signaling data. Finally, we summarize this chapter in Section 3.5.

3.1 Standard Copula Gaussian Graphical Models

We denote the observed non-Gaussian variables and latent Gaussian variables as $\mathbf{x} = [x_1, \dots, x_P]^T$ and $\mathbf{z} = [z_1, \dots, z_P]^T$ respectively. Following from the two-layer construction of Gaussian copulas (cf. Eq. (2.55) and (2.56)), a standard Copula Gaussian graphical model can be defined as [29]:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, K^{-1}) \quad (3.1)$$

$$x_i = F_i^{-1}(\Phi(z_i)), \quad (3.2)$$

where K is the precision matrix whose inverse (the covariance matrix) has normalized diagonal, Φ is the CDF of the standard Gaussian distribution, and F_i is the CDF of x_i . The latter is often approximated by the empirical distributions $\hat{F}_i$. The pseudo-inverse F_i^{-1} of F_i is defined as:

$$F_i^{-1}(y) = \inf_{x_i \in \mathcal{X}} \{F_i(x_i) \geq y\}, \quad (3.3)$$

where $\mathcal{X}$ is the domain of x_i such that $F_i(x_i) \leq 1$ for all $x_i \in \mathcal{X}$. Note that we use a precision matrix as the dependence parameter of the Gaussian copula instead of the correlation matrix in (2.55). Next, we prove that for a CGGM, the conditional independence between variables is specified by the zero elements in the precision matrix K [22].

Proposition 3.1. *Define $z_i = h_i(x_i) = \Phi^{-1}(\hat{F}_i(x_i))$ and let K be the precision matrix of the latent Gaussian variables $\mathbf{z}$. If the functions $h_i(x_i)$ are differentiable with respect to x_i , then $p(x_i, x_j | \mathbf{x}_{\mathcal{V} \setminus \{i,j\}}) = p(x_i | \mathbf{x}_{\mathcal{V} \setminus \{i,j\}})p(x_j | \mathbf{x}_{\mathcal{V} \setminus \{i,j\}})$ if and only if $K_{ij} = 0$.*

Proof. If the functions $h_i(x_i)$ are differentiable, it follows from Eq. (2.54) and (2.45) that the PDF of the CGGM is given by:

$$p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}h(\mathbf{x})^T K h(\mathbf{x})\right\} \prod_{i=1}^P h'_i(x_i). \quad (3.4)$$

The first factor corresponds to a GGM, and thus can be further decomposed as the product of node and edge potentials as defined in Eq. (2.11) and (2.12). Thus, the CGGM only modifies the node potentials of the corresponding GGM. By applying the Hammersley-Clifford Theorem (i.e., Theorem 2.1), we can find that the conditional dependence between variables is characterized by the precision matrix. $\square$

As a result, we aim to estimate a sparse precision matrix (so as to learn a sparse graphical model) from the observed non-Gaussian data $\mathbf{x}$. To move forward this objective, we first map the non-Gaussian variables $\mathbf{x}$ to the latent Gaussian layer:

$$z_i = \Phi^{-1}(\hat{F}_i(x_i)). \quad (3.5)$$

We then learn the sparse precision matrix based on the latent Gaussian variables $\mathbf{z}$, by solving the ℓ_1 -norm penalized optimization problem introduced in Section 2.1.4 [22]:

$$\hat{K} := \underset{K \succ 0}{\operatorname{argmin}} \operatorname{tr}(SK) - \log \det K + \lambda \|K\|_1, \quad (3.6)$$

where S is the empirical covariance of $\mathbf{z}$.

3.2 Hidden Variable Copula Gaussian Graphical Models

As mentioned in Section 1.1.1, standard graphical models are only applicable to the case where all the relevant variables are observed. In other words, unobserved or missing variables are not allowed. However, in practice, it is very difficult to observe all the relevant variables (cf. the example of gene networks in Section. 1.1.1). We therefore propose a novel hidden variable copula Gaussian graphical model (HVCGGM). Before presenting the proposed model, we first introduce hidden variable Gaussian graphical models.

Figure 3.1: A graphical model with hidden variables (yellow nodes): (a) the joint graphical model; (b) the marginal graphical model of observed variables (green nodes); (c) the conditional graphical model of observed variables.

3.2.1 Hidden Variable Gaussian Graphical Model

Here, we consider the simplest case of hidden variable graphical models, where both the observed variables and hidden variables are Gaussian. Suppose we have observable variables $\mathbf{z}_O$ (green nodes in Fig. 3.1(a)) and hidden variables $\mathbf{z}_H$ (yellow nodes in Fig. 3.1(a)) which are jointly Gaussian distributed. The joint precision matrix of $\mathbf{z}_O \cup \mathbf{z}_H$, $K_{O \cup H}$, which characterizes the graphical model in Fig. 3.1(a), can be expressed as:

$$K_{O \cup H} = \begin{bmatrix} K_{OO} & K_{OH} \\ K_{HO} & K_{HH} \end{bmatrix}. \quad (3.7)$$

Then according to the Schur complement [104], the marginalized precision matrix $\tilde{K}_{OO}$ of $\mathbf{z}_O$ can be written as:

$$\tilde{K}_{OO} = K_{OO} - K_{OH} K_{HH}^{-1} K_{HO} = K_{OO} - L, \quad (3.8)$$

where $L = K_{OH} K_{HH}^{-1} K_{HO}$. Note that given the joint covariance matrix $\Sigma_{O \cup H}$, the marginal precision matrix of observed variables $\tilde{K}_{OO} = ([\Sigma_{O \cup H}]_{OO})^{-1}$.

The two components of $\tilde{K}_{OO}$ have their own properties [19]. K_{OO} is the conditional precision matrix of $\mathbf{z}_O$, conditioned on $\mathbf{z}_H$ (see Fig. 3.1(c)). It is supposed to be sparse as it describes the interactions only between observed variables. L summarizes the effect of marginalization over the hidden variables. The rank of L is equal to

the number of hidden variables, and it is assumed to be low since the number of hidden variables is supposed to be small. Since $\mathbf{z}_H$ are connected to many of $\mathbf{z}_O$, K_{OH} and K_{HO} are not sparse, thus making the product matrix dense. Resulting from the subtraction, $\tilde{K}_o$ is also dense, as shown in Fig. 3.1(b). CGGMs cannot yield a sparse graph in this case because they can only estimate $\tilde{K}_o$.

Given *i.i.d.* samples of $\mathbf{z}_O$, our objective is to estimate K_{OO} and L ; we are especially interested in the rank of L since it equals the number of hidden variables. Those matrices may be recovered by solving the convex relaxation [19]:

$$(\hat{K}_{OO}, \hat{L}) = \underset{K_{OO} \succ 0, L \succeq 0}{\operatorname{argmin}} \operatorname{tr}((K_{OO} - L)S_{OO}) - \log \det(K_{OO} - L) + \lambda(\gamma \|K_{OO}\|_1 + \operatorname{tr}(L)). \quad (3.9)$$

where $\hat{K}_{OO}$ and $\hat{L}$ are estimates of K_{OO} and L respectively, and S_{OO} is the empirical marginal covariance of $\mathbf{z}_O$.

Note that similar to (3.6), $\operatorname{tr}((K_{OO} - L)S_{OO}) - \log \det(K_{OO} - L)$ is the divergence between the observed data and the estimated model. The two regularization parameters λ and γ can be interpreted as follows. The product of λ and γ is the regularization parameter of the ℓ_1 norm; it controls the trade-off between the sparsity of K_{OO} and the fidelity of the data. On the other hand, λ alone is the regularization parameter of the nuclear norm (which reduces to the trace norm for symmetric, positive-semidefinite matrices), and therefore it penalizes the rank of L . In addition, γ alone balances the trade-off between the rank of L and the sparsity.

The convex problem (3.9) can be solved efficiently by the Newton-CG primal proximal point algorithm [105]. To recover the correct K_{OO} and L , the parameters λ and γ need to be chosen properly, and this will be discussed in Section 3.3.

3.2.2 Copula Gaussian Graphical Models with Hidden Variables

We now present the proposed hidden variable copula Gaussian graphical model. The observed (continuous) variables $\mathbf{x}$ (blue nodes in Fig. 3.2) are non-Gaussian, and each of them is associated with a Gaussian distributed latent variable z_{O_i} (green nodes in Fig. 3.2), as in the standard copula Gaussian graphical model. However, besides the latent variables $\mathbf{z}_O$, there exist several hidden variables $\mathbf{z}_H$ (yellow nodes in Fig. 3.2) that are not associated with observed variables. In the corresponding graphical model, the nodes $\mathbf{z}_H$ are only connected to latent variables $\mathbf{z}_O$; they are not connected to observed variables $\mathbf{x}$.

Figure 3.2: A copula Gaussian graphical model with hidden variables: blue nodes denote observed non-Gaussian variables, green nodes denote latent Gaussian variables and yellow nodes denote hidden variables in the latent layer.

In other words, the variables $\mathbf{x}$ and $\mathbf{z}_O$ constitute a Gaussian copula, while the variables $\mathbf{z}_O$ and $\mathbf{z}_H$ form a hidden variable Gaussian graphical model. Together, the variables $(\mathbf{x}, \mathbf{z}_O, \mathbf{z}_H)$ form a hidden variable copula Gaussian graphical model with associated conditional precision matrix K_{OO} and product matrix L (cf. (3.8)).

Given *i.i.d.* samples of the non-Gaussian variables $\mathbf{x}$, we wish to infer the conditional precision matrix K_{OO} of $\mathbf{z}_O$ conditioned on $\mathbf{z}_H$ (corresponding to the red edges in Fig. 3.2), and the product matrix L , associated with the HVCGGM.

As a first step, we transform the non-Gaussian observed variables $\mathbf{x}$ into Gaussian

distributed hidden variables $\mathbf{z}_O$ associated with the observed variables $\mathbf{x}$:

$$z_{O_i} = \Phi^{-1}(\hat{F}_i(x_i)), \quad (3.10)$$

where Φ is the CDF of the standard Gaussian distribution and $\hat{F}_i$ is the empirical CDF of x_i . As a result, we are dealing with Gaussian variables $\mathbf{z}_O$ which together with $\mathbf{z}_H$ constitute a HVGGM.

In the second step, we follow the procedure of (3.9) to infer the sparse conditional precision matrix K_{OO} of $\mathbf{z}_O$ and the low-rank product matrix L :

$$(\hat{K}_{OO}, \hat{L}) = \underset{K_{OO} \succ 0, L \succeq 0}{\operatorname{argmin}} \operatorname{tr}((K_{OO} - L)S_{OO}) - \log \det(K_{OO} - L) + \lambda(\gamma \|K_{OO}\|_1 + \operatorname{tr}(L)). \quad (3.11)$$

The theoretical computational complexity of the proposed method is $\mathcal{O}(P^3)$, where P is the number of variables. The computational bottleneck stems from solving the above convex problem. In practice, the algorithm proposed in [105] can efficiently deal with problems of dimension up to several thousands. Note that the selection of regularization parameters λ and γ in Eq. (3.11) needs special attention, as will be discussed in Section 3.3

3.3 Learning and Inference

A suitable choice of regularization parameters λ and γ in (3.11) can recover the underlying true graphical model. Standard approaches for regularization selection, including cross validation (cv), the Akaike information criterion (AIC), and the Bayesian information criterion (BIC), are known to overfit the data, and they typically result in graphs that are too dense [106]. As an alternative, we circumvent the delicate issue of regularization selection by first learning the graph structure and next inferring the parameters. Specifically, stability selection [107] is used to learn the structure (sparsity pattern) of the conditional precision matrix K_{OO} . We

further specify K_{OO} and L by solving a convex problem subject to the structure constraints. The estimated number of hidden variables equals the rank of L .

3.3.1 Structure Learning: Stability Selection

The objective of stability selection is to estimate a stable graph from the data. It generates multiple sample sets by resampling the original data, and then determines the graphical model that appears consistently over those sample sets.

Suppose we have a P -dimensional dataset S with sample size N . We denote the undirected graphical model associated with K_{OO} as $\mathcal{G}_{OO} = (\mathcal{V}_{OO}, \mathcal{E}_{OO})$.

First, we randomly draw M sample sets $S_1, S_2, \dots, S_M$ without replacement from the dataset S , each of size $\lfloor N/2 \rfloor$. In our case, we choose $M = 100$.

Second, we select a range of λ and γ (cf. (3.11)). Now let us concentrate on one pair of parameters (λ, γ) in that range. For each sample set S_m (for $m = 1, \dots, M$), we estimate one conditional precision matrix K_{OO} using (3.11), resulting in M precision matrices $K_1, \dots, K_M$. For each element (i, j) in the matrix K_m , the number of times it is nonzero ($[K_m]_{ij} \neq 0$) among the M matrices is counted and divided by M ; as a result, we obtain the probability (referred to as “stability”) that this edge exists in the graphical model associated with (λ, γ) . By varying λ and γ through the chosen range, we can draw a surface of the stability for each edge.

At last, we include edge (i, j) in the edge set $\mathcal{E}_{OO}$, if the probability of that edge, for at least one pair (λ, γ) in the selected range, is larger than a threshold π_{thr} . It has been proven in [107] that given the expected number of falsely selected edges E , the threshold π_{thr} is upper bounded by:

$$\pi_{\text{thr}} \leq \frac{\bar{p}^2}{P(P-1)E} + 0.5. \quad (3.12)$$

The parameter $\bar{p}$ is the *average* number of edges in the graphs associated with each pair (λ, γ) in the selected range, inferred from the entire data set S through (3.11). In practice, the threshold π_{thr} is often set to be the upper bound (3.12).

We also applied a randomized approach suggested in [107]. When inferring the matrices K_m , we replace the parameter pair (λ, γ) by a random pair $(\lambda/\alpha_1, \gamma/\alpha_2)$, where α_1 and α_2 are different for each sample set and are uniformly distributed on $[0.2, 1]$. In the following, we will only report results for the randomized approach, since it yields the best results.

3.3.2 Parameter Learning

With the learned sparsity pattern of K_{OO} , we proceed to estimate K_{OO} and L . Specifically, we solve a problem similar to (3.11), subject to the learned graphical model structure $\mathcal{E}_{OO}$:

$$\begin{aligned} (\hat{K}_{OO}, \hat{L}) = \underset{K_{OO} \succ 0, L \succ 0}{\operatorname{argmin}} \quad & \log \det(K_{OO} - L) - \operatorname{tr}((K_{OO} - L)S_{OO}) + \lambda \operatorname{tr}(L) \quad (3.13) \\ \text{s.t.} \quad & [K_{OO}]_{ij} = 0 \quad \forall (i, j) \notin \mathcal{E}_{OO}, \end{aligned}$$

The parameter λ is selected as the mean of the values of λ in the chosen range in the second step of stability selection. The number of hidden variables can be estimated easily as the rank of L . The convex problem (3.13) can be solved using the solver LogdetPPA [105].

3.4 Numerical Results

We test on a synthetic and two real datasets our proposed graphical model and existing graphical models, including Gaussian graphical models, copula Gaussian graphical models, and hidden variable Gaussian graphical models. We also compare the performance of stability selection and cross validation for learning the graphical model structure.

3.4.1 Synthetic Data

We generate non-Gaussian synthetic data as follows:

1. Generate a random precision matrix by the method of [22]: we first uniformly sample $r_1, \dots, r_n$ from a unit square. The precision matrix is initialized as a unit matrix. Next, we set the elements K_{ij} and K_{ji} of precision matrix equal to a constant ρ with probability $(\sqrt{2\pi})^{-1} \exp(-4\|r_i - r_j\|^2)$, and equal to zero otherwise. We follow [22] to set $\rho = 0.245$ in order to guarantee the positive definiteness of the precision matrix.
2. Add a few variables and connect each of them to at least 80% of other variables. The corresponding elements in precision matrix are nonzero.
3. Generate samples from the Gaussian distribution with the above precision matrix and discard all the samples of variables added in Step 2 (hidden variables).
4. Transform the Gaussian variables to continuous non-Gaussian variables via Eq. (3.10) by means of different marginals, e.g., beta, exponential, and chi-square marginal distributions.

We apply the following graphical models to the synthetic data:

- the Gaussian graphical model (GGM), cf. Section 2.1.4;
- the copula Gaussian graphical model (CGGM), cf. Section 3.1;
- the hidden variable Gaussian graphical model inferred by cross validation (HVGGM(cv));
- the hidden variable Gaussian graphical model inferred by stability selection (HVGGM(ss)), cf. Section 3.2.1;
- the hidden variable copula Gaussian graphical model inferred by cross validation (HVCGGM(cv));
- the proposed hidden variable copula Gaussian graphical model learned by stability selection (HVCGGM(ss)), cf. Section 3.2.2;

Table 3.1: Quantitative comparison of different methods

Methods	20-dimensional dataset				60-dimensional dataset			
	Precision	Recall	F_1 -score	Prm No.	Precision	Recall	F_1 -score	Prm No.
GGM	0.1195	0.9048	0.2111	179	0.0238	0.9655	0.0465	1723
CGGM	0.1105	1.0000	0.1990	210	0.0194	1.0000	0.0381	1761
HVGGM(cv)	0.1324	0.4286	0.2023	488	0.0336	0.1379	0.0540	2130
HVGGM(ss)	0.0000	0.0000	0.0000	0	0.0000	0.0000	0.0000	0
HVCGGM(cv)	0.2877	1.0000	0.4468	173	0.1677	1.0000	0.2858	576
HVCGGM(ss)	0.8750	1.0000	0.9333	64	0.9667	1.0000	0.9831	95
GGM(m)	1.0000	0.6737	0.8050	179	0.9466	0.7737	0.8515	1723
CGGM(m)	1.0000	1.0000	1.0000	210	0.9792	0.9739	0.9765	1761

We evaluate those methods through various criteria including precision, recall, F_1 score and number of parameters (Prm No.). Their definitions are as follows:

1. Precision is defined as the proportion of correctly estimated edges to all the edges in the estimated graph;
2. Recall is defined as the proportion of correctly estimated edges to all the edges in the true graph;
3. F_1 -score is defined as $2 \cdot \text{precision} \cdot \text{recall} / (\text{precision} + \text{recall})$, which is a weighted average of the precision and recall.

The results for a 20-dimensional dataset and a 60-dimensional dataset are summarized in Table. 3.1, Fig. 3.3, and Fig. 3.4.

The results show that HVCGGM(ss) achieves the best performance with the least number of parameters. On the other hand, the HVCGGM(cv) generates a dense graph, whereas the HVGGM(ss) produces an empty graph without edges.

The rank of the inferred L in the HVCGGM(ss) equals the number of hidden variables. However, the rank estimated by the HVCGGM(cv) is about four times larger than the true value, while the HVGGM(cv) yields a full-rank matrix L .

The GGM and CGGM infer the marginal precision matrix $\tilde{K}_{OO}$ (3.8) instead of the conditional precision matrix K_{OO} ; we also list the results for that inference problem

Figure 3.3: Results of different methods on the 20-dimensional synthetic dataset.

Figure 3.4: Results of different methods on the 60-dimensional synthetic dataset.

under GGM(m) and CGGM(m) in Table 3.1, where “m” refers to marginal. The performance of CGGM is the best, since the latter is designed for Gaussian data. However, the marginal precision matrix $\tilde{K}_{OO}$ is much denser than the conditional precision matrix K_{OO} , and therefore, it is a more complicated model with more parameters.

Figure 3.5: Histogram of the cell signaling data

3.4.2 Cell Signaling Data

We now apply the aforementioned methods to a real dataset. This dataset is composed of 11 variables, corresponding to 11 proteins. The amounts of the 11 proteins contained in 7466 different cells are measured [108]. Therefore, the sample size is 7466. Two proteins (PKA and PCA) seem to interact with most of the 9 other proteins, and they can be considered as “hubs” (see Fig. 3.6(a), indicated in yellow). We remove those proteins from the dataset. The graphical models should infer those two proteins as hidden variables. The histogram of three cell signals are shown in Fig. 3.5. As can be seen from this figure, the protein amounts clearly do not follow Gaussian distribution, and therefore, one would expect the copula Gaussian model with hidden variables to perform well on this dataset.

The results are shown in Fig. 3.6. The connectivity of the 11 proteins has been inferred by experiments in [108]; we consider this graph as the ground truth. The HVCGGM(cv) approach overfits the data. Both the GGM and the CGGM are trying to estimate the marginalized dense graph. The HVGGM(cv) generates an incorrect full graph, and the HVGGM(ss) leads to a fully disconnected graph. In contrast, the proposed method yields the most accurate conditional graph. Moreover, the rank of the inferred matrix L equals 2, which is indeed the true number of hidden variables (PKA and PCA).

Figure 3.6: Results of different methods on the cell signaling data.

3.4.3 Electroencephalogram (EEG) Data

In this section, we aim to infer functional brain networks from brain signal recordings. Inferring the interactions between different brain areas is an important step towards understanding brain activity. Most often, signals can only be measured from some specific brain areas (e.g., cortex in the case of scalp electroencephalograms). However, those signals may be affected by brain areas from which no measurements are available (e.g., deeper areas such as hippocampus). Therefore, introducing hidden variables in this case often leads to simpler models (networks) than standard graphical models without hidden variables.

Our EEG data set consists of 24 healthy control subjects (age: 69.4 $\pm$ 11.5 years old; 10 males) and 17 patients with mild AD (age: 77.6 $\pm$ 10.0 years old; 9 males) [109]. The patient group underwent full battery of cognitive tests (Mini Mental State Examination, Rey Auditory Verbal Learning Test, Benton Visual Retention Test, and memory recall tests). The EEG time series were recorded using 19 electrodes positioned according to Maudsley system, similar to the 10-20 international system,

Figure 3.7: Conditional precision matrix K_0 (HVCGGM) of first three AD patients and control subjects.

at a sampling frequency of 128 Hz. All recording sessions were conducted with the subjects in an awake but resting state with eyes closed, and the length of the EEG recording was about 5 minutes, for each subject. Only those subjects were retained in the analysis whose EEG recordings contained at least 20s of artifact-free data. We only consider EEG signals between 4 and 30Hz to filter the noise.

It is well known that the EEG of AD patients is generally less synchronous (less correlated) [110]. Therefore, we would expect the (functional) brain networks to be sparser in AD patients. Indeed, we found that the conditional precision matrix K_{OO} in HVCGGM is sparser for AD EEG than healthy EEG ($p = 0.0121$ for Mann-Whitney test; see Fig. 3.7). Based on sparsity only as input feature, we can classify the two populations with a success rate of about 80% (linear discriminant analysis with leave-one-out cross validation). In other words, conditioned on the hidden factors, the AD EEG signals are substantially less mutually dependent than healthy EEG signals. The correlations between the AD EEG signals are to a larger extent due to hidden factors.

Figure 3.8: Boxplots for the sparsity measure of HVCGGM and HVGGM (The corresponding p -value are 0.0121, 0.9048, 0.9894 and 0.1561)

Interestingly, the number of hidden variables is not significantly different in healthy subjects and AD patients (typically between 3 and 6). No significant differences were observed for HVGGM, which seems to suggest that modeling the non-Gaussian nature of the data with a copula is crucial. Likewise, no significant differences (see Fig. 3.8) between the AD patients and healthy subjects were found for graphical models without hidden variables, i.e., monoscale Gaussian graphical models (GGM) and copula monoscale Gaussian graphical models (CGGM). Those models are also much denser (and hence more complicated) than the models with hidden variables, as can be seen from Fig. 3.8.

3.5 Summary

In this chapter, we introduced the copula Gaussian graphical model with hidden variables; such model can provide a simple description of high-dimensional non-Gaussian data, where correlations can be captured through a few hidden variables. We used stability selection to learn the structure of the model and inferred the parameters of model and the number of hidden variables by solving a convex problem. Both synthetic and real data on cell signaling show that the model can handle non-Gaussian data with unobserved hidden variables.

Chapter 4

Copula Gaussian Graphical Models for Discrete Data

In this chapter, we consider learning copula Gaussian graphical models from discrete data. So far, structure learning algorithm of discrete graphical models can only deal with binary data [23, 24], and may be extended to the case where all the variables take on values in the same alphabet [25, 26]. To remedy the issue, we explore discrete copula Gaussian graphical models (DCGGM), which are applicable to an arbitrary combination of heterogeneous discrete variables (i.e., variables with different alphabets). Previous works related to DCGGMs [27, 28, 29] often employ full MCMC approaches, sampling both the Gaussian latent variables and the precision matrix for the corresponding posterior distributions. However, such models can be quite computationally intensive, and thus are prohibitive for large-scale problems. In this chapter, we propose an algorithm to directly infer point estimates of the precision matrix; we apply Monte Carlo expectation maximization (MCEM) [111]. In the E-step, we draw samples for the latent Gaussian variables by efficient Gibbs sampling [112]. In the M-step, we learn the sparse Gaussian graphical model through the glasso method [18]. The proposed method is reliable yet much more efficient than the full MCMC approach, as it avoids the costly Monte Carlo simulations for the posterior distribution of the precision matrix.

We begin this chapter by introducing the DCGGM in Section 4.1. We further outline the proposed learning algorithm for this model in Section 4.2. In Section 4.3, we deal with the problem of regularization selection, and in Section 4.4, we present numerical results on synthetic and real datasets. We offer concluding remarks in Section 4.5.

4.1 Discrete Copula Gaussian Graphical Models

Let us now revisit the definition of CGGMs. For non-Gaussian observed variables $\mathbf{x}$ and Gaussian latent variables $\mathbf{z}$, the CGGM is defined as:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, K^{-1}) \quad (4.1)$$

$$x_i = F_i^{-1}(\Phi(z_i)), \quad (4.2)$$

where K is the precision matrix whose inverse (the covariance matrix) has normalized diagonal, Φ is the CDF of the standard Gaussian distribution, F_i is the CDF of Y_i , and F_i^{-1} is the pseudo-inverse of F_i , that is,

$$F_i^{-1}(y) = \inf_{x_i \in \mathcal{X}} \{F_i(x_i) \geq y\}. \quad (4.3)$$

According to the above definition, F_i^{-1} is a one-to-one mapping for a continuous variable. However, it is a one-to-many mapping if x_i is discrete [28]. Therefore, unlike the case in the previous chapter, z_i cannot be determined uniquely from the observed discrete x_i .

The overall statistical model can be written as:

$$p(\mathbf{x}, \mathbf{z}, K; F) = \prod_{i=1}^P p(x_i | z_i; F_i) \mathcal{N}(\mathbf{z}; \mathbf{0}, K^{-1}) p(K). \quad (4.4)$$

The cumulative distributions F_i are unknown, but they can be approximated by the

empirical distributions $\hat{F}_i$. Model (4.4) then simplifies as:

$$p(\mathbf{x}, \mathbf{z}, K; \hat{F}) = \prod_{i=1}^P p(x_i | z_i; \hat{F}_i) \mathcal{N}(\mathbf{z}; \mathbf{0}, K^{-1}) p(K). \quad (4.5)$$

4.2 Monte-Carlo Expectation Maximization Algorithm

In the section, we develop the MCEM algorithm to learn the discrete copula Gaussian graphical model. In order to learn the graphical model, our objective is to infer K from N independent observations $\mathbf{x}^{(1:N)} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}]$. We denote the associated latent variables as $\mathbf{z}^{(1:N)} = [\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(N)}]$. A reasonable approach to infer K is maximum a posteriori (MAP) estimation:

$$\hat{K} = \operatorname{argmax}_{K \succ 0} p(K | \mathbf{x}^{(1:N)}), \quad (4.6)$$

where the marginal $p(K | \mathbf{x}^{(1:L)})$ is given by:

$$\begin{aligned} p(K | \mathbf{x}^{(1:N)}) &\propto \int_{\mathbf{z}} p(\mathbf{x}^{(1:L)}, \mathbf{z}^{(1:L)}, K; \hat{F}) d\mathbf{z}^{(1:L)} \\ &= p(K) \prod_{n=1}^N \int_{\mathbf{z}^{(n)}} \left[\prod_{i=1}^P p(x_i^{(n)} | z_i^{(n)}; \hat{F}_i) \mathcal{N}(\mathbf{z}^{(n)}; \mathbf{0}, K^{-1}) \right] d\mathbf{z}^{(n)}. \end{aligned} \quad (4.7)$$

In contrast, in a full Bayesian approach one would infer the posterior distribution of K numerically [27, 28, 29]. Due to the large number of entries in K when the dimension is high, a great quantity of samples has to be drawn in order to well approximate the posterior distribution. The resulting MCMC algorithm is therefore computationally expensive.

Let us return to the main thread. If the observed variables $\mathbf{x}$ are continuous, the conditional distributions $p(x_i | z_i; \hat{F}_i)$ are Dirac deltas, and the integrals in the right hand side (RHS) of (4.7) are trivial. If $\mathbf{x}$ are discrete, those integrals are intractable,

and instead we solve (4.6) by expectation maximization [111]:

$$\hat{K}^{(\kappa+1)} = \operatorname{argmax}_K \log p(K) + Q(K; \hat{K}^{(\kappa)}), \quad (4.8)$$

where the Q -function can be computed as:

$$Q(K; \hat{K}^{(\kappa)}) = \sum_{n=1}^N \int_{\mathbf{z}^{(n)}} \left[p(\mathbf{z}^{(n)} | \mathbf{y}^{(n)}; \hat{K}^{(\kappa)}) \log \mathcal{N}(\mathbf{z}^{(n)}; \mathbf{0}, K^{-1}) \right] d\mathbf{z}^{(n)}, \quad (4.9)$$

and κ is the iteration number.

Since the integrals in the RHS of (4.9) are intractable, we calculate them numerically by Monte Carlo integration:

$$\int_{\mathbf{z}^{(n)}} p(\mathbf{z}^{(n)} | \mathbf{x}^{(n)}; \hat{K}^{(\kappa)}) \log \mathcal{N}(\mathbf{z}^{(n)}; \mathbf{0}, K^{-1}) d\mathbf{z}^{(n)} \approx \frac{1}{M} \sum_{m=1}^M \log \mathcal{N}((\hat{\mathbf{z}}^{(n,m)})^{(\kappa)}; \mathbf{0}, K^{-1}), \quad (4.10)$$

where the M samples $(\hat{\mathbf{z}}^{(n,m)})^{(\kappa)}$ with $m = 1, \dots, M$, are drawn from $p(\mathbf{z}^{(n)} | \mathbf{x}^{(n)}; \hat{K}^{(\kappa)})$. The latter is a truncated Gaussian in the case of discrete $\mathbf{x}$. Efficient block Gibbs sampling methods have been developed for truncated Gaussian distributions [112]. When some components of $\mathbf{x}^{(n)}$ are missing, we can still handle them by drawing Gibbs samples from the entire domain of Gaussian distribution: $(-\infty, +\infty)$. Substitute (4.13) into (4.9), we obtain the following approximation of $Q(K; \hat{K}^{(\kappa)})$:

$$\begin{aligned} \hat{Q}(K; \hat{K}^{(\kappa)}) &\approx \frac{1}{M} \sum_{n=1}^N \sum_{m=1}^M \log \mathcal{N}((\hat{\mathbf{z}}^{(n,m)})^{(\kappa)}; \mathbf{0}, K^{-1}) \\ &= \frac{N}{2} (\log \det K - \operatorname{tr}(S^{(\kappa)} K)) + C, \end{aligned} \quad (4.11)$$

where C summarizes all irrelevant constants, and $S^{(\kappa)}$ is the empirical covariance matrix:

$$S^{(\kappa)} = \frac{1}{N} \sum_{n=1}^N (S^{(n)})^{(\kappa)} = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M (\hat{\mathbf{z}}^{(n,m)})^{(\kappa)} ((\hat{\mathbf{z}}^{(n,m)})^{(\kappa)})^T, \quad (4.12)$$

where $(S^{(n)})^{(\kappa)}$ may be viewed as the empirical covariance matrix computed from

the n -th observation $\mathbf{x}^{(n)}$.

A common choice for the prior $p(K)$ is the Laplace distribution:

$$p(K) = \frac{\tilde{\lambda}^P}{2} \exp \left(-\tilde{\lambda} \|K\|_1 \right), \quad (4.13)$$

since it favors a sparse precision matrix K and hence a sparse latent Gaussian graphical model. With that choice of prior, the update rule (4.8) boils down to:

$$\hat{K}^{(\kappa+1)} = \operatorname{argmax}_{K \succ 0} \log \det K - \operatorname{tr}(S^{(\kappa)} K) - \lambda \|K\|_1, \quad (4.14)$$

where $\lambda = 2\tilde{\lambda}/N$. The resulting Monte-Carlo EM algorithm is summarized in Table 4.1.

1. Initialize $\hat{K}^{(0)}$.
2. Iterate the following steps till convergence:
 - (a) Draw M samples $(\hat{\mathbf{z}}^{(n,m)})^{(\kappa)}$ with $m = 1, \dots, M$ from $p(\mathbf{z}^{(n)} | \mathbf{x}^{(n)}; \hat{K}^{(\kappa)})$.
 - (b) Compute the empirical covariance matrix $S^{(\kappa)}$:

$$S^{(\kappa)} = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M (\hat{\mathbf{z}}^{(n,m)})^{(\kappa)} ((\hat{\mathbf{z}}^{(n,m)})^{(\kappa)})^T.$$

- (c) Update the estimate of K :

$$\hat{K}^{(\kappa+1)} = \operatorname{argmax}_{K \succ 0} [\log \det K - \operatorname{tr}(S^{(\kappa)} K) - \lambda \|K\|_1].$$

Table 4.1: Monte-Carlo EM algorithm for learning the discrete copula Gaussian graphical model.

Convergence of the EM algorithm for penalized likelihood problems (4.14) has been proven in [113]. In our numerical experiments, the algorithm usually converges in less than 10 iterations. Note that the regularization parameter λ needs to be chosen appropriately to recover the correct precision matrix $\hat{K}^{(\kappa+1)}$ in each EM iteration, which is a critical issue to be addressed in the following section.

4.3 Regularization Selection

A suitable choice of regularization parameter λ in (4.14) can produce the graphical model with true sparsity pattern of K . However, standard procedures for selecting λ are known to overfit the data and result in graphs that are too dense [106]. As an alternative, we employ a two-step procedure of structure learning and parameter learning. We first apply the BINCO method [114] to structure learning, resulting in the sparsity pattern of $K^{(\kappa+1)}$. Next, we adopt IPF (cf. Section 2.1.4) to learn the specific value of nonzero entries in $K^{(\kappa+1)}$. Note that we do not use stability selection (cf. Section 3.3.1) here, since it only provides an upper bound of the threshold (3.12) for selecting “stable” edges. As a consequence, the estimated graph is often more sparse than the true one. On the other hand, the BINCO algorithm cannot handle the case of multiple regularization parameters, and hence, we resort to stability selection when solving the HVCGGM in Chapter 3.

4.3.1 Structure Learning

We apply the BINCO procedure [114] to infer the sparsity pattern of the precision matrix $K^{(\kappa+1)}$, from $L = NM$ Gibbs samples drawn in the E-step. This method is motivated by stability selection [107], however, a more reasonable criterion is proposed to determine which edge in the candidate graph is chosen.

We randomly divide all the L samples into M sample sets S_m , $m = 1, \dots, M$, of size N . For each λ , we estimate one precision matrix K_m for each sample set S_m by solving (4.14), resulting in M precision matrices. For each element (i, j) in the matrix K_m , the number of times it is nonzero for each sample set S_m is counted and divided by M . As a result, we obtain the selection frequency (or stability) $\hat{p}_{ij}$ of the corresponding edge associated with λ . Note that $\hat{p}_{ij}$ can only take values in $\{0, 1/M, 2/M, \dots, 1\}$. Typically, a proper selection of λ will result in a U-shaped empirical density function $f^\lambda(x)$ of selection frequencies of all the candidate edges, as illustrated in Fig. 4.1(a).

Figure 4.1: The distribution of selection frequencies based on a simulated dataset (a) all the edges; (b) null edges and true edges separately.

The selection frequencies fall into two categories, i.e., “true” or “null”, depending on whether the edge exists in the true graphical model. Consequently, the density function $f^\lambda(x)$ can be decomposed as

$$f^\lambda(x) = (1 - \pi)f_0^\lambda(x) + \pi f_1^\lambda(x), \quad x \in \left\{0, \frac{1}{M}, \frac{2}{M}, \dots, 1\right\}, \quad (4.15)$$

where π is the proportion of true edges, and f_0^λ and f_1^λ are the density functions of selection frequencies when the edge belongs to the “null” and “true” category respectively. As illustrated in Fig. 4.1(b), the distribution of f_0^λ is monotonically decreasing and it almost equals f^λ when the selection frequencies are small. In other words, we can obtain the following proper condition:

Proper Condition 1. [114] *There exists V_1 and V_2 , $0 < V_1 < V_2 < 1$, such that,*

1. $f_1^\lambda \rightarrow 0$ on $(V_1, V_2]$;
2. f_0^λ is monotonically decreasing on $(V_1, 1]$.

This proper condition is satisfied by a class of procedures as described in the theorem below:

Theorem 4.1. [114] *A selection procedure satisfies the proper condition if, as the sample size increases, $\hat{p}_{ij}$ tends to one uniformly for all true edges and has a limit superior strictly less than one for all null edges.*

It has been verified in [114] that all consistent procedures for graphical model selection (e.g., the glasso algorithm) satisfy the condition in Theorem 4.1.

The proper condition motivates us to estimate f_0^λ and π using the empirical distribution of selection frequencies f^λ in the region $(V_1, V_2]$. Specifically, we posit a parametric model over f_0^λ [114]:

$$f_0^\lambda = \int_0^1 b_1(x|\tau)b_2(\tau|a, b)d\tau, \quad (4.16)$$

where $b_1(x|\tau)$ is a binomial distribution, and $b_2(\tau|a, b)$ is a Beta distribution that is conjugate to the binomial distribution. It is intuitive to model the selection frequency by a rescaled binomial distribution, since whether an edge is selected can be described by a binomial distribution, and a beta distribution can have the shape with f_0^λ in the region $(V_1, 1]$. We then estimate the parameters a , b and π by minimizing the Kullback-Leibler (KL) divergence between $(1 - \pi)f_0^\lambda$ and f^λ in the region $(V_1, V_2]$.

Next, we extrapolate the parametric model (4.16) of f_0^λ to the region $(V_2, 1]$, and compute the False Detection Error (FDR) for possible threshold $c \in (V_2, 1]$ as:

$$\text{FDR}^\lambda(c) = \frac{\sum_{x \geq c} f_0^\lambda(x)}{\sum_{x \geq c} f^\lambda(x)}. \quad (4.17)$$

For each λ , we choose the threshold c that minimizes the FDR.

We then find the optimal λ that maximizes the estimated number of true edges whose definition is:

$$N_e(\lambda) = (1 - \text{FDR}^\lambda(c)) \sum_{x \geq c} f^\lambda(x). \quad (4.18)$$

Eventually, the sparsity pattern K is obtained by retaining those edges with selection frequencies above threshold c . The selected edges are then stored in the edge set $\mathcal{E}_K$ of the graphical model associated with K .

4.3.2 Parameter Learning

After learning the structure of the graphical model, the problem of learning the nonzero elements in the precision matrix $K^{(\kappa+1)}$ can be formulated as:

$$\begin{aligned} K^{(\kappa+1)} = \underset{K \succ 0}{\operatorname{argmax}} \log \det K - \operatorname{tr}(S^{(\kappa)} K), \\ \text{s.t. } K_{ij} = 0 \quad \forall (i, j) \notin \mathcal{E}_K. \end{aligned} \tag{4.19}$$

This parameter learning problem can be solved efficiently by Iterative Proportional Fitting [85].

4.4 Numerical Results

In this section, we benchmark the proposed learning method with existing methods on both synthetic and real data.

4.4.1 Synthetic Data

We generate synthetic discrete data as follows:

1. We generate a random 15 by 15 precision matrix through the method described in Section 3.4.1.
2. From the resulting precision matrix, we generate Gaussian data and then apply Beta marginals with different shape parameters to transform each Gaussian variable to the corresponding non-Gaussian continuous variable (4.2).

Table 4.2: Quantitative comparison of Gaussian graphical models (GGM), copula Gaussian graphical models (CGGM), and discrete copula Gaussian graphical models (DCGGM).

	Methods	Precision	Recall	F_1 -score
binary	GGM	0.96(0.09)	0.24(0.08)	0.37(0.11)
	CGGM	0.96(0.09)	0.25(0.09)	0.38(0.11)
	DCGGM	0.96(0.07)	0.40(0.10)	0.56(0.10)
3-ary	GGM	0.99(0.03)	0.54(0.17)	0.68(0.17)
	CGGM	0.99(0.04)	0.54(0.17)	0.69(0.16)
	DCGGM	0.99(0.04)	0.69(0.18)	0.79(0.15)
5-ary	GGM	0.99(0.02)	0.83(0.09)	0.90(0.05)
	CGGM	1.00(0.01)	0.85(0.10)	0.91(0.06)
	DCGGM	1.00(0.01)	0.92(0.08)	0.95(0.04)
10-ary	GGM	1.00(0.01)	0.93(0.05)	0.96(0.03)
	CGGM	1.00(0.01)	0.95(0.05)	0.97(0.03)
	DCGGM	1.00(0.007)	0.97(0.04)	0.99(0.02)

3. We partition the domain of each continuous variable into d equidistant intervals, where d is the number of discrete values for each variable. We replace the samples within the i -th interval ($i = 1, \dots, d$) by the value i , leading to discrete data.

We generate four groups of 100 datasets with 2, 3, 5, 10 discrete values respectively, all generated from the same precision matrix. The sample size for each dataset is 750. In our experiment, the number of burn-in iterations in the Gibbs sampler is set to 12,000, and we use the last 1,000 Gibbs samples. Our results are summarized in Table 4.2. We list the mean value and standard deviation (in brackets) for all the criteria (including precision, recall and F_1 -score as defined in Section 3.4.1) averaged over 100 datasets.

The proposed the DCGGM always outperforms the GGM (cf. Section 2.1.4) [18] and the CGGM (cf. Section 3.1) [22], especially when the alphabet is small. The performance of all three methods improves, when the alphabet size increases and

hence more information about the precision matrix is contained in the discrete data. When the number of discrete values increases to 10 and beyond, the discrete distribution becomes a pseudo-continuous distribution, which may explain why the GGM and the CGGM perform well for 10-ary data. We also considered smaller sample sizes (not shown here), and observed that the performance gain of the DCGGM over the other two methods becomes larger. We also tested the MCMC algorithm of [29] on the 100 binary data sets (first group). The resulting mean value (standard deviation) for the three criteria are 0.76(0.11), 0.48(0.09), and 0.58(0.09) respectively. We can see that the precision of the MCMC algorithm is much smaller than that of the proposed method, indicating that the MCMC algorithm involves more false positives in the estimated graph. On the other hand, the recall of the MCMC algorithm is slightly larger, implying that the MCMC algorithm successfully infers few more true edges than the proposed algorithm. In other words, in comparison with the MCMC algorithm, the proposed method yields a relatively sparse graph, greatly reducing the false detection rate at the cost of slightly lower recall. Additionally, in our simulations, the running time of the proposed algorithm (MATLAB implementation) is about 5 times shorter than the MCMC approach of [29] (C++ implementation).

4.4.2 Real Data

4.4.2.1 General Social Survey Data

Here we consider data from the 1994 General Social Survey (GSS), concerning 1002 males in the U.S. labor force [115]. There are 6 relevant variables: income of the respondent in 1000s of dollars (INC), highest degree ever obtained (DEG), number of children ever had (CHILD), financial status of respondent's parents when respondent was 16 (PINC), maximum of mother's or father's highest degree (PDEG), and number of siblings of the respondent plus one (PCHILD). Therefore, $P = 6$ in this case. Missing data rates among each of the non-income variables was less than 4%.

Figure 4.2: Results of different methods on the GSS data: (a) Hoff's method [28]; (b) the GGM; (c) the CGGM; (d) DCGGM.

The missing data rates for INC and PINC were 10% and 48% respectively. However, the question PINC was asked on only half of the surveys, and so missing values for this variable can reasonably be considered as missing at random. Age of the survey respondent is additionally included, as it is typically strongly related to income and number of children.

The results for all three methods are shown in Fig. 4.2, in addition to results for Hoff's method [28], which is a discrete copula Gaussian graphical model learned by standard MCMC. The networks inferred by the GGM and the CGGM are quite different from that by Hoff's method [28]. In contrast, the graphical model obtained through the proposed DCGGM is identical to the one via Hoff's method (apart from one edge). Hoff's method [28], however, is much more computationally complex than the proposed method. In each iteration, the computational complexity of Gibbs sampling in Hoff's algorithm is $\mathcal{O}\{LP^4\}$, compared to $\mathcal{O}\{LP^2\}$ in our algorithm. The complexity of learning dependence structure is not comparable due to different method used in those two algorithms. Moreover, Hoff's method needs 25,000 iterations to simulate the stable posterior distribution while the MCEM algorithm for the DCGGM only needs to iterate 5 times until the dependence structure has converged.

Figure 4.3: Results of different methods on the Rochdale data: (a) Whittaker's method [116]; (b) the GGM; (c) the CGGM; (d) the DCGGM.

4.4.2.2 The Rochdale Data

Here we consider a social survey data set previously analyzed in [116]. This observational study was conducted in Rochdale and attempted to explore factors affecting women's economic activity. The corresponding eight variables are as follows: a) Wife economically active (no, yes); b) Age of wife > 38 (no, yes); c) Husband unemployed (no, yes); d) Number of children ≤ 4 (no, yes); e) Wife's education, high-school+ (no, yes); f) Husband's education, high-school+ (no, yes); g) Asian origin (no, yes); h) Other household member working (no, yes). The results are shown in Fig. 4.3 for the GGM, the CGGM, and the proposed DCGGM, in addition to a log-linear model with minimum sufficient statistics, proposed by Whittaker [116].

The result of the proposed DCGGM (see Fig. 4.3(d)) is almost identical to Whittaker's result (see Fig. 4.3(a)), as only the edge (d, h) is missing. However, the GGM and the CGGM fail to infer the graphical model, probably because both models are mostly designed for continuous data.

Interestingly, in the DCGGM, the variables connected to a ("Wife economically active") are c ("Husband unemployed"), d ("Number of children ≤ 4 "), e ("Wife's education"), and g ("Asian origin"), which is identical to Whittaker's results [116] and results in [29, 117]. In the latter, full MCMC approaches are employed to infer all variables and hence are computationally complex.

We also investigated the partial correlations for pairs of variables. Whittaker [116] observed that the strongest pairwise interaction is $\{b, d\}$, followed by $\{b, h\}$, $\{e, f\}$

and $\{a, g\}$. The order of those pairs is identical in the proposed DCGGM and the models of [29, 117]. The latter two models learn the order of interactions based on the Cramér’s V statistic [31].

The results for this dataset seem to suggest that the proposed DCGGM method yields very similar results as Whittaker’s log-linear model and the approach of [29, 117]. Whittaker’s log-linear model, however, is not able to learn the sparsity structure automatically in contrast to the proposed method. The MCMC approach of [29, 117] is computationally much more complex than the proposed approach.

4.4.2.3 US Senate Voting Records Data

We close our numerical experiments by validating our DCGGM on the voting records of US Senate [118]. This dataset involves the voting records of 102 senators for 645 bills from the 109th congress (2004 - 2006). Therefore, the sample size is 645 and the number of variables is 102. The votes are recorded as 1 for “yes” and -1 for “no”. We further regard “not voting” and “absence” as missing data. Note that there exist some missing records for each bill, thus, approaches such as the GGM and CGGM are not applicable directly under this scenario, since they cannot tackle missing components in every sample.

Our objective is to infer the network between the 102 senators. We depict the estimated graphical model resulting from the DCGGM in Fig 4.4. The red, blue, and yellow nodes correspond to Democratic, Republican, and Independent senators respectively. Interestingly, the proposed DCGGM can well distinguish between Democrats and Republicans in an automated manner: except for the Independent senator (the yellow node), Democrats (red nodes) only have connections with Democrats, while Republicans (blue nodes) only have other Republicans as neighbors.

Banerjee *et al.* [17] applied the GGM to the dataset after replacing all missing votes by -1. However, the resulting graphical model includes some edges between Demo-

Figure 4.4: Network of US senators: Democratic, Republican, and Independent senators are respectively represented by red, blue, and yellow nodes.

cratic and Republican senators. Again, the proposed DCGGM outperforms the GGM when analyzing discrete data with missing values.

4.5 Summary

We proposed novel learning algorithms for discrete copula Gaussian graphical models in this chapter, which can be applied to any combinations of heterogeneous variables. Numerical results for synthetic data show that the proposed method can produce better estimates than the GGM and CGGM. Moreover, results on real data suggest that the proposed method yields similar results as for existing methods, while being more computationally efficient and more generally applicable.

Chapter 5

Network Inference and Change Point Detection for Piecewise-Stationary Time Series

This chapter is concerned with piecewise-stationary data. The resulting model may be applied to real-world data in different regimes, such as stock indices, biomedical signals, and climate measurements. In particular, we extend copula Gaussian graphical models with and without hidden variables to model piecewise stationary data. To this end, we first detect abrupt change points in the time series using the method proposed in Section 5.1. In Section 5.2, we further describe a method to determine the number and position of the change points in an automatic manner. We then infer the graphical model for each stationary time segment using the learning methods described in previous sections. Specifically, we employ the stability selection (cf. Section 3.3.1) learn the structure of HVCGGMs, since it can deal with multiple regularization parameters. However, stability selection often leads to a graphical model that is too sparse. As a consequence, we exploit the BINCO method (cf. Section 4.3.1) for CGGMs with one single regularization parameter, since it determines the edge selection threshold reasonably by minimizing the false detection rate. We present the numerical results for both synthetic and real data in Section 5.3. We

finally offer concluding remarks in Section 5.4.

5.1 Change Point Detection

In this section, we address the problem of detecting change points in piecewise stationary multivariate time series. Specifically, we aim to detect changes in the statistical dependence (a.k.a. connectivity or network structure) among the time series. The data between the change points is assumed to be stationary, and will be modeled by copula Gaussian graphical models.

Let us assume that we have an ordered sequence of N samples for each of the P variables, $x_i^{(n)}$, where $n = 1, \dots, N$ and $i = 1, \dots, P$. We wish to infer an unknown number M of change points $\tau_{1:M} = (\tau_1, \dots, \tau_M)$. Each change point is an integer between ℓ and $N - \ell$, where ℓ is minimum length of one segment. We further define $\tau_0 = 0$ and $\tau_{M+1} = n$, and thus the M change points will split the data into $M + 1$ segments, where the k th segment is given by $\mathbf{x}_{1:P}^{(\tau_{k-1}+1:\tau_k)}$.

As a first step, we transform the non-Gaussian observed variables x_i into Gaussian latent variables z_i (associated with the observed variables x_i), i.e., $z_i = \Phi^{-1}(\hat{F}_i(x_i))$.

We next solve the problem of identifying all the change points together in the Gaussian latent layer. Concretely, we minimize a cost function with a penalty on the number of change points, as suggested in the literature [37, 119]:

$$\sum_{k=1}^{M+1} L_k \left(\mathbf{z}_{1:P}^{(\tau_{k-1}+1:\tau_k)} \right) + \beta M, \quad (5.1)$$

where β is the cost associated with each change point, in order to limit overfitting.

The negative log-likelihood L_k is defined as:

$$L_k(\mathbf{z}_{1:P}^{(\tau_{k-1}+1:\tau_k)}) = \frac{\tau_k - \tau_{k-1}}{n} \log \det S_k, \quad (5.2)$$

where S_k is the empirical covariance of segment k . We apply the Pruned Exact Linear Time (PELT) method [37] to efficiently find the global minimum of (5.1).

Specifically, let $F_M(t)$ denote the global minimum of (5.1) for data $\mathbf{z}_{1:P}^{(1:t)}$ and let $T_t = \{\tau : 0 = \tau_0 < \tau_1 < \dots < \tau_M < \tau_{M+1} = t\}$ be the set of candidate change points at time t . It therefore follows that [120]:

$$F_M(t) = \min_{\tau \in T_t} \sum_{k=1}^{M+1} \left[L\left(\mathbf{z}_{1:P}^{(\tau_{k-1}+1:\tau_k)}\right) + \beta \right] \quad (5.3)$$

$$= \min_s \left\{ \min_{\tau \in T_s} \sum_{k=1}^M \left[L\left(\mathbf{z}_{1:P}^{(\tau_{k-1}+1:\tau_k)}\right) + \beta \right] + L(\mathbf{z}_{1:P}^{(s+1:t)}) + \beta \right\} \quad (5.4)$$

$$= \min_s \left[F_{M-1}(s) + L(\mathbf{z}_{1:P}^{(s+1:t)}) + \beta \right]. \quad (5.5)$$

The interpretation is that the optimal value $F_M(t)$ of the cost function at time t can be derived using the optimal partition of the data prior to the last change point plus the cost for the segment from the last change point to t . In other words, the expression (5.5) offers a recursion expressing the minimum for data $\mathbf{z}_{1:P}^{(1:t)}$ in terms of the minimum for $\mathbf{z}_{1:P}^{(1:s)}$ for $s < t$. We then set $F(0) = -\beta$ so that the penalty equals βM in (5.3), and (5.5) can be solved in turn for $t = \ell + 1, \dots, N$ by finding the most recent change point s before t . The computational cost of solving (5.5) is linear in t , since we need to consider all candidate positions of the last change point before time t , namely, $s \in \{\ell + 1, \ell + 2, \dots, t - \ell\}$. Therefore, the overall computational complexity of finding the optimal partition is $\mathcal{O}(N^2)$.

The computational complexity can be further reduced by pruning the candidate set of last change points at time t as t increases. Such pruning can be done by removing those candidates that can never be change points after t . The following theorem asserts a condition to determine which time points to be pruned:

Theorem 5.1. [37] *We assume that when introducing a change point into a sequence of observations, the cost L of the sequence reduces. More formally, for all $r < s < t$,*

$$L(\mathbf{z}^{(r+1:s)}) + L(\mathbf{z}^{(s+1:t)}) \leq L(\mathbf{z}^{(r+1:t)}). \quad (5.6)$$

Therefore, if it holds that

$$F(r) + L(\mathbf{z}^{(r+1:s)}) \geq F(s), \quad (5.7)$$

then at a future time $t > s$, r can never be the optimal last change point prior to t .

By discarding time points satisfying the above condition in each iteration, such procedure successfully removes computations that are irrelevant to obtaining the final set of change points and accelerates the algorithm. The resulting computational cost the PELT method can be reduced to be linear under certain conditions as proven in [37]. In summary, the PELT method is proceeded as follows:

1. Initialize $F(0) = -\beta$. The set $cp(t)$ of previous change points at $t \leq \ell$ is initialized as $cp(t) = \emptyset$, and the set $R_{\ell+1}$ of candidate change points at $t = \ell+1$ is initialized as $R_{\ell+1} = \{0\}$.
2. Compute $F(t) = \min_{\tau \in R_t} [F(\tau) + L(\mathbf{z}_{1:P}^{(\tau+1:t)}) + \beta]$ and let $\tau^* = \operatorname{argmin}_{\tau \in R_t} F(t)$.
3. Update the set $cp(t)$ of previous change points at time t : $cp(t) = cp(\tau^*) \cup \tau^*$.
4. Prune the set R_t by removing $\{\tau \in R_t : F(\tau) + L(\mathbf{z}_{1:P}^{(\tau+1:t)}) > F(t)\}$. If $t \geq 2\ell - 1$, then update R_{t+1} of candidate change points at the next time position $t + 1$: $R_{t+1} = R_t \cup \{t + 1 - \ell\}$.
5. Return to Step 2 if $t < N$ and increase t by 1.

The resulting $cp(N)$ is the optimal set of change points.

5.2 Regularization Selection

Another issue with solving (5.1) is the selection of regularization parameter β , which determines the final number of change points. There is no uniform rule to compute β for all kinds of data. An adaptive method to determine the number of change points

is introduced in [119], which keeps increasing the number of change points until the negative log-likelihood of the entire time series ceases to decrease significantly. In the example of Fig. 5.1(a), the optimal number of change points is three, as the negative log-likelihood starts decreasing more slowly for larger number of change points. Since we implicitly infer the number of change points by tuning the value of β , we modify the abovementioned approach to choose β in the same spirit.

Figure 5.1: Adaptive regularization selection.

Let $L = \sum_{k=1}^{M+1} L_k$ and define a general penalty $f(M)$, which equals M in our case. It is apparent that the estimated number of change points $\hat{M}(\beta)$ is a piecewise constant function of β . As such, if $\hat{M}(\beta) = a$, $L(a) + \beta f(a) < \min_{b \neq a} (L(b) + \beta f(b))$. Therefore, β satisfies the following condition [119]:

$$\max_{b>a} \frac{L(a) - L(b)}{f(b) - f(a)} < \beta < \min_{b<a} \frac{L(b) - L(a)}{f(a) - f(b)}. \quad (5.8)$$

In the example of Fig. 5.1, for β such that $\hat{m}(\beta) = 3$, $L(3) - L(4) < \beta < L(2) - L(3)$. There hence are ordered sequences $1 = m_1 < m_2 < \dots$ and $\infty = \beta_0 > \beta_1 \dots$ defined as [119]:

$$\beta_i = \frac{L(m_i) - L(m_{i+1})}{f(m_{i+1}) - f(m_i)}, \quad i \geq 1, \quad (5.9)$$

such that $\hat{m}(\beta) = m_i, \forall \beta \in [\beta_i, \beta_{i-1})$ (as marked by the red circles in Fig. 5.1(b)).

To find the m for which L ceases to decrease significantly, we need to look for a *break* in the slope of the function $L(f)$, which is defined by β according to (5.9). Therefore, the change of the slope is determined by the length h_{m_i} of the interval $[\beta_i, \beta_{i-1})$. As a result, the regularization selection procedure can be executed as follows:

1. Select an arithmetic sequence of $\beta = (\beta_1, \beta_2, \dots)$ (denoted by black plus signs in Fig. 5.1(b)). In practice, we set the interval between two β values to be:

$$c \log(N)P(P+1)/4N, \quad (5.10)$$

where $c = 0.02$ is a user-defined constant and $\log(N)P(P+1)/4N$ is the regularization parameter when the BIC is employed for regularization selection.

2. For each β_i , estimate the number of change points $\hat{M}(\beta_i)$ by solving (5.1) using the PELT method.
3. Compute the length h_{M_j} of the interval by counting the number of β_i that generates M_j change points, which is represented by the length between two circles with the same number of change points in Fig. 5.1(b).
4. Choose the smallest value of β such that $h_{M_j} \gg h_{M_k}$, for $k > j$. In Fig. 5.1(b), as an example, β corresponding to $h_{\hat{M}(\beta)=3}$ is optimal since $h_{\hat{M}(\beta)=3} \gg h_{\hat{M}(\beta)=4}$.

Once we obtain the optimal segmentation of the time series, we can learn graphical models based on the empirical covariance S_k for each time segment. Here we consider copula Gaussian graphical models without and with hidden variables (i.e., CGGMs and HVCGMs), which are inferred by solving (3.6) and (3.11) respectively.

5.3 Numerical Results

In this section, we validate the proposed model on both synthetic and real data.

5.3.1 Synthetic Data

The piecewise stationary non-Gaussian data are generated as follows:

1. We generate $M + 1$ random precision matrices K_k ($k = 1, \dots, m + 1$) with $P = 25$ as in previous chapters, where $M = 3$ is the predefined number of change points. Specifically, first we uniformly sample $x_1, \dots, x_n$ from a unit square. The precision matrix is initialized as a unit matrix. Next, we set the element $[K_k]_{jk} = [K_k]_{jk}$ of precision matrix equal to $\rho = 0.245$ with probability $(\sqrt{2\pi})^{-1} \exp(-4\|x_i - x_j\|^2)$, and equal to zero otherwise. For assessing the hidden variable graphical models (cf. Section 3.4.1), we add a few variables and connect each of them to at least 80% of other variables; the corresponding elements in precision matrix are nonzero.
2. From each precision matrices K_k , we generate N_k Gaussian samples. Consequently, N_k is the length of the k th segment. We then concatenate the Gaussian data corresponding to all the precision matrices. For assessing the hidden variable graphical models, we discard all the samples of variables added in Step 1 (hidden variables).
3. Transform the P Gaussian variables to continuous non-Gaussian variables following (3.2), where F_i can be any non-Gaussian continuous marginal. Normalize the non-Gaussian data set to have unit variance, so that its empirical covariance becomes more tractable.

We generate 100 25-dimensional data sets with 2220 non-Gaussian samples for each variable. The common true values of change points are (661, 1561, 1861). In order to assess both the copula Gaussian graphical models with and without hidden variables, the first half of 100 data sets does not contain hidden variables while the other half does.

We first test the accuracy of change point detection. Specifically, we benchmark the proposed copula PELT method with adaptive regularization selection (denoted

Table 5.1: Comparison of different methods for accuracy of change point (cp) detection

Methods	Distribution of detected cp no.				MAE (cp no. = 3)
	0	1	2	3	
CPELT-A	0%	0%	0%	100%	1.4
CPELT-B	100%	0%	0%	0%	
PELT-A	0%	31%	1%	68%	2.8
PELT-B	91%	9%	0%	0%	

as “CPELT-A”) against three other approaches: copula PELT method with the regularization parameter selected via the BIC (denoted as “CPELT-B”), the original PELT method with adaptive regularization selection (“PELT-A”), and the original PELT method with the BIC (“PELT-B”). The results are listed in Table 5.1, where we show the distribution of the detected number of change points. We also report the mean absolute error (MAE) between the estimated position of change points and the ground truth, for the cases where the correct number of change points is inferred. Clearly, the proposed CPELT-A method greatly outperforms other methods, reliably identifying the number and the position of change points. On the other hand, CPELT-B and PELT-B seriously underestimate the number of change points; the BIC does not seem to be suitable here for regularization selection. The PELT-A method can often detect the correct number of change points (in 68% of the cases), but the inferred position of change points is not as accurate compared to CPELT-A. We also validated the graphical models inferred from the stationary segments between change points obtained from CPELT-A. From our results (not shown here for the sake of conciseness), we observe that the proposed methods can reliably recover the true graph (precision matrices generated in Step 1), as reported in Section 3.4.1 and Section 4.4.1.

5.3.2 Real Data

We now apply the proposed model to scalp electroencephalograms (EEG) recorded during epilepsy seizures [121]. The EEG time series were collected from 23 pediatric patients (5 males, ages 3-22; and 17 females, ages 1.5-19) with intractable seizures, using the international 10-20 system at a sampling frequency of 256 Hz. During the recordings, the patients experienced 173 events that were judged to be clinical seizures by experts. Here, we randomly select five events and extract EEG data starting 60s before seizure onset and ending 30s after seizure termination. The EEGs are band-pass filtered with a digital third-order Butterworth filter between 3 and 40Hz. We then down sample the signal to retain one fourth of all the samples, to limit the computational complexity.

Table 5.2: Results for seizure data

No.	Seizure period		Estimated results: change points and network density								
1	60	76	change points	64.34	72.09	77.45	90.75				
			no. of edges (CGGM)	142	106	104	101	130			
			no. of edges (HVCGGM)	34	12	40	23	23			
2	60	109	change points	35.23	60.78	72.69	110.44				
			no. of edges (CGGM)	140	131	111	119	126			
			no. of edges (HVCGGM)	18	50	18	33	29			
3	60	100	change points	60.88	69.42	98.52	119.52				
			no. of edges (CGGM)	146	108	149	135	125			
			no. of edges (HVCGGM)	25	26	28	33	33			
4	60	123	change points	61.48	117.02	118.48	120.25	122.66	141.05		
			no. of edges (CGGM)	137	135	59	48	52	106	91	
			no. of edges (HVCGGM)	23	34	32	3	22	50	24	
5	60	142	change points	30.00	59.16	66.63	94.20	141.47			
			no. of edges (CGGM)	104	90	62	105	123	114		
			no. of edges (HVCGGM)	16	30	16	19	54	19		

The results are summarized in Table. 5.2. We observe that the proposed method detects change points near the start and end of seizures, as one would expect. Typically there is change point within two seconds from the start or end of a seizure,

Figure 5.2: Results of functional networks resulting from graphical models without hidden variables (a)-(g) and with hidden variables (h)-(n) (with the vertical lines denoting the onset and end of one example seizure)

indicating that the proposed method may be further developed for detecting the onset and ending of seizures.

We also inferred the functional network in each stationary time segment by using the Gaussian copula graphical models with and without hidden variables. As an illustration, we show the networks without and with hidden variables for the 4th data set in Fig. 5.2. Kramer *et al.* [32] analyzed the dynamics of functional networks through the entire seizure in intracranial electrocorticogram (ECoG) recordings. They found that the networks are dense at seizure onset and termination, but sparse during the middle portion of the seizure. Interestingly, although scalp EEG is more noisy than intracranial ECoG recordings, the proposed method can find the same pattern as observed in [32]. Moreover, the inferred number of stationary segments for all 5 data sets are similar, implying that seizures undergo characteristic progressions that can be stretched or compressed in time, as assumed in [32]. It is noteworthy that, in contrast to [32], the proposed method allows us to infer the number and location of change points in an automated fashion.

5.4 Summary

An effective method is proposed to infer abruptly changing network structures from piecewise-stationary time series. A low-complexity PELT-based algorithm is introduced to detect change points. Next graphical models (with and without hidden variables) are inferred for each stationary time segment. Numerical results for synthetic and real data demonstrate the utility of the proposed model.

As an illustration, we have shown that the proposed model can be an effective tool to determine “when” and “how” the functional brain networks change during epileptic seizures, which may lead to novel insights in the phenomenology of seizures. The proposed method may find use in other biomedical applications and beyond.

Part II

Graphical Models for Marginal Analysis of Extreme-Value Data

Chapter 6

Extreme-Value Graphical Models with Multiple Covariates

In this chapter, we shift our attentions from nominal events to marginal analysis of extreme events. Extreme events often display heterogeneity (i.e., non-stationarity), depending continuously on a number of covariates. Previous studies have suggested that models considering covariate effects lead to reliable estimates of extreme events distributions. In this chapter, we develop a novel statistical model to incorporate the effects of multiple covariates. More explicitly, we aim to model extreme events with multiple covariates by means of graphical models. Our model is theoretically significant because it is among the first approaches to exploit the framework of graphical models to analyze the marginal behavior of extreme events. As an example, we model the storm-wise maxima of significant wave heights in the Gulf of Mexico (see Fig. 6.6), where the covariates are longitude, latitude, and wind direction. Note that the significant wave height is a standard measure of sea surface roughness; it is defined as the mean of the highest one third of waves (typically in a three-hour period). Theoretically, significant wave heights are affected by dominant wave direction (i.e., storm direction). However, we use wind direction as a surrogate since wind and wave direction are generally fairly well correlated, especially for extreme events.

The proposed model is derived from the following ideas: Motivated by the extreme value theory [41], the extreme events are assumed to follow the fat-tailed Generalized Extreme Value (GEV) distributions. The parameters of those GEV distributions are further assumed to depend smoothly on the covariates. To facilitate the use of graphical models, we discretize the continuous covariates within a finite range. In the example of extreme wave heights in the Gulf of Mexico, space is discretized as a finite homogeneous two-dimensional lattice, and the wind direction is discretized in a finite number of equal-sized sectors. More generally, the GEV distributed variables (and hence also the GEV parameters) are defined on a finite number of points indexed by the (discretized) covariates. We characterize the dependence between the GEV parameters through a graphical model prior, in particular, a multidimensional thin-membrane model where edges are only present between pairs of neighboring points (see Fig. 6.1). We demonstrate that the multidimensional model can be constructed flexibly from one-dimensional thin-membrane models for each covariate. The proposed model can therefore easily be extended to cope with an arbitrary number of covariates. We follow the empirical Bayes approach to learn the parameters and hyper-parameters. Specifically, both the smoothed GEV parameters and the smoothness parameters are inferred via Expectation Maximization. A major challenge lies in the scalability of the algorithm since the dimension of the model is usually quite large. Instead of using the aforementioned algorithms for solving cyclic graphical models (cf. Section 2.1.5), we take advantage of the special pattern of the eigenvalues and eigenvectors corresponding to the one-dimensional thin-membrane models, and derive an efficient inference algorithm specialized for the proposed model.

This chapter is organized as follows. In Section 6.1, we introduce thin-membrane models and their properties. In Section 6.2, we first construct the 3D thin-membrane model based on simple 1D thin-membrane models to model extreme wave heights, and then illustrate how to generalize the model to incorporate any number of covariates. In Section 6.3, we discuss the efficient learning and inference algorithms at length and provide theoretical guarantees. Numerical results for both synthetic

and real data are presented in Section 6.4. Lastly, we end this chapter with some concluding remarks in Section 6.5.

6.1 Thin-membrane models

In this section, we first give a brief introduction to thin-membrane models, and then analyze two concrete examples of thin-membrane models that will be used as building blocks in the proposed extreme-value graphical model: chain and cycle models.

The thin-membrane model [122] is a Gaussian graphical model that is commonly used as smoothness prior, as it minimizes the difference between values at neighboring nodes:

$$P(\mathbf{z}) \propto \exp\left\{-\frac{1}{2}\alpha \sum_{i \in \mathcal{V}} \sum_{j \in N(i)} (z_i - z_j)^2\right\} \quad (6.1)$$

$$\propto \exp\left\{-\frac{1}{2}\alpha \mathbf{z}^T K_{\text{tm}} \mathbf{z}\right\}, \quad (6.2)$$

where $N(i)$ denotes the neighboring nodes of node i , K_{tm} is a graph Laplacian matrix such that $[K_{\text{tm}}]_{i,i}$ is the number of neighbors of the i^{th} node while the off-diagonal elements $[K_{\text{tm}}]_{i,j}$ are equal to -1 if nodes i and j are adjacent and 0 otherwise, and α is the smoothness parameter which controls the smoothness across the domain defined by the thin-membrane model. By comparing (6.2) with (2.10), we can see that the precision matrix of the thin-membrane model is $K = \alpha K_{\text{tm}}$. Since K_{tm} is a Laplacian matrix, K is rank deficient, i.e., $\det K = 0$. As such, the thin-membrane model is classified as a partially informative normal prior [123] or intrinsic Gaussian Markov random field [77]. To make the distribution well-defined, the improper density function is usually applied in practice:

$$P(\mathbf{z}) \propto |K|_+^{0.5} \exp\left\{-\frac{1}{2}\mathbf{z}^T K \mathbf{z}\right\} \quad (6.3)$$

$$= |\alpha K_{\text{tm}}|_+^{0.5} \exp\left\{-\frac{1}{2}\alpha \mathbf{z}^T K_{\text{tm}} \mathbf{z}\right\}, \quad (6.4)$$

where $|K|_+$ denotes the product of nonzero eigenvalues of K . Note that $K_{\text{tm}}\mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ is a vector of all ones, indicating the eigenvalue associated with the eigenvector $\mathbf{1}$ equals 0. Thus, the thin-membrane model is invariant to the addition of $c\mathbf{1}$, where c is an arbitrary constant, and it allows the deviation from any overall mean level without having to specify the overall mean level itself. As an illustration, we can easily find that the conditional mean of variable z_i is [77]

$$E(z_i|z_{\mathcal{V}|i}) = -\frac{1}{[K_{\text{tm}}]_{i,i}} \sum_{j \in N(i)} [K_{\text{tm}}]_{i,j} z_j, \quad (6.5)$$

i.e., the mean of its neighbors, but it does not involve an overall level. This special behavior is often desirable in applications.

We now turn our attention to the thin-membrane models of the chain and the cycle graph, as shown in Fig. 6.1(a) and Fig. 6.1(b) respectively. The former can well characterize the dependence structure of nonperiodic covariates (e.g., longitude and latitude), while the latter is highly suitable for periodic covariates (e.g., directional and seasonal patterns). The corresponding Laplacian matrices are denoted as K_B and K_C respectively. It is easy to prove that the eigenvalues and eigenvectors of K_B and K_C have the following special pattern [124]:

$$\begin{aligned} \lambda_{Bk} &= 2 - 2 \cos \left(\frac{k\pi}{P_K} \right), \\ \mathbf{v}_{Bk} &= \left[\cos \left(\frac{k\pi}{2P_K} \right), \cos \left(\frac{3k\pi}{2P_K} \right), \dots, \cos \left(\frac{(2P_K-1)k\pi}{2P_K} \right) \right]^T, \\ \lambda_{Ck} &= 2 - 2 \cos \left(\frac{2k\pi}{P_K} \right), \\ \mathbf{v}_{Ck} &= [1, \omega^k, \omega^{2k}, \dots, \omega^{(P_K-1)k}]^T, \end{aligned}$$

for $k = 1, \dots, P_K$, where P_K is the dimension of K_B and K_C , $\omega = \exp(2\pi i/P_K)$ and i is the imaginary unit. An important property of the eigenvectors is as follows: Let V_B be the eigenvector matrix of K_B , i.e., $V_B = [\mathbf{v}_{B1}, \mathbf{v}_{B2}, \dots, \mathbf{v}_{BP_K}]$, and $\mathbf{x}$ be a $P_K \times 1$ column vector, then $V_B \mathbf{x}$ is identical to the discrete cosine transform of x [124, Ch. 1]. Similarly, let V_C denote the eigenvector matrix of K_C , then $V_C \mathbf{x}$

can be computed as the discrete Fourier transform of $\mathbf{x}$ [124, Ch. 1]. Note that other smoothness priors, such as thin-plate models (cf. Section 7.1), do not have such nicely structured eigendecomposition. This motivates us to use thin-membrane models for the sake of computational convenience.

6.2 Modeling Multiple Covariates

In this section, we present a novel extreme-value statistical model that incorporates multiple covariates. We assume that the block maxima (e.g., monthly or annual maxima) associated with every possible set of values for the covariates follow the Generalized Extreme Value (GEV) distribution, according to the Fisher-Tippett-Gnedenko theorem in extreme value theory [41]. The GEV parameters z are assumed to vary smoothly with the covariates. The latter are discretized within a finite range. Therefore, the GEV parameters can be indexed as $z_{i_1 i_2 \dots i_m}$, where m is the number of covariates, and i_1 corresponds to a discretized value of the first covariate and likewise for the other indices $i_2, \dots, i_m$. In summary, the GEV parameters are defined on a finite number of points $(i_1, i_2, \dots, i_m)$ and those parameters are supposed to vary smoothly from one point to a nearby point. The resulting dependence structure can be well represented by a multidimensional thin-membrane model. Furthermore, we show that the multidimensional model can be constructed from one-dimensional thin-membrane models for each covariate, thereby making the proposed model generalizable to accommodate as many covariates as required. We employ the multidimensional thin-membrane model as the prior and estimate the GEV parameters through an empirical Bayes approach.

As an illustration, we present in detail the spatio-directional model to quantify the extreme wave heights. Suppose that we have N samples $x_{ijk}^{(n)}$ (block maxima) at each location, indexed by its longitude and latitude (i, j) , and directional sector k , where $n = 1, \dots, N$, $i = 1, \dots, P$, $j = 1, \dots, Q$, $k = 1, \dots, D$ and P, Q, D are the number of longitude indices, latitude indices and directional sectors respectively.

Consequently, the dimension M of the proposed model is given by $M = PQD$. Hence, our objective is to accurately infer the GEV parameters with the consideration of both spatial and directional dependence. Specifically, we first locally fit the GEV distribution to block maxima at each location and in each directional sector. The local estimates are further smoothed across the spatio-directional domain by means of 3D thin-membrane models.

6.2.1 Local estimates of GEV parameters

We assume that the block maxima x_{ijk} follow a Generalized Extreme Value (GEV) distribution [41], whose cumulative probability distribution (CDF) equals:

$$F(x_{ijk}) = \begin{cases} \exp\{-[1 + \frac{\xi_{ijk}}{\sigma_{ijk}}(x_{ijk} - \mu_{ijk})]^{-\frac{1}{\xi_{ijk}}}\}, & \xi_{ijk} \neq 0 \\ \exp\{-\exp[-\frac{1}{\sigma_{ijk}}(x_{ijk} - \mu_{ijk})]\}, & \xi_{ijk} = 0, \end{cases}$$

for x_{ijk} satisfying $1 + \xi_{ijk}/\sigma_{ijk}(x_{ijk} - \mu_{ijk}) \geq 0$ if $\xi_{ijk} \neq 0$ and $x_{ijk} \in \mathbb{R}$ if $\xi_{ijk} = 0$, where $\mu_{ijk} \in \mathbb{R}$ is the location parameter, $\sigma_{ijk} > 0$ is the scale parameter and $\xi_{ijk} \in \mathbb{R}$ is the shape parameter.

The Probability-Weighted Moment (PWM) method [125] is employed here to yield the estimates of GEV parameters $\hat{\mu}_{ijk}$, $\hat{\sigma}_{ijk}$ and $\hat{\xi}_{ijk}$ locally at each location (i, j) and each direction k . The goal of the PWM method is to match the PWMs $E[x_{ijk}(F(x_{ijk}))^r]$ with the empirical ones b_r , where r is a real number. For the GEV distribution, $E[x_{ijk}(F(x_{ijk}))^r]$ can be written as:

$$\frac{1}{r+1} \left\{ \mu_{ijk} - \frac{\sigma_{ijk}}{\xi_{ijk}} [1 + (r-1)\xi_{ijk}\Gamma(1-\xi_{ijk})] \right\}, \quad (6.6)$$

where $\xi_{ijk} < 1$ and $\xi_{ijk} \neq 0$, and $\Gamma(\cdot)$ is the gamma function. Here we choose to compute the PWMs for $r = 0, 1, 2$, and the resulting PWM estimates $\hat{\mu}_{ijk}$, $\hat{\sigma}_{ijk}$ and

$\hat{\xi}_{ijk}$ are the solution of the following system of equations:

$$\begin{cases} b_0 &= \mu_{ijk} - \frac{\sigma_{ijk}}{\xi_{ijk}}(1 - \Gamma(1 - \xi_{ijk})), \\ 2b_1 - b_0 &= \frac{\sigma_{ijk}}{\xi_{ijk}}\Gamma(1 - \xi_{ijk})(2^{\xi_{ijk}} - 1), \\ \frac{3b_2 - b_0}{2b_1 - b_0} &= \frac{3^{\xi_{ijk}} - 1}{2^{\xi_{ijk}} - 1}. \end{cases} \quad (6.7)$$

In practice, since solving the last equation in (6.7) is time-consuming, it can be approximated by [125]:

$$\hat{\xi}_i = -(7.859c + 2.9554c^2), \quad (6.8)$$

where $c = (2b_1 - b_0)/(2b_2 - b_0) - \log 2/\log 3$. The PWM method generates good estimates even when the sample size is small, as demonstrated in [125]. Therefore, the method is suitable for extreme-events modeling, especially in our case where the number of samples with the same values of covariates is limited. However, the linear approximation (6.8) is accurate only when $|\xi_{ijk}| < 0.5$. Moreover, the moments in (6.6) do not exist when $\xi_{ijk} \geq 1$.

To address these concerns, we utilize the two-stage procedure proposed by Castillo *et al.* [126], which is referred to as the median (MED) method, when the PWM estimates $|\hat{\xi}_{ijk}| \geq 0.5$ or is not a number (NAN). In the first stage of the MED method, the extreme-value samples can be ordered $x_{ijk}^{(1)} \leq x_{ijk}^{(2)} \leq \dots \leq x_{ijk}^{(N)}$. A set of GEV estimates is obtained by equating the GEV CDF $F(x_{ijk}^{(n)})$ to its corresponding empirical CDF $P(x_{ijk}^{(n)}) = (n - c_0)/N$, where $c_0 \in (0, 1)$ and we choose $c_0 = 0.35$ as

in [126]. More explicitly, for each n such that $2 \leq n \leq N - 1$ in turn, we have:

$$\begin{cases} x_{ijk}^{(1)} &= \frac{\sigma_{ijk}^{(n)}}{\xi_{ijk}^{(n)}} \{ [-\log P(x_{ijk}^{(1)})]^{-\xi_{ijk}^{(n)}} - 1 \} + \mu_{ijk}^{(n)}, \\ x_{ijk}^{(n)} &= \frac{\sigma_{ijk}^{(n)}}{\xi_{ijk}^{(n)}} \{ [-\log P(x_{ijk}^{(n)})]^{-\xi_{ijk}^{(n)}} - 1 \} + \mu_{ijk}^{(n)}, \\ x_{ijk}^{(N)} &= \frac{\sigma_{ijk}^{(n)}}{\xi_{ijk}^{(n)}} \{ [-\log P(x_{ijk}^{(N)})]^{-\xi_{ijk}^{(n)}} - 1 \} + \mu_{ijk}^{(n)}. \end{cases} \quad (6.9)$$

Note that the equations associated with the first and last sample, i.e., $x_{ijk}^{(1)}$ and $x_{ijk}^{(N)}$, will be used multiple times. By solving the three equations in (6.9) *w.r.t.* the GEV parameters independently for each n ($2 \leq n \leq N - 1$) in turn, we can obtain a set of $(\xi_{ijk}^{(n)}, \sigma_{ijk}^{(n)}, \mu_{ijk}^{(n)})$ for each of the $N - 2$ occurrences $x_{ijk}^{(n)}$. In the second stage, we determine the median of the sets of GEV parameters as the final MED estimates $(\hat{\mu}_{ijk}, \hat{\sigma}_{ijk}, \hat{\xi}_{ijk})$. As shown in [126], although the MED method is more computationally demanding than the PWM method, it yields reliable estimates when the shape parameter $|\xi_{ijk}| \geq 0.5$.

6.2.2 Prior distribution

We assume that each of the three parameter vectors $\boldsymbol{\mu} = [\mu_{ijk}]^T$, $\boldsymbol{\xi} = [\xi_{ijk}]^T$ and $\boldsymbol{\sigma} = [\sigma_{ijk}]^T$ has a 3D thin-membrane model (see Fig. 6.1(d)) as prior. Since the thin-membrane models of $\boldsymbol{\mu}$, $\boldsymbol{\xi}$, and $\boldsymbol{\sigma}$ share the same structure and inference methods, we present the three models in a unified form. Let $\mathbf{z}$ denote the true GEV parameters, that is, $\mathbf{z}$ is either $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\xi}$.

We next illustrate how to construct the 3D thin-membrane model priors from the constituent chain and cycle graphs. As a first step, we build a regular lattice (see Fig. 9.1(a)) from Markov chains. Since both the longitude and the latitude of the measurements are nonperiodic, either of them can be characterized by the chain graph shown in Fig. 6.1(a). Let K_{B_x} and K_{B_y} denote the Laplacian matrices corresponding to the graph of one row (i.e., the longitude) and one column (i.e., the

Figure 6.1: Thin-membrane models: (a) chain graph; (b) cycle graph; (c) lattice; (d) spatio-directional model.

latitude) of sites respectively. We further assume that the smoothness across the longitude and the latitude are the same, thus, they can share one common smoothness parameter α_z . The resulting precision matrix of the regular lattice is given by:

$$K_p = (\alpha_z K_{B_x}) \oplus (\alpha_z K_{B_y}) = \alpha_z (K_{B_x} \oplus K_{B_y}) = \alpha_z K_L,$$

where $\oplus$ represents the Kronecker sum. It is easy to show that $K_L = K_{B_x} \oplus K_{B_y}$ is the graph Laplacian matrix of the lattice. According to the property of the Kronecker sum [127, Ch. 13], the eigenvalue matrix $\Lambda_L = \Lambda_{B_x} \oplus \Lambda_{B_y}$ and eigenvector matrix $V_L = V_{B_y} \otimes V_{B_x}$, where $\otimes$ denotes the Kronecker product.

In the second step, we accommodate the effect of wind direction. We discretize the wind direction into D sectors and assume that the true GEV parameters are constant in each sector. As GEV parameters of neighboring sectors are similar, the directional dependence can be encoded in a cycle graph (see Fig. 6.1(b)), whose Laplacian matrix is K_C . Another smoothness parameter β_z is introduced to dictate the directional dependence since the smoothness across wind direction and space can be different. We can then seamlessly combine the lattice and the cycle, leading to the 3D thin-membrane model (see Fig. 6.1(d)) with precision matrix:

$$K_{\text{prior}} = (\alpha_z K_L) \oplus (\beta_z K_C) = \alpha_z K_s + \beta_z K_d. \quad (6.10)$$

Interestingly, $K_s = K_L \otimes I_C$ and $K_d = I_L \otimes K_C$ corresponds to the lattices and cycle graphs in the graph respectively, indicating that the former only characterizes the spatial dependence while the latter the directional dependence. Note that I_* is an identity matrix with the same dimension as K_* . Based on the property of the Kronecker sum [127, Ch. 13], the eigenvalue matrix of K_{prior} equals:

$$\Lambda_{\text{prior}} = (\alpha_z \Lambda_L) \oplus (\beta_z \Lambda_C) = \alpha_z \Lambda_s + \beta_z \Lambda_d, \quad (6.11)$$

where $\Lambda_s = \Lambda_L \otimes I_C$ and $\Lambda_d = I_L \otimes \Lambda_C$. The eigenvector matrix equals:

$$V_{\text{prior}} = V_L \otimes V_C = V_{B_x} \otimes V_{B_y} \otimes V_C. \quad (6.12)$$

By substituting (6.10) into (6.3), the density function of the 3D thin-membrane model becomes:

$$P(\mathbf{z}) \propto |K_{\text{prior}}|_+^{0.5} \exp\left\{-\frac{1}{2} \mathbf{z}^T K_{\text{prior}} \mathbf{z}\right\} \quad (6.13)$$

$$= |\alpha_z K_s + \beta_z K_d|_+^{0.5} \exp\left\{-\frac{1}{2} \mathbf{z}^T (\alpha_z K_s + \beta_z K_d) \mathbf{z}\right\}. \quad (6.14)$$

The specific structure of the model is extendable to any number of covariates. Specifically, the precision matrix can be generalized as:

$$K_{\text{prior}} = (\alpha K_a) \oplus (\beta K_b) \oplus (\gamma K_c) \oplus \dots \quad (6.15)$$

$$= \alpha K_a \otimes I_b \otimes I_c \otimes \dots + \beta I_a \otimes K_b \otimes I_c \otimes \dots + \gamma I_a \otimes I_b \otimes K_c \otimes \dots + \dots, \quad (6.16)$$

where $K_i \in \{K_B, K_C\}$ for $i \in \{a, b, c, \dots\}$ is the graph Laplacian matrix associated with the dependence structure of covariate i , and where α , β and γ are corresponding smoothness parameters. The eigenvalue and eigenvector matrix of K_{prior} can be computed as:

$$\Lambda_{\text{prior}} = (\alpha \Lambda_a) \oplus (\beta \Lambda_b) \oplus (\gamma \Lambda_c) \oplus \dots \quad (6.17)$$

$$= \alpha \Lambda_a \otimes I_b \otimes I_c \otimes \cdots + \beta I_a \otimes \Lambda_b \otimes I_c \otimes \cdots + \gamma I_a \otimes I_b \otimes \Lambda_c \otimes \cdots + \cdots, \quad (6.18)$$

$$= \alpha \tilde{\Lambda}_a + \beta \tilde{\Lambda}_b + \gamma \tilde{\Lambda}_c + \cdots, \quad (6.19)$$

$$V_{\text{prior}} = V_a \otimes V_b \otimes V_c \otimes \cdots, \quad (6.20)$$

where $\Lambda_i \in \{\Lambda_B, \Lambda_C\}$ and $V_i \in \{V_B, V_C\}$ are the eigenvalue and eigenvector matrix of K_i . The dependence structure of nonperiodic and periodic covariates can usually be described by chain and cycle graphs respectively, thus it is easy to calculate Λ_i and V_i as discussed in Section 6.1.

6.2.3 Posterior distribution

Let $\mathbf{y}$ denote the local estimates of $\mathbf{z}$, where $\mathbf{y}$ is either $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\sigma}}$, or $\hat{\boldsymbol{\xi}}$, and $\mathbf{z}$ denotes the true GEV parameters. We further assume that local estimates for some locations (i, j) and directions k are missing, probably due to the two reasons listed below:

1. There are insufficient observations of extremes in some direction sectors or at some sites. Note that the PWM method needs at least three samples to solve (6.7). In practice, for data collected by satellites, there are always missing parts in satellite images due to the limited satellite path or the presence of clouds. For data collected by sensors, the sensors may fail during the extreme events, resulting in missing measurements of extreme-value samples. On the other hand, the rate of occurrence of events with respect to direction in particular is non-uniform. For example, for locations sheltered by land, the number of extreme events emanating from the direction of the land will be smaller than from other directions in general.
2. There exist unmonitored sites where no observations are available. For instance, the measuring stations are often irregularly distributed across space, whereas the proposed method is more applicable to the case of regular lattice (see Fig. 9.1(a)). As a result, we can introduce unmonitored sites such that

all the sites, including both observed and unobserved ones, are located on a regular lattice. In addition, in the case of wave heights analysis, people may have particular interest in some unmonitored locations since it is easy and convenient to build offshore facilities there.

We therefore only have measurements (i.e., the local estimates $\mathbf{y}$) at a subset of variables $\mathbf{z}$ in the 3D graphical model. Furthermore, the local estimates given by the PWM method are asymptotically Gaussian distributed with respect to the sample size [125], thus motivating us to employ a Gaussian approximation. In other words, we assume that the local estimates are corrupted by Gaussian noise due to the limited number of samples available in each site and directional sector. To summarize, the local estimates are modeled as $\mathbf{y} = C\mathbf{z} + \mathbf{b}$, where $\mathbf{b} \sim N(\mathbf{0}, R_z)$ is zero-mean Gaussian random vector (Gaussian white noise) with diagonal covariance matrix R_z , and C is the selection matrix that only selects $\mathbf{z}$ at which the noisy observations y are available. C has a single nonzero value (equal to 1) in each row. If there are adequate observations available at all the locations and directions, C would simply be an identity matrix. As a consequence, the conditional distribution of the observed value $\mathbf{y}$ given the true value $\mathbf{z}$ can be written as:

$$P(\mathbf{y}|\mathbf{z}) \propto \exp\left\{-\frac{1}{2}(\mathbf{y} - C\mathbf{z})^T R_z^{-1}(\mathbf{y} - C\mathbf{z})\right\}. \quad (6.21)$$

Since we assume that the prior distribution of $\mathbf{z}$ is the 3D thin-membrane model (6.14), the posterior distribution is given by:

$$P(\mathbf{z}|\mathbf{y}) \propto P(\mathbf{z})P(\mathbf{y}|\mathbf{z}) \quad (6.22)$$

$$\propto |K_{\text{post}}|_+^{0.5} \exp\left\{-\frac{1}{2}\mathbf{z}^T K_{\text{post}} \mathbf{z} + \mathbf{z}^T C^T R_z^{-1} \mathbf{y}\right\}, \quad (6.23)$$

where $K_{\text{post}} = K_{\text{prior}} + C^T R_z^{-1} C$ is the precision matrix of the posterior distribution.

6.3 Learning and Inference

We describe here the proposed learning and inference algorithm. Concretely, we discuss how the smoothed GEV parameters, the noise covariance matrix R_z , and the smoothness parameters α_z and β_z for each of the three parameters $\boldsymbol{\mu}$, $\boldsymbol{\xi}$, and $\boldsymbol{\sigma}$ are computed.

6.3.1 Inferring smoothed GEV Parameters

Given the covariance matrix R_z and the smoothness parameters α_z and β_z , the maximum a posteriori estimate of $\mathbf{z}$ is given by:

$$\hat{\mathbf{z}} = \operatorname{argmax} P(\mathbf{z}|\mathbf{y}) = K_{\text{post}}^{-1} C^T R_z^{-1} \mathbf{y}. \quad (6.24)$$

The dimension M of K_{post} is equal to $M = PQD$. In most practical scenarios, it is intractable to compute the inverse of K_{post} due to the $\mathcal{O}(M^3)$ complexity. In the following, we explain how we can significantly reduce the computational complexity by exploiting the special configuration of the eigenvectors of K_{prior} . When the diagonal matrix $C^T R_z C$ can be well approximated by a scaled identity matrix cI_z , we have:

$$K_{\text{post}} \approx K_{\text{prior}} + cI_z \quad (6.25)$$

$$= V_{\text{prior}}^T (\alpha_z \Lambda_s + \beta_z \Lambda_d + cI_z) V_{\text{prior}} \quad (6.26)$$

As a result, $\hat{\mathbf{z}}$ can be computed as:

$$\hat{\mathbf{z}} = V_{\text{prior}}^T (\alpha_z \Lambda_s + \beta_z \Lambda_d + cI_z)^{-1} V_{\text{prior}} C^T R_z^{-1} \mathbf{y}. \quad (6.27)$$

We propose a fast thin-membrane model (FTM) solver to evaluate (6.27) in three steps.

1. Let $\mathbf{z}_0 = C^T R_z^{-1} \mathbf{y}$, and the first step computes $\mathbf{z}_1 = V_{\text{prior}} \mathbf{z}_0$. Note that $V_{\text{prior}} = V_{B_x} \otimes V_{B_y} \otimes V_C$ (6.20), thus, $\mathbf{z}_1 = V_{\text{prior}} \mathbf{z}_0$ is a three-dimensional integration transform as defined in [129]. More explicitly, we can first reshape the vector $\mathbf{z}_0$ into a $P \times Q \times D$ array Z_0 such that $\mathbf{z}_0 = \text{vec}(Z_0)$, where $\text{vec}(Z_0)$ denotes the vectorization operation. In the spatio-directional model, the three dimensions represent longitude, latitude and direction respectively. Next, we perform the fast cosine transform (FCT) in the first and second dimension (corresponding to V_{B_x} and V_{B_y}) and the fast Fourier transform (FFT) in the third one (corresponding to V_C). The detailed derivation is shown in Appendix A.1. The resulting computational complexity is $\mathcal{O}(M \log(M))$. Moreover, the computation is amenable to parallelization.
2. In the second step, $(\alpha_z \Lambda_s + \beta_z \Lambda_d + cI_z)$ is a diagonal matrix, so the complexity of the operation $\mathbf{z}_2 = (\alpha_z \Lambda_s + \beta_z \Lambda_d + cI_z)^{-1} \mathbf{z}_1$ is linear in M .
3. The operation $V_{\text{prior}}^T \mathbf{z}_2$ in the final step amounts to performing the inverse FCT and FFT in the proper dimensions of the $P \times Q \times D$ array reshaped from $\mathbf{z}_2$, and the computational effort is the same with the first step.

In summary, the computational complexity of evaluating $\mathbf{z}$ is $\mathcal{O}(M \log(M))$. A similar algorithm can easily be designed for the general case of multiple covariates. Since the generalized $V_{\text{prior}} = V_a \otimes V_b \otimes V_c \otimes \cdots$ for $V_i \in \{V_B, V_C\}$ ($i \in \{a, b, c, \dots\}$) (6.20), we can perform FFT in the i -th dimension if V_i belongs to the V_B family, and perform FCT otherwise.

When cI_z is not a good approximation to $C^T R_z C$, we decompose K_{post} into two parts K_1 and K_2 as follows:

$$K_1 = K_{\text{prior}} + cI_z, \quad K_2 = K_{\text{post}} - K_1,$$

where c is chosen as the largest entry in $C^T R_z C$. The Richardson iteration [92] is then used to solve $\mathbf{z}^{(\kappa+1)} = K_1^{-1}(C^T R_z^{-1} \mathbf{y} - K_2 \mathbf{z}^{(\kappa)})$ until convergence. In each iteration, computing the inverse of K_1 can be circumvented by using the FTM

solver mentioned above. The following two theorems guarantee the convergence of the proposed method.

Theorem 6.1. *Given $K_{post} = K_{prior} + C^T R_z^{-1} C$, $K_1 = K_{prior} + cI_z$ and $K_2 = K_{post} - K_1$, where c equals the largest diagonal element in $C^T R_z^{-1} C$,*

1. K_{post} is strictly positive definite,
2. the spectral radius $\rho(K_1^{-1} K_2) < 1$ and the resulting Richardson iterations $\mathbf{z}^{(\kappa+1)} = K_1^{-1}(C^T R_z^{-1} \mathbf{y} - K_2 \mathbf{z}^{(\kappa)})$ are guaranteed to converge to $K_{post}^{-1} C^T R_z^{-1} \mathbf{y}$.

Proof. See Appendix C.1. □

Theorem 6.2. *The convergence rate of the proposed algorithm $\rho(K_1^{-1} K_2)$ is bounded below and above by:*

$$\rho(K_1^{-1} K_2) \geq \frac{\max(C^T R_z^{-1} C) - \min(C^T R_z^{-1} C)}{\max(\Lambda_{prior}) + \max(C^T R_z^{-1} C)}, \quad (6.28)$$

$$\rho(K_1^{-1} K_2) \leq \frac{\max(C^T R_z^{-1} C) - \min(C^T R_z^{-1} C)}{\min(\Lambda_{prior}) + \max(C^T R_z^{-1} C)}, \quad (6.29)$$

where $\max(K)$ and $\min(K)$ denote the maximum and minimum diagonal element of the matrix K respectively.

Proof. See Appendix A.3. □

According to Theorem 6.2, decreasing the difference between the largest and smallest diagonal entries of matrix $C^T R_z^{-1} C$ will reduce the upper bound on $\rho(K_1^{-1} K_2)$, resulting in faster convergence. In the limit case where $\min(C^T R_z^{-1} C) = \max(C^T R_z^{-1} C)$, that is, $C^T R_z^{-1} C = cI_z$, the Richardson iteration converges in one step as in (6.27).

Note that the MAP estimate of $\mathbf{z}$ is dependent on the covariance matrix R_z and the smoothness parameters, the calculation of which is described in the sequel.

6.3.2 Estimating Covariance Matrices R_z

We use the parametric bootstrap approach to infer the (diagonal) noise covariance matrices R_μ , R_ξ and R_σ ; this method is suitable when the number of available samples is small [128]. Concretely, we proceed as follows:

1. We generate the local GEV estimates $\hat{\mu}_{ijk}$, $\hat{\sigma}_{ijk}$ and $\hat{\xi}_{ijk}$ using the method discussed in Section 6.2.1.
2. We draw M sample sets $S_1, \dots, S_M$, each with N GEV distributed samples based on the local estimates of GEV parameters $(\hat{\mu}_{ijk}, \hat{\sigma}_{ijk}, \hat{\xi}_{ijk})$. We set $M = 3000$ in our experiments.
3. For each S_m , where $m = 1, \dots, M$, we estimate the GEV parameters locally again using the method in Section 6.2.1. The resulting GEV estimates are denoted as $(\hat{\mu}_{ijk}^{[m]}, \hat{\sigma}_{ijk}^{[m]}, \hat{\xi}_{ijk}^{[m]})$.
4. The variance of $\hat{\mu}_{ijk}^{[m]}$ ($m = 1, \dots, M$) at site (i, j) and wind direction k is our estimate of the corresponding diagonal element in R_μ . Similarly, we can obtain estimates of the diagonal covariance matrices R_ξ and R_σ .

6.3.3 Learning Smoothness Parameters

Since α_z and β_z are usually unknown, we need to infer them based on the local estimates $\mathbf{y}$. However, since $\mathbf{z}$ is unknown, directly inferring α_z and β_z is impossible, and instead we solve (6.23) by Expectation Maximization (EM):

$$(\hat{\alpha}_z^{(\kappa)}, \hat{\beta}_z^{(\kappa)}) = \operatorname{argmax} Q(\alpha_z, \beta_z; \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}). \quad (6.30)$$

In Appendix A.4, we derive the Q -function:

$$Q(\alpha_z, \beta_z; \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}) \\ \propto -\text{tr} \left(K_{\text{prior}} \left(K_{\text{post}}^{(\kappa-1)} \right)^{-1} \right) - (\mathbf{z}^{(\kappa-1)})^T K_{\text{prior}} \mathbf{z}^{(\kappa-1)} + \log |K_{\text{prior}}|_+ + c, \quad (6.31)$$

$$= -c_1 \alpha_z - c_2 \beta_z + \log |K_{\text{prior}}|_+ + c, \quad (6.32)$$

where

$$c_1 = \text{tr} \left(K_s \left(K_{\text{post}}^{(\kappa-1)} \right)^{-1} \right) + (\mathbf{z}^{(\kappa-1)})^T K_s \mathbf{z}^{(\kappa-1)}, \quad (6.33)$$

$$c_2 = \text{tr} \left(K_d \left(K_{\text{post}}^{(\kappa-1)} \right)^{-1} \right) + (\mathbf{z}^{(\kappa-1)})^T K_d \mathbf{z}^{(\kappa-1)}, \quad (6.34)$$

$\mathbf{z}^{(\kappa-1)}$ is computed as in (8.4) with $\hat{\alpha}_z^{(\kappa-1)}$ and $\hat{\beta}_z^{(\kappa-1)}$ obtained from the previous iteration, and c stands for all the unrelated terms. In (6.31), we need to evaluate $\log |K_{\text{prior}}|_+$. Since K_{prior} can be regarded as a generalized Laplacian matrix corresponding to the connected graph of the 3D thin-membrane model, according to the properties of Laplacian matrices, $|K_{\text{prior}}|_+ = M \det S(K_{\text{prior}})$, where $S(K_{\text{prior}})$ denotes the first $M-1$ rows and columns of the $M \times M$ matrix K_{prior} and $S(K_{\text{prior}})$ is positive definite [130]. As a result,

$$\log |K_{\text{prior}}|_+ = \log \det S(K_{\text{prior}}) + \log M, \quad (6.35)$$

$$= \log \det(\alpha_z S(K_s) + \beta_z S(K_d)) + c. \quad (6.36)$$

Taking the partial derivatives of Q function with regard to α_z and β_z , we can obtain:

$$\frac{\partial Q}{\partial \alpha_z} = -c_1 + \text{tr}(S(K_{\text{prior}})^{-1} S(K_s)), \quad (6.37)$$

$$\frac{\partial Q}{\partial \beta_z} = -c_2 + \text{tr}(S(K_{\text{prior}})^{-1} S(K_d)), \quad (6.38)$$

where the Jacobi's formula is applied, i.e.,

$$\frac{\partial \log \det K}{\partial x} = \text{tr} \left(K^{-1} \frac{\partial K}{\partial x} \right). \quad (6.39)$$

By equating the partial derivatives (6.37) and (6.38) to zero, we have the following two identities:

$$c_1 = \text{tr}(S(K_{\text{prior}})^{-1}S(K_s)), \quad (6.40)$$

$$c_2 = \text{tr}(S(K_{\text{prior}})^{-1}S(K_d)). \quad (6.41)$$

As a consequence, we obtain:

$$\begin{aligned} \alpha c_1 + \beta c_2 &= \alpha \text{tr}(S(K_{\text{prior}})^{-1}S(K_s)) + \beta \text{tr}(S(K_{\text{prior}})^{-1}S(K_d)) \\ &= \text{tr}\{S(K_{\text{prior}})^{-1}(\alpha_z S(K_s) + \beta_z S(K_d))\} \end{aligned} \quad (6.42)$$

Recall that $\alpha_z S(K_s) + \beta_z S(K_d) = S(K_{\text{prior}})$, and therefore,

$$\alpha c_1 + \beta c_2 = M - 1. \quad (6.43)$$

By substituting (6.43) into the Q -function (6.31), it follows:

$$Q(\alpha_z, \beta_z; \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}) = \log |K_{\text{prior}}|_+ - (M - 1) + c. \quad (6.44)$$

At this point, the expression (6.30) can be succinctly formulated as:

$$\begin{aligned} (\alpha_z^{(\kappa)}, \beta_z^{(\kappa)}) &= \text{argmax} \log |K_{\text{prior}}|_+, \\ \text{s.t. } c_1 \alpha_z + c_2 \beta_z &= M - 1, \quad \alpha_z \geq 0, \quad \beta_z \geq 0, \end{aligned} \quad (6.45)$$

where the constants c_1 and c_2 are computed using (6.33) and (6.34) respectively. Since the eigenvalue matrix Λ_{prior} of K_{prior} equals $\Lambda_{\text{prior}} = \alpha_z \Lambda_s + \beta_z \Lambda_d$ as in (6.19),

Table 6.1: The learning and inference algorithm for modeling multiple covariates.

<ol style="list-style-type: none"> 1. Estimate the GEV parameters locally using the combination of PWM and MED estimator. 2. Approximate the relation between locally fitting and true GEV parameters by a Gaussian model, that is, $\mathbf{y} = C\mathbf{z} + \mathbf{b}$. Estimate the diagonal noise covariance R_z using the parametric bootstrap approach. 3. Initialize the smoothness parameters $\hat{\alpha}_z^{(0)}$ and $\hat{\beta}_z^{(0)}$. Iterate the following steps till convergence: <ol style="list-style-type: none"> (a) E-step: update the MAP estimates of GEV parameters z using the methods described in Section 6.3.1: $\hat{\mathbf{z}}^{(\kappa-1)} = \left\{ K_{\text{post}}^{(\kappa-1)} \right\}^{-1} C^T R_z^{-1} \mathbf{y},$ <p>where $K_{\text{post}}^{(\kappa-1)} = K_{\text{prior}}^{(\kappa-1)} + C^T R_z^{-1} C = (\hat{\alpha}_z^{(\kappa-1)} K_a) \oplus (\hat{\beta}_z^{(\kappa-1)} K_b) \oplus (\hat{\xi}_z^{(\kappa-1)} K_c) \oplus \dots + C^T R_z^{-1} C$. The last expression follows from (6.15).</p> (b) M-step: update the estimate of smoothness parameters by solving (6.47).

we can further simplify the objective function (6.45) as:

$$\log |K_{\text{prior}}|_+ = \log |\Lambda_{\text{prior}}|_+ = \sum_{k: \lambda_{sk} + \lambda_{dk} > 0} \log(\alpha_z \lambda_{sk} + \beta_z \lambda_{dk}), \quad (6.46)$$

Consequently, the constrained convex optimization problem (6.45) can be solved efficiently via the bisection method [131].

In the scenario of multiple covariates, (6.45) can be extended as:

$$(\alpha_z^{(\kappa)}, \beta_z^{(\kappa)}, \xi_z^{(\kappa)}, \dots) = \operatorname{argmax} \sum_k \log(\alpha_z \tilde{\lambda}_{ak} + \beta_z \tilde{\lambda}_{bk} + \gamma_z \tilde{\lambda}_{ck} + \dots), \quad (6.47)$$

s.t. $c_1 \alpha_z + c_2 \beta_z + c_3 \xi_z + \dots = M - 1, \alpha_z \geq 0, \beta_z \geq 0, \gamma_z \geq 0, \dots$

where $\tilde{\lambda}_{ik} (i \in \{a, b, c, \dots\})$ is the k -th diagonal element of $\tilde{\lambda}_i$ in (6.19). The overall learning and inference algorithm for the generalized model is summarized in Table 6.1.

6.3.4 Bootstrapping the uncertainty of GEV estimates

In addition to the point estimates of GEV parameters, we also have particular interest in the uncertainty of the estimates. As demonstrated in [42], nonparametric bootstrapping provides a reliable tool to quantify the uncertainty associated with extreme-value models. The bootstrap procedure consists of the following steps:

1. Generate a large number M of sample sets $S_1, \dots, S_m$, each with N occurrences, by resampling at random with replacement from the original N observations. $M = 1000$ in our experiments.
2. For each S_m , where $m = 1, \dots, M$, apply the algorithm in Table 6.1 to yield smoothed GEV parameters $z^{(m)}$, which denotes either ξ , σ , or μ .
3. The 95% confidence interval for GEV estimates is computed as the values corresponding to the 2.5% and 97.5% quantiles of $z^{(m)}$ with $m = 1, \dots, M$.

6.4 Numerical Results

In this section, we test the proposed spatio-directional model (SDM) on both synthetic and real data against 5 other models, that is, a locally fit model (LFM) (i.e., assuming $\alpha_z = 0$ and $\beta_z = 0$), a directionally constant model (DCM) (assuming GEV parameters to be constant across different directions, i.e., $\beta_z = \infty$), a directionally independent model (DIM) (only considering possible spatial dependence and assuming GEV parameters corresponding to different directional sectors are independent, i.e., $\beta_z = 0$), a spatially constant model (SCM) (i.e., $\alpha_z = \infty$), and a spatially independent model (SIM) (i.e., $\alpha_z = 0$).

6.4.1 Synthetic Data

Here we draw samples from GEV distributions with parameters that depend on both location and direction. Concretely, we select 256 sites arranged in a two-dimensional

Figure 6.2: Estimates of scale parameter σ across all sites in one directional sector.

16 $\times$ 16 lattice. We then discretize the direction into 15 directional sectors. Next, we predefine GEV parameters for each site and each direction sector. More explicitly, we characterize the spatial dependence and directional dependence by a quadratic polynomial and a Fourier series expansion respectively, as suggested in [47] and [44]. Finally, we randomly generate 50 GEV distributed occurrences for each site and direction sector.

Our results are listed in Fig. 6.2, Fig. 6.3 and Table 6.2. Fig. 6.2 shows the scale parameters estimated by the aforementioned 6 models across space, while Fig. 6.3(a) shows the estimated shape parameters with 95% confidence interval across different wind directions. We further magnify the section in the black square in Fig. 6.3(a) and show it in Fig. 6.3(b). For other GEV parameters in the spatial or directional domain, the results are qualitatively similar, so we omit them. We can see that estimates resulting from the proposed SDM follow the ground truth closely, across both space and wind direction. On the contrary, local estimates (resulting from the LFM) fluctuate substantially and exhibit the largest uncertainty among the 6 models, since the limited number of occurrences at each location and direction is insufficient to yield reliable estimates of GEV parameters.

Figure 6.3: (a) Estimates of the shape parameters (solid lines) with 95% confidence intervals (dashed lines) across different directions at a randomly selected site; (b) magnified image corresponding to the part in the black square.

Next, let us focus on the DCM and the SCM, which ignore one of the two covariates. The DCM is able to infer the varying trend of GEV parameters across space, but generates biased results; specifically, it overestimates the scale parameters (see Fig. 6.2(c)). Moreover, the DCM mistakenly ignores the directional variation of GEV parameters as shown in Fig. 6.3(a). Similarly, the SCM can capture the overall trend across different directions (cf. Fig. 6.3(a)) but fails to model the spatial variation (cf. Fig. 6.2(e)). Note that the confidence intervals of the DCM and the SCM are narrower than that of the SDM due to the larger sample size by combining the data from different directions (or sites) after ignoring the possible variation.

On the other hand, the DIM and the SIM perform much better, yielding similar results to that of the SDM. This implies the necessity of accommodating the heterogeneity in the data. However, since these two models only capture one of the two types of dependence (i.e., directional and spatial dependence) and mistakenly assume the other independent, we can find in Fig. 6.3(b) that the DIM estimates fluctuate more across direction, and in Fig. 6.2(f) that the estimates across space given by the SIM are not as smooth as that of the SDM.

Table 6.2 summarizes the overall mean square error (MSE) for each of the three GEV parameters and the Akaike Information Criterion (AIC) of model fitting. The

Figure 6.4: Mean square error (MSE) of the GEV estimates as a function of sample size (averaged over 100 data sets).

Table 6.2: Quantitative comparison of different models

Models	Mean Square Error (MSE)			AIC
	ξ	σ	μ	
LFM	0.0119	0.1002	0.1539	1.0327×10^6
DCM	0.0090	0.0626	0.3659	9.0111×10^5
DIM	0.0021	0.0276	0.0505	8.7767×10^5
SCM	0.1033	0.6405	3.9919	1.0065×10^6
SIM	0.0026	0.0235	0.0487	8.9400×10^5
SDM	0.0016	0.0192	0.0436	8.7524×10^5

proposed model yields the smallest MSE and AIC score. The latter implies that the SDM fits the data best notwithstanding the penalty on the effective number of parameters. Note that the effective number of parameters (i.e. the degree of freedom) in the AIC score can be computed as $\text{tr}\{(C^T R_z^{-1} C + \alpha_z K_s + \beta_z K_d)^{-1} C^T R_z^{-1} C\}$ in the proposed spatio-directional model, according to the definition given in [132] and [133]. By comparing the SDM with the other 5 models, we can tell that the proposed model can capture the proper amount of spatial and directional dependence in an automatic manner.

Next, we investigate the impact of sample size on the 6 models. In this set of experiments, we consider the MSE of the GEV estimates for varying sample size. Concretely, we consider sample size 10, 30, 50, 100, 150, and 200 per location per

Figure 6.5: MSE for varying proportion of observed sites (averaged over 100 trials). The MSE decreases with increasing number of observed sites, as expected.

directional bin. For each sample size, we generate 100 data sets with the same parameterization as before. We then compute the MSE of GEV estimates averaged over the 100 sets, as shown in Fig. 6.4. The proposed SDM usually performs the best. The only exception is for the scale parameter when the number of samples is 10. In this case, the DCM slightly outperforms the SDM, whereas the SDM generates significantly better results in other cases. The proposed model yields reasonably accurate estimates even when the sample size is as small as 10, which demonstrates the utility of the proposed model in the case of small sample size.

Finally, we test the performance of the proposed model when dealing with randomly distributed unmonitored sites and wind directions, i.e., sites and directional sectors for which no observations are available. The selection matrix C is not an identity matrix in this case. We depict in Fig. 7.8 the MSE for each GEV parameter and for the observed and unobserved variables respectively as a function of the percentage of missing variables across 100 trials. We can see that the MSE increases with the number of unmonitored sites and directional sectors, in agreement with our expectation. However, the MSE is still small even when only 10% variables are observed in the 3D graphical models. In conclusion, the proposed model can successfully tackle missing data.

(a) Distribution of extreme wave heights at 20 randomly selected sites.

(b) Scatter plot of wave height *w.r.t* direction.

Figure 6.6: Heterogeneity in GOMOS data. The wave heights clearly depend on the location in the Gulf of Mexico and the wind direction.

6.4.2 Real Data

We now investigate the extreme wave heights in the Gulf of Mexico. Such analysis could be beneficial when constructing oil platforms in the gulf to withstand extreme waves. The GOMOS (Gulf of Mexico Oceanographic Study) data [134] is used in this study, which cover the period from September 1900 to September 2005 inclusive, at thirty-minute intervals. It measures the significant wave heights. The hindcasts are produced by a physical model, calibrated to observed hurricane data. We isolate 315 maximum peak wave height values; each corresponds to a hurricane event in

Figure 6.7: Scatter plots of wave height at pairs of sites.

the Gulf of Mexico. We also extract the corresponding vector mean direction of the wind at the time of the peak significant wave height. We then select 176 sites arranged on a 8×22 lattice with spacing 0.5° (approximately 56km), which almost covers the entire U.S. Gulf of Mexico.

An initial analysis (as shown in Fig. 6.6) shows the heterogeneity of the data *w.r.t.* location and direction. We can also see from Fig. 6.6 that the distribution of storm-wise maxima at each site and direction has a heavy tail, supporting the use of the GEV marginals. We further depict in Fig. 6.7 the scatter plot of extreme wave heights from two neighboring sites and two distant sites in the lattice. Strong spatial dependence exists between two nearby sites, however, the dependence is clearly weaker between sites that are far apart. This observation provides support for the choice of thin-membrane models, since the latter only capture the direct dependence between neighbors. In summary, the preliminary study suggests that the proposed model is well suited for this data set.

We next apply the 6 models to the data. To test the influence of the selected number of directional sectors on the results, we consider different numbers of sectors, i.e., 6, 8, 10, 12, 15 and 18 in sequence. We then compute the AIC score of the 6 models for each number of directional sectors. The results are summarized in Fig. 6.8.

Again, the SDM always achieves the best AIC score. Moreover, the performance of

Figure 6.8: AIC score as a function of the selected number of directional sectors.

this model is not sensitive to the chosen number of directional sectors. In practice, we propose to choose the number that minimizes the AIC score, which is 12 for the GOMOS data. In this case, the estimated shape parameters ξ_{ijk} resulting from the SDM ranges from -0.30 to 0.34 , hence, all the three classes of extreme-value distributions are employed to describe the extreme wave heights. Interestingly, the DIM performs well when the number of directional sectors is small (≤ 10). However, the performance deteriorates significantly as the number of directional sectors increases. This indicates the need to capture the directional dependence, especially when there is a relatively large number of directional sectors and thus the directional dependence between neighboring sectors is strong. On the other hand, the SIM fails to model the spatial dependence as shown in Fig. 7.10(a). In contrast with the SIM, the SCM mistakenly ignores the spatial variation, which is essential to model the extreme wave heights in the Gulf of Mexico [47] (see Fig 6.6(a) and 7.10(b)). Similarly, the DCM does not properly consider the strong directional variation visible in Fig. 6.6(b), and therefore leads to the highest AIC score. Finally, the LFM overfits the data by introducing too many parameters. This can be concluded from the fact that its AIC increases with the number of directional sectors. Taken together, it is evident that the proposed SDM is preferred for this GOMOS data since it models

the proper amount of dependence across space and wind direction simultaneously.

6.5 Summary

In this chapter, we proposed a novel extreme-value graphical model to accommodate the effects of multiple covariates. More explicitly, we assume that marginal extreme values follow GEV distributions. The GEV parameters are then coupled together through a multidimensional thin-membrane model. The multidimensional thin-membrane model can be constructed flexibly from one-dimensional thin-membrane models, and thus, the proposed model is generalizable to incorporate any number of covariates. Furthermore, the special eigendecomposition corresponding to thin-membranes simplifies the learning and inference process. As a result, the proposed method scales gracefully with the problem size. Numerical results for both synthetic and real data support the proposed model; it achieves the most accurate estimates of GEV parameters, and models the data best. Therefore, the approach may prove to be a practical tool for modeling extreme events with covariates.

Chapter 7

Modeling Spatio-Temporal Extreme Events using Graphical Models

In this chapter, we construct a spatio-temporal model for marginal extreme events. Such model usually involves thousands or even millions of variables in the spatio-temporal domain, whereas only one single observation is available for each location and time point. Previous works usually employ learning and inference methods that are computationally burdensome, and therefore are prohibitive for large-scale data. Moreover, they assume that the shape and the scale parameters of the extreme-value distributions are constant across the spatio-temporal domain, which is often too restrictive in practice. By contrast, we aim to exploit undirected graphical models to describe the highly structured dependencies among all three GEV parameters (the shape, scale, and location parameter) across both space and time, and to further develop efficient learning and inference algorithms that are applicable to large datasets.

To move forward to this goal, we first assume that the single monthly maximum at each site and time position (i.e., each month) follows a GEV distribution. We further stipulate that each of the three GEV parameters (the shape, scale, and location

parameter) corresponding one site and time point consists of two components: a spatial and a temporal component. The proposed model is therefore similar in spirit to the generalized additive models that are popular in the literature of extreme events modeling, cf. [55, 57, 51]. Next, we impose Gaussian graphical model priors, particularly, thin-plate model priors, respectively on the spatial components across space and the time components across time. The amount of dependence is then determined by the smoothness parameters of the thin-plate models. In order to infer all the parameters, we follow the empirical Bayes approach: we generate point estimates of the smoothness parameters while inferring the posterior distribution of the GEV parameters. Specifically, we approximate the posterior distribution of the GEV parameters by a variational multivariate Gaussian distribution with a diagonal covariance, and exploit efficient stochastic optimization methods [135, 136] to learn both the smoothness parameters and the parameters of the variational distribution. As only noisy gradients (rather than the exact ones) are required in each iteration of the stochastic variational inference algorithm, the computational complexity can be reduced to be sublinear in the number of variables. Numerical results show that the proposed model can automatically recover the underlying pattern of GEV parameters across both space and time, given one single sample observed at each location and time point. Moreover, it also provides an effective tool to predict the future distribution of extreme events.

The chapter is organized as follows. In Section 7.1, we review thin-plate models, since those models play a central role in our approach. In Section 7.2, we present the proposed spatio-temporal model for extreme events in detail. The efficient stochastic variational inference algorithm is then developed in Section 7.3. We also explain how to predict the distribution of extreme events in the future in this section. In Section 7.4, we assess the proposed model and benchmark it with other models by means of synthetic and real data. We conclude in Section 7.5 with a discussion.

7.1 Thin-Plate Models

In this section, we first give a short description of thin-plate models. In particular, we introduce thin-plate models with zero curvature and zero gradient boundary conditions respectively. The former is useful to predict future trend while the latter can deal with multi-dimensional data. As a result, we utilize the former to capture temporal dependence among GEV parameters and the latter for modeling spatial dependence.

In an undirected graphical model (i.e., a Markov random field), the probability distribution is represented by an undirected graph $\mathcal{G}$ which consists of P nodes $\mathcal{V}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. Each node i is associated with a random variable z_i . An edge (i, j) is absent if the corresponding two variables z_i and z_j are conditionally independent: $P(z_i, z_j | \mathbf{z}_{\mathcal{V} \setminus \{i, j\}}) = P(z_i | \mathbf{z}_{\mathcal{V} \setminus \{i, j\}})P(z_j | \mathbf{z}_{\mathcal{V} \setminus \{i, j\}})$, where $\mathbf{z}_{\mathcal{V} \setminus \{i, j\}}$ denotes all the variables except z_i and z_j .

The thin-plate model [77, 137] is a GMRF that is commonly used as smoothness prior as it penalizes the second-order difference. In other words, we model the second-order differences as a Gaussian distribution:

$$\Delta^2 z_i \sim \mathcal{N}(0, \alpha_z^{-1}). \quad (7.1)$$

For a one-dimensional problem where the variables z_i 's are evenly located on a chain, the second-order difference at z_i can be defined as $\Delta^2 z_i = z_{i-1} - 2z_i + z_{i+1}$. As a result, the density function of a thin-plate model with a chain structure can be written as [77]:

$$p(\mathbf{z}) \propto \exp \left\{ -\frac{\alpha_z}{2} \sum_{i=2}^{P-1} (z_{i-1} - 2z_i + z_{i+1})^2 \right\} \quad (7.2)$$

$$= \exp \left\{ -\frac{\alpha_z}{2} \mathbf{z}^T K_{\text{tp}} \mathbf{z} \right\}, \quad (7.3)$$

where the smoothness parameter α_z controls the curvature, and K_{tp} has the following

form:

$$K_{\text{tp}} = A^T A, \quad (7.4)$$

$$A = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix}. \quad (7.5)$$

Note that A is a $P - 2 \times P$ matrix. Apparently, the precision of a thin-plate model $K = \alpha_z K_{\text{tp}}$. It is easy to tell from (7.2) that the thin-plate model is invariant to the addition of a constant, and more importantly, a linear function along the Markov chain. In other words, $K_{\text{tp}} \mathbf{1} = \mathbf{0}$ and $K_{\text{tp}} \mathbf{s}_1 = \mathbf{0}$, where $\mathbf{1}$ is a column vector of all ones, and $\mathbf{s}_1 = [1, 2, \dots, P]^T$. As such, this prior can accommodate the linear trends without penalty. We can also conclude that K_{tp} is rank deficient with two zero eigenvalues. As a result, the improper density is often used in practice [77, 123], that is,

$$p(\mathbf{z}) \propto |K|_+^{0.5} \exp \left\{ -\frac{1}{2} \mathbf{z}^T K \mathbf{z} \right\}, \quad (7.6)$$

where $|K|_+$ denotes the product of the positive eigenvalues of the precision matrix K . We can also read from (7.4) that the conditional mean of one variable z_i conditioned on other variables $\mathbf{z}_{\mathcal{V}|i}$ is [77]:

$$E(z_i | \mathbf{z}_{\mathcal{V}|i}) = \frac{4}{6}(z_{i+1} + z_{i-1}) - \frac{1}{6}(z_{i+2} + z_{i-2}), \quad (7.7)$$

which can be regarded as second-order polynomial interpolation based on four nearby variables z_{i-2} , z_{i-1} , z_{i+1} , and z_{i+2} without an overall level. Therefore, the thin-plate model allows the deviation from any overall mean level without having to specify the overall mean level itself. Such property is often favored in practice. Furthermore, the zero curvature boundary condition of (7.2) (i.e., $\Delta^2 z_1 = \Delta^2 z_P = 0$)

aids in predicting (or extrapolating) future values [77], i.e.,

$$E(z_{P+1}|z_1, \dots, z_P; \alpha_z) = z_P + (z_P - z_{P-1}). \quad (7.8)$$

Therefore, the conditional mean of z_{P+1} is simply the linear extrapolation based on the last two observations z_{P-1} and z_P . Such models will be exploited to model temporal trend of GEV parameters.

A zero gradient boundary condition is also often applied in thin-plate models, i.e., $z_0 = z_1$ and $z_{P+1} = z_P$. Consequently, we can simplify the second-order difference at the boundary variable z_1 and z_P respectively as:

$$\Delta^2 z_1 = z_0 - 2z_1 + z_2 = z_2 - z_1, \quad (7.9)$$

$$\Delta^2 z_P = z_{P-1} - 2z_P + z_{P+1} = z_{P-1} - z_P. \quad (7.10)$$

Hence, the resulting thin-plate model with constant boundary condition is [137]:

$$p(\mathbf{z}) \propto \exp \left\{ -\frac{\alpha}{2} \sum_{i=1}^P (|N(i)| z_i - \sum_{j \in N(i)} z_j)^2 \right\}, \quad (7.11)$$

where $N(i)$ denotes the neighboring nodes of z_i and $|N(i)|$ is the number of neighbors z_i have. In (7.11), each node is modeled to be close to the average of its neighbors. Note that the resulting K_{tp} has rank $P - 1$. This model can be easily extended to the case of multiple dimensions, and coincides with the boundary conditions proposed in [138] to address the problem of extending (7.2) to a two-dimensional spatial domain. Moreover, it is shown in [139] that such models perform better when modeling spatial dependence of spatial data than thin-membrane models that penalize gradient (cf. Section 6.1). As a result, we will employ this type of thin-plate models to capture spatial dependence among GEV parameters in the sequel.

7.2 Spatio-Temporal Models for Extreme Events

In this section, we present the proposed spatio-temporal graphical model for extreme events. Suppose that we have N block maxima x_{ij} at each of the P locations, where $i = 1, \dots, N$ and $j = 1, \dots, P$. The resulting number of dimensions of the spatio-temporal model is $D = NP$. We further assume the observations are missing at random, in order to demonstrate that the proposed model is capable of dealing with missing data. The set of observed spatio-temporal indices is denoted as $\mathcal{V}_O$.

7.2.1 Likelihood: Generalized Extreme Value Distributions

Motivated by the extreme value theory, we assume that each observed x_{ij} follows a Generalized Extreme Value (GEV) distribution with cumulative distribution function (CDF) [41]:

$$F(x_{ij}|\xi_{ij}, \sigma_{ij}, \mu_{ij}) = \exp \left\{ - \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\sigma_{ij}} \right) \right]^{-\frac{1}{\xi_{ij}}} \right\}, \quad (7.12)$$

where $\mu_{ij} \in \mathbb{R}$ is the location parameter, $\sigma_{ij} > 0$ is the scale parameter and $\xi_{ij} \neq 0$ is the shape parameter. We further use $\zeta_{ij} = \log \sigma_{ij}$ to parameterize the GEV distribution such that $\zeta_{ij} \in \mathbb{R}$ and a Gaussian prior can be imposed on ζ_{ij} . Taking derivatives of (7.12) with regard to x_{ij} yields the probability density function (PDF):

$$\begin{aligned} f(x_{ij}|\xi_{ij}, \zeta_{ij}, \mu_{ij}) &= \frac{1}{\exp(\zeta_{ij})} \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}-1} \\ &\quad \cdot \exp \left\{ - \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}} \right\}. \end{aligned} \quad (7.13)$$

Note that the support of both (7.12) and (7.13) is $\{x_{ij} : 1 + \xi_{ij}(x_{ij} - \mu_{ij})/\exp(\zeta_{ij}) > 0\}$. For $\xi_{ij} = 0$, the above functions are undefined, and thus are replaced by the results obtained by taking the limit as $\xi_{ij} \rightarrow 0$. In this case, the resulting CDF and

PDF are:

$$F(x_{ij}|0, \zeta_{ij}, \mu_{ij}) = \exp \left[-\exp \left(-\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right], \quad (7.14)$$

$$f(x_{ij}|0, \zeta_{ij}, \mu_{ij}) = \frac{1}{\exp(\zeta_{ij})} \exp \left(-\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \exp \left\{ -\exp \left(-\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right\}, \quad (7.15)$$

where x_{ij} is defined in $\mathbb{R}$.

Note that the key parameter of GEV distributions is the shape parameter, which determines the subfamily. Specifically, $\xi_{ij} = 0$ yields Gumbel distributions with light upper tails, $\xi_{ij} > 0$ corresponds to Fréchet distributions with heavy upper tails, while $\xi_{ij} < 0$ corresponds to Weibull distributions with bounded upper tails.

7.2.2 Prior: Thin-plate Models

We now turn our attention to the prior distributions of the GEV parameters. Since the three parameters share the same dependence structure, we present them in a unified form. Let z_{ij} denote either ξ_{ij} , ζ_{ij} or μ_{ij} . We assume that each parameter z_{ij} at time instant i and site j can be decomposed as $z_{ij} = z_{Ti} + z_{Sj}$, where z_{Ti} is the temporal component at time instant i and z_{Sj} is the spatial component at site j . Note that we construct the model in a similar fashion to the generalized additive models that have seen broad applications in the literature of extreme value analysis due to their simplicity, flexibility and utility, cf. [55, 57, 51] and references therein. However, different from the generalized additive models that are often decomposed as the deterministic (spline) functions for the spatial and the temporal component plus a noise term, we only put smoothness priors on the spatial and temporal component without specifying their functional form. Thus, the resulting model can be more flexible. In addition, we include the noise term implicitly in $\mathbf{z}_S$ and $\mathbf{z}_T$. In the following, we describe the priors on $\mathbf{z}_S = [z_{S1}, \dots, z_{SP}]^T$ and $\mathbf{z}_T = [z_{T1}, \dots, z_{TN}]^T$ individually. Such priors are constructed according to the highly structured dependence between the GEV parameters.

Figure 7.1: Neighborhood structure of the graphical model for spatial components.

Without loss of generality, we assume that the measuring stations are deployed on a regular lattice as shown in Fig. 7.1. As a result, we employ the thin-plate model with zero gradient boundary condition (7.11) to capture the spatial dependence among the sites (variables) in the spatial component $\mathbf{z}_S$:

$$\begin{aligned} p(\mathbf{z}_S | \alpha_z) &\propto \exp \left\{ -\frac{\alpha_z}{2} \sum_{i=1}^P (|N(i)| z_{Si} - \sum_{j \in N(i)} z_{Sj})^2 \right\} \\ &\propto |\alpha_z K_S|_+^{0.5} \exp \left\{ -\frac{\alpha_z}{2} \mathbf{z}_S^T K_S \mathbf{z}_S \right\}. \end{aligned} \quad (7.16)$$

Here, $N(i)$ typically includes four neighbors (two vertical and two horizontal) of node i .

Next, we consider the temporal dependence. We first deal with the periodicity and the trend separately, and then integrate them together to construct the temporal graphical model. More specifically, we partition $\mathbf{z}_T$ according to the period as $(z_{T1}, z_{T2}, \dots, z_{T\tau}), (z_{T\tau+1}, z_{T\tau+2}, \dots, z_{T2\tau}), \dots$, where τ is the period. Since we focus on block maxima, τ is automatically determined by the block size. For example, if we analyze monthly maxima, $\tau = 12$, whereas for seasonal maxima, $\tau = 4$. For variables in each group, e.g., $(z_{T1}, z_{T2}, \dots, z_{T\tau})$, we couple them together via a cycle graph as shown in Fig. 7.2(a) to accommodate the periodicity in the time series. In this case, the thin-plate model with zero curvature and zero gradient boundary condition takes on the same form; we use $\beta_z K_{pr}$ to denote the precision matrix of the thin-plate model, where β_z is the smoothness parameter. On the other hand, we capture the trend by coupling $z_{Ti}, z_{Ti+\tau}, z_{Ti+2\tau}, \dots$ together via a chain

Figure 7.2: (a) cycle graph to describe periodicity; (b) chain graph to describe trend; (c) neighborhood structure of the graphical model for temporal components.

graph (see Fig. 7.2(b)), for $i = 1, \dots, \tau$. Here, we utilize the thin-plate model with zero curvature boundary condition for the sake of future forecast, and represent the corresponding precision matrix as $\gamma_z K_{tr}$. As a consequence, the neighborhood structure of the temporal model can be specified as in Fig. 7.2(c), and the overall precision matrix is given by:

$$K_T = (\gamma_z K_{tr}) \oplus (\beta_z K_{pr}) \quad (7.17)$$

$$= \gamma_z K_{tr} \otimes I_{pr} + \beta_z I_{tr} \otimes K_{pr} \quad (7.18)$$

$$= \gamma_z \tilde{K}_{tr} + \beta_z \tilde{K}_{pr}, \quad (7.19)$$

where $\oplus$ and $\otimes$ denote the Kronecker sum and Kronecker product respectively, I_* is an identity matrix with the same dimension as K_* , $\tilde{K}_{pr} = I_{pr} \otimes K_{pr}$ characterizes the dependence within each period (corresponding to the blue edges in Fig. 7.2(c)), and $\tilde{K}_{tr} = K_{tr} \otimes I_{tr}$ characterizes the dependence between contiguous periods (corresponding to the green edges in Fig. 7.2(c)).

As mentioned in Section 7.1, thin-plate models do not specify the overall mean level for $\mathbf{z}_S$ and $\mathbf{z}_T$, and thus, there can be an infinite number of combinations (z_{Ti}, z_{Sj}) with $z_{ij} = z_{Ti} + z_{Sj}$ unchanged. To remedy the problem, we explicitly add a constraint in the prior of $\mathbf{z}_T$ that $\sum_i z_{Ti} = 0$. Taken together, the prior density of

$\mathbf{z}_T$ can be written as:

$$p(\mathbf{z}_T|\beta_z, \gamma_z) \propto |\gamma_z \tilde{K}_{tr} + \beta_z \tilde{K}_{pr}|_+^{0.5} \exp \left\{ -\frac{1}{2} \mathbf{z}_T^T \left(\gamma_z \tilde{K}_{tr} + \beta_z \tilde{K}_{pr} + \mathbf{1}\mathbf{1}^T \right) \mathbf{z}_T \right\}, \quad (7.20)$$

where $\mathbf{1}$ is a column vector of all ones. Recall that the eigenvalue of K_T corresponding to eigenvector $\mathbf{1}$ is 0 (see Section 7.1). By adding $\mathbf{1}\mathbf{1}^T$ to K_T , we only modify the eigenvalue to 1 with other eigenvalues unchanged. Therefore, $|\gamma_z \tilde{K}_{tr} + \beta_z \tilde{K}_{pr}|_+$ equals $|\gamma_z \tilde{K}_{tr} + \beta_z \tilde{K}_{pr} + \mathbf{1}\mathbf{1}^T|_+$ in Eq. (7.20).

7.2.3 Spatio-Temporal Graphical Models for Extreme Values

The joint PDF of the overall spatio-temporal model can be written as:

$$\begin{aligned} & p(\mathbf{x}, \boldsymbol{\xi}_S, \boldsymbol{\xi}_T, \boldsymbol{\zeta}_S, \boldsymbol{\zeta}_T, \boldsymbol{\mu}_S, \boldsymbol{\mu}_T | \alpha_\xi, \beta_\xi, \gamma_\xi, \alpha_\zeta, \beta_\zeta, \gamma_\zeta, \alpha_\mu, \beta_\mu, \gamma_\mu) \\ &= \prod_{\{i,j\} \in \mathcal{V}_O} f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \prod_{z \in \{\xi, \zeta, \mu\}} p(\mathbf{z}_S | \alpha_z) p(\mathbf{z}_T | \beta_z, \gamma_z), \end{aligned} \quad (7.21)$$

where $f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj})$ is the GEV density function (i.e., the likelihood of the GEV parameters, cf. (7.13) and (7.15)) introduced in Subsection 7.2.1.

Since the GEV densities are non-Gaussian, the resulting overall graphical model is non-Gaussian as well. Therefore, we exploit variational inference methods to estimate both the GEV parameters and the smoothness parameters given observed extreme values $\mathbf{x} = [x_{ij}]^T$, which is explained in the next section.

7.3 Learning and Inference

In this section, we first elaborate on how to learn both the GEV and smoothness parameters given the extreme-value observations. We then employ the model to predict future GEV distributions.

7.3.1 Learning GEV and Smoothness Parameters

Here, we estimate all the parameters through an empirical Bayes approach [140]. Specifically, we infer the smoothness parameters by maximum likelihood estimation. Let $\mathbf{y} = [\boldsymbol{\xi}_S; \boldsymbol{\xi}_T; \boldsymbol{\zeta}_S; \boldsymbol{\zeta}_T; \boldsymbol{\mu}_S; \boldsymbol{\mu}_T]$ and $\boldsymbol{\theta} = [\alpha_\xi; \beta_\xi; \gamma_\xi; \alpha_\zeta; \beta_\zeta; \gamma_\zeta; \alpha_\mu; \beta_\mu; \gamma_\mu]$. The likelihood is given by:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int_{\mathbf{y}} p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) d\mathbf{y} = \int_{\mathbf{y}} p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y}, \quad (7.22)$$

where $p(\mathbf{x}|\mathbf{y})$ are the GEV densities, and $p(\mathbf{y}|\boldsymbol{\theta})$ are the thin-plate model priors. Since maximizing $p(\mathbf{x}|\boldsymbol{\theta})$ (7.22) directly is intractable, we instead find $q(\mathbf{y})$ and $\boldsymbol{\theta}$ to maximize the lower bound of $\log p(\mathbf{x}|\boldsymbol{\theta})$:

$$L = \int_{\mathbf{y}} q(\mathbf{y}) \log \frac{p(\mathbf{x}|\mathbf{y}) p(\mathbf{y}|\boldsymbol{\theta})}{q(\mathbf{y})} d\mathbf{y} \leq \log p(\mathbf{x}|\boldsymbol{\theta}). \quad (7.23)$$

Ideally, we choose $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ such that the lower bound L is maximized. However, in our case, we cannot obtain the closed-form expression of the posterior distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ due to the complicated functional form of the GEV densities. Alternatively, we resort to the variational inference algorithm [141], in which we find the variational distribution $q(\mathbf{y})$ with a fixed but tractable functional form that maximizes L . More precisely, we set $q(\mathbf{y})$ to be a multivariate Gaussian distribution with a diagonal covariance matrix. Specifying $q(\mathbf{y})$ to such a simple function can dramatically speed up the learning process. At the same time, each factor $q(y_i)$ can reliably approximate the corresponding marginal posterior distribution $p(y_i|\mathbf{x}, \boldsymbol{\theta})$ given by Gibbs sampling, especially the mean, as shown in our numerical experiments.

We estimate both the smoothness parameters and the parameters of $q(\mathbf{y})$ via stochastic optimization [142], which is discussed below at length. Since the variational distribution $q(\mathbf{y})$ is given by $\mathcal{N}(\mathbf{y}; \mathbf{m}, C C^T)$, where $\mathbf{m}$ is the mean vector, and C is a diagonal matrix with the standard deviation vector $\boldsymbol{\nu}$ on the diagonal, $\mathbf{y}$ can be

equivalently parameterized as [135]:

$$\mathbf{y} = C\mathbf{y}_e + \mathbf{m}, \quad (7.24)$$

where $\mathbf{y}_e$ is a random vector drawn from a multivariate Gaussian distribution with zero mean and unit variance $\phi(\mathbf{y}_e) = \mathcal{N}(\mathbf{y}_e; \mathbf{0}, I)$. By changing variables according to $\mathbf{y}_e = C^{-1}(\mathbf{y} - \mathbf{m})$, L can be expressed as [135]:

$$\begin{aligned} L &= \int_{\mathbf{y}_e} \phi(\mathbf{y}_e) \log \frac{p(\mathbf{x}|C\mathbf{y}_e + \mathbf{m})p(C\mathbf{y}_e + \mathbf{m}|\boldsymbol{\theta})|C|}{\phi(\mathbf{y}_e)} d\mathbf{y}_e \\ &= E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{x}|C\mathbf{y}_e + \mathbf{m})] + E_{\phi(\mathbf{y}_e)} [\log p(C\mathbf{y}_e + \mathbf{m}|\boldsymbol{\theta})] + \log |C| + c, \end{aligned} \quad (7.25)$$

where c is a constant that summarizes all irrelevant terms. Since $p(C\mathbf{y}_e + \mathbf{m}|\boldsymbol{\theta})$ corresponds to a Gaussian graphical model, we can obtain the closed-form expression of the second term in (7.25):

$$E_{\phi(\mathbf{y}_e)} [\log p(C\mathbf{y}_e + \mathbf{m}|\boldsymbol{\theta})] = \sum_{z \in \{\xi, \zeta, \mu\}} E_{\phi(\mathbf{z}_{Se})} [\log p(\mathbf{z}_S|\alpha_z)] + E_{\phi(\mathbf{z}_{Te})} [\log p(\mathbf{z}_T|\beta_z, \gamma_z)],$$

and

$$E_{\phi(\mathbf{z}_{Se})} [\log p(\mathbf{z}_S|\alpha_z)] = \frac{1}{2} \log |\alpha_z K_S|_+ - \frac{\alpha_z}{2} \mathbf{m}_{\mathbf{z}_S}^T K_S \mathbf{m}_{\mathbf{z}_S} - \frac{\alpha_z}{2} \boldsymbol{\nu}_{\mathbf{z}_S}^T \text{diag}(K_S) \boldsymbol{\nu}_{\mathbf{z}_S}, \quad (7.26)$$

$$\begin{aligned} E_{\phi(\mathbf{z}_{Te})} [\log p(\mathbf{z}_T|\beta_z, \gamma_z)] &= \frac{1}{2} \log |\beta_z \tilde{K}_{pr} + \gamma_z \tilde{K}_{tr}|_+ - \frac{1}{2} \mathbf{m}_{\mathbf{z}_T}^T (K_T + \mathbf{1}\mathbf{1}^T) \mathbf{m}_{\mathbf{z}_T} \\ &\quad - \frac{1}{2} \boldsymbol{\nu}_{\mathbf{z}_T}^T \text{diag}(K_T + \mathbf{1}\mathbf{1}^T) \boldsymbol{\nu}_{\mathbf{z}_T}, \end{aligned} \quad (7.27)$$

where $\text{diag}(K)$ is a diagonal matrix whose diagonal equals that of K .

Our objective is to find the smoothness parameters $\boldsymbol{\theta}$ and the variational parameters $\mathbf{m}$ and $\boldsymbol{\nu}$ to maximize the lower bound L (7.25). To this end, we consider the gradients with respect to those parameters. For the spatial smoothness parameter

α_z , the gradient is given by:

$$\frac{\partial L}{\partial \alpha_z} = \frac{P-1}{2\alpha_z} - \frac{1}{2} \mathbf{m}_{z_S}^T K_S \mathbf{m}_{z_S} - \frac{1}{2} \boldsymbol{\nu}_{z_S}^T \text{diag}(K_S) \boldsymbol{\nu}_{z_S}. \quad (7.28)$$

For the smoothness parameters β_z and γ_z that characterize the temporal dependence, the gradient appears to be complicated because of the log-determinant term. However, recall that $\beta_z \tilde{K}_{pr} + \gamma_z \tilde{K}_{tr} = K_T = (\gamma_z K_{tr}) \oplus (\beta_z K_{pr})$ (7.17). Due to the properties of Kronecker sum [127, Ch. 13], the eigenvalue matrix Λ_T of K_T boils down to:

$$\Lambda_T = (\gamma_z \Lambda_{tr}) \oplus (\beta_z \Lambda_{pr}) = \beta_z \tilde{\Lambda}_{pr} + \gamma_z \tilde{\Lambda}_{tr}, \quad (7.29)$$

where Λ_{tr} and Λ_{pr} are the eigenvalue matrices corresponding respectively to K_{tr} and K_{pr} , $\tilde{\Lambda}_{pr} = I_{tr} \otimes \Lambda_{pr}$, and $\tilde{\Lambda}_{tr} = \Lambda_{tr} \otimes I_{pr}$. Consequently, the log-determinant term in (7.27) can be simplified as:

$$\log |\beta_z \tilde{K}_{pr} + \gamma_z \tilde{K}_{tr}|_+ = \log |\beta_z \tilde{\Lambda}_{pr} + \gamma_z \tilde{\Lambda}_{tr}|_+ = \sum_{\{i: \tilde{\lambda}_{pri} + \tilde{\lambda}_{tri} > 0\}} \log(\beta_z \tilde{\lambda}_{pri} + \gamma_z \tilde{\lambda}_{tri}),$$

and therefore the gradient of L with regard to β_z and γ_z equals:

$$\frac{\partial L}{\partial \beta_z} = \frac{1}{2} \sum_{\{i: \tilde{\lambda}_{pri} + \tilde{\lambda}_{tri} > 0\}} \frac{\tilde{\lambda}_{pri}}{\beta_z \tilde{\lambda}_{pri} + \gamma_z \tilde{\lambda}_{tri}} - \frac{1}{2} \mathbf{m}_{z_T}^T \tilde{K}_{pr} \mathbf{m}_{z_T} - \frac{1}{2} \boldsymbol{\nu}_{z_T}^T \text{diag}(\tilde{K}_{pr}) \boldsymbol{\nu}_{z_T}, \quad (7.30)$$

$$\frac{\partial L}{\partial \gamma_z} = \frac{1}{2} \sum_{\{i: \tilde{\lambda}_{pri} + \tilde{\lambda}_{tri} > 0\}} \frac{\tilde{\lambda}_{tri}}{\beta_z \tilde{\lambda}_{pri} + \gamma_z \tilde{\lambda}_{tri}} - \frac{1}{2} \mathbf{m}_{z_T}^T \tilde{K}_{tr} \mathbf{m}_{z_T} - \frac{1}{2} \boldsymbol{\nu}_{z_T}^T \text{diag}(\tilde{K}_{tr}) \boldsymbol{\nu}_{z_T}. \quad (7.31)$$

On the other hand, the gradient *w.r.t* the variational mean and standard deviation corresponding to the spatial components $\mathbf{z}_S$ of the GEV parameters can be computed as:

$$\begin{aligned} \nabla_{\mathbf{z}_S} L = & E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \right\} \\ & - \alpha_z K_S \mathbf{m}_{z_S}, \end{aligned} \quad (7.32)$$

$$\begin{aligned} \nabla_{\boldsymbol{\nu}_{\mathbf{z}_S}} L = & E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \odot \mathbf{z}_{Se} \right\} \\ & - \alpha_z \text{diag}(K_S) \boldsymbol{\nu}_{\mathbf{z}_S} + 1 \oslash \boldsymbol{\nu}_{\mathbf{z}_S}, \end{aligned} \quad (7.33)$$

where $\odot$ and $\oslash$ denote componentwise product and division respectively, and $\phi(\mathbf{y}_e) = \prod_{z \in \{\xi, \zeta, \mu\}} \phi(\mathbf{z}_{Se}) \phi(\mathbf{z}_{Te})$. The detailed derivation is presented in Appendix B.1. Similarly, for the temporal components,

$$\begin{aligned} \nabla_{\mathbf{m}_{\mathbf{z}_T}} L = & E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_T} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \right\} - K_T \mathbf{m}_{\mathbf{z}_T} \\ & - \left(\sum \mathbf{m}_{\mathbf{z}_T} \right) \mathbf{1}, \end{aligned} \quad (7.34)$$

$$\begin{aligned} \nabla_{\boldsymbol{\nu}_{\mathbf{z}_T}} L = & E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_T} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \odot \mathbf{z}_{Te} \right\} \\ & - \text{diag}(K_T + \mathbf{1}\mathbf{1}^T) \cdot \boldsymbol{\nu}_{\mathbf{z}_T} + 1 \oslash \boldsymbol{\nu}_{\mathbf{z}_T}. \end{aligned} \quad (7.35)$$

The gradient of the logarithm of the GEV densities $\log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj})$ in the above expressions is listed in Appendix B.2.

Since the expectations in (7.32)-(7.35) are intractable, we approximate them stochastically using Monte Carlo integration. The resulting unbiased stochastic approximation of the gradients are called stochastic gradient. Replacing the exact gradients in (7.32)-(7.35) leads to a stochastic optimization algorithm for inferring the optimal variational parameters [142]. Concretely, in each iteration κ , we use one realization of the exact gradients, namely, we draw one sample $\hat{\mathbf{y}}_e$ from $\phi(\mathbf{y}_e)$ and evaluate the gradients at $\hat{\mathbf{y}}^{(\kappa)} = \boldsymbol{\nu}^{(\kappa)} \odot \hat{\mathbf{y}}_e + \mathbf{m}^{(\kappa)}$, where $a^{(\kappa)}$ denotes the value of parameter a in iteration κ . Therefore,

$$\tilde{\nabla}_{\mathbf{m}} L|_{\mathbf{m}=\mathbf{m}^{(\kappa)}} = \nabla_{\mathbf{y}} \log p(\mathbf{x}|\mathbf{y})|_{\mathbf{y}=\hat{\mathbf{y}}^{(\kappa)}} + \nabla_{\mathbf{m}} E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{y}|\boldsymbol{\theta})]|_{\mathbf{m}=\mathbf{m}^{(\kappa)}}, \quad (7.36)$$

$$\begin{aligned} \tilde{\nabla}_{\boldsymbol{\nu}} L|_{\boldsymbol{\nu}=\boldsymbol{\nu}^{(\kappa)}} = & \nabla_{\mathbf{y}} \log p(\mathbf{x}|\mathbf{y})|_{\mathbf{y}=\hat{\mathbf{y}}^{(\kappa)}} \odot \hat{\mathbf{y}}_e + \nabla_{\boldsymbol{\nu}} E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{y}|\boldsymbol{\theta})]|_{\boldsymbol{\nu}=\boldsymbol{\nu}^{(\kappa)}} + 1 \oslash \boldsymbol{\nu}^{(\kappa)}, \end{aligned} \quad (7.37)$$

where $\tilde{\nabla}_{\mathbf{m}} L$ and $\tilde{\nabla}_{\boldsymbol{\nu}} L$ represent the stochastic gradients. Only the first terms on

the right hand side of the above two equations are approximated stochastically, whereas the second terms can be computed in closed form. We then update all the parameters following a gradient ascent approach:

$$\mathbf{m}^{(\kappa+1)} = \mathbf{m}^{(\kappa)} + \rho^{(\kappa)} \tilde{\nabla}_{\mathbf{m}} L|_{\mathbf{m}=\mathbf{m}^{(\kappa)}}, \quad (7.38)$$

$$\boldsymbol{\nu}^{(\kappa+1)} = \boldsymbol{\nu}^{(\kappa)} + \rho^{(\kappa)} \tilde{\nabla}_{\boldsymbol{\nu}} L|_{\boldsymbol{\nu}=\boldsymbol{\nu}^{(\kappa)}}, \quad (7.39)$$

$$\boldsymbol{\theta}^{(\kappa+1)} = \boldsymbol{\theta}^{(\kappa)} + \rho^{(\kappa)} \nabla_{\boldsymbol{\theta}} L|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(\kappa)}}, \quad (7.40)$$

where $\rho^{(\kappa)}$ is the learning rate (or step size) in iteration κ .

When the learning rate schedule follows the Robbins-Monro conditions [142]:

$$\sum_{\kappa=1}^{\infty} \rho^{(\kappa)} = \infty, \quad \sum_{\kappa=1}^{\infty} (\rho^{(\kappa)})^2 < \infty, \quad (7.41)$$

the stochastic optimization algorithm converges to a local maximum of L . Due to the noisy gradient used in each iteration, the SVI algorithm can easily escape from shallow local maxima of the complex objective function, and converges to at least a significant local maximum. The proposed algorithm can be viewed as a stochastic variational extension of the expectation conjugate gradient algorithm proposed in [143].

The computational complexity of the SVI algorithm is now linear in $|\mathcal{V}_O|$, where $|\mathcal{V}_O|$ denotes the number of nodes in $\mathcal{V}_O$, and it equals $\mathcal{O}(D)$ when there is no missing data. As pointed out in [136], the stochastic gradient can be obtained using a mini-batch of $M \leq |\mathcal{V}_O|$ factors of the joint distribution $p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ (7.21). The resulting computational cost can be further reduced to be sublinear. As an example, let us focus on the stochastic gradient of $m_{z_{Sj}}$ ($j = 1, \dots, P$):

$$\frac{\partial L}{\partial m_{z_{Sj}}} = \sum_{i=1}^N \frac{\partial \log f(x_{ij}|\xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj})}{\partial z_{Sj}} - \alpha_z [K_S]_{j,:} \mathbf{m}_{\mathbf{z}_S}, \quad (7.42)$$

where $[K_S]_{j,:}$ is the j th row of matrix K_S . The first term on the right hand side of

the above equation can be equivalently expressed as:

$$NE_{p(i)} \left[\frac{\partial \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj})}{\partial z_{Sj}} \right], \quad (7.43)$$

where $p(i)$ is a discrete uniform distribution on the set $\{1, \dots, N\}$. Thus, we randomly draw a mini-batch of i from the set, compute the corresponding partial derivatives, and approximate the expectation in (7.43) by the mean value of the partial derivatives. The stochastic gradients *w.r.t* other parameters can be computed in a similar fashion. In the most extreme case, to compute all the required stochastic gradients, we only need to draw $M = \max(N, P)$ samples x_{ij} uniformly from $\mathcal{V}_O$ such that all the indices i ($i = 1, \dots, N$) and j ($j = 1, \dots, P$) appear at least once. Then the computational complexity is only $\mathcal{O}(\max(N, P))$. Our numerical experiments demonstrate that using a mini-batch of $\mathcal{V}_O$ can greatly accelerate the algorithm, reducing the computational time from hours to minutes.

7.3.1.1 Selecting the Step Size

One challenge with stochastic optimization methods is setting the learning rate. As the parameters in our problem have completely different scales, if we use a unified step size to update all the parameters, the step size has to be small enough to tackle the smallest scale. The resulting algorithm would converge slowly. To address this concern, we exploit the ADADELTA method [144], which adaptively sets individual dynamic step size for each component of the parameter vector. Specifically, ADADELTA defines the step size as:

$$\boldsymbol{\rho}^{(\kappa)} = \rho_0 \oslash \sqrt{\tilde{E}[(\nabla L)^2]^{(\kappa)} + \epsilon}, \quad (7.44)$$

where ϵ is a small constant that servers the purpose to better condition the denominator, and $\tilde{E}[(\nabla L)^2]$ is an exponentially decaying moving average of the squared

gradients which can be updated as:

$$\tilde{E}[(\nabla L)^2]^{(\kappa)} = \eta \tilde{E}[(\nabla L)^2]^{(\kappa-1)} + (1 - \eta)(\nabla L^{(\kappa)})^2, \quad (7.45)$$

where $(\nabla L^{(\kappa)})^2$ denotes componentwise square of $\nabla L^{(\kappa)}$. Since the denominator employs the squared gradient information, large gradients have smaller learning rates and vice versa. The ADADELTA method has the nice property as in second-order methods (e.g., Newton's method) that the progress along each dimension evens out over time. On the other hand, as shown in [145, 146], the moving average of squared gradient $\tilde{E}[(\nabla L)^2]$ is a good approximation to $E[(\nabla L)^2]$, which can be further decomposed as:

$$E[(\nabla L)^2] = E[\nabla L]^2 + V[\nabla L], \quad (7.46)$$

where $V[\nabla L]$ is the variance of the gradient. As a result, the step size decreases with the growing of the variance of the gradient, thus mitigating the risk of taking a large step in a wrong direction.

In our experiments, we follow [144] to set $\eta = 0.95$ and $\epsilon = 10^{-6}$. Additionally, we initialize $\rho_0 = 10^{-3}$ and scale it every 1000 iterations by a factor of 0.99 in a similar manner as in [135], so as to guarantee the convergence of the algorithm.

7.3.1.2 Reducing the Variance of the Gradient

In order to increase the step size and improve the convergence rate, one has to design methods that can reduce the variance of the stochastic gradient. It has been demonstrated in [147]-[149], both empirically and theoretically, that utilizing a fixed-window moving average of stochastic gradients can effectively reduce the variance and highly speed up the stochastic gradient algorithm both empirically and theoretically. Here, we further extend the idea and employ an exponentially decaying moving average in which the decaying rate depends on the variance of the gradient. As in [149], we only compute the moving average of the Monte Carlo approximation

part of the stochastic gradient (e.g., the first terms in Eq. (7.32)-(7.35)), since other terms are deterministic values that have no influence on the variance. Specifically, let $E[g]$ denote the exact value of the first terms in Eq. (7.32)-(7.35), g the Monte Carlo approximation of $E[g]$, and $\tilde{E}[g]$ the exponentially decaying moving average. $\tilde{E}[g]$ can be updated as:

$$\tilde{E}[g]^{(\kappa)} = \left(1 - \frac{1}{\tau_g^{(\kappa)}}\right) \tilde{E}[g]^{(\kappa-1)} + \frac{1}{\tau_g^{(\kappa)}} g^{(\kappa)}, \quad (7.47)$$

where $\tau_g^{(\kappa)}$ can be viewed as the window size of the moving average in iteration κ . We want the window size to increase when the variance is large, and to decay if the variance becomes small. Note that a good measure of the variance given the stochastic gradients in each iteration is:

$$\omega = \frac{\sum \tilde{E}[\nabla L]^2}{\sum \tilde{E}[(\nabla L)^2]} \approx \frac{\sum E[\nabla L]^2}{\sum E[\nabla L]^2 + \sum V[\nabla L]}, \quad (7.48)$$

where $\tilde{E}[\nabla L]$ and $\tilde{E}[(\nabla L)^2]$ are the exponentially decaying moving average of the gradient and the squared gradient respectively with the decaying rate η as defined in (7.45). It is evident that ω grows with the inverse of the variance. Given the current measure of the variance $\omega^{(\kappa)}$, we update the window size $\tau_g^{(\kappa+1)}$ for the next iteration as:

$$\tau_g^{(\kappa+1)} = (1 - \omega^{(\kappa)}) \tau_g^{(\kappa)} + 1. \quad (7.49)$$

As such, $\tau_g \geq 1$ as the algorithm proceeds, and it changes with the variance of the gradient as desired. Interestingly, it can be observed that the length of the moving window will decrease if we take a big step (i.e., ∇L is large and $E[\nabla L]$ increases) in the current iteration. In this case, the gradients in the previous iterations are unreliable, and we can see that they will contribute less to $\tilde{E}[g]$ in the next iteration according to the proposed method. Therefore, such mechanism makes $\tilde{E}[g]$ approximate $E[g]$ more accurately. Note that similar methods are applied in [145, 146] for step size selection. We find that this method works as well for variance reduction.

Moreover, it is straightforward to combine the method with ADADELTA to achieve better performance. The method can be further extended to yield different τ_g for different sets of parameters, cf. [145]. Here we only compute a unified τ_g for simplicity. In our experiments, we initialize $\tau_g^{(1)} = 1$. We then replace $g^{(\kappa)}$ by $\tilde{E}[g]^{(\kappa)}$ when computing the stochastic gradient in each iteration.

7.3.1.3 Bounding the GEV Parameters

Note that when the shape parameter $\xi_{ij} = 0$, the expression of the PDF (7.15) does not involve ξ_{ij} . In order to obtain the partial derivatives with respect to ξ_{ij} , we approximate it by a small value, e.g., $\xi_{ij} = 10^{-6}$, and use the PDF in (7.13) instead. Additionally, given an observation x_{ij} , the GEV parameters must satisfy the constraint $1 + \xi_{ij}(x_{ij} - \mu_{ij})/\exp(\zeta_{ij}) > 0$, so as to guarantee that the log-likelihood and the corresponding gradient are well defined. However, the variational distribution $q(\mathbf{y})$ is defined in $\mathbb{R}$. To address this issue, we borrow the idea of Lagrangian multipliers, and extend the domain of the likelihood of the GEV parameters to $\mathbb{R}$ as follows:

$$\tilde{f}(x_{ij}|\xi_{ij}, \zeta_{ij}, \mu_{ij}) = \begin{cases} f(x_{ij}|\xi_{ij}, \zeta_{ij}, \mu_{ij}) & \text{if } 1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) > 0, \\ \exp \left\{ c_1 \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right] - c_2 \right\} & \text{otherwise,} \end{cases} \quad (7.50)$$

where c_1 and c_2 are sufficiently large positive constants. According to the above definition, when a sample $\hat{\mathbf{y}}$ from the variational distribution $q(\mathbf{y})$ fails to satisfy the constraint, the corresponding stochastic gradient will move the mean vector of $q(\mathbf{y})$ in the direction where the constraints can be satisfied, since in this case the gradient of $\tilde{f}(x_{ij})$ always points to the direction in which the value of $1 + \xi_{ij}(x_{ij} - \mu_{ij})/\exp(\zeta_{ij})$ increases. Moreover, when the constraint is not satisfied, $\tilde{f}(x_{ij})$ is close to zero due to the large positive constant c_2 . Therefore, the shape of the original and the extended likelihood are almost the same.

After replacing $f(x_{ij})$ with $\tilde{f}(x_{ij})$ in (7.21), the value $\mathbf{y}^*$ that maximizes $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ is the same as before, as indicated by the following proposition:

Table 7.1: Stochastic variational inference of the spatio-temporal model.

<p>Initialize $E[\nabla_{\boldsymbol{\theta}} L^2]^{(0)} = 0$, $E[\Delta \boldsymbol{\theta}^2]^{(0)} = 0$. Iterate the following steps until $\mathbf{m}$ and $\boldsymbol{\nu}$ converge.</p> <ol style="list-style-type: none"> 1. Draw one sample $\hat{\mathbf{y}}^{(\kappa)}$ from the multivariate Gaussian distribution $\mathcal{N}(\mathbf{y}; \mathbf{m}^{(\kappa)}, C^{(\kappa)}(C^{(\kappa)})^T)$: $\hat{\mathbf{y}}_e \sim \phi(\mathbf{y}_e), \quad \hat{\mathbf{y}}^{(\kappa)} = \boldsymbol{\nu}^{(\kappa)} \odot \hat{\mathbf{y}}_e + \mathbf{m}^{(\kappa)}.$ 2. Compute the gradient <i>w.r.t</i> the smoothness parameters $\nabla_{\boldsymbol{\theta}} L _{\boldsymbol{\theta}=\boldsymbol{\theta}^{(\kappa)}}$ (cf. Eq. (7.28), (7.30) and (7.31)) and the stochastic gradients <i>w.r.t</i> the variational parameters (cf. Eq. (7.32)-(7.35)): $\begin{aligned} \tilde{\nabla}_{\mathbf{m}} L _{\mathbf{m}=\mathbf{m}^{(\kappa)}} &= \nabla_{\mathbf{y}} \log p(\mathbf{x} \mathbf{y}) _{\mathbf{y}=\hat{\mathbf{y}}^{(\kappa)}} + \nabla_{\mathbf{m}} E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{y} \boldsymbol{\theta})] _{\mathbf{m}=\mathbf{m}^{(\kappa)}}, \\ \tilde{\nabla}_{\boldsymbol{\nu}} L _{\boldsymbol{\nu}=\boldsymbol{\nu}^{(\kappa)}} &= \nabla_{\mathbf{y}} \log p(\mathbf{x} \mathbf{y}) _{\mathbf{y}=\hat{\mathbf{y}}^{(\kappa)}} \odot \hat{\mathbf{y}}_e + \nabla_{\boldsymbol{\nu}} E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{y} \boldsymbol{\theta})] _{\boldsymbol{\nu}=\boldsymbol{\nu}^{(\kappa)}} \\ &\quad + 1 \oslash \boldsymbol{\nu}^{(\kappa)}. \end{aligned}$ 3. Compute the moving average the first terms of the above two equations to reduce the variance of the stochastic gradients, as described in Section 7.3.1.2. 4. Set the componentwise step size $\rho^{(\kappa)}$ as described in Section 7.3.1.1. 5. Update $\mathbf{m}$, $\boldsymbol{\nu}$, $\boldsymbol{\theta}$ following Eq. (7.38)-(7.40)

Proposition 7.1. *Let $\mathbf{y}^*$ correspond to a local maximum of $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$. If the positive constant c_1 is sufficiently large, then $\mathbf{y}^*$ satisfies the constraint that $1 + \xi_{ij}(x_{ij} - \mu_{ij})/\exp(\zeta_{ij}) > 0$, $\forall \{i, j\} \in \mathcal{V}_O$, and $\log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ also attains a local maximum at $\mathbf{y}^*$.*

Proof. See Appendix B.3. □

As a result, we can safely replace $\log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ with $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ during the learning process. We summarize the overall algorithm in Table 7.1.

7.3.2 Prediction of Future GEV Parameters

A primary goal of the spatio-temporal model is to predict future GEV distributions. Since the spatial component of each GEV parameter is time invariant, we only need to extrapolate the temporal component to time points $N+1, N+2, \dots$. Recall that the thin-plate model with the zero curvature boundary condition serves as a natural tool to predict the future by means of the current trend (cf. Section 7.1). We thus incorporate the future temporal components of GEV parameters at the right end of the neighborhood structure of the temporal model (see Fig. 7.2(c)). As a result, the future parameters $\mathbf{z}_{Tf}$ and the observed parameters $\mathbf{z}_{To}$ together form a Gaussian graphical model with precision matrix $\tilde{K}_T$. According to Schur complement, we can obtain the MAP estimates:

$$\mathbf{z}_{Tf} = -[\tilde{K}_T]_{ff}^{-1}[\tilde{K}_T]_{fo}\mathbf{z}_{To}. \quad (7.51)$$

Due to the special structure of the temporal thin-plate model, the expression (7.51) boils down to:

$$\begin{aligned} \mathbf{z}_{TN+(i-1)\tau+1:N+i\tau} &= \gamma_z (\beta_z K_{pr} + \gamma_z I_{pr})^{-1} \\ &\cdot (2\mathbf{z}_{TN+(i-2)\tau+1:N+(i-1)\tau} - \mathbf{z}_{TN+(i-3)\tau+1:N+(i-2)\tau}), \end{aligned} \quad (7.52)$$

for $i = 1, 2, \dots$. Since $(\beta_z K_{pr} + \gamma_z I_{pr})$ is a sparse matrix, the computational complexity of solving the linear system is linear in τ , when applying algorithms such as the belief propagation [87] and the embedded subgraphs [91]. Therefore, the proposed model provides an efficient tool for forecasting future GEV distributions. The final estimation of GEV parameters at time point $N+i$ and site j is $z_{N+i,j} = z_{TN+i} + z_{Sj}$.

7.4 Numerical Results

In this section, we apply our model to synthetic and real data. We first show the sub-linear computational complexity of the proposed SVI algorithm. We then compare

the SVI algorithm with Gibbs sampling when learning the spatio-temporal model. In addition, we also benchmark the proposed spatio-temporal model (STM) against a spatial model (SM; without considering the temporal variation) [47], a temporal model (TM; without considering the spatial variation) [55], and a model with the same shape and scale parameter for all locations and time points (SSSM) [57]. The proposed SVI algorithm is employed to learn the parameters of all four models. We compare the four models using the deviance information criterion (DIC) [65]:

$$\text{DIC} = \bar{D} + p, \quad (7.53)$$

where

$$\bar{D} = E_{q(\mathbf{y})}[-2 \log p(\mathbf{x}|\mathbf{y})] \quad (7.54)$$

$$p = E_{q(\mathbf{y})}[-2 \log p(\mathbf{x}|\mathbf{y})] + 2 \log p(\mathbf{x}|E_{q(\mathbf{y})}[\mathbf{y}]). \quad (7.55)$$

The first term $\bar{D}$ is defined as the posterior expectation of the deviance. It can be regarded as a Bayesian measure of model fit, which attains smaller values for better models. The second term p measures the model complexity by the effective number of parameters. The DIC is a hierarchical model generalization of the AIC and the BIC, and it is particularly useful in Bayesian model selection problems [60, 61, 64]. In addition, for synthetic data, we compute mean squared error (MSE) between the estimated GEV parameter and the ground truth for both observed time series and future GEV distributions in the next year. For real data, we assess the predictive performance by evaluating the averaged absolute fractional prediction errors (AAFPE) [63]. More concretely, let $\hat{x}_{N+i,j}$ be the median of the M samples $\hat{\mathbf{x}}_{N+i,j}^{(1:M)}$ drawing from the estimated future GEV distribution at time instant $N+i$ and location j . Then, the AAFPE is given by [63]:

$$\text{AAFPE} = \frac{1}{N_f P} \sum_{i=1}^{N_f} \sum_{j=1}^P \left\| \frac{\hat{x}_{N+i,j} - x_{N+i,j}}{x_{N+i,j}} \right\|, \quad (7.56)$$

Figure 7.3: Estimates of the spatial component of a scale parameter *w.r.t.* number of iterations using a proportion of all the observations.

where $x_{N+i,j}$ is the observed block maximum at time point $N + i$ and location j , and $\|\cdot\|$ denotes the absolute value.

7.4.1 Synthetic Data

We generate synthetic data by first specifying the GEV parameters in the spatio-temporal domain and then drawing one single sample at each location and time instant. The GEV parameters are defined as quadratic Legendre polynomials of the latitude and longitude of the measuring stations. We then specify the temporal variation by means of trigonometric functions with period $\tau = 12$. Finally, we add an overall polynomial trend to the GEV parameters across time. Concretely, we consider 256 sites arranged on a 16×16 regular lattice with 360 monthly maximum observations for each site. We use the data of the first 348 months to learn the model, and retain the rest 12-month data to test the prediction algorithm. Therefore, $D = 89,088$ in this case.

We first explore how the performance of the SVI algorithm changes when using a smaller mini-batch of samples from $\mathcal{V}_O$ to compute the stochastic gradient. Con-

Table 7.2: Performance of the SVI algorithm when the size of the mini-batch changes.

Size of mini-batch	MSE			Computational time (s)	No. of iterations
	ξ	ζ	μ		
1%	6.08×10^{-4}	1.98×10^{-3}	1.24×10^{-1}	4.93×10^2	80000
2%	6.71×10^{-4}	2.00×10^{-3}	1.25×10^{-1}	7.09×10^2	75000
5%	6.22×10^{-4}	1.95×10^{-3}	1.25×10^{-1}	1.23×10^3	55000
10%	6.01×10^{-4}	1.93×10^{-3}	1.25×10^{-1}	2.45×10^3	55000
30%	5.97×10^{-4}	1.92×10^{-3}	1.25×10^{-1}	6.11×10^3	55000
50%	6.07×10^{-4}	1.92×10^{-3}	1.25×10^{-1}	1.09×10^4	60000
70%	6.03×10^{-4}	1.91×10^{-3}	1.25×10^{-1}	1.51×10^4	65000
90%	6.21×10^{-4}	1.92×10^{-3}	1.25×10^{-1}	1.94×10^4	65000

cretely, we use 1%, 2%, 5%, 10%, 30%, 50%, 70%, and 90% of all the samples sequentially. We show how the spatial component of a randomly selected scale parameter ζ_{Sj} varies as the algorithm proceeds in Fig 7.3. Other related information, such as the accuracy of estimation, the computational time, and the total number of iterations, is listed in Table 7.2. It can be seen that the SVI algorithm converges to the same optimal point regardless of the size of the mini-batch. More importantly, although gradients resulting from a very small minibatch (i.e., 1% and 2%) are very noisy and therefore it takes more iterations before the algorithm can converge, the small computational complexity in each iteration successfully shortens the overall computational time, from hours to minutes. Therefore, unless otherwise stated, we only use 1% of observations to compute the stochastic gradient in the following simulations.

Next, we compare the proposed SVI algorithm with Gibbs sampling to investigate how well the variational distribution can approximate the simulated true posterior distribution. The Gibbs sampling procedure is outlined in Appendix B.4. Similar methods are employed in [56]. Here, we draw 500,000 samples. We discard the first 5000 samples as burn-in iterations, and further thin the rest samples by a factor 20. We depict in Fig. 7.4 the estimated distributions of randomly selected spatial and temporal components of GEV parameters resulting from the two methods. As shown in the figure, although the variances of the variational distributions are less consistent with those of the simulated posterior distributions, the mean values are

Figure 7.4: Comparison the results from the MCMC method and the SVI method: the distribution of the spatial component of a shape (a), a scale (b), and a location parameter (c) and the distribution of the temporal component of a shape (d), a scale (e), and a location parameter (f).

almost identical. Indeed, the MSE between the mean value of the Gibbs samples and the ground truth for the three GEV parameters (ξ, ζ, μ) is 2.87×10^{-4} , 1.45×10^{-3} and 1.10×10^{-1} respectively, while the corresponding MSE for the SVI algorithm is 6.08×10^{-4} , 1.98×10^{-3} , and 1.24×10^{-1} respectively (see Table 7.2), indicating that the SVI algorithm performs comparably with the Gibbs sampling algorithm in terms of MSE. However, it takes 4.09×10^5 seconds to generate all the Gibbs samples, whereas the SVI algorithm only runs for 4.93×10^2 seconds. The computational time of the proposed SVI algorithm is three orders of magnitude shorter than that of the Gibbs sampling. Indeed, the MSE between the mean value of the Gibbs samples and the ground truth for the three GEV parameters (ξ, ζ, μ) are 2.87×10^{-4} , 1.45×10^{-3} and 1.10×10^{-1} respectively. In comparison with Table 7.2, we can find that the SVI algorithm performs comparably with the Gibbs sampling algorithm in terms of MSE. However, it takes 4.09×10^5 seconds to generate all the Gibbs samples, whereas

Figure 7.5: Estimates of GEV parameters across time: (a) estimates of scale parameter σ across time; (b) estimates of location parameter μ across time.

the SVI algorithm only takes 4.93×10^2 seconds (see Table 7.2). The computational time of the proposed SVI algorithm is several orders of magnitude shorter than that of the Gibbs sampling.

Now we compare the proposed model with three other models, including a SM, a TM, and a SSSM. The results are summarized in Fig. 7.5 to Fig. 7.7 and in Table 7.3. Specifically, Fig. 7.5 shows the estimated scale and location parameters across time resulting from the four models, while Fig. 7.6 and Fig. 7.7 shows the estimated scale and location parameters respectively across space. The results of the shape parameters are qualitatively similar to that of the scale parameters, so we omit them. Table 7.3 lists the DIC scores, and the MSE for the estimated GEV

Figure 7.6: Estimates of scale parameter σ across all sites at one time point.

Figure 7.7: Estimates of location parameter μ across all sites at one time point.

parameters with respect to the observed monthly maxima as well as for the predicted GEV parameters in the next year.

As shown in Table 7.3, the proposed STM outshines the competing models in terms of the MSE and the DIC score, and also provides a reliable tool to forecast the GEV distributions in the future. Moreover, we can observe from Fig. 7.6(e), Fig. 7.7(e), and Fig. 7.5 that the STM yields estimates that closely follow the true temporal and spatial pattern. By contrast, the SM mistakenly ignores the temporal variation (see Fig. 7.5), and yields biased estimates of the scale and location parameters across space at the randomly selected time instant (see Fig. 7.6(b) and Fig. 7.7(b)). Similarly, the TM fails to explain the spatial variation of the GEV parameters (see Fig. 7.6(c) and Fig. 7.7(c)), while wrongly estimating the location and scale parameters in time domain (see Fig. 7.5). In addition, since the location parameters are assumed to be the same across time and space respectively in the SM and the TM, the observations are more different from the corresponding location parameters in these two models than in other models. In order to capture such large deviance from the location parameters, the scale parameters are overestimated, as demonstrated in Fig. 7.6 and Fig. 7.5(a). The SSSM, on the other hand, performs better than the above mentioned two models when describing the spatio-temporal dependence

Table 7.3: Quantitative comparison of different models for spatio-temporal extreme-value data

Models	MSE (observed)			MSE (predicted)			DIC
	ξ	ζ	μ	ξ	ζ	μ	
SM	1.68×10^{-2}	2.30×10^{-1}	6.44×10^1	3.05×10^{-2}	2.98×10^{-1}	6.24×10^1	6.93×10^5
TM	2.76×10^{-2}	2.70×10^{-1}	5.23×10^1	2.29×10^{-2}	3.31×10^{-1}	5.22×10^1	7.11×10^5
SSSM	3.18×10^{-2}	6.63×10^{-2}	1.01	5.43×10^{-2}	8.67×10^{-2}	1.38	6.29×10^5
STM	6.08×10^{-4}	1.98×10^{-3}	1.24×10^{-1}	2.51×10^{-3}	3.22×10^{-3}	6.93×10^{-1}	6.15×10^5

Figure 7.8: MSE for varying proportion of missing data (averaged over 100 trials). The MSE increases with growing proportion of missing data, as expected.

among the location parameters. Unfortunately, the assumption that the shape and scale parameters are constant seriously limits the modeling power, and therefore, the resulting DIC score is larger than that of the STM. In addition, the assumption influences the estimation of the location parameters as well. As a consequence, the corresponding estimates are less accurate than that of the STM, which can be seen from Fig. 7.7(d), Fig. 7.5, and Table 7.3. In summary, we can conclude that it is essential to consider the spatio-temporal variation for all the three GEV parameters when modeling the synthetic data at hand.

Finally, we investigate whether the proposed SVI algorithm can yield accurate estimates of GEV parameters when observations are missing at random. Here, we use the first data set whose GEV parameters vary across both space and time. In

this case, $|\mathcal{V}_O| < D$. Fig. 7.8 shows the MSE for each GEV parameter and for the observed variables ($\{i, j\} \in \mathcal{V}_O$) and the unobserved ones ($\{i, j\} \notin \mathcal{V}_O$) respectively as a function of the percentage of missing variables across 100 trials. We can see that the MSE increases with the number of unmonitored sites and time points, in agreement with our expectation. However, the MSE is still small even when only 10% variables are observed. In conclusion, the proposed model is applicable to cases with missing data.

7.4.2 Real Data

7.4.2.1 Nigeria Precipitation Data

We now consider the extreme precipitation in south Nigeria. The daily rainfall data available at measuring stations from 1979-2005 is merged into the full gridded data set with resolution 0.1° in [150]. We choose 256 sites arranged on a 16×16 lattice, and extract the monthly maxima for each site. We fit the four models (i.e., the SM, the TM, the SSSM, and the STM) to the first 26-year data, and retain the monthly maxima in 2004 to check the predictive performance.

We first conduct an exploratory study on the data. Fig. 7.9 illustrates the non-stationarity of the data across space and time. We can see that the distribution of monthly maximum rainfall amount varies significantly more across time than across space. Additionally, Fig. 7.10 shows that there exists strong spatial association in monthly maximum rainfall amount for pairs of nearby sites, but less strong dependence for pairs of sites situated at opposite points of the lattice. This indicates that the GEV distributions, or equivalently the GEV parameters, are similar at nearby sites.

We next estimate parameters of the four models using the SVI algorithm developed in Section 7.3. The DIC scores of the four models are respectively 7.6654×10^5 , 5.6087×10^5 , 6.2440×10^5 , and 5.5151×10^5 . It is obvious that the proposed STM fits the data the best. We further depict in Fig. 7.11 the estimated shape parameters

Figure 7.9: Non-stationarity in extreme rainfall data: (a) Distribution of monthly maximum rainfall of all months in 26 years at 12 randomly selected sites; (b) Distribution of monthly maximum rainfall from January to December of all 26 years at a random site. The distribution of extreme rainfall clearly depends on the location and the month.

Figure 7.10: Scatter plots of monthly maximum rainfall amount at pairs of sites: (a) two contiguous sites; (b) two distant sites.

given by the STM for different months at the same site as in Fig. 7.9(a). It can be seen that shape parameters tend to be larger than zero (i.e., the extreme values follow Fréchet distributions) only in January, February, July, August, and December, which correspond to those months with heavy-tailed distributions in Fig. 7.9(a). The remaining months are more likely to have distributions with bounded upper tails. As different months exhibit different tail behaviors, it is improper to assume that the shape parameter is constant across time. This results in the relatively large DIC score of the SSSM in comparison with the STM. Moreover, since there is less heterogeneity across space than across time as demonstrated in Fig. 7.9, the TM that

Figure 7.11: Estimated shape parameters in different months.

ignores the spatial variation performs the second best. The spatial model, however, cannot capture well the temporal non-stationarity, leading to the worst fitting. On the other hand, the AAFPEs for the four models are 1.3522, 0.8271, 0.7996, and 0.7651. The proposed model attains the smallest prediction error, suggesting that it can forecast extreme-value distributions in the future more reliably.

7.4.2.2 Japan Precipitation Data

We next analyze a real data set of the monthly maximum rainfall amount in Japan. The daily rainfall data from 1900-2011 is compiled and interpolated onto a grid with resolution 0.05° [151]. We select one 32×32 regular grid in central Japan, where heavy rainfall is often the cause of floods. Once again, we extract the monthly maxima from 1900 to 2009 to learn the model and hold out the data in 2010 to 2011 for validation purpose. Note that the number of dimensions in this case is $D = 1024$ sites $\times$ 110 years $\times$ 12 months = 1,351,680. The computational time of the proposed SVI algorithm for such large-scale data is only 2.22×10^4 seconds. Similar to the results of the Nigeria data, the DIC scores of the four models are 1.22×10^7 (SM), 1.18×10^7 (TM), 1.07×10^7 (SSSM), and 1.01×10^7 (STM) respectively, while the AAFPEs are 2.07, 2.02, 1.72, 1.68. Hence, the proposed STM achieves the best performance.

7.5 Summary

In this chapter, a novel statistical model is proposed to describe the spatio-temporal extreme-value data. This model allows all the three GEV parameters (shape, location, and scale) to change in the spatio-temporal domain, thus characterizing the spatial and temporal dependence in a flexible manner. More explicitly, we assume each GEV parameter can be decomposed as the sum of a spatial and a temporal component, as in a generalized additive model. Graphical models, particularly, thin-plate models, are then imposed on the spatial and the temporal components to capture the spatial and temporal dependence. A stochastic variational inference algorithm is developed to learn the model parameters. Due to the stochastic nature of the algorithm, the computational complexity is sublinear in the number of variables. Thus, as demonstrated in the numerical experiments, the model can handle thousands or even millions of variables in the spatio-temporal domain. Results of both synthetic and real data show that the proposed model can recover the underlying spatio-temporal pattern in an automatic manner, given one single observation at each site and time point. Furthermore, it can reliably predict distributions of extreme events in the future.

Part III

Graphical Models for Joint Analysis of Extreme-Value Data

Chapter 8

Modeling Spatial Extremes by Copula Gaussian Graphical Models

Let us turn our attention to joint modeling of extreme events, in particular, spatial extreme events. Besides spatial dependence between GEV parameters (as discussed in the previous two chapters), there also exists spatial dependence between extreme values; extreme values at nearby locations are similar to each other.

In this chapter, we employ the method proposed in Chapter 6 to couple the GEV parameters across space by regarding the location of the measuring sites as the only covariate. Furthermore, we couple the extreme values across space using a copula Gaussian graphical model (CGGM). We refer to the proposed model here as the copula MRF-GEV model. In addition, we derive interpolation algorithms from the copula MRF-GEV model. The resulting interpolation schemes strongly resemble inverse distance weighted (IDW) interpolation [152], and are quite simple and efficient, due to the sparse thin-membrane structure.

More explicitly, we first generalize thin-membrane models to irregular grids in Section 8.1. This allows us to handle the situation where measurements are collected at

random locations. In Section 8.2, we discuss the GEV marginals, and describe algorithms to infer the spatial dependent GEV parameters. In Section 8.3, we describe how we incorporate dependencies among the extreme events by means of a copula Gaussian graphical model. In Section 8.4, we explain how our proposed model can be used for interpolating extreme values at sites without observations. In Section 8.5, we assess the proposed model and benchmark it with other spatial models by means of synthetic and real data. We offer concluding remarks in Section 8.6.

8.1 Thin-Membrane Models

Let us first review the definition of thin membrane models. The thin-membrane model is a Gaussian graphical model that is commonly used as smoothness prior as it minimizes the difference between values at neighboring nodes. The thin-membrane model is usually defined for regular grids (i.e., lattice), as illustrated in Fig. 8.1(a), and its PDF can be written as:

$$P(\mathbf{z}) \propto \exp\left\{-\frac{1}{2}\alpha \sum_{i \in \mathcal{V}} \sum_{j \in N(i)} (z_i - z_j)^2\right\} \quad (8.1)$$

$$\propto \exp\left\{-\frac{1}{2}\alpha \mathbf{z}^T K_{\text{tm}} \mathbf{z}\right\}, \quad (8.2)$$

where $N(i)$ denotes the neighboring nodes of node i , and α is the smoothness parameter. The matrix K_{tm} is a Laplacian matrix with its diagonal elements $[K_{\text{tm}}]_{ii}$ equal to the number of neighbors of site i , while its off-diagonal elements $[K_{\text{tm}}]_{ij}$ are -1 if the sites i and j are adjacent and 0 otherwise. Note that $K = \alpha K_{\text{tm}}$ is the precision matrix of $P(\mathbf{z})$ (8.2). The parameter α controls the smoothness of the whole thin-membrane model, imposing the same extent of smoothness for all pairs of neighbors.

Thin-membrane models can be extended to irregular grids, as illustrated in Fig. 8.1(b). The adjacency structure $\mathcal{E}_{\text{tm}}$ may be generated automatically by Delaunay triangulation [153], which maximizes the minimum angle for all the triangles in the grid.

Figure 8.1: Generalized thin-membrane model: (a) Regular grid; (b) Irregular grid.

In this case, $N(i)$ denotes all the nodes that have direct connection with node i . As a natural extension of (8.2), the nonzero entries in K_{tm} may be defined as $[K_{\text{tm}}]_{ij} = -1/d_{ij}^2$, where d_{ij} is the distance between node i and j . The diagonal elements in the adjacency matrix are given by $[K_{\text{tm}}]_{ii} = -\sum_{j=1, j \neq i}^p [K_{\text{tm}}]_{ij}$. We refer to this model as the irregular thin-membrane model, and its PDF can still be expressed by (8.2). Note that the regular thin-membrane model is a special case of the irregular thin-membrane model, where all the nodes are located on a regular grid, and all distances d_{ij} are identical.

8.2 GEV Marginals

In this section, we describe how we infer the marginal GEV distribution at each site (i.e., the first stage of spatial extremes modeling). Here, we consider maxima over a particular time period, e.g., annual maxima. Suppose that we have N samples $x_i^{(n)}$ at each of P sites, where $i = 1, \dots, P$ and $n = 1, \dots, N$. We assume that the block maxima at each site follow a GEV distribution with shape parameter ξ_i , scale parameter σ_i , and location parameter μ_i . As in Section 6.2.1, we can produce local estimates of GEV parameters $\hat{\mu}$, $\hat{\sigma}$, and $\hat{\xi}$ via the PWM method.

To improve the accuracy of the estimated GEV parameters, we couple those local estimates in space by means of a thin-membrane model; we utilize regular or irregular

thin-membrane models, depending on the arrangement of the measuring stations. Let $\mathbf{y} = [y_1, \dots, y_P]^T$ denote the local estimates of $\mathbf{z}$, that is, $\mathbf{y}$ is either $\hat{\boldsymbol{\mu}}$, $\hat{\boldsymbol{\sigma}}$, or $\hat{\boldsymbol{\xi}}$ and z is either $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, or $\boldsymbol{\xi}$ after considering spatial dependence. We assume that local estimates can be estimated as $\mathbf{y} = \mathbf{z} + \mathbf{b}$, where $\mathbf{b} \sim N(\mathbf{0}, R_z)$ is zero-mean Gaussian random vector with diagonal covariance matrix R_z . We estimate both the smooth GEV parameters and the smoothness parameter α_z via EM (cf. Section 6.3.3). In the E-step, we compute:

$$\begin{aligned} Q(\alpha_z, \hat{\alpha}_z^{(\kappa)}) &= E_{p(\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa)})}[\log P(\mathbf{y}, \mathbf{z}|\alpha_z)] \\ &= -\frac{1}{2}\alpha_z \{\text{tr}[K_{\text{tm}}(\hat{\alpha}_z^{(\kappa)} K_{\text{tm}} + R_z^{-1})^{-1}] + (\hat{\mathbf{z}}^{(\kappa)})^T K_{\text{tm}} \hat{\mathbf{z}}^{(\kappa)}\} + \frac{1}{2} \log \det(\alpha_z K_{\text{tm}}), \end{aligned} \quad (8.3)$$

where $\hat{\mathbf{z}}^{(\kappa)}$ is computed as in (8.4). Since the posterior distribution of $\mathbf{z}$ is Gaussian, the MAP estimate is also the mean of the posterior, which is given by:

$$\hat{\mathbf{z}} = \arg\max p(\mathbf{z}|\mathbf{y}) = (\alpha_z K_{\text{tm}} + R_z^{-1})^{-1} R_z^{-1} \mathbf{y}. \quad (8.4)$$

The noise covariance R_z can be estimated by the bootstrap approach described in Section 6.3.2. Note that $\hat{\mathbf{z}}$ in (8.4) can be evaluated efficiently with complexity $\mathcal{O}(P \log P)$ as shown in Section 6.3.1.

In the M-step, we select the value $\hat{\alpha}_z^{(\kappa+1)}$ of α_z that maximizes $Q(\alpha_z, \hat{\alpha}_z^{(\kappa)})$. A closed form expression of $\hat{\alpha}_z^{(\kappa+1)}$ exists:

$$\hat{\alpha}_z^{(\kappa+1)} = \frac{P - 1}{\text{tr}[K_{\text{tm}}(\hat{\alpha}_z^{(\kappa)} K_{\text{tm}} + R_z^{-1})^{-1}] + (\hat{\mathbf{z}}^{(\kappa)})^T K_{\text{tm}} \hat{\mathbf{z}}^{(\kappa)}}, \quad (8.5)$$

where P is the number of sites.

8.3 Copula Gaussian Graphical Models

We capture the spatial dependence between the extreme values (block maxima) $x_i^{(n)}$ at the each of the P locations by means of a copula Gaussian graphical models

with the thin-membrane model structure $\mathcal{E}_{\text{tm}}$. The nonzero entries of the precision matrix K of latent Gaussian variables are inferred from the data. An equivalent interpretation is that we impose a thin-membrane model on the latent Gaussian variables, but the smoothness parameter α_{ij} in this case varies for each pair of neighbors $(i, j) \in \mathcal{E}_{\text{tm}}$ in the grid, thus adding extra flexibility to the model. We refer to such models as generalized thin-membrane models, as opposed to the standard thin-membrane models (both regular and irregular) in which only one smoothness parameters is used to describe the spatial dependence.

In the following, we denote the observed GEV variables and hidden Gaussian variables as $\mathbf{x}$ and $\mathbf{z}$ respectively. As introduced in Section 2.3.1.1, the copula Gaussian graphical model can be expressed as:

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, K^{-1}) \quad (8.6)$$

$$x_i = F_i^{-1}(\Phi(z_i)). \quad (8.7)$$

In other words, the copula Gaussian graphical model is determined by the marginals F_i and the precision matrix K . The marginals F_i are GEV distributions, as described in Section 8.2.

In order to infer the precision matrix K , we first transform the non-Gaussian observed variables $\mathbf{x}$ into Gaussian distributed latent variables $\mathbf{z}$:

$$z_i = \Phi^{-1}(F_i(x_i)). \quad (8.8)$$

In the second step, for the given thin-membrane model sparsity structure $\mathcal{E}_{\text{tm}}$, the precision matrix K is estimated from the latent Gaussian variables $\mathbf{z}$:

$$\begin{aligned} \hat{K} &= \underset{K \succ 0}{\operatorname{argmax}} \log \det K - \operatorname{tr}(SK), \\ \text{s.t. } K_{i,j} &= 0 \quad \forall (i, j) \notin \mathcal{E}_{\text{tm}}, \end{aligned} \quad (8.9)$$

where S is the empirical covariance of latent variables $\mathbf{z}$. The convex optimization

problem (8.9) can be solved efficiently by the Newton-CG primal proximal point algorithm [105] or iterative proportional fitting [85].

8.4 Interpolation

Here we explain how extreme values can be inferred at any location L_0 in space, including sites without observations. We assume that the surface of the extreme values is smooth across space. Since both spatial-dependent GEV parameters $\boldsymbol{\mu}$, $\boldsymbol{\xi}$, $\boldsymbol{\sigma}$ (cf. Section 8.2) and the latent variables $\mathbf{z}$ in (8.6) (cf. Section 8.3) share a (generalized) thin-membrane structure, we will first formulate the interpolation problem in a unified form, and then describe the minor differences.

Let $\mathbf{s}$ represent the parameter vectors $\boldsymbol{\mu}$, $\boldsymbol{\xi}$, $\boldsymbol{\sigma}$, or hidden variables $\mathbf{z}$, associated to the sites with measurements, and let s_0 denote the interpolated value at site L_0 (without measurements). The random variables $\mathbf{s}$ and s_0 are assumed to form a thin-membrane model with joint precision matrix:

$$K_0 = \begin{bmatrix} K_{s_0, s_0} & K_{s_0, \mathbf{s}} \\ K_{\mathbf{s}, s_0} & K_{\mathbf{s}, \mathbf{s}} \end{bmatrix}. \quad (8.10)$$

The conditional expected value of s_0 therefore equals:

$$E[s_0 | \mathbf{s}] = -K_{s_0, s_0}^{-1} K_{s_0, \mathbf{s}} \mathbf{s}. \quad (8.11)$$

Since we consider thin-membrane models, s_0 is conditionally dependent on its neighbors only. For a standard thin-membrane model (on regular or irregular grid) as illustrated in Fig. 8.2, the expected value (8.11) can be simplified as:

$$E[s_0 | \mathbf{s}] = \frac{K_{s_0, s_1} s_1 + K_{s_0, s_2} s_2 + K_{s_0, s_3} s_3 + K_{s_0, s_4} s_4}{K_{s_0, s_1} + K_{s_0, s_2} + K_{s_0, s_3} + K_{s_0, s_4}}. \quad (8.12)$$

Clearly, due to the sparse precision matrix corresponding to a thin-membrane model, the expression (8.12) is quite simple. It is natural to choose $K_{s_0, s_j} = -\alpha/d_{0,j}^2$ in stan-

Figure 8.2: Illustration of the modified interpolation method

dard thin-membrane models (both on regular and irregular grids). Consequently, the weighted average (8.12) boils down to inverse distance weighted (IDW) interpolation [152]. We therefore use IDW to interpolate the GEV parameters.

For the latent Gaussian variables $\mathbf{z}$, however, the thin-membrane precision matrix is not a function of distance. On the other hand, for interpolation purposes, we need to consider the distance from sites with observations, and potentially also other parameters. Here we propose a modified inverse distance weighted (MIDW) interpolation method, where K_{s_0, s_j} is not only a function of distance but of direction as well. We assume that K_{s_0, s_j} changes linearly with direction when the distance remains unchanged and is proportional to the inverse square distance when the direction remains unchanged. As illustrated in Fig. 8.2, K_{s_0, s_1} for instance can be computed as:

$$\alpha_{s_0, s_1} = \frac{1}{\frac{\pi}{2}} \left(\alpha_1 \theta_4 + \alpha_4 \theta_1 \right), \quad (8.13)$$

$$K_{s_0, s_1} = -\frac{\alpha_{s_0, s_1}}{d_{s_0, s_1}^2}. \quad (8.14)$$

After interpolating the parameters μ_0 , ξ_0 , σ_0 , and the value of the hidden variable z_0 for site L_0 , we obtain the GEV distributed value of site L_0 through (8.7).

8.5 Numerical Results

In this section, we benchmark the proposed copula MRF-GEV model against a MRF-GEV model (without modeling the extreme value dependence), a copula GEV model (without modeling the GEV parameter dependence), and a Gaussian graphical model (GGM) with the same thin-membrane model structure, both on synthetic and real data sets. We compare all four models by means of three criteria: the mean square error between the interpolated extreme value and the true value, the KL-divergence, and the number of parameters.

For synthetic data, we also compute the mean square error (MSE) for inferring the GEV parameters. Specifically, we report the MSE for (i) local PWM estimates; (ii) spatial-dependent estimates from copula MRF-GEV model; (iii) IDW interpolation resulting from the copula MRF-GEV model at unobserved sites.

8.5.1 Synthetic Data

We generate spatially dependent GEV distributed synthetic data as follows:

1. We generate the coordinates of the 256 observed and 400 unobserved sites. We consider two cases: First, the observed sites are arranged in a regular grid (e.g., wave height measuring stations in the Gulf of Mexico [134]) while unobserved sites are randomly distributed across the grid. Second, both observed and unobserved sites are randomly distributed in the same spatial domain (e.g., precipitation measuring stations in South Africa [67]).
2. We generate 315 samples from a zero-mean multivariate Gaussian distribution, both at observed and unobserved sites. The covariance matrix of that distribution is defined as $\Sigma_{i,j} = \exp(-d_{i,j}^2/\phi)$, where ϕ is the range parameter.

Figure 8.3: True GEV parameters for synthetic data: (a) Location parameter surface; (b) Scale parameter surface.

3. We select GEV parameters for each site. All GEV parameters vary smoothly across space. For both case studies, the location parameter surface is a quadratic Legendre polynomial, as shown in Fig. 8.3(a). The other parameters are chosen differently in each case, as we will explain later.
4. We transform the Gaussian samples generated in Step 2 to GEV distributed samples with the GEV parameters chosen in Step 3 using (2.56).

8.5.1.1 Case Study 1

The observed sites are located on a 16x16 grid, whereas the unobserved sites are randomly distributed, as shown in Fig. 8.4. The shape and scale parameters ξ and σ are chosen to be constant, and are equal to 0.4 and 2 respectively. The results for GEV parameter estimation are summarized in Table 8.1. As mentioned earlier, we report the MSE for (i) local PWM estimates; (ii) spatial-dependent estimates given by the copula MRF-GEV model; (iii) IDW interpolation given by the copula MRF-GEV model at unobserved sites. The GEV estimates by the copula MRF-GEV model are slightly more accurate than the local PWM estimates. The corresponding smoothness parameter $\alpha_\mu = 0.3370$ while α_ξ and α_σ converge to infinity. As a consequence,

Figure 8.4: Coordinates of observed and unobserved sites for Case Study 1.

Table 8.1: Mean square estimation error for GEV parameters in Case Study 1

GEV parameter	PWM	copula MRF-GEV	IDW interpolation
shape parameter ξ	0.0013	9.0334×10^{-4}	9.0334×10^{-4}
scale parameter σ	0.0093	0.0084	0.0084
location parameter μ	0.0114	0.0113	0.0260

the shape and scale parameter do not depend on location, in agreement with the true parameter values. As can also be seen from Table 8.1, IDW parameter interpolation, based on estimates from the copula MRF-GEV model, generates accurate estimates of the GEV parameters at unobserved sites.

Table 8.2 summarizes the performance of the four methods for inferring extreme values. The proposed copula MRF-GEV model has the smallest MSE and KL divergence. Both the MRF-GEV and the copula GEV model fail to describe the spatial extreme values suitably, since they only capture one of the two types of spatial dependence, motivating our approach to model the spatial dependence both for the parameter and extreme values. Interestingly, the GGM has the second best performance in terms of MSE, probably due to the smooth nature of the extreme value surface. However, the KL divergence for the GGM is large compared to the

Table 8.2: Comparison for Case Study 1

Models	MSE	KL-divergence	No. of Parameters
copula MRF-GEV	0.0497	301.8436	738
MRF-GEV	81.0440	646.8706	258
copula GEV	116.0777	302.3067	1248
Gaussian	0.7025	804.4967	736

other models, since GGMs are not capable of capturing extreme events; such models mostly describe fluctuations around the mean value. Compared with the copula GEV model, the copula MRF-GEV model achieves a smaller KL divergence with fewer parameters, suggesting that it is beneficial to model the spatial dependence of the GEV parameters.

We further set the scale parameter surface to be quadratic instead of constant, as shown in Fig. 8.3(b). The results are qualitatively similar. The only difference is that α_σ is now finite (4.9831), implying that the estimates of σ are no longer independent of location. This is not surprising since the true parameter σ follows a quadratic surface. In other words, by inferring smoothness parameters, the smoothness of the GEV parameters can be automatically and suitably adjusted.

8.5.1.2 Case Study 2

In the second scenario, both the observed and unobserved sites are randomly distributed in space, as shown in Fig. 8.5. The adjacency structure of the corresponding irregular thin-membrane model is generated automatically using Delaunay triangulation [153], indicated by the blue lines in the figure.

As in the first scenario, we first set the shape and scale parameters ξ and σ to be constant, and equal to 0.4 and 2 respectively. The results of parameter estimation are presented in Table 8.3. The proposed copula MRF-GEV model yields smaller estimation error than the PWM local estimates. The smoothness parameters α_ξ and α_σ converge to infinity while α_μ is finite (0.2188), in agreement with the true

Figure 8.5: Coordinates of observed and unobserved sites for Case Study 2.

Table 8.3: Mean square estimation error for GEV parameters in Case Study 2

GEV parameter	PWM	copula MRF-GEV	IDW interpolation
shape parameter ξ	2.8527×10^{-4}	9.9447×10^{-5}	9.9447×10^{-5}
scale parameter σ	0.0126	0.0126	0.0084
location parameter μ	0.0157	0.0155	0.0650

underlying model. Table 8.4 summarizes the comparison of the four models. It can be seen that the proposed copula MRF-GEV outperforms the other methods both in terms of MSE and KL divergence, which suggests that the method is also suitable for irregular grids.

Comparing the results with Case Study 1, we notice that by introducing more parameters, the KL-divergence between the copula MRF-GEV model and the data is reduced. Meanwhile, due to the random location of the observed sites, there is a lack of information in some areas compared to the regular grid, resulting in a larger mean square error of interpolation.

Next, as in the first scenario, we set the scale parameter surface to be a quadratic Legendre polynomial as shown in Fig. 8.3(b). Consistent with the true model, the resulting α_σ is finite (2.8716).

Table 8.4: Comparison for Case Study 2

Models	MSE	KL-divergence	No. of Parameters
copula MRF-GEV	0.1869	150.7595	1009
MRF-GEV	95.0509	652.2792	258
copula GEV	99.1437	151.0429	1519
Gaussian	0.2283	690.6725	1007

Figure 8.6: The maximum wave height corresponding to one storm for 256 sites located on 16x16 lattice.

8.5.2 Real Data

In this section, we consider the GOMOS (Gulf of Mexico Oceanographic Study) data [134], which consists of 315 maximum peak wave height values; each corresponds to a hurricane event in the Gulf of Mexico. The distance between each pair of neighbors is 0.125° (approximately 14km). As can be seen from Fig. 8.6, the extreme wave heights vary smoothly over space, indicating strong spatial dependence between extreme values at neighboring sites.

First we analyze a 31x31 lattice in a central region of the Gulf of Mexico (see Fig. 8.7(b)). The nodes of a 16x16 regular lattice are chosen as locations with observations (indicated by black circles in the figure), while the other nodes in the 31x31 lattice are treated as sites without observations. The interpolation results for one storm event are shown in Fig. 8.7. The figure suggests that the copula MRF-

Figure 8.7: Interpolation of the maximum wave heights caused by a storm (in small regular grid). (a) True graph; (b) Observed sites (black) and unobserved sites (red) on the grid; Interpolation by (c) the copula MRF-GEV model; (d) the MRF-GEV model; (e) the copula GEV model; (f) the GGM.

Table 8.5: Comparison for regular grid in the Gulf of Mexico

Models	MSE	KL-divergence	No. of Parameters
copula MRF-GEV	0.0011	238.3892	993
MRF-GEV	17.0022	513.9034	258
copula GEV	1.3863	240.3537	1248
Gaussian	0.0023	506.9120	736

GEV model outperforms the other three models in terms of interpolation accuracy. Results for all 315 storms are summarized in Table 8.5, listing the interpolation mean square error (MSE), KL divergence, and number of parameters for each model. The proposed copula MRF-GEV model achieves the smallest KL divergence and MSE with a relatively small number of parameters. The resulting values of smoothness parameters are $\alpha_\xi = \infty$, $\alpha_\sigma = 127.1875$ and $\alpha_\mu = 78.5047$. The shape parameter ξ is constant, whereas both the scale parameter σ and the location parameter μ change across space.

Figure 8.8: Interpolation of the maximum wave heights caused by a storm (irregular grid covering the Gulf of Mexico). (a) True graph; (b) Observed sites (black) and unobserved sites (red) on the grid; Interpolation by (c) the copula MRF-GEV model; (d) the MRF-GEV model; (e) the copula GEV model; (f) the GGM.

Now we consider the irregular grid with all 4363 measuring sites in the Gulf of Mexico. We randomly select 1000 sites as the locations with measurements, while the remaining sites are regarded as locations without measurements. Fig. 8.8 shows the interpolation by all four methods, for one storm. Again, the copula MRF-GEV model seems to yield the lowest interpolation error. The GGM also performs well but its contour plot fluctuates more than the copula MRF-GEV model, suggesting that the estimates are less reliable. On the other hand, the other two methods

Table 8.6: Comparison for irregular grid in the Gulf of Mexico

Models	MSE	KL-divergence	No. of Parameters
copula MRF-GEV	0.0130	524.4927	5944
MRF-GEV	7.6933	1.7159×10^3	3000
copula GEV	1.6094	529.2659	5944
Gaussian	0.0254	791.1372	3944

cannot correctly interpolate the values for the unobserved sites, since they fail to capture the spatial dependencies among the GEV parameters or the wave heights at different locations.

Table 8.6 shows quantitative results of all four models, computed from all hurricane events. The corresponding smoothness parameters for the copula MRF-GEV model are $\alpha_\xi = 4.8860 \times 10^3$, $\alpha_\sigma = 148.0747$ and $\alpha_\mu = 83.9102$. None of the smoothness parameters converge to infinity, suggesting significant spatial variations in the GEV parameters across the Gulf of Mexico.

8.6 Summary

In this study, we have presented a new model for quantifying dependencies among spatial extreme events. The dependency of the extreme events across space is modeled through a thin-membrane Gaussian copula. Also the parameters of the GEV marginals are coupled in space through thin-membrane models. Numerical results show that the proposed model not only provides accurate estimates and predictions, it is also computationally efficient and practical for both regular and irregular grids, even for large regions such as the entire Gulf of Mexico.

Chapter 9

Modeling Spatial Extremes via Ensemble-of-Trees of Pairwise Copulas

As mentioned in Section 1.1.3, due to the asymptotic tail independence property of the Gaussian copula, CGGMs might not be favorable theoretically when considering all types of extreme phenomena. Instead, in this chapter, after capturing the spatial dependence among GEV parameters (see Section 8.2), we proceed to construct a graphical model for extreme values based on pairwise copulas in which the copula family can be determined from the data. Here, our main idea is to construct graphical models with pairwise (extreme-value) copulas as building blocks. Concretely, we develop ensemble-of-trees models [154] of pairwise copulas (ETPC). As a starting point, the sites in spatial domain are arranged on a lattice (cf. Fig. 9.1(a)). The probability density function (PDF) of the ETPC model is a weighted sum the PDF of all possible spanning trees on that lattice. The PDF of these trees in turn are constructed from pairwise copulas [155]. Different from the original mixture of trees model [156], the number of trees can be determined automatically in the ETPC model. On the other hand, the proposed model is more computationally

efficient compared with the infinite mixtures of trees based on Bayesian nonparametrics [157, 158].

It is worthwhile to emphasize that, under the setting of the ETPC model, the extremes can be modeled as either asymptotically tail dependent or independent [159, 160] by choosing the pairwise copulas appropriately. Furthermore, we have proven that tail dependence in the ETPC model is preserved if all the pairwise copulas in the model are tail dependent. Additionally, to the best of our knowledge, we are the first to propose a scalable inference algorithm called stochastic smoothing-based optimization (SSO) for the ETPC model, so as to impute extreme values at unobserved sites. It is a stochastic approximation of the smoothing-based optimization method, and thus only the gradient of the objective function needs to be evaluated in each iteration. As a result, the SSO algorithm may offer opportunities to further reduce the computational complexity when solving general optimization problems in which gradients can be efficiently computed. Numerical results for extreme precipitation data in Japan suggest that the proposed ETPC model is suitable for spatial extreme-event analysis, in terms of model fitting and extreme value imputation.

This chapter is structured as follows. In Section 9.1, we first explain how to learn the ETPC model, which couples the extreme-value marginal distributions through pairwise copulas, arranged in an ensemble of trees. Inference methods for the ETPC model are then developed in Section 9.2. A theoretical guarantee on tail dependence of the ETPC model is presented in Section 9.3, and numerical results for precipitation data from Japan are provided in Section 9.4. Finally, we offer concluding remarks in Section 9.5.

9.1 Ensemble-of-Trees of Pairwise Copulas

After estimating the spatial dependent GEV marginals as in Section 8.2, we tie the GEV marginal distributions together by means of statistical copulas, i.e., ensemble-of-trees of pairwise copulas (ETPC). As it is convenient to work with tree structured

graphical models (cf. Section 2.1.2), we use trees and pairwise copulas, aside from the marginal density functions, to build the joint density function. Since the absence of cycles in trees renders it restrictive when modeling dependence, we average over all possible spanning trees of the graph. In this way, we can account for all the dependencies in the original structure of the graphical model.

The ETPC model can thus be proceeded as follows. Let $\mathcal{T}_i = (\mathcal{V}, \mathcal{E}_i)$ (see Fig. 9.1(b)-9.1(e)) be a spanning three of the lattice $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (see Fig. 9.1(a)). As illustrated in Section 2.1.2, the PDF of a tree structured graphical model can be written as:

$$f(\mathbf{x}|\mathcal{T}_i) = \prod_{j \in \mathcal{V}} f_j(x_j) \prod_{(j,k) \in \mathcal{E}_i} \frac{f_{jk}(x_j, x_k)}{f_j(x_j)f_k(x_k)}. \quad (9.1)$$

Interestingly, as shown in Section 2.3, the density of pairwise copulas has the following form:

$$c(F_i(x_i), F_j(x_j)) = \frac{f(x_i, x_j)}{f_i(x_i)f_j(x_j)}. \quad (9.2)$$

Therefore, it follows from (9.1) and (9.2) that the PDF of a tree can be simplified as [155]:

$$f(\mathbf{x}|\mathcal{T}_i) = \prod_{j \in \mathcal{V}} f_j(x_j) \prod_{(j,k) \in \mathcal{E}_i} c_{jk}(F_j(x_j), F_k(x_k)), \quad (9.3)$$

that is, the edge potentials of a tree are the corresponding pairwise copulas. Since trees have limited flexibility in modeling dependence, we construct the Ensemble-of-Trees of Pairwise Copulas (ETPC) model by computing the weighted sum over all possible spanning trees of Fig. 9.1(a) as follows:

$$f(\mathbf{x}) = \sum_{\mathcal{T}_i} P(\mathcal{T}_i) f(\mathbf{x}|\mathcal{T}_i), \quad (9.4)$$

where $P(\mathcal{T}_i)$ is a decomposable prior proposed in [161] which assigns the weight of each spanning tree $\mathcal{T}_i$ as the product of the weights β_{jk} of all the edges $(j, k) \in \mathcal{E}_i$

Figure 9.1: ET model: the lattice (a) and several decomposed spanning trees (b) - (e).

in the tree, i.e.,

$$P(\mathcal{T}_i) = \frac{1}{\sum_{\mathcal{T}_i=(\mathcal{V}, \mathcal{E}_i)} \prod_{(j,k) \in \mathcal{E}_i} \beta_{jk}} \prod_{(j,k) \in \mathcal{E}_i} \beta_{jk} \quad (9.5)$$

$$= \frac{1}{Z} \prod_{(j,k) \in \mathcal{E}_i} \beta_{jk}. \quad (9.6)$$

In order to compute the normalizing term Z , we first introduce Kirchhoff's matrix-tree theorem, cf. [162].

Theorem 9.1. (Matrix-tree theorem) *For a graph $\mathcal{G}$, denote by $L(\boldsymbol{\beta})$ its Laplacian matrix with $\boldsymbol{\beta}$ being the edge weight matrix. For any vertices v and w of $\mathcal{G}$, the (weighted) number of spanning trees in $\mathcal{G}$ is given by*

$$\omega(\mathcal{G}) = \sum_{\mathcal{T}_i} \prod_{\{j,k\} \in \mathcal{T}_i} \beta_{jk} = \det Q(\boldsymbol{\beta})_v^u$$

where $Q(\boldsymbol{\beta})_v^u$ is the resulting matrix obtained from deleting the corresponding column of u and row of v .

According to the matrix-tree theorem, the normalizing constant

$$Z = \sum_{\mathcal{T}_i} \prod_{(j,k) \in \mathcal{E}_i} \beta_{jk} = \det[Q(\boldsymbol{\beta})], \quad (9.7)$$

where $\boldsymbol{\beta}$ is the edge weight matrix that is symmetric with the diagonal entries being zero and with the (j, k) entry being β_{jk} . For simplicity, we set $Q(\boldsymbol{\beta})$ to be the first

$P - 1$ rows and columns of the $P \times P$ Laplacian matrix corresponding to a graph with edge weight matrix defined by β .

By substituting (9.3) and (9.5) into (9.4), the ETPC model can be succinctly formulated as [154, 155, 161]:

$$\begin{aligned}
 f(\mathbf{x}) &= \sum_{\mathcal{T}_i} \frac{1}{Z} \prod_{j \in \mathcal{V}} f_j(x_j) \prod_{(j,k) \in \mathcal{E}_i} \beta_{jk} c_{jk}(F_j(x_j), F_k(x_k)) \\
 &= \frac{1}{Z} \left[\prod_{i \in \mathcal{V}_i} f_i(x_i) \right] \sum_{\mathcal{T}_i} \left[\prod_{(j,k) \in \mathcal{E}_i} \beta_{jk} c_{jk}(F_j(x_j), F_k(x_k)) \right] \\
 &= \prod_{j \in \mathcal{V}} f_j(x_j) \frac{\det[Q(\beta \odot \mathbf{c})]}{\det[Q(\beta)]}, \tag{9.8}
 \end{aligned}$$

where $\odot$ denotes componentwise multiplication and $\mathbf{c}$ is the copula density matrix whose (j, k) entry equals $c_{jk}(F_j(x_j), F_k(x_k))$.

In order to learn the ETPC model (9.8), we need to learn three groups of parameters, that is, parameters of the marginal GEV distributions $f_j(x_j)$, parameters of pairwise copulas $c_{jk}(u_j, u_k)$, and the edge weight matrix β . The GEV parameters are estimated using the algorithms in Section 8.2. Under the assumption of the lattice graphical structure (Fig. 9.1(a)), we only need to estimate the parameters of the pairwise copulas $c_{jk}(u_j, u_k)$ corresponding to the pairs of adjacent sites $(j, k) \in \mathcal{E}_{\text{tm}}$. For the sake of efficiency, we apply the Inference for Margins (IFM) method in which the parameters of the copulas are estimated separately from those of the marginal distributions [96]. Specifically, we use the estimated marginals to compute $u_i = F_i(x_i)$, and the copula parameters are further estimated via the Maximum Likelihood method. Moreover, we commit to the Akaike Information Criterion (AIC) when selecting the proper type of copula among the 8 families introduced in Section 2.3, as suggested in [96].

We now proceed to infer β . We intend to estimate β by maximum likelihood given

N samples. The resulting optimization problem is:

$$\begin{aligned} \hat{\beta} = \underset{\beta}{\operatorname{argmin}} \quad & N \log \det[Q(\beta)] - \sum_{n=1}^N \log \det[Q(\beta \odot \mathbf{c}^{(n)})], \\ \text{s. t.} \quad & \beta_{jk} = 0 \quad \forall (j, k) \notin \mathcal{E} \quad , \quad \beta_{jk} \geq 0 \quad \forall (j, k) \in \mathcal{E} \quad \text{and} \quad \|U(\beta)\|_2 = 1, \end{aligned} \quad (9.9)$$

where $\|U(\beta)\|_2 = 1$ is the Euclidean norm for the upper triangular part of the matrix β . We have altered the projection from the unit simplex, as in [154], to the Euclidean norm, as it allows for a larger space of solutions which will aid the optimization algorithm used to solve this problem. Note that the first term in the objective function $N \log \det[Q(\beta)]$ serves as a penalty on the weighted number of trees, hence, the model will automatically choose a small number of trees and reduce the number of parameters. The optimization problem can be solved by applying the Projected Gradient Descent algorithm described in [154] and [163], wherein every iteration computes $\beta_{jk}^{\text{new}} = [\beta_{jk}^{\text{old}} - \alpha(\nabla_{\beta} g)_{jk}]^+$ with g being the objective function. Here, $[\cdot]^+$ denotes projection onto the constrained space and α is the step size determined via Armijo's rule, which is a line search method [163].

As derived in [154], the gradient of the objective function g with respect to β , $(\nabla_{\beta} g)_{jk}$, is computed as follows:

$$(\nabla_{\beta} g)_{jk} = \mathbf{e}_{jk}^T \left\{ N[Q(\beta)]^{-1} - \sum_{n=1}^N c_{jk}^{(n)} (F_j(x_j), F_k(x_k)) [Q(\beta \odot \mathbf{c}^{(n)})]^{-1} \right\} \mathbf{e}_{jk}, \quad (9.10)$$

where $\mathbf{e}_{jk}$ is a $P \times 1$ vector whose only nonzero entries are the j^{th} row with value 1 and the k^{th} row with value -1 for $j, k < P$, and whose only nonzero entry is the 1 at the j^{th} row for $j < P$ and $k = P$. We emphasize that $\nabla_{\beta} g$ is a symmetric matrix and that both $\nabla_{\beta} g$ and the initial $\hat{\beta}_{jk}$ are set to 0 if the edge $\{j, k\}$ does not appear in the regular lattice, satisfying the graph structure constraint as well as reducing the computational complexity. Since the objective function (9.9) is non-convex, a reliable initial estimate can be obtained by solving a convex upper bound that ignores the first term of (9.9) [154]. The overall learning algorithm is summarized

Table 9.1: The learning algorithm of the ETPC model.

1. Estimate the GEV marginals with spatially dependent parameters using the method proposed in Section 8.2:
 - (a) Estimate the GEV parameters locally for each site via the PWM method (cf. Eq. (6.7)).
 - (b) Smooth the GEV parameters across space using the EM algorithm (cf. Eq. (8.3) and (8.5)).
2. Learn the pairwise copulas corresponding to edges in the lattice (see Fig. 9.1(a)) using maximum likelihood method. Choose copula family using AIC.
3. Learn the edge weight matrix β of the ETPC model by solving (9.9) using the projected gradient descent method (cf. Section 9.1). The gradient is given by (9.10).

in Table 9.1.

9.2 Inference of Missing Data

We now turn our attention to inference. To the best of our knowledge, no inference methods have been proposed for the ensemble-of-trees models [154]. Here we consider the inference problem of imputing missing data. Concretely, we aim to infer missing values $\mathbf{x}_M$ at a set of sites given observed data $\mathbf{x}_O$ at other sites. A reasonable approach is maximum a posteriori (MAP) estimation:

$$\hat{\mathbf{x}}_M = \operatorname{argmax}_{\mathbf{x}_M} \log f(\mathbf{x}_M | \mathbf{x}_O) \quad (9.11)$$

$$= \operatorname{argmax}_{\mathbf{x}_M} \sum_{x_j \in \mathbf{x}_M} \log f_j(x_j) + \log \det[Q(\beta \odot \mathbf{c}(\mathbf{x}_M, \mathbf{x}_O))], \quad (9.12)$$

where the notation $\mathbf{c}(\mathbf{x}_M, \mathbf{x}_O)$ signifies that the copula density matrix $\mathbf{c}$ is a function of both $\mathbf{x}_M$ and $\mathbf{x}_O$. In general graphical models (e.g., trees), the problem can be simplified as estimating $\mathbf{x}_M$ given the values of the respective adjacent nodes (cf. Section 2.1). However, for ET model, such simplification is not possible as

indicated by the following theorem.

Theorem 9.2. *Every pair of variables (x_j, x_k) in the ensemble of trees (ET) model corresponding to a connected graph $\mathcal{G}$ with more than one spanning tree are conditionally dependent given other variables in the model.*

Proof. See Appendix C.1. □

As a result, the only alternative is to solve (9.12). Unfortunately, the $\mathcal{O}(P^3)$ computational complexity of evaluating the determinant in (9.12) precludes the use of general convex optimization algorithms, which are prohibitive for high-dimensional problems. Furthermore, due to the complex functional form of copulas (cf. Section 2.3), the objective function often has multiple modes. In order to remedy these two problems, we propose a novel algorithm, named stochastic smoothing-based optimization (SSO). The resulting algorithm does not require the computation of the objective function. Moreover, it can at least attain a significant local maximum. To this end, we first introduce the smoothing-based optimization algorithm, and subsequently, we elaborate on the proposed SSO algorithm and its application to the ETPC model.

9.2.1 Smoothing-based Optimization

Let $h(\mathbf{x})$ be a d -dimensional non-negative function. Its corresponding scale space function can be defined as [164]:

$$H(\mathbf{m}, \nu) = \int h(\mathbf{x}) \mathcal{N}(\mathbf{x}; \mathbf{m}, \nu) d\mathbf{x}, \quad (9.13)$$

where $\mathcal{N}(\mathbf{x}; \mathbf{m}, \nu)$ is a multivariate Gaussian with mean $\mathbf{m}$ and covariance $\nu^2 I$. The right hand side of (9.13) can be viewed as smoothing $h(\mathbf{x})$ with a Gaussian blur kernel with zero mean and covariance $\nu^2 I$ and further evaluating the value of the

smoothed function at point $\mathbf{m}$. Leordeanu *et al.* [165] proposed to find the mode of $h(\mathbf{x})$ by updating $\mathbf{m}$ and ν in each iteration simply as:

$$\mathbf{m}^{(\kappa+1)} = \frac{\int \mathbf{x} h(\mathbf{x}) N(\mathbf{x}; \mathbf{m}^{(\kappa)}, \nu^{(\kappa)}) d\mathbf{x}}{\int h(\mathbf{x}) N(\mathbf{x}; \mathbf{m}^{(\kappa)}, \nu^{(\kappa)}) d\mathbf{x}}, \quad (9.14)$$

$$\nu^{(\kappa+1)} = \sqrt{\frac{\int 1/P \sum_{i=1}^P (x_i - m_i^{(\kappa)})^2 h(\mathbf{x}) N(\mathbf{x}; \mathbf{m}^{(\kappa)}, \nu^{(\kappa)}) d\mathbf{x}}{\int h(\mathbf{x}) N(\mathbf{x}; \mathbf{m}^{(\kappa)}, \nu^{(\kappa)}) d\mathbf{x}}}, \quad (9.15)$$

such that $H(\mathbf{m}^{(\kappa+1)}, \nu^{(\kappa)}) \geq H(\mathbf{m}^{(\kappa)}, \nu^{(\kappa)})$, $H(\mathbf{m}^{(\kappa)}, \nu^{(\kappa+1)}) \geq H(\mathbf{m}^{(\kappa)}, \nu^{(\kappa)})$.

Smoothing-based optimization is in fact a mean shift procedure with adaptively updated kernel variance. The scale space function H gradually approaches h as the standard deviation ν decreases, and recovers h when $\nu = 0$. It is obvious that the maximization of H would localize $\mathcal{N}(\mathbf{x}; \mathbf{m}, \nu)$ as a Dirac delta function (i.e., $\nu = 0$) at the maximum of h . Therefore, the global maximum of H is the same as that of h . However, smoothing h in each iteration with a Gaussian kernel facilitates the convergence to the global maximum, since scale space theory [164] implies that for most functions the local maxima disappear very fast with the increase of the variance of the Gaussian blurring. By initially smoothing h with a large-variance Gaussian kernel and slowly decreasing the variance, the global maximum or at least a significant local maximum can be found [166]. As shown in [165], starting from a large ν , the smoothing-based optimization algorithm can automatically reach (at least) a significant local maximum of h (i.e., escaping from shallow local optima) when $\nu \rightarrow 0$.

9.2.2 Stochastic Smoothing-based Optimization

Notice that the update rules (9.14) and (9.15) are implemented by means of Monte Carlo approximations in [165], thus requiring numerous expensive evaluations of the objective function. Consequently, it is intractable in our case of imputation in the ETPC model.

Let us relax the assumption that the standard deviation ν is the same for all the

variables, and the resulting multivariate Gaussian distribution can be represented as $\mathcal{N}(x; \mathbf{m}, CC^T)$, where C is a diagonal matrix with the vector $\boldsymbol{\nu}$ on the diagonal and therefore CC^T is a diagonal covariance matrix. As a result, $\mathbf{x} = C\mathbf{z} + \mathbf{m}$, where $\mathbf{z}$ follows $\phi(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, I)$ with zero mean and unit variance for all the components. As discussed in Section 7.3.1, we can change variables in (9.13) according to $\mathbf{z} = C^{-1}(\mathbf{x} - \mathbf{m})$, resulting in:

$$H(\mathbf{m}, \boldsymbol{\nu}) = \int h(\mathbf{x}) \mathcal{N}(x; \mathbf{m}, CC^T) d\mathbf{x} \quad (9.16)$$

$$= \int h(C\mathbf{z} + \mathbf{m}) \phi(\mathbf{z}) d\mathbf{z}. \quad (9.17)$$

To maximize $H(\mathbf{m}, \boldsymbol{\nu})$, we consider the gradients over $\mathbf{m}$ and $\boldsymbol{\nu}$, that is,

$$\nabla_{\mathbf{m}} H(\mathbf{m}, \boldsymbol{\nu}) = E_{\phi(\mathbf{z})}[\nabla_{\mathbf{x}} h(\mathbf{x})], \quad (9.18)$$

$$\nabla_{\boldsymbol{\nu}} H(\mathbf{m}, \boldsymbol{\nu}) = E_{\phi(\mathbf{z})}[\nabla_{\mathbf{x}} h(\mathbf{x})] \odot \mathbf{z}. \quad (9.19)$$

In other words, the gradients in (9.18) and (9.19) can be expressed as the expectation of exact quantities. As such, when $\nabla_{\mathbf{x}} h(\mathbf{x})$ can be calculated easily, $H(\mathbf{m}, \boldsymbol{\nu})$ can be maximized efficiently via stochastic optimization [142]. Such algorithms iteratively update the solution by means of noisy estimates of the gradient. The latter are often easier to compute than the true gradient. In particular, we approximate the expectation in (9.18) and (9.19) by one realization of $\nabla_{\mathbf{x}} h(\mathbf{x})$, namely, drawing one sample $\hat{\mathbf{z}}$ from $\phi(\mathbf{z})$ and evaluating $\nabla_{\mathbf{x}} h(\mathbf{x})$ at $\hat{\mathbf{x}} = \boldsymbol{\nu} \odot \hat{\mathbf{z}} + \mathbf{m}$. More precisely, beginning with a large $\boldsymbol{\nu}$, the SSO algorithm performs the following steps in each iteration:

$$\hat{\mathbf{z}} \sim \phi(\mathbf{z}), \quad (9.20)$$

$$\hat{\mathbf{x}}^{(\kappa)} = \boldsymbol{\nu}^{(\kappa)} \odot \hat{\mathbf{z}} + \mathbf{m}^{(\kappa)}, \quad (9.21)$$

$$\mathbf{m}^{(\kappa+1)} = \mathbf{m}^{(\kappa)} + \rho_{\kappa} \nabla_{\mathbf{x}} h(\mathbf{x})|_{\mathbf{x}=\hat{\mathbf{x}}^{(\kappa)}}, \quad (9.22)$$

$$\boldsymbol{\nu}^{(\kappa+1)} = \boldsymbol{\nu}^{(\kappa)} + \rho_{\kappa} \nabla_{\mathbf{x}} h(\mathbf{x})|_{\mathbf{x}=\hat{\mathbf{x}}^{(\kappa)}} \odot \mathbf{z}, \quad (9.23)$$

where $\nabla_{\mathbf{x}}h(\hat{\mathbf{x}}^{(\kappa)})$ is the value of $\nabla_{\mathbf{x}}h(\mathbf{x})$ at $\hat{\mathbf{x}}^{(\kappa)}$, and ρ_{κ} is the step size in iteration κ . As stated by the theory of stochastic optimization [142], the algorithm is convergent if using a series of step sizes ρ_{κ} such that $\sum \rho_{\kappa} = \infty$ and $\sum \rho_{\kappa}^2 = 0$. In our case of SSO, we simply initialize $\rho_1 = 1e^{-4}$ and multiple it by 0.95 every 5000 iterations, in order to guarantee that the variance of the Gaussian kernel changes slowly. By iterating (9.20) to (9.23) until all the components of $\boldsymbol{\nu}$ are sufficiently small, the resulting $\mathbf{m}$ is a significant local optimum that maximizes $h(\mathbf{x})$.

9.2.3 SSO for Imputation in the ETPC Model

Let us denote the objective function in (9.12) as $h(\mathbf{x}_M)$. The only challenge of applying the SSO algorithm lies in efficient computation of the gradient $\nabla_{\mathbf{x}_M}h(\mathbf{x}_M)$. For each $x_j \in \mathbf{x}_M$, the corresponding partial derivative can be computed as:

$$\frac{\partial h(\mathbf{x}_M)}{\partial x_j} = \left\{ \sum_{k \in N(j)} \frac{\partial \log \det[Q(\boldsymbol{\beta} \odot \mathbf{c})]}{\partial c_{jk}} \frac{\partial c_{jk}}{\partial u_j} \right\} \frac{du_j}{dx_j} + \frac{df_j(x_j)}{dx_j}, \quad (9.24)$$

where $c_{jk} = c_{jk}(u_j, u_k)$, $u_j = F_j(x_j)$, and $N(j)$ includes all the neighbors of node j in the lattice. We next analyze each term in (9.24) in turn. It is easy to compute $du_j/dx_j = f_j(x_j)$ and $df_j(x_j)/dx_j$ in closed-form expressions, so we will not provide further details. In addition, due to the complicated functional form copula densities c_{jk} , we instead evaluate the partial derivative numerically as:

$$\frac{\partial c_{jk}}{\partial u_j} = \frac{c_{jk}(u_j + t, u_k) - c_{jk}(u_j - t, u_k)}{2t}, \quad (9.25)$$

for $t = 10^{-4}$. The remaining term $\partial \log \det[Q(\boldsymbol{\beta} \odot \mathbf{c})]/\partial c_{jk}$ equals:

$$\frac{\partial \log \det[Q(\boldsymbol{\beta} \odot \mathbf{c})]}{\partial c_{jk}} = \beta_{jk} \mathbf{e}_{jk}^T Q(\boldsymbol{\beta} \odot \mathbf{c})^{-1} \mathbf{e}_{jk}, \quad (9.26)$$

where $\mathbf{e}_{jk}$ is the $P \times 1$ vector defined in Expression (9.10). Note that $Q(\boldsymbol{\beta} \odot \mathbf{c})$ is a sparse matrix, corresponding to a lattice excluding the last node and all the edges containing the node. Consequently, there are many methods available (cf. Sec-

Table 9.2: The imputation algorithm of the ETPC model.

<p>Begin with a large $\boldsymbol{\nu}$. Iterate the following steps until all the components of $\boldsymbol{\nu}$ are sufficiently small.</p> <ol style="list-style-type: none"> 1. Simulate $\mathbf{x}_M^{(\kappa)}$ from the multivariate Gaussian distribution $\mathcal{N}(x; \mathbf{m}, CC^T)$: $\hat{\mathbf{z}} \sim \phi(\mathbf{z}), \quad \mathbf{x}_M^{(\kappa)} = \boldsymbol{\nu}^{(\kappa)} \odot \hat{\mathbf{z}} + \mathbf{m}^{(\kappa)}.$ 2. Compute the gradient $\nabla_{\mathbf{x}_M} h(\mathbf{x}_M)$ at $\mathbf{x}_M^{(\kappa)}$ (cf. Eq. (9.24)-(9.26)) and update $\mathbf{m}$ and $\boldsymbol{\nu}$ as follows: $\begin{aligned} \mathbf{m}^{(\kappa+1)} &= \mathbf{m}^{(\kappa)} + \rho_\kappa \nabla_{\mathbf{x}_M} h(\mathbf{x}_M) \big _{\mathbf{x}_M = \hat{\mathbf{x}}_M^{(\kappa)}}, \\ \boldsymbol{\nu}^{(\kappa+1)} &= \boldsymbol{\nu}^{(\kappa)} + \rho_\kappa \nabla_{\mathbf{x}_M} h(\mathbf{x}_M) \big _{\mathbf{x}_M = \hat{\mathbf{x}}_M^{(\kappa)}} \odot \mathbf{z}. \end{aligned}$

tion 2.1.5) to obtain the linear system solution $Q(\boldsymbol{\beta} \odot \mathbf{c})^{-1} \mathbf{e}_{jk}$ with complexity linear in the number of variables (sites) P . Under the assumption that each missing variable has four neighbors, the overall computational complexity of the SSO imputation algorithm is $\mathcal{O}(P_M P)$, where P_M is the number of missing variables. Recall that the complexity of solving (9.12) via general optimization approaches is $\mathcal{O}(P^3)$, therefore, the proposed inference method yields significant savings in computational complexity. The overall imputation algorithm is summarized in Table 9.2.

9.3 Theoretical Results

In this section, we explore the tail properties of the ETPC model. Theoretically, tail dependence is essential to a multivariate extreme value model, as multivariate extreme value theory and methods can be regarded as the characterization, estimation and extrapolation of the joint tails of multidimensional distributions. Practically, one often considers the probability of extreme events occurring at one site given extreme extremes occurring at other sites. Such type of dependence is well encoded in the tail dependence of a model (cf. Eq. (2.49)). In the following, we start this section with a key proposition about the tail dependence between two variables that

are conditionally independent given a third variable [167]. We then generalize the results to two arbitrary nodes in a tree and further in the ETPC model.

Proposition 9.1. *Suppose that (X_1, X_2, X_3) form a Markov chain where X_1 and X_3 are conditionally independent given X_2 and (X_1, X_2) and (X_2, X_3) have upper tail dependence with coefficients λ_{12} and λ_{23} respectively. Moreover, assume that X_1 and X_3 are stochastically increasing with respect to X_2 . Then (X_1, X_3) has upper tail dependence with coefficient λ_{13} bounded below by $\lambda_{12}\lambda_{23}$.*

Proof. See Appendix C.2 □

Proposition 9.1 implies that two conditionally independent variables can be tail dependent, therefore, it sets up the foundation of analyzing tail dependence in an extreme-value graphical model. Based on Proposition 9.1, we next consider the lower bound on the upper tail dependence of two arbitrary variables in a tree.

Proposition 9.2. *Given a tree graphical model $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with the corresponding set of random variables $\mathbf{X} = (X_i)_{i \in \mathcal{V}}$ whose joint PDF can be written as in (9.3), the upper tail dependence between all components of $\mathbf{X}$ exist if all the pairwise copulas c_{ij} corresponding to the edges $\mathcal{E}$ in the tree are upper tail dependent and X_i and X_j (variables conditionally dependent in the graph) are stochastically increasing with each other. Specifically, for any two variables X_i and X_j in the graph, the lower bound of their tail dependence coefficient is the product of the tail dependence coefficients of all the pairwise copulas corresponding to the edges in the path connecting X_i and X_j .*

Proof. Proposition 9.2 can be proven based on Proposition 9.1 via induction. See Appendix C.3. □

Proposition 9.2 allows us to prove the existence of tail dependence between any two variables in a tree graphical model. Based on the definition of upper tail dependence,

it is easy to prove that the tail dependence still exists in the Ensemble-of-Trees model if it exists in one tree graphical model. More explicitly, for a mixture of trees model,

$$\lambda_{ij} = \lim_{u \rightarrow 1^-} P(U_i > u | U_j > u) \quad (9.27)$$

$$= \sum_i \lim_{u \rightarrow 1^-} P(U_i > u | U_j > u; \mathcal{T}_i) P(\mathcal{T}_i). \quad (9.28)$$

We now present a closed form expression for the lower bound of the upper tail dependence of any two nodes in the ETPC model. This is a consequence of the following lemma.

Lemma 9.1. *A finite weighted sum of bivariate copulas, of which at least one is upper tail dependent, preserves upper tail dependence.*

Proof. This follows directly from the fact that the limit of a finite sum is the sum of the limit of each addend. $\square$

Proposition 9.3. *The upper tail dependence between any two nodes in the Ensemble-of-Trees model is bounded below by $\det[Q(\boldsymbol{\beta} \odot \boldsymbol{\lambda})] / \det[Q(\boldsymbol{\beta})]$ where $Q(X)$ takes the first $P - 1$ rows and columns of the Laplacian matrix corresponding to the graph defined by the $P \times P$ edge weight matrix X , $\boldsymbol{\lambda}$ is the pairwise upper tail dependence coefficient matrix, and $\odot$ denotes componentwise multiplication.*

Proof. See Appendix C.4. $\square$

Therefore, by properly selecting pairwise copulas, the ETPC model can flexibly describe the dependencies between extreme values, both in a tail dependent and independent manner. In practice, hurricane-induced extreme wave heights are deemed to be tail independent [168], whereas extreme temperature and precipitation often exhibit tail dependence [169]. Instead of using different models for different type of extreme value data, the ETPC model can handle various extreme value data in a unified framework.

Figure 9.2: Normalized histogram (blue bars) and estimated GEV distribution (red lines) for one randomly selected site in each of the four regions.

9.4 Experimental Results

To assess our model, we consider the extreme precipitation in the Japanese archipelago. The daily rainfall data from 1900-2011 is compiled and interpolated onto a grid with resolution 0.05° [151]. We select four 10×10 regular grids in South Japan, where heavy rainfall is often the cause of floods. We extract the extreme precipitation values from the sites in each of the 4 regions as follows: We compute the overall daily rainfall amount in that region. We next choose a high enough threshold and retain those precipitation events (possibly lasting for several days) with rainfall amount above a threshold, resulting in fewer than 10 events per year. For each rainfall event in the region, we isolate the maximum daily rainfall amount for each site.

As shown in Fig. 9.2, we observe that the resulting histogram of each site accurately resembles the GEV distribution. Moreover, we also verify empirically whether pairwise upper tail dependence exists in the data using the method in [170]. Concretely, if x_i, x_j denote extreme precipitation at two different sites, we transform them to be Gaussian distributed via $y_i = \Phi^{-1}(F_i(x_i))$ and $y_j = \Phi^{-1}(F_j(x_j))$ where Φ is the standard normal CDF and F_i, F_j are the marginal GEV distributions for x_i, x_j . The plots for y_i and y_j are shown in Fig. 9.3 for the four arbitrarily selected pairs of sites (each in one region). If Gaussian copula fits the data well, the distribution of y_i and y_j would be Gaussian in the latent Gaussian layer. This further implies that there is no tail dependence between x_i and x_j . In our case, however, the pointed contour lines at the upper right corner suggest that the pairs of sites manifest upper tail dependence. Thus, we expect that the upper tail dependent copulas are the most

Figure 9.3: Scatter plot of transformed extreme precipitation values for a randomly selected pair of sites at the Gaussian layer for each region.

suitable to model such type of dependence.

For each of the four regions, we compare two possible configurations under the Ensemble-of-Trees model: 1) the case where pairwise copula selection is performed through the minimization of the AIC (ETPC model with a mixture of copulas, denoted as “ETPCm”) and 2) the case when all pairwise copulas are Gaussian (ETPC model with Gaussian copula, denoted as “ETPCG”). Additionally, we also benchmark the proposed model with a lattice-structured copula Gaussian graphical model (CGGM) (cf. Chapter 8). Note that all three models are based on the same set of spatially dependent GEV marginals (cf. Section 8.2). As the testing data, we retain five samples corresponding to the 99th, 90th, 80th, 70th, and 60th quantiles of overall rainfall in the region, while the remaining data points are treated as training data. Then, we compute the AIC of the three models (when fitting the training data), and also the mean absolute error (MAE) of the imputation results, computed as the averaged absolute difference between the true and estimated data over all unobserved sites.

9.4.1 Model Fitting

The results of model fitting are listed in Table 9.3. In particular, for ETPCm model, almost all of the chosen copulas are tail dependent which is consistent with the empirical analysis we have performed earlier. Furthermore, most of the selected copulas belong to the extreme-value family, especially for Region 2 and 3. We then compute

Table 9.3: Comparison of goodness of fit of three models.

Region No.	ETPCm									ETPCG	CGGM
	Gaussian	t	Gumbel	Clayton	Frank	Galambos	t -EV	Hüsler-Reiss	AIC	AIC	AIC
Region 1	2	85	30	0	0	8	0	55	-2.71×10^5	-2.61×10^5	-2.69×10^5
Region 2	0	0	45	0	0	11	2	122	-3.01×10^5	-2.86×10^5	-2.90×10^5
Region 3	0	0	38	0	0	24	4	114	-3.75×10^5	-3.51×10^5	-3.59×10^5
Region 4	1	30	44	0	0	17	3	85	-2.27×10^5	-2.17×10^5	-2.20×10^5

the overall value of the AIC for the two ETPC models and the CGGM. Apparently, the ETPCm model has the smallest AIC value, implying that it possesses the best fit among the three models. Moreover, we can see that the AIC score of the CGGM is smaller than that of the ETPCG model, since the cyclic structure of the CGGM can better capture the spatial dependence.

9.4.2 Imputation of Missing Data

We next present the results for the imputation algorithm of Section 9.2 on the Japanese precipitation data set. Note that for the CGGM, we simply maximize the conditional density $p(\mathbf{x}_M|\mathbf{x}_O)$ via the smoothing-based optimization algorithm to obtain the estimates of the missing data. For each testing sample, we consider 5 instances in which the data is missing in a 2×2 , 3×3 , 4×4 , 5×5 , and 6×6 area at the middle of the 10×10 grid. The resulting MAE are summarized in Table 9.4. We also show the color plots in Fig. 9.4 for the imputation results of the most challenging case, i.e., a missing area of 6×6 grid for the 99th quantile inside Region 1.

There are three major trends that can be gleaned from the results. First, most MAE values for the ETPCm model are relatively small, since the precipitation values are typically a few hundred mm. This implies that the imputation algorithm proposed in Section 9.2 yields reliable estimates. Second, for large quantiles (e.g., 90th and 99th), as the area of the region containing the missing values increases, the performance of the ETPCG model and the CGGM deteriorates more seriously compared to that of the ETPCm model. As the region with missing values becomes larger,

Table 9.4: Summary of the MAE (mm) of imputed precipitation values with respect to the length of the square region of missing values for 5 quantiles.

Area	99 th quantile			90 th quantile			80 th quantile			70 th quantile			60 th quantile			
	ETPCm	ETPCG	CGGM	ETPCm	ETPCG	CGGM	ETPCm	ETPCG	CGGM	ETPCm	ETPCG	CGGM	ETPCm	ETPCG	CGGM	
Region 1	2×2	3.06	7.20	3.37	6.31	11.39	9.51	11.25	11.18	10.05	2.84	4.27	3.13	2.12	1.89	2.18
	3×3	3.83	8.74	6.06	6.52	10.88	7.52	12.82	8.53	7.53	2.32	3.79	2.69	4.09	1.93	2.68
	4×4	5.59	13.63	5.92	8.28	11.66	8.15	14.76	12.12	11.66	2.03	3.96	2.27	3.44	6.16	3.59
	5×5	6.18	12.27	6.90	9.77	16.69	12.71	12.89	11.67	11.69	4.61	7.43	5.02	7.63	6.99	3.22
	6×6	7.06	17.12	12.56	11.65	17.05	15.66	13.73	13.92	11.56	4.80	8.61	8.23	6.24	6.75	4.18
Region 2	2×2	10.96	16.92	21.25	13.05	16.36	13.81	4.98	14.50	15.37	6.28	3.47	2.95	1.64	1.89	3.87
	3×3	12.28	13.72	15.49	13.29	18.68	15.03	14.09	13.06	12.85	3.06	1.80	1.74	7.54	4.17	4.76
	4×4	13.54	30.54	20.13	11.66	17.60	14.08	12.79	12.81	11.95	4.29	5.54	4.32	7.07	4.85	5.21
	5×5	17.50	43.37	44.94	19.36	26.44	28.15	10.57	9.63	9.48	8.04	9.93	7.52	5.73	4.70	5.51
	6×6	35.65	59.64	62.14	16.82	23.90	27.51	9.92	10.00	10.56	8.10	9.73	8.29	5.39	5.79	6.76
Region 3	2×2	2.70	4.68	4.99	3.52	4.77	3.02	2.46	1.82	1.44	8.40	8.94	3.73	3.02	2.86	2.22
	3×3	4.92	9.31	6.52	8.74	11.50	7.89	2.03	2.51	3.13	11.12	13.50	5.34	3.10	4.29	2.88
	4×4	6.18	12.39	8.05	10.11	12.06	10.95	3.18	2.85	4.12	11.63	13.25	6.44	4.42	3.99	3.76
	5×5	7.95	15.39	8.40	11.55	13.58	13.52	4.71	4.21	3.74	11.93	12.70	8.14	4.99	4.59	4.80
	6×6	9.50	20.66	13.94	14.98	18.71	18.17	8.35	6.04	4.62	13.94	16.66	10.47	7.73	6.59	6.12
Region 4	2×2	17.22	19.18	22.86	9.95	15.65	12.59	2.95	3.35	2.99	1.03	1.32	1.30	0.94	2.39	1.93
	3×3	21.32	27.88	29.12	15.47	18.18	20.33	2.55	3.98	4.42	3.50	3.78	2.15	3.45	4.68	4.51
	4×4	37.38	48.57	46.24	13.00	18.08	18.86	2.71	4.85	4.63	5.69	5.12	3.05	3.76	4.66	5.04
	5×5	41.90	52.84	45.91	15.17	25.75	24.28	3.95	7.96	5.11	3.99	4.04	3.50	6.27	7.08	6.92
	6×6	42.04	58.69	56.71	18.01	27.55	24.08	4.05	7.19	5.45	7.47	9.43	4.42	9.23	9.75	9.95

Figure 9.4: Color plots of the imputed 6×6 area in the first region for the 99th quantile.

the correlation between two distant nodes decreases. As a result, the Gaussian copula is unable to capture the upper tail dependence in the data, rendering high values for the MAE over the mixture of copulas model comprised of upper tail dependent copulas. In particular, as manifested in Fig 9.4, the ETPCG model and the CGGM are unable to recover reliable estimates for the maximum precipitation values, especially for the 6×6 region with missing values. While all three models underestimate the largest precipitation amount, the ETPCm model yields more reliable estimates. Intuitively, this is because the rainfall amounts for this case are extreme and reside in the tail part of the distribution, which the Gaussian copula is incapable of modeling. This also implies that the proposed model can reliably simulate the extreme events in a spatial domain given the boundary conditions. Third, the CGGM tends to outperform the ETPCG model, since cyclic graphs offer a more natural description of the spatial dependence than the ensemble-of-trees configuration. Unfortunately, the cyclic structure can only be imposed on Gaussian copulas so far, whereas Gaussian copulas fail to capture tail dependence both empirically (as shown in our experiments) and theoretically. Therefore, the ETPCm model is currently preferable in terms of computational convenience and tail properties.

9.5 Summary

We have proposed the ETPC model for multivariate analysis of spatial extremes, together with tractable and efficient learning and imputation algorithms. The ETPC model enables us to construct the joint PDF in a flexible manner to accommodate various types of copulas for different modeling purposes, depending on the data at hand. Furthermore, it enjoys the attractive property of preserving the upper tail dependence between any two nodes under certain mild conditions, which assists in the analysis of extreme events in a spatial domain. To the best of our knowledge, this is the first work on tail dependence in graphical models. Our experimental results on real Japanese rainfall data further shows the ETPCm model outperforms the ETPCG model and the CGGM in terms of both model fitting and inferring extreme missing data, indicating that the ETPC model serves as a powerful and flexible tool to model spatial extremes in practice.

Chapter 10

Conclusions and Outlook

This thesis investigates graphical models for non-Gaussian data, including both nominal and extremal data. Particularly, copulas play an important role in our approach to constructing those graphical models. In this chapter, we close this thesis by summarizing the main contributions and suggesting directions for future research.

10.1 Summary of Contributions

10.1.1 Graphical Models for Nominal Data

We first consider non-Gaussian nominal data in this thesis; we propose novel copula Gaussian graphical models to handle continuous, discrete and piecewise stationary data respectively in Chapter 3, Chapter 4, and Chapter 5.

For continuous data, we develop hidden variable copula Gaussian graphical models; such models allow us to infer hidden variables from non-Gaussian multivariate data, and therefore provide a simple explanation for the statistical relations between the observed variables. Specifically, we first transform non-Gaussian observed variables to Gaussian latent variables, according to the definition of Gaussian copulas. We

then infer a hidden variable Gaussian graphical model in the latent Gaussian layer by solving a penalized maximum likelihood problem. The resulting model enjoys the advantages of Gaussian graphical models, such as efficient learning and inference, while being applicable to a much wider variety of continuous distributions. In addition, we solve the problem of regularization selection by first learning the graphical model structure via stability selection, and then estimating the parameters. The results of both synthetic and real data shows that the proposed algorithm can effectively recover the underlying graphical model structure and the number of hidden variables.

For discrete data, we propose to learn discrete copula Gaussian graphical models using the Monte-Carlo Expectation Maximization (MCEM) algorithm. Concretely, in the E-step, we employ an efficient Gibbs sampler to draw samples for the latent Gaussian variables; in the M-step, we adopt the efficient glasso algorithm to learn the precision matrix of the latent Gaussian variables. The resulting algorithm is more efficient than the existing full MCMC algorithm as shown in Chapter 4, since only point estimates rather than the posterior distribution of the precision matrix are estimated. The selection of the regularization parameter is again a critical issue in this model; we circumvent such a delicate problem as in the previous chapter by splitting the graphical model learning procedure into two steps: structure and parameter learning. The structure is determined using BINCO method, since it provides a more reasonable method to select “stable” edges than stability selection. The parameters, on the other hand, are inferred via iterative proportional fitting (IPF). We further apply the model to one synthetic data set and three real data sets. The results indicate that the proposed method produces very similar results to alternative approaches, while being more efficient and wider applicable.

In the last chapter for nominal data, we build a model to detect change points in piecewise-stationary data, and then infer graphical models for each stationary time segments. Previous works on learning abruptly changing dependency structure for multivariate time series usually suffer from the issue of high computational cost. Instead, we show that a method with linear time complexity (under certain condi-

tions) can be leveraged to identify the change points. We also develop an approach to specify the number of change points in an automatic fashion. After detecting the change points, graphical models are then inferred for each time segment using the methods described in the previous two chapters. The crux of the proposed approach is that it determines the number and location of the change points as well as the graphical models automatically without manually tuning any parameters. Numerical results for both synthetic and real data show that the proposed method provides an effective and efficient tool to identify change points and infer networks.

10.1.2 Graphical Models for Marginal Analysis of Extremal Data

After developing models for nominal data, we turn our attention to extremal data. Modeling extreme events can be divided into two stages: marginal and joint analysis. The objective of marginal analysis is to accurately estimate the parameters of marginal distributions. We construct graphical models for extreme events with multiple covariates in Chapter 6, and subsequently, pay special attention to spatio-temporal extreme events in Chapter 7.

Chapter 6 is devoted to the novel extreme-value model which accommodates the effects of multiple covariates. More explicitly, marginal extreme values are assumed to follow GEV distributions. The GEV parameters are then coupled together through a multidimensional thin-membrane model. The proposed model possesses three advantages: First, the multidimensional thin-membrane model can be constructed from one-dimensional thin-membrane models in an flexible manner, thus, the proposed model is generalizable to an arbitrary number of covariates. Second, component eigenvector structures provide efficient inference of smoothed GEV parameters with computational complexity $\mathcal{O}(M \log(M))$. Third, the eigenvalues of the overall multidimensional model can be computed easily, and help to simplify the determinant maximization problem in the learning process of smoothness parameters. Numerical results for both synthetic and real data support the proposed model;

it achieves the most accurate estimates of GEV parameters, and models the data best. Therefore, the approach may prove to be a practical tool for modeling extreme events with covariates.

On the other hand, we propose a new spatial-temporal model for marginal extremes in Chapter 7. Specifically, we assume each GEV parameter can be decomposed as the sum of a spatial and a temporal component, as in a generalized additive model. Graphical models, in particular, thin-plate models, are then imposed on the spatial and the temporal components to capture the spatial and temporal dependence. Compared with previous works that restrict the shape and scale parameters to be constant, the proposed model has the virtue that all the three GEV parameters (shape, scale, and location) are able to change flexibly in the spatio-temporal domain. Furthermore, an efficient algorithm whose computational complexity is sublinear is derived to learn the GEV parameters as well as the smoothness parameters of the thin-plate models. Numerical results show that the proposed model can reliably estimate the GEV distributions corresponding to the observed time series. It also provides an effective tool to predict GEV distributions in the future.

10.1.3 Graphical Models for Joint Analysis of Extremal Data

Finally, we proceed to the second stage of extreme-events modeling. Here, we focus on spatial extremes, and we aim to model the spatial dependence between extreme values rather than GEV parameters.

In Chapter 8, we investigate the performance of copula Gaussian graphical models on extreme events. The resulting model captures the spatial dependence in two complementary ways. The GEV parameters are coupled through a thin-membrane model. The extreme events, on the other hand, are coupled through a copula Gaussian graphical model (CGGM) with the precision matrix corresponding to a generalized thin-membrane model. We then derive inference and interpolation algorithms for the proposed model. We consider both regular and irregular grids. The approach is validated on synthetic data as well as real data related to hurricanes in the Gulf

of Mexico. Numerical results suggest that it can accurately describe extreme events in spatial domain, and can reliably interpolate extreme values at arbitrary sites.

Since Gaussian copulas are asymptotically tail independent and are therefore theoretically unfavorable to analysis of extreme events, we consider in Chapter 9 an extreme-value graphical model built on other copulas, in particular, the ensemble-of-trees of pairwise copulas (ETPC). We describe the learning algorithm and further propose an efficient inference with linear computational complexity. More importantly, we prove that the ETPC model can preserve the upper tail dependence between any two nodes under mild conditions, thus, it is suitable for extreme-value analysis. The experimental results on real Japanese rainfall data also verifies that the ETPC model wins over the CGGM in terms of both model fitting and inferring extreme missing data, suggesting that the ETPC model can well characterize the spatial dependence between extremes in practice.

10.2 Recommendations for Future Works

10.2.1 Cyclic Pairwise Copula Graphical Models

As demonstrated in Table 9.3 and Table 9.4 in Chapter 9, cyclic graphs can be a better choice to analyze spatial dependence than the ensemble of trees. So far, it is unclear how to build cyclic graphical models from non-Gaussian pairwise copulas. Therefore, we would like to employ non-Gaussian copulas as the building blocks of cyclic graphical models in our future work.

10.2.2 Multiscale Pairwise Copula Graphical Models

Another interesting direction is to build a multiscale graphical model based on pairwise copulas. Multiscale models are equipped with the richest modeling power and are capable of capturing complex dependence, such as the long-range dependence between far-away sites (e.g., the dependency between nodes X_1 and X_P in

Figure 10.1: Graphical models with multiscale structure: (a) a monoscale graphical model; (b) a latent tree (i.e., a quadtree); (c) a multiscale graphical model.

Fig. 10.1(a)) [171, 172]. As a starting point, we may consider a multi-scale tree model, as shown in Fig. 10.1(b). However, the multi-scale tree model induces blocky artifacts, since some spatially adjacent variables are widely separated in the tree structure [172]. To alleviate that issue, we will consider the ensemble of multi-scale trees [122] or models with inscale structure [172], which capture short-range correlation in each scale of different resolution

Appendix A

Proofs and Derivations of Results in Chapter 6

A.1 Derivation of the first step in the FTM solver

Due to the mixed product property of the Kronecker product [127, Ch. 13], V_{prior} in (6.20) can be expressed alternatively as the matrix product:

$$\begin{aligned} V_{\text{prior}} = & (I_{B_x} \otimes I_{B_y} \otimes V_C)(I_{B_x} \otimes V_{B_y} \otimes I_C) \\ & \times (V_{B_x} \otimes I_{B_y} \otimes I_C). \end{aligned} \quad (\text{A.1})$$

Therefore, the calculation of $\mathbf{z}_1 = V_{\text{prior}}\mathbf{z}_0$ can be further divided into three substeps. Before elaborating on the three substeps, we first introduce another property of the Kronecker product: for two arbitrary matrices J and K that can be multiplied together,

$$\text{vec}(JK) = (I_K \otimes J)\text{vec}(K) \quad (\text{A.2})$$

$$= (K^T \otimes I_J)\text{vec}(J). \quad (\text{A.3})$$

Now let us focus on the first substep (as denoted by the superscript), that is, $\mathbf{z}_1^1 =$

$(V_{B_x} \otimes I_{B_y} \otimes I_C) \mathbf{z}_0$. Given the above mentioned property, $\mathbf{z}_1^1$ can be computed as:

$$\mathbf{z}_1^1 = \text{vec}(Z_0^1 V_{B_x}^T) = \text{vec}\{(V_{B_x} Z_0^{1T})^T\}, \quad (\text{A.4})$$

where Z_0^1 is a P by QD matrix such that $\text{vec}(Z_0^1) = \mathbf{z}_0$, and P , Q and D are the dimension of V_{B_x} , V_{B_y} and V_C respectively. In addition, recall that we define Z_0 to be a 3D P by Q by D array such that $\text{vec}(Z_0) = \mathbf{z}_0$ as in Section 6.3.1. As a result, it is easy to express Z_0^{1T} using the entries of Z_0 :

$$Z_0^{1T} = \begin{bmatrix} [Z_0]_{111} & [Z_0]_{121} & \cdots & [Z_0]_{1Q1} & \cdots & [Z_0]_{1QD} \\ [Z_0]_{211} & [Z_0]_{221} & \cdots & [Z_0]_{2Q1} & \cdots & [Z_0]_{2QD} \\ \vdots & \vdots & & \vdots & & \vdots \\ [Z_0]_{P11} & [Z_0]_{P21} & \cdots & [Z_0]_{PQ1} & \cdots & [Z_0]_{PQD} \end{bmatrix}.$$

As mentioned in Section 6.1, $V_{B_x} Z_0^{1T}$ is equivalent to discrete Fourier transform on each column of Z_0^{1T} , and therefore, it coincides with performing FFT along the first dimension of the 3D array Z_0 .

Similarly, in the second substep, we can first apply the property in Expression (A.2), and subsequently, in Expression (A.3). As a result, we can find that $\mathbf{z}_1^2 = (I_{B_x} \otimes V_{B_y} \otimes I_C) \mathbf{z}_1^1$ is identical to FFT in the second dimension of the 3D P by Q by Q array Z_1^1 which satisfies $\text{vec}(Z_1^1) = \mathbf{z}_1^1$. The operation in the third substep, i.e., $\mathbf{z}_1 = (I_{B_x} \otimes I_{B_y} \otimes V_C) \mathbf{z}_1^2$, can be proven likewise. Note that the ordering of the three integration transforms can be arbitrary, because the three matrices $(I_{B_x} \otimes I_{B_y} \otimes V_C)$, $(I_{B_x} \otimes V_{B_y} \otimes I_C)$, and $(V_{B_x} \otimes I_{B_y} \otimes I_C)$ commute with each other. This algorithm resembles the popular row-column algorithm in the literature of multidimensional Fourier transform, cf. [173].

A.2 Proof of Theorem 6.1

Since K_{prior} is a Laplacian matrix, it is positive semi-definite and singular. In addition, $C^T R_z^{-1} C$ is a diagonal matrix with positive diagonal elements corresponding

to the observed variables. According to the properties of Laplacian matrices, the resulting summation K_{post} is strictly positive definite.

The second conclusion follows from the following theorem of standard Richardson iteration, proved by Adams [174].

Theorem A.1. *Let $K_{\text{post}} = K_1 + K_2$ be a symmetric positive definite matrix and let K_1 be symmetric and nonsingular. Then $\rho(K_1^{-1}K_2) < 1$ if and only if $K_1 - K_2$ is positive definite.*

Since $K_1 = K_{\text{prior}} + cI_z$ and $c > 0$, K_1 is symmetric and nonsingular. Note that

$$K_2 = K_{\text{post}} - K_1 \tag{A.5}$$

$$= C^T R_z^{-1} C - cI_z, \tag{A.6}$$

where c equals the largest diagonal element in $C^T R_z^{-1} C$. Therefore, K_2 is a diagonal matrix with diagonal elements smaller than or equal to 0. As a result, $K_1 - K_2$ is positive definite.

A.3 Proof of Theorem 6.2

Computing the eigenvalues of $K_1^{-1}K_2$ amounts to solving the generalized eigenvalue problem:

$$\lambda K_1 \mathbf{x} = K_2 \mathbf{x}. \tag{A.7}$$

It follows from Theorem 2.2 in [175] that

$$\frac{\lambda_{\max}(K_2)}{\lambda_{\max}(K_1)} \leq \rho(K_1^{-1}K_2) \leq \frac{\lambda_{\max}(K_2)}{\lambda_{\min}(K_1)}, \tag{A.8}$$

where $\lambda_{\max}(K)$ and $\lambda_{\min}(K)$ denote the largest and smallest absolute eigenvalues of K . By substituting $K_1 = K_{\text{prior}} + cI_z$ and $K_2 = C^T R_z^{-1} C - cI_z$ into (A.8), we can obtain the bounds specified for the proposed algorithm.

A.4 Derivation of the Q -function

We aim to learn the smoothness parameters α_z and β_z by maximum likelihood estimation, i.e., by maximizing

$$L(\alpha_z, \beta_z) = \log p(\mathbf{y}|\alpha_z, \beta_z) \quad (\text{A.9})$$

$$= \log \int_{\mathbf{z}} p(\mathbf{y}, \mathbf{z}|\alpha_z, \beta_z) d\mathbf{z}. \quad (\text{A.10})$$

Since maximizing $\log p(\mathbf{y}|\alpha_z, \beta_z)$ is intractable, we apply Expectation Maximization (EM) instead.

In the E-step, we compute the Q -function, which is defined as:

$$\begin{aligned} Q(\alpha_z, \beta_z; \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}) &= \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}) \log p(\mathbf{y}, \mathbf{z}|\alpha_z, \beta_z) d\mathbf{z} \\ &= E_{\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}} \{ \log p(\mathbf{y}, \mathbf{z}|\alpha_z, \beta_z) \} \\ &= E_{\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}} \left\{ -\frac{1}{2} \text{tr}(\mathbf{z}^T K_{\text{prior}} \mathbf{z}) + \frac{1}{2} \log |K_{\text{prior}}|_+ + c \right\} \\ &= -\frac{1}{2} E_{\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}} \{ \text{tr}(K_{\text{prior}} \mathbf{z} \mathbf{z}^T) \} + \frac{1}{2} \log |K_{\text{prior}}|_+ + c. \end{aligned}$$

Note that the trace and the expectation operator commute. Consequently,

$$\begin{aligned} &Q(\alpha_z, \beta_z; \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}) \\ &= -\frac{1}{2} \text{tr} \left(K_{\text{prior}} E_{\mathbf{z}|\mathbf{y}, \hat{\alpha}_z^{(\kappa-1)}, \hat{\beta}_z^{(\kappa-1)}} (\mathbf{z} \mathbf{z}^T) \right) + \frac{1}{2} \log |K_{\text{prior}}|_+ + c \\ &= -\frac{1}{2} \text{tr} \left(K_{\text{prior}} \left(K_{\text{post}}^{(\kappa-1)} \right)^{-1} \right) - \frac{1}{2} (\mathbf{z}^{(\kappa-1)})^T K_{\text{prior}} \mathbf{z}^{(\kappa-1)} + \frac{1}{2} \log |K_{\text{prior}}|_+ + c. \end{aligned}$$

If we ignore the common coefficient $1/2$, we obtain the Q -function in (6.31).

Appendix B

Proofs and Derivations of Results in Chapter 7

B.1 Derivation of the Gradients

The gradient of the lower bound L with respect to the variational parameters $\mathbf{m}_{\mathbf{z}_S}$ and $\boldsymbol{\nu}_{\mathbf{z}_S}$ can be derived as follows:

$$\begin{aligned}\nabla_{\mathbf{m}_{\mathbf{z}_S}} L &= \nabla_{\mathbf{m}_{\mathbf{z}_S}} \left\{ E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{x} | C\mathbf{y}_e + \mathbf{m})] + E_{\phi(\mathbf{z}_{S_e})} [\log p(\mathbf{z}_S | \alpha_z)] \right\} \\ &= E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \nabla_{\mathbf{m}_{\mathbf{z}_S}} \mathbf{z}_S \right\} \\ &\quad - \alpha_z K_S \mathbf{m}_{\mathbf{z}_S} \\ &= E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \right\} \\ &\quad - \alpha_z K_S \mathbf{m}_{\mathbf{z}_S}, \tag{B.1}\end{aligned}$$
$$\begin{aligned}\nabla_{\boldsymbol{\nu}_{\mathbf{z}_S}} L &= \nabla_{\boldsymbol{\nu}_{\mathbf{z}_S}} \left\{ E_{\phi(\mathbf{y}_e)} [\log p(\mathbf{x} | C\mathbf{y}_e + \mathbf{m})] + E_{\phi(\mathbf{z}_{S_e})} [\log p(\mathbf{z}_S | \alpha_z)] \right\} + 1 \oslash \boldsymbol{\nu}_{\mathbf{z}_S} \\ &= E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \nabla_{\boldsymbol{\nu}_{\mathbf{z}_S}} \mathbf{z}_S \right\} \\ &\quad - \alpha_z K_S \boldsymbol{\nu}_{\mathbf{z}_S} + 1 \oslash \boldsymbol{\nu}_{\mathbf{z}_S}\end{aligned}$$

$$\begin{aligned}
&= E_{\phi(\mathbf{y}_e)} \left\{ \nabla_{\mathbf{z}_S} \left[\sum_{\{i,j\} \in \mathcal{V}_O} \log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \right] \odot \mathbf{z}_{Se} \right\} \\
&\quad - \alpha_z K_S \boldsymbol{\nu}_{\mathbf{z}_S} + 1 \odot \boldsymbol{\nu}_{\mathbf{z}_S}.
\end{aligned} \tag{B.2}$$

Note that $\phi(\mathbf{y}_e) = \prod_{z \in \{\xi, \sigma, \mu\}} \phi(\mathbf{z}_{Se}) \phi(\mathbf{z}_{Te})$.

B.2 Gradients of the Logarithm of GEV densities

The logarithm of the PDF of a GEV distribution can be written as:

$$\begin{aligned}
&\log f(x_{ij} | \xi_{Ti} + \xi_{Sj}, \zeta_{Ti} + \zeta_{Sj}, \mu_{Ti} + \mu_{Sj}) \\
&= -\zeta_{ij} - \left(\frac{1}{\xi_{ij}} + 1 \right) \log \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right] - \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}}, \tag{B.3}
\end{aligned}$$

where $\xi_{ij} = \xi_{Ti} + \xi_{Sj}$, $\zeta_{ij} = \zeta_{Ti} + \zeta_{Sj}$, and $\mu_{ij} = \mu_{Ti} + \mu_{Sj}$. As a result, the partial derivatives can be computed as:

$$\begin{aligned}
\frac{\partial \log f(x_{ij})}{\partial \xi_{Ti}} &= \frac{\partial \log f(x_{ij})}{\partial \xi_{Sj}} = - \left(\frac{1}{\xi_{ij}} + 1 \right) \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-1} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \\
&\quad + \frac{1}{\xi_{ij}^2} \log \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right] + \frac{1}{\xi_{ij}} \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}-1} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \\
&\quad - \frac{1}{\xi_{ij}^2} \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}} \log \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right], \tag{B.4}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \log f(x_{ij})}{\partial \zeta_{Ti}} &= \frac{\partial \log f(x_{ij})}{\partial \zeta_{Sj}} = -1 + (1 + \xi_{ij}) \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-1} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \\
&\quad - \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}-1} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right), \tag{B.5}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \log f(x_{ij})}{\partial \mu_{Ti}} &= \frac{\partial \log f(x_{ij})}{\partial \mu_{Sj}} = (1 + \xi_{ij}) \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-1} \left(\frac{1}{\exp(\zeta_{ij})} \right) \\
&\quad - \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right]^{-\frac{1}{\xi_{ij}}-1} \left(\frac{1}{\exp(\zeta_{ij})} \right). \tag{B.6}
\end{aligned}$$

B.3 Proof of Proposition 7.1

We can prove Proposition 7.1 via contradiction. Let $a_{ij} = 1 + \xi_{ij}(x_{ij} - \mu_{ij})/\exp(\zeta_{ij})$. Suppose that there is a local maximizer $\hat{\mathbf{y}}$ of $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ that breaks the constraint that $a_{ij} > 0$. According the definition in (7.50),

$$\tilde{f}(x_{ij}|\xi_{ij}, \zeta_{ij}, \mu_{ij}) = \exp \left\{ c_1 \left[1 + \xi_{ij} \left(\frac{x_{ij} - \mu_{ij}}{\exp(\zeta_{ij})} \right) \right] - c_2 \right\}. \quad (\text{B.7})$$

Thus, we can obtain:

$$\frac{\partial \log \tilde{f}(x_{ij})}{\partial \hat{z}_{Ti}} = \frac{\partial \log \tilde{f}(x_{ij})}{\partial \hat{z}_{Sj}} = \frac{\partial \log \tilde{f}(x_{ij})}{\partial a_{ij}} \frac{\partial a_{ij}}{\partial \hat{z}_{ij}} = c_1 \frac{\partial a_{ij}}{\partial \hat{z}_{ij}}, \quad (\text{B.8})$$

where $z \in \{\xi, \zeta, \mu\}$. For scale parameters $\hat{\zeta}_{ij}$, it follows from the inequality $1 + \hat{\xi}_{ij}(x_{ij} - \hat{\mu}_{ij})/\exp(\hat{\zeta}_{ij}) \leq 0$ that

$$\frac{\partial a_{ij}}{\partial \hat{\zeta}_{ij}} = -\hat{\xi}_{ij} \left(\frac{x_{ij} - \hat{\mu}_{ij}}{\exp(\hat{\zeta}_{ij})} \right) \geq 1. \quad (\text{B.9})$$

Therefore,

$$\frac{\partial \log \tilde{f}(x_{ij})}{\partial \hat{\zeta}_{Ti}} = \frac{\partial \log \tilde{f}(x_{ij})}{\partial \hat{\zeta}_{Sj}} = c_1 \frac{\partial a_{ij}}{\partial \hat{\zeta}_{ij}} \geq c_1. \quad (\text{B.10})$$

As a result, if c_1 is sufficiently large, for example,

$$c_1 > \left[[K_T \hat{\boldsymbol{\zeta}}_T]_i - \sum_{\{j: \{i,j\} \notin \mathcal{V}_-\}} \frac{\partial \log f_{ij}}{\partial \hat{\zeta}_{ij}} \right]_+, \quad (\text{B.11})$$

$\exists i \in \{i : \{i, j\} \in \mathcal{V}_-\}$, where $t_+ = \max(t, 0)$, then

$$\begin{aligned} \frac{\partial \log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})}{\partial \hat{\zeta}_{Ti}} &= c_1 \sum_{\{j: \{i,j\} \in \mathcal{V}_-\}} \frac{\partial a_{ij}}{\partial \hat{\zeta}_{ij}} + \sum_{\{j: \{i,j\} \notin \mathcal{V}_-\}} \frac{\partial \log f_{ij}}{\partial \hat{\zeta}_{ij}} - [K_T \hat{\boldsymbol{\zeta}}_T]_i \\ &\geq c_1 + \sum_{\{j: \{i,j\} \notin \mathcal{V}_-\}} \frac{\partial \log f_{ij}}{\partial \hat{\zeta}_{ij}} - [K_T \hat{\boldsymbol{\zeta}}_T]_i > 0. \end{aligned} \quad (\text{B.12})$$

The above inequality contradicts the assumption that $\hat{\mathbf{y}}$ is a local maximizer of $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$. Therefore, $\hat{\mathbf{y}}$ can be a maximum of $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ only if it satisfies the constraint that $a_{ij} > 0$ for all $\{i, j\} \in \mathcal{V}_O$. Furthermore, if the constraint is satisfied, then $\tilde{f}(x_{ij}) = f(x_{ij})$ for all $\{i, j\} \in \mathcal{V}_O$, and therefore, $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = \log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$. As a result, a local maximizer of $\log \tilde{p}(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ is also a local maximizer of $\log p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$.

B.4 Gibbs Sampling Method to Learn the STM

In order to employ Gibbs sampling, we construct a full Bayesian model here by imposing Gamma priors on the smoothness parameters. More specifically, we set the shape and rate parameters (a, b) of the Gamma priors to be very small (i.e., 10^{-6}) such that the priors are non-informative. The detailed steps of the Gibbs sampling algorithm are as follows:

1. Updating the spatial components of GEV parameters at each site

Each component of $\mathbf{z}_S = [z_{Sj}]^T$ is updated individually via the Metropolis-Hastings (MH) algorithm. Let us take a location parameter μ_{Sj} as an example. In iteration κ , we first generate a proposal $\mu_{Sj}^{(p)}$ from a Gaussian distribution with mean value $\mu_{Sj}^{(\kappa-1)}$, and then compute the acceptance probability:

$$r = \min(1, \alpha), \quad (\text{B.13})$$

where α is a ratio between GEV likelihoods times the thin-plate model likelihood when $\mu_{Sj}^{(\kappa)} = \mu_{Sj}^{(p)}$ and when $\mu_{Sj}^{(\kappa)} = \mu_{Sj}^{(\kappa-1)}$. Note that other parameters are set to their most recent values. With probability r , $\mu_{Sj}^{(\kappa)}$ is set to $\mu_{Sj}^{(p)}$; otherwise it remains at $\mu_{Sj}^{(\kappa-1)}$. The spatial components of the scale and shape parameters are updated similarly.

2. Updating the temporal components of GEV parameters at each site

Each component of $\mathbf{z}_T = [z_{Ti}]^T$ is updated singly in a similar vein as in the previous step.

3. Updating the smoothness parameters α_z

The conditional distribution of α_z conditioned on other parameters has a closed form, that is, a Gamma distribution $\text{Gamma}(\alpha_z; a + (P-1)/2, b + \mathbf{z}_S^T K_S \mathbf{z}_S / 2)$.

We therefore draw one sample from this distribution and set $\alpha_z^{(\kappa)}$ to the value of this sample.

4. Updating the smoothness parameters β_z and γ_z

We update β_z and γ_z using the MH approach since the Gamma priors are not conjugate to the likelihood of β_z and γ_z . Here, we specify the proposal distributions as Gamma distributions. Due to its asymmetry, we need the Hastings correction when computing the ratio.

Appendix C

Proofs of Results in Chapter 9

C.1 Proof of Theorem 9.2

We can prove the conditional dependence of the ETPC model via contradiction.

Suppose that we are interested in performing inference on one arbitrary site x_a given data at the other sites $\mathbf{x}_O$ with $x_a \cup \mathbf{x}_O = \mathbf{x}$. Let f denote the probability density function of the ETPC model. If the conditional independence assumption holds, then $f(x_a|\mathbf{x}_O) = f(x_a|\mathbf{x}_{N(a)})$, where $N(a)$ refers to the neighboring nodes of a . However,

$$f(x_a, \mathbf{x}_O) = \sum_i P(\mathcal{T}_i) f(x_a, \mathbf{x}_O | \mathcal{T}_i) \quad (\text{C.1})$$

$$= \sum_i P(\mathcal{T}_i) f(x_a | \mathcal{T}_i, \mathbf{x}_O) f(\mathbf{x}_O | \mathcal{T}_i). \quad (\text{C.2})$$

Since $f(x_a | \mathcal{T}_i, \mathbf{x}_O) = f(x_a | \mathcal{T}_i, \mathbf{x}_{N(a)})$ in a tree graphical model, $f(x_a, \mathbf{x}_O)$ can be further written as:

$$f(x_a, \mathbf{x}_O) = \sum_i P(\mathcal{T}_i) f(x_a | \mathcal{T}_i, \mathbf{x}_{N(a)}) f(\mathbf{x}_O | \mathcal{T}_i). \quad (\text{C.3})$$

As a result,

$$f(x_a|\mathbf{x}_O) = \sum_i f(x_a|\mathcal{T}_i, \mathbf{x}_{N(a)})P(\mathcal{T}_i|\mathbf{x}_O), \quad (\text{C.4})$$

while

$$f(x_a|\mathbf{x}_{N(a)}) = \sum_i f(x_a|\mathcal{T}_i, \mathbf{x}_{N(a)}) \frac{P(\mathcal{T}_i, \mathbf{x}_O)}{f(\mathbf{x}_{N(a)})}, \quad (\text{C.5})$$

which contradicts the assumption that $f(x_a|\mathbf{x}_O) = f(x_a|\mathbf{x}_{N(a)})$. Therefore, the absence of an edge between two nodes in the ETPC model (i.e., zero elements in the edge weight matrix β) does not imply conditional independence.

C.2 Proof of Proposition 9.1

Before proving the proposition, we would like to introduce the stochastic increasing condition first: for two continuously distributed variables X_1 and X_2 , X_1 is said to be stochastically increasing with X_2 if the probability $P(X_1 > x_1|X_2 = x_2)$ increases with x_2 for all x_1 in the support of X_1 [96]. Note that this condition is satisfied by all the copulas we used in this thesis when modeling positive correlation.

Returning to the main thread, we can write the joint PDF of a three-node Markov chain (X_1, X_2, X_3) by applying (9.3):

$$p(x_1, x_2, x_3) = c_{12}(u_1, u_2)c_{23}(u_2, u_3)f_1(x_1)f_2(x_2)f_3(x_3).$$

According to the definition of upper tail dependence (2.49), we obtain:

$$\begin{aligned} P(U_1 > u|U_3 > u) &= \frac{P(U_1 > u, U_3 > u)}{P(U_3 > u)} \\ &= \frac{1}{1-u} \int_u^1 \int_u^1 \int_0^1 c_{12}(u_1, u_2)c_{23}(u_2, u_3)du_2du_1du_3 \\ &= \frac{1}{1-u} \int_0^1 \left(\int_u^1 c_{12}(u_1, u_2)du_1 \int_u^1 c_{23}(u_2, u_3)du_3 \right) du_2 \\ &= \frac{1}{1-u} \int_u^1 C_{12}(U_1 > u|u_2)C_{23}(U_3 > u|u_2)du_2 \\ &\quad + \frac{1}{1-u} \int_0^1 C_{12}(U_1 > u|u_2)C_{23}(U_3 > u|u_2)du_2 \end{aligned} \quad (\text{C.6})$$

Here we define $C_{12}(U_1 > u|u_2) = \int_u^1 c_{12}(u_1, u_2) du_1$ following the literature of copulas [71, 167]. The first term in the above expression can be regarded as computing the expected value of the product of two functions of a random variable uniformly distributed in $[u, 1]$ with probability density function $1/(1-u)$, i.e.

$$E \{C_{12}(U_1 > u|u_2)C_{2,3}(U_3 > u|u_2)\} = \frac{1}{1-u} \int_u^1 C_{12}(U_1 > u|u_2)C_{23}(U_3 > u|u_2)du_2. \quad (\text{C.7})$$

With the stochastically increasing assumption, $C_{12}(U_1 > u|u_2)$ and $C_{2,3}(U_3 > u|u_2)$ are increasing with respect to $u_2 \in [u, 1)$. Since the covariance of two increasing functions of a random variable is non-negative, assuming that it exists, we can obtain the follow inequality [167]:

$$E \{C_{12}(U_1 > u|u_2)C_{23}(U_3 > u|u_2)\} - E \{C_{12}(U_1 > u|u_2)\} E \{C_{23}(U_3 > u|u_2)\} \geq 0. \quad (\text{C.8})$$

Correspondingly,

$$\begin{aligned} & \lim_{u \rightarrow 1^-} \frac{1}{1-u} \int_u^1 C_{12}(U_1 > u|u_2)C_{23}(U_3 > u|u_2) du_2 \\ & \geq \lim_{u \rightarrow 1^-} \left(\frac{1}{1-u} \int_u^1 C_{12}(U_1 > u|u_2)du_2 \right) \left(\frac{1}{1-u} \int_u^1 C_{23}(U_3 > u|u_2)du_2 \right) \\ & = \lim_{u \rightarrow 1^-} P(U_1 > u|U_2 > u)P(U_3 > u|U_2 > u) \\ & = \lambda_{12}\lambda_{23}. \end{aligned} \quad (\text{C.9})$$

On the other hand, the second term in (C.6) can be simplified as follows:

$$\begin{aligned} & \frac{1}{1-u} \int_0^u C(U_1 > u|u_2)C(U_3 > u|u_2) du_2 \\ & = \frac{u}{1-u} \int_0^u \frac{1}{u} C(U_1 > u|u_2)C(U_3 > u|u_2) du_2 \\ & = \frac{u}{1-u} E[C(U_1 > u|u_2)C(U_3 > u|u_2)] \\ & \geq \frac{u}{1-u} E[C(U_1 > u|u_2)]E[C(U_3 > u|u_2)] \end{aligned}$$

$$\begin{aligned}
&= \frac{u}{1-u} \left[\int_0^u \frac{1}{u} C(U_1 > u|u_2) du_2 \right] \left[\int_0^u \frac{1}{u} C(U_3 > u|u_2) du_2 \right] \\
&= \frac{1}{u(1-u)} \left[\int_0^u (1 - C(U_1 < u|u_2)) du_2 \right] \left[\int_0^u (1 - C(U_3 < u|u_2)) du_2 \right] \\
&= \frac{[u - C_{12}(u, u)][u - C_{23}(u, u)]}{u(1-u)}. \tag{C.10}
\end{aligned}$$

To compute the limit of the above, We then apply L'hôpital's rule as $[u - C_{12}(u, u)] \times [u - C_{23}(u, u)]$ and $u(1-u)$ both approach 0 when $u \rightarrow 1^-$. Noting that

$$C_{jk}(u, u) = \int_0^u \int_0^u c_{jk}(u_j, u_k) du_j du_k, \tag{C.11}$$

Leibniz' Integral Rule can be employed to yield:

$$\begin{aligned}
\frac{d}{du} C_{jk}(u, u) &= \int_0^u c_{jk}(u, u_k) du_k + \int_0^u c_{jk}(u_j, u) du_j \\
&= C_{jk}(U_j < u|u) + C_{jk}(U_k < u|u). \tag{C.12}
\end{aligned}$$

Consequently,

$$\begin{aligned}
&\lim_{u \rightarrow 1^-} \frac{1}{1-u} \int_0^u C(U_1 > u|u_2) C(U_3 > u|u_2) du_2 \\
&= \lim_{u \rightarrow 1^-} \frac{[u - C_{12}(u, u)][1 - C_{23}(U_2 < u|u) - C_{23}(U_3 < u|u)]}{1 - 2u} \\
&\quad + \frac{[1 - C_{12}(U_1 < u|u) - C_{12}(U_2 < u|u)][u - C_{23}(u, u)]}{1 - 2u} \\
&= 0, \tag{C.13}
\end{aligned}$$

because $\lim_{u \rightarrow 1^-} C_{jk}(U_j < u|u) < \infty$ by the definition of conditional cumulative distribution function.

This implies that, from (C.9) and (C.13),

$$\lambda_{13} = \lim_{u \rightarrow 1^-} P(U_1 > u|U_3 > u) \geq \lambda_{12}\lambda_{23}, \tag{C.14}$$

where λ_{13} denotes the upper tail dependence coefficient between X_1 and X_3 .

C.3 Proof of Proposition 9.2

We proceed via an induction argument. Consider an undirected tree graphical model $\mathcal{T}$ with N nodes. There exists a unique path in the tree between two arbitrary distinct nodes, X_{i_1} and X_{i_n} , which we denote as $\{i_1, i_2, \dots, i_n\}$. We label the indices of the remaining nodes in the tree absent in the path as $i_{n+1}, \dots, i_N$.

Figure C.1: An example of a tree graphical model with pairwise upper tail dependence.

Note that from the fact that the marginals of a copula distribution are uniformly distributed we obtain:

$$\int_0^1 c_{ij}(u_i, u_j) du_i = 1. \quad (\text{C.15})$$

Therefore, the pairwise copulas which are not on the path between node i_1 and i_n will be integrated out when computing the tail dependence coefficient. As an example, the joint probability density function corresponding to the tree in Fig. C.1 is

$$p(x_1, x_2, \dots, x_{10}) = c_{13}c_{23}c_{34}c_{45}c_{46}c_{67}c_{78}c_{79} \prod_{i=1}^9 f_i(x_i),$$

where we use c_{ij} to denote $c_{ij}(u_i, u_j)$ for simplicity. Let us now analyze, for instance, the tail dependence between X_3 and X_7 . According to the definition of upper tail dependence,

$$P(U_3 > u | U_7 > u) = \frac{P(U_3 > u, U_7 > u)}{P(U_7 > u)}$$

$$\begin{aligned}
&= \frac{1}{1-u} \int_u^1 \int_u^1 \left(\int_0^1 \cdots \int_0^1 c_{13} c_{23} c_{34} c_{45} c_{46} c_{67} c_{78} c_{79} du_{1:9|3,7} \right) du_3 du_7 \\
&= \frac{1}{1-u} \int_u^1 \cdots \int_0^1 \left(\int_0^1 c_{13} du_1 \int_0^1 c_{23} du_2 \int_0^1 c_{45} du_5 \int_0^1 c_{78} du_8 \int_0^1 c_{79} du_9 \right) \\
&\quad \cdot c_{34} c_{46} c_{67} du_4 du_6 du_3 du_7.
\end{aligned}$$

By utilizing (C.15), $P(U_3 > u | U_7 > u)$ can be further simplified as:

$$\frac{1}{1-u} \int_u^1 \int_u^1 \int_0^1 \int_0^1 c_{34} c_{46} c_{67} du_4 du_6 du_3 du_7, \quad (\text{C.16})$$

which only involves pairwise copulas comprising the path between X_3 and X_7 .

Returning to the main thread, we first deal with the case when $n = 3$, that is, there are only two distinct edges which comprise the path between X_{i_1} and X_{i_3} . Let $\lambda_{i_1 i_3}$ denote the upper tail dependence coefficient between X_{i_1} and X_{i_3} . From Proposition 9.1, $\lambda_{i_1 i_3} = \lim_{u \rightarrow 1^-} P(U_{i_1} > u | U_{i_3} > u) \geq \lambda_{i_1 i_2} \lambda_{i_2 i_3}$, assuming that $\lambda_{i_1 i_2}$ and $\lambda_{i_2 i_3}$ both exist.

We now tackle the general case when $n > 3$. Suppose that the proposition holds for the nodes X_{i_1} and $X_{i_{n-1}}$ where the length of the path between the nodes is $n - 1$. More explicitly, suppose that the upper tail dependence between X_{i_1} and $X_{i_{n-1}}$, denoted by $\lambda_{i_1 i_{n-1}}$, satisfies $\lambda_{i_1 i_{n-1}} \geq \prod_{t=1}^{n-2} \lambda_{i_t i_{t+1}}$. By the definition of upper tail dependence,

$$\begin{aligned}
P(U_{i_1} > u | U_{i_n} > u) &= \frac{P(U_{i_1} > u, U_{i_n} > u)}{P(U_{i_n} > u)} \\
&= \frac{1}{1-u} \int_u^1 \int_u^1 \left(\int_0^1 \cdots \int_0^1 \prod_{\{i_s, i_t\} \in T} c_{i_s i_t} du_{i_2} \cdots du_{i_{n-1}} du_{i_{n+1}} \cdots du_{i_N} \right) du_{i_1} du_{i_n} \\
&= \frac{1}{1-u} \int_u^1 \int_u^1 \left(\int_0^1 \cdots \int_0^1 \prod_{t=1}^{n-1} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \right) du_{i_1} du_{i_n} \\
&= \frac{1}{1-u} \int_0^1 \cdots \int_0^1 \left(\int_u^1 c_{i_1 i_2} du_{i_1} \int_u^1 c_{i_{n-1} i_n} du_{i_n} \right) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \\
&= \frac{1}{1-u} \int_0^1 \cdots \int_0^1 C_{i_1 i_2}(U_{i_1} > u | u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u | u_{i_{n-1}}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{1-u} \int_u^1 \int_0^1 \underbrace{\cdots}_{n-3} \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \\
&+ \frac{1}{1-u} \int_0^u \int_0^1 \underbrace{\cdots}_{n-3} \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}}.
\end{aligned} \tag{C.17}$$

We analyze each term in (C.17) separately. The first term in (C.17) can be regarded as the expected value of a product of two functions of $n-2$ jointly distributed random variables with joint pdf $\prod_{t=2}^{n-2} c_{i_t i_{t+1}}(u_{i_t}, u_{i_{t+1}})/(1-u)$ in the region $[0, 1]^{n-3} \times [u, 1]$, i.e. $E\{C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}})\}$. The stochastically increasing assumption coupled with the fact that all the copulas with upper tail dependence only model positive dependence guarantee $\text{Cov}\{C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}})\} \geq 0$. Thus,

$$\begin{aligned}
&E\{C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}})\} \\
&\geq E\{C_{i_1 i_2}(U_{i_1} > u|u_{i_2})\} E\{C_{i_{n-1} i_n}(U_{i_n} > u|u_{i_{n-1}})\}
\end{aligned} \tag{C.18}$$

We then find that:

$$\begin{aligned}
&E\{C_{i_1 i_2}(U_{i_1} > u|u_{i_2})\} \\
&= \frac{1}{1-u} \int_u^1 \int_0^1 \underbrace{\cdots}_{n-3} \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \\
&= \frac{1}{1-u} \int_0^1 \cdots \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) \prod_{t=2}^{n-3} c_{i_t i_{t+1}} \left(\int_u^1 c_{i_{n-2} i_{n-1}} du_{i_{n-1}} \right) du_{i_2} \cdots du_{i_{n-2}} \\
&= \frac{1}{1-u} \int_0^1 \cdots \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-2} i_{n-1}}(U_{i_{n-1}} > u|u_{i_{n-2}}) \prod_{t=2}^{n-3} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-2}} \\
&= P(U_{i_1} > u|U_{i_{n-1}} > u)
\end{aligned} \tag{C.19}$$

which is bounded below by $\prod_{t=1}^{n-2} \lambda_{i_t i_{t+1}}$ as $u \rightarrow 1^-$ by the induction assumption.

On the other hand,

$$E\{C_{i_{n-1}i_n}(U_{i_n} > u|u_{i_{n-1}}) = P(U_{i_n} > u|U_{i_{n-1}} > u) \quad (\text{C.20})$$

which equals $\lambda_{i_{n-1}i_n}$ as $u \rightarrow 1^-$. Therefore, by taking (C.19) and (C.20) together, we have the following lower bound for the first term in (C.17):

$$\begin{aligned} & \lim_{u \rightarrow 1^-} \frac{1}{1-u} \int_u^1 \int_0^1 \underbrace{\cdots}_{n-3} \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) \\ & \quad \cdot C_{i_{n-1}i_n}(U_{i_n} > u|u_{i_{n-1}}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \\ & \geq \prod_{t=1}^{n-1} \lambda_{i_t i_{t+1}}. \end{aligned} \quad (\text{C.21})$$

The second term in (C.17), on the other hand, can be simplified in a similar fashion as:

$$\begin{aligned} & \frac{1}{1-u} \int_0^u \int_0^1 \underbrace{\cdots}_{n-3} \int_0^1 C_{i_1 i_2}(U_{i_1} > u|u_{i_2}) C_{i_{n-1}i_n}(U_{i_n} > u|u_{i_{n-1}}) \prod_{t=2}^{n-2} c_{i_t i_{t+1}} du_{i_2} \cdots du_{i_{n-1}} \\ & = \frac{[u - C_{i_1 i_{n-1}}(u, u)] [u - C_{i_{n-1}i_n}(u, u)]}{u(1-u)}, \end{aligned} \quad (\text{C.22})$$

which approaches 0 as $u \rightarrow 1^-$ using a similar argument as presented in the proof of Proposition 9.1.

Hence, from (C.21) and (C.22),

$$\lambda_{i_1 i_n} = \lim_{u \rightarrow 1^-} P(U_{i_1} > u|U_{i_n} > u) \geq \prod_{t=1}^{n-1} \lambda_{i_t i_{t+1}}.$$

C.4 Proof of Proposition 9.3

Denote by $\mathcal{G}$ the graph under consideration and let x_1 and x_2 be distinct and arbitrary nodes. We label each spanning tree of $\mathcal{G}$ by $\mathcal{T}_i$ and label the path between x_1

and x_2 in $\mathcal{T}_i$ by $P_{x_1 x_2}^i$. By the definition of upper tail dependence,

$$\begin{aligned}
& \lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u) \\
&= \lim_{u \rightarrow 1^-} \sum_i P(U_1 > u | U_2 > u, \mathcal{T}_i) P(\mathcal{T}_i) \\
&= \frac{1}{\det[Q(\boldsymbol{\beta})]} \sum_i \left[\prod_{\{j,k\} \in \mathcal{T}_i} \beta_{jk} \right] \lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u, \mathcal{T}_i) \\
&\geq \frac{1}{\det[Q(\boldsymbol{\beta})]} \sum_i \left[\prod_{\{j,k\} \in \mathcal{T}_i} \beta_{jk} \right] \left[\prod_{\{j,k\} \in P_{x_1 x_2}^i} \lambda_{jk} \right]. \tag{C.23}
\end{aligned}$$

Since all $\lambda_{jk} \in [0, 1]$, the above expression can be further relaxed by including other λ_{jk} outside the path $P_{x_1 x_2}$, that is,

$$\begin{aligned}
& \lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u) \\
&\geq \frac{1}{\det[Q(\boldsymbol{\beta})]} \sum_i \left[\prod_{\{j,k\} \in \mathcal{T}_i} \beta_{jk} \right] \left[\prod_{\{j,k\} \in \mathcal{T}_i} \lambda_{jk} \right] \\
&= \frac{1}{\det[Q(\boldsymbol{\beta})]} \sum_i \left[\prod_{\{j,k\} \in \mathcal{T}_i} \beta_{jk} \lambda_{jk} \right] \\
&= \frac{\det[Q(\boldsymbol{\beta} \odot \boldsymbol{\lambda})]}{\det[Q(\boldsymbol{\beta})]}, \tag{C.24}
\end{aligned}$$

where the final equality is due to the matrix-tree theorem.

We remark that because $\sum_i P(\mathcal{T}_i) = 1$ and that $\lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u, \mathcal{T}_i) \in [0, 1]$, $\lim_{u \rightarrow 1^-} P(U_1 > u | U_2 > u) \in [0, 1]$.

Author's Publications

Journal Papers:

- [J1] J. Dauwels, H. Yu, X. Wang, F. Vialatte, C. Latchoumane, J. Jeong, and A. Cichocki, "Inferring Brain Networks through Graphical Models with Hidden Variables," *Machine Learning and Interpretation in Neuroimaging, Lecture Notes in Computer Science*, vol. 7263, pp. 194-201, 2012.
- [J2] H. Yu, J. Dauwels, and P. Johnathan, "Extreme-Value Graphical Models with Multiple Covariates," *IEEE Transaction on Signal Processing*, vol. 62, no. 21, pp. 5734-5747, 2014.
- [J3] H. Yu, W. I. T. Uy, and J. Dauwels, "Modeling Spatial Extremes via Ensemble-of-Trees of Pairwise Copulas," submitted to *IEEE Transaction on Signal Processing*, 2014.
- [J4] H. Yu, and J. Dauwels, "Modeling Spatial-Temporal Extreme Events using Graphical Models," submitted to *IEEE Transaction on Signal Processing*, 2014.

Conference Papers:

- [C1] H. Yu, J. Dauwels, and X. Wang, "Copula Gaussian Graphical Model with Hidden Variables," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 2177-2180, 2012.

-
- [C2] H. Yu, Z. Choo, J. Dauwels, P. Jonathan, and Q. Zhou, “Modeling Spatial Extreme Events using Markov Random Field Priors,” in *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, pp.1453-1457, 2012.
 - [C3] H. Yu, J. Dauwels, X. Zhang, S. Xu, and W. I. T. Uy, “Copula Gaussian Multiscale Graphical Model,” in *Proc. 15th Int. Conf. Inf. Fusion*, pp. 1741-1748, 2012.
 - [C4] H. Yu, Z. Choo, W. I. T. Uy, J. Dauwels, and P. Jonathan, “Modeling Extreme Events in Spatial Domain using Copula Graphical Models,” in *Proc. 15th Int. Conf. Inf. Fusion*, pp. 1761-1768, 2012. (**Best Paper Award**)
 - [C5] J. Dauwels, H. Yu, S. Xu, and X. Wang, “Copula Gaussian Graphical Model for Discrete Data,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 6283-6287, 2013.
 - [C6] H. Yu, J. Cheng, and J. Dauwels, “Extreme-Value Graphical Models with Multiple Covariates,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4553-4557, 2014.
 - [C7] H. Yu, W. I. T. Uy, and J. Dauwels, “Modeling Spatial Extremes via Ensemble-of-Trees of Pairwise Copulas,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 2415-2419, 2014.
 - [C8] H. Yu, C. Li, and J. Dauwels, “Network Inference and Change Point Detection for Piecewise-Stationary Time Series,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 4498-4502, 2014.
 - [C9] B. Li, H. Yu, J. Dauwels, and K. S. Low, “Sensor Fault Detection by Sparsity Optimization,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 7614-7618, 2014.
 - [C10] H. Yu, L. Zhang, and J. Dauwels, “Spatio-temporal Graphical Models for Extreme Events,” in *Proc. 2014 IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2032-2036, 2014.

-
- [C11] H. Yu, and J. Dauwels, “Variational Bayes Learning of Multiscale graphical Models,” to be appear in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2015.
- [C12] H. Yu, and J. Dauwels, “Variational Inference for Graphical Models of Multivariate Piecewise-Stationary Time Series”, to be appear in *Proc. 18th Int. Conf. Inf. Fusion*, 2015.
- [C13] H. Yu, and J. Dauwels, “Variational Bayes Learning of Graphical Models with Hidden Variables,” to be appear in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, 2015.

Bibliography

- [1] S. L. Lauritzen, *Graphical Models*, Oxford University Press, Oxford, 1996.
- [2] L. R. Rabiner, “A tutorial in hidden Markov models and selected applications in speech recognition,” in *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [3] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, P. Li, and F. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, no. 6, pp. 1295-1322, 2007.
- [4] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision,” *International Journal of Computer Vision (IJCV)*, vol. 40, no. 1, pp. 25-47, 2000.
- [5] M. Beal, N. Jojic, H. Attias, “A graphical model for audiovisual object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 828-836, 2003.
- [6] Z. W. Tu, X. R. Chen, A. L. Yuille, and S. C. Zhu, “Image parsing: Unifying segmentation, detection and recognition,” *International Journal of Computer Vision (IJCV)*, vol. 63, no. 2, pp 113-140, 2005
- [7] M. Fishelson, and D. Geiger, “Exact genetic linkage computations for general pedigrees,” *Bioinformatics*, vol. 18, pp. 189-198, 2002.
- [8] A. Raval, Z. Ghahramani, and D. L. Wild, “A Bayesian network model for protein fold and remote homologue recognition,” *Bioinformatics*, vol. 18, pp.788-801, 2002.

- [9] S. L. Lauritzen, and N. A. Sheehan, “Graphical models for genetic analyses,” *Statist. Sci.*, vol.18, pp. 489-514, 2003.
- [10] E. Segal, M. Shapira, A. Regev, D. Peer, D. Botstein, D. Koller, and N. Friedman, “Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data,” *Nature Gene.*, vol.34, no. 2, pp. 166-176, 2003.
- [11] P. W. Fieguth, W. C. Karl, A. S. Willsky, and C. Wunsch, “Multiresolution optimal interpolation and statistical analysis of TOPEX/POSEDON satellite altimetry,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 33, no. 2, pp. 280-292, 1995.
- [12] T. M. Chin, A. Mariano, and E. Chassignet, “Spatial regression and multiscale approximations for sequential data assimilation in ocean models,” *Journal of Geophysical Research: Oceans*, vol. 104(C4), pp. 7991-8014, 1999.
- [13] M. Butenuth, and D. Straub, “Probabilistic Graphical Models for Flood State Detection of Roads Combining Imagery and DEM,” *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 6, pp. 1051-1055, 2012.
- [14] R. McEliece, D. Mackay, and J. Cheng, “Turbo decoding as an instance of Pearls belief propagation algorithm,” *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 2, pp. 140-152, 1998.
- [15] R. B. Nelsen, *An Introduction to Copulas*, New York: Springer-Verlag, 2006.
- [16] S. Celik, B. Logsdon, and S.-I. Lee, “Efficient Dimensionality Reduction for High-Dimensional Network Estimation,” *Journal of Machine Learning Research*, W&CP vol. 31, no. 1, pp. 1953-1961, 2014
- [17] O. Banerjee, L. E. Ghaoui, and A. d’Aspremont, “Model Selection Through Sparse Maximum Likelihood Estimation”, *Journal of Machine Learning Research*, vol. 9, pp. 485-516, 2008.

- [18] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432-441, 2008.
- [19] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, “Latent Variable Graphical Model Selection via Convex Optimization,” *The Annals of Statistics*, vol. 40, no. 4, pp. 1935-1967, 2012.
- [20] E. N. Lorenz, “The problem of deducing the climate from the governing equations,” *Tellus*, vol. 16, pp. 1-11, 1964.
- [21] C. Schözel, and P. Friederichs, “Multivariate non-normally distributed random variables in climate research an introduction to the copula approach,” *Nonlinear Processes in Geophysics*, vol. 15, pp. 761-772, 2008.
- [22] H. Liu, J. Lafferty, and L. Wasserman, “The Nonparanormal: Semiparametric Estimation of High Dimensional Undirected Graphs,” *Journal of Machine Learning Research*, pp. 2295-2328, 2010
- [23] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty, “High-dimensional Ising model selection using ℓ_1 -regularized logistic regression,” *Ann. Statist.*, vol. 38, pp. 1287-1319, 2010.
- [24] A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky, “High-dimesional structure esimtaion in Ising models: Local separation criterion,” *Ann. Statist.*, vol. 40, pp. 1346-1375, 2012.
- [25] A. Jalali, P. D. Ravikumar, V. Vasuku, and S. Sanghavi, “On learning discrete graphical models using group-sparse regularization”, *Journal of Machine Learning Research*, W&CP vol. 15, pp. 378-387, 2011.
- [26] P.-L. Loh, and M. J. Wainwright, “Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses,” *Ann. Statist.*, vol. 41, no. 6, pp. 3022-3049, 2013.

- [27] M. Pitt, D. Chan, and R. Kohn, "Efficient Bayesian Inference for Gaussian Copula Regression Models," *Biometrika*, vol. 93, no. 3, pp. 537-554, 2006.
- [28] P. D. Hoff, "Extending the Rank Likelihood for Semiparametric Copula Estimation," *Annals of Applied Statistics*, vol. 1, no. 1, pp. 265-283, 2007.
- [29] A. Dobra and A. Lenkoski, "Copula Gaussian Graphical Models and their Application to Modeling Functional Disability Data," *Annals of Applied Statistics*, vol. 5, No. 2A, pp. 969-993, 2011.
- [30] J. S. Murray, D. B. Dunson, L. Carin, and J. E. Lucas, "Bayesian Gaussian copula factor models for mixed data," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 656-665, 2013.
- [31] H. Cramér, *Mathematical Methods of Statistics*. Princeton Univ. Press, Princeton, NJ, 1946.
- [32] M. A. Kramer, U. T. Eden, E. D. Kolaczyk, R. Zepeda, E. N. Eskandar, and S. S. Cash, "Coalescence and Fragmentation of Cortical Networks during Focal Seizures", *The Journal of Neuroscience*, pp. 10076-10085, 2010.
- [33] X. Xuan, and K. Murphy, "Modeling Changing Dependency Structure in Multivariate Time Series", *Proceedings of the 24th ICML*, 2007.
- [34] I. Cribben, R. Haraldsdottir, L. Y. Atlas, T. D. Wager, and M. A. Lindquist, "Dynamic Connectivity Regression: Determining State-related Changes in Brain Conectivity", *NeuroImage* 61, pp. 907-920, 2012.
- [35] M. Lavielle, and G. Teyssière, "Adaptive Detection of Multiple Change-Points in Asset Price Volatility", in: G. Teyssière and A. Kirman (Eds), *Long-Memory in Economics*, Springer, pp. 129-156, 2005.
- [36] D. Angelosante, and G. B. Giannakis, "Sparse Graphical Modeling of Piecewise-Stationary Time Series", *Proceedings of ICASSP 2011*, pp. 1960-1963, 2011.

- [37] R. Killick, P. Fearnhead, I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association* vol. 107, no. 500, pp. 1590-1598, 2012.
- [38] S. J. Spignesi, *Catastrophe!: the 100 greatest disasters of all time*, Citadel Press. ISBN 0-8065-2558-4. pp. 37, 2004.
- [39] R. D. Knabb, J. R. Rhome, D. P. Brown, "Tropical Cyclone Report: Hurricane Katrina, August 23 - 30, 2005," *United States National Oceanic and Atmospheric Administration's National Weather Service*, Retrieved Dec. 10, 2012.
- [40] M. L. Parry, O. F. Canziani, J. P. Palutikof, van der Linden, and C. E. Hanson, Eds., IPCC, 2007: summary for policymakers, In: *Climate Change 2007: Impacts, Adaptation, and Vulnerability*. Contribution of Working Group II to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, UK, 2007.
- [41] S. G. Coles, *An Introduction to Statistical Modeling of Extreme Values*, Springer, London, 2001.
- [42] P. Jonathan and K. Ewans, "Uncertainties in extreme wave height estimates for hurricane-dominated regions," *J. Offshore Mech. Arct.*, vol. 129(4), pp. 300-305, 2007.
- [43] P. Jonathan, K. Ewans, and G. Forristall, "Statistical estimation of extreme ocean environments: the requirement for modelling directionality and other covariate effects," *Ocean Eng.*, vol. 35, pp. 1211-1225, 2008.
- [44] P. Jonathan and K. C. Ewans, "The effect of directionality on extreme wave design criteria," *Ocean Eng.*, vol. 34, pp. 1977-1994, 2007.
- [45] K. C. Ewans and P. Jonathan, "The effect of directionality on Northern North Sea extreme wave design criteria," *J. Offshore Mech. Arct.*, vol. 130, 041604, 2008.

- [46] P. Jonathan, K. Ewans, "Modelling the seasonality of extreme waves in the Gulf of Mexico," *J. Offshore Mech. Arct.*, vol. 133, 021104, 2011.
- [47] P. J. Northrop and P. Jonathan, "Threshold modelling of spatially dependent non-stationary extremes with application to hurricane-induced wave heights," *Environmetrics*, vol. 22, pp. 799-809, 2011.
- [48] D. Cooley, and S. Sain, "Spatial hierarchical modeling of precipitation extremes from a regional climate model," *Journal of Agricultural, Biological and Environmental Statistics* **15**, pp. 381-402, 2010.
- [49] E. F. Eastoe and J. A. Tawn, "Modelling non-stationary extremes with application to surface level ozone," *J. Roy. Stat. Soc. C-App.*, vol. 58, pp. 25-45, 2009.
- [50] P. Jonathan, K. Ewans, "A spatio-directional model for extreme waves in the Gulf of Mexico," *J. Offshore Mech. Arct.*, vol. 133, 011601, 2011.
- [51] V. Chavez-Demoulin and A. C. Davison. "Generalized additive modelling of sample extremes," *J. Roy. Stat. Soc. C-App.*, vol. 54, pp. 207-222, 2005.
- [52] D. Randell, Y. Wu, P. Jonathan and K. Ewans, "Modelling Covariate Effects in Extremes of Storm Severity on the Australian North West Shelf," *Proc. Int. Conf. Ocean Offshore Arctic Eng.*, 2013.
- [53] N. P. Galatsanos and A. K. Katsaggelos, "Methods for Choosing the Regularization Parameter and Estimating the Noise Variance in Image Restoration and Their Relation," *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 322-336, 1992.
- [54] G. Kuhn, S. Khan, A. R. Ganguly, and M. L. Branstetter, " Geospatial-temporal dependence among weekly precipitation extremes with applications to observations and climate model simulations in South America," *Advances in Water Resources*, vol. 30, pp. 2401-2423.

- [55] V. Chavez-Demoulin and A. C. Davison, “Modelling Times Series Extremes,” *Statistical Journal*, vol. 10, pp. 109-133, 2012.
- [56] F. Laurini, and F. Pauli, “Smoothing sample extremes: The mixed model approach,” *Computational Statistics & Data Analysis*, vol. 53, no. 11, pp. 3842-3864, 2009.
- [57] S. E. Neville, M. J. Palmer, and M. P. Wand, “Generalized Extreme Value Additive Model Analysis Via Mean Field Variational Bayes,” *Australian & New Zealand Journal of Statistics*, vol. 53, no. 3, pp. 305-330, 2011.
- [58] M. P. Wand, J. T. Ormerod, S. A. Padoan, and R. Frühwirth, “Mean Field Variational Bayes for Elaborate Distributions,” *Bayesian Analysis*, vol. 7, no. 2, pp. 847-900, 2012.
- [59] G. Huerta and B. Sansó, “Time-Varying Models for Extreme Values,” *Environmental and Ecological Statistics*, vol. 14, pp. 285-299, 2007.
- [60] H. Sang, and A. E. Gelfand, “Hierarchical modeling for extreme values observed over space and time,” *Environmental and Ecological Statistics* vol. 16, pp. 407–426, 2009.
- [61] S. Ghosh and B. K. Mallick, “A hierarchical Bayesian spatio-temporal model for extreme precipitation events,” *Environmetrics*, vol. 22, pp. 192-204, 2011.
- [62] Y. Liu, M. T. Bahadori, and H. Li, “Sparse-GEV: Sparse Latent Space Model for Multivariate Extreme Value Time Series Modeling,” *Proc. ICML*, 2012.
- [63] B. Mahmoudian, and M. Mohammadzadeh, “A spatio-temporal dynamic regression model for extreme wind speeds,” *Extremes*, vol. 17, no. 2, pp. 221-245, 2014.
- [64] D. Cooley, D. Nychka, P. Naveau, “Bayesian spatial modeling of extreme precipitation return levels,” *J Am Stat Assoc* vol. 102, pp. 824-840, 2007.

- [65] D. J. Spiegelhalter, N. G. Best, B. P. Carlin, A. van der Linde “Bayesian measures of model complexity and fit,” *J Roy Stat Soc, Ser B*, vol. 64, no. 4, pp. 583-639, 2002.
- [66] D. Cooley, J. Cisewski, R. J. Erhardt, S. Jeon, E. Mannshardt, B. O. Omolo, and Y. Sun, “A survey of spatial extremes: measuring spatial dependence and modeling spatial effects,” *REVSTAT* **10**, pp. 135-165, 2012.
- [67] H. Sang, and A. E. Gelfand, “Continuous Spatial Process Models for Spatial Extreme Values,” *Journal of Agricultural, Biological and Environmental Statistics* **15**, pp. 49-65, 2010.
- [68] A. C. Davison, and M. M. Gholamrezaee, “Geostatistics of extremes,” *Proc. R. Soc. Lond. Ser. A* **468**, pp. 581-08, 2012.
- [69] A. C. Davison, S. Padoan, and M. Ribatet, “Statistical modeling of spatial extremes,” *Statist. Sci.* **27** (2):161-186, 2012.
- [70] C. Varin, N. Reid, and D. Firth, “An overview of composite likelihood methods,” *Statistica Sinica* **21**(1):5-42, 2011.
- [71] K. Aas, C. Czado, A. Frigessi, and H. Bakken, “Pair-copula constructions of multiple dependence,” *Insurance: Mathematics and Economics* **44**, pp. 182-198, 2009.
- [72] H. Joe, H. Li, and A. K. Nikoloulopoulos, “Tail dependence functions and vine copulas,” *Journal of Multivariate Analysis*. vol. 101, no. 1, pp. 252-270, 2010.
- [73] A. T. Ihler, S. Kirshner, M. Chil, A. W. Robertson and P. Smyth, “Graphical models for statistical inference and data assimilation,” *Physica D*, vol. 230, pp. 72-87, 2007.
- [74] F. V. Jensen, *An introduction to Bayesian networks*, Springer, 1996.
- [75] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhater, *Probabilistic Networks and Expert Systems*, Springer, 2003.

- [76] R. Chellappa and A. K. Jain, *Markov Random Fields: Theory and Application*, Academic Press, 1993.
- [77] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Chapman & Hall, 2005.
- [78] F. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 498-519, 2001.
- [79] J. M. Hammersley and P. E. Clifford, “Markov fields on finite graphs and lattices,” unpublished manuscript, 1971.
- [80] J. Ma, J. Peng, S. Wang, and J. Xu, “Estimating the Partition Function of Graphical Models Using Langevin Importance Sampling,” *Journal of Machine Learning Research*, W&CP, vol. 31, pp. 433-441, 2013.
- [81] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, “Model selection in Gaussian graphical models: High-dimensional consistency of ℓ_1 -regularized MLE,” In *Advances in Neural Information Processing Systems 22*, MIT Press, 2009.
- [82] C. T. Ireland and S. Kullback, “Contingency tables with given marginals,” *Biometrika*, vol. 55, pp. 179-188, 1968.
- [83] I. Csiszár and P. C. Shields, *Information Theory and Statistics: A Tutorial*, Foundations and Trends in Communication and Information Theory. NOW Publisher, Inc., 2004.
- [84] T. P. Speed and H. T. Kiiveri, “Gaussian Markov distributions over finite graphs,” *The Annals of Statistics*, vol. 14, no. 1, pp. 138-150, 1986.
- [85] P.-F. Xu, J. Guo, and M.-L. Tang, “An improved Hara-Takamura procedure by sharing computations on junction tree in Gaussian graphical models,” *Statistics and Computing*, vol. 22, no. 5, pp. 1125-1133, 2012.

- [86] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kauffman, 1988.
- [87] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Comput.*, vol. 13, pp. 2173-2200, 2001.
- [88] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-Sums and Belief Propagation in Gaussian Graphical Models," *Journal of Machine Learning Research*, vol. 7, pp. 2031-2064, 2006.
- [89] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky, "Embedded Trees: Estimation of Gaussian processes on graphs with cycles," *IEEE Trans. Signal Process.*, vol. 52, no. 11, pp. 3136-3150, 2004.
- [90] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust Distributed Estimation using the Embedded Subgraphs Algorithm," *IEEE Trans. Signal Process.*, vol. 54, pp. 2998-3010, 2006.
- [91] V. Chandrasekaran, J. K. Johnson, and A. S. Willsky, "Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1916-1930, 2008.
- [92] Y. Saad, *Iterative Methods for Sparse Linear Systems, second edn*, SIAM, Philadelphia, PA, 2003.
- [93] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modeling Extremal Events for Insurance and Finance*, Springer, 1997.
- [94] J. R. M. Hosking, J. R. Wallis, "Parameter and Quantile Estimation for the Generalized Pareto Distribution," *Technometrics* vol. 29, pp. 339-349, 1987.
- [95] M. North, "Time-dependent stochastic models of floods," *Journal of Hydraulic Division ASCE*, vol. 106, pp. 649C655, 1980.

- [96] P. K. Trivedi, and D. M. Zimmer, *Copula Modeling: An Introduction for Practitioners*. Hanover, MA, USA: now Publishers Inc, 2007.
- [97] A. Sklar, “Fonctions de répartition à n dimensions et leurs marges,” *Publications de l’Institut de Statistique de L’Université de Paris* 8, pp. 229-231, 1959.
- [98] E. Bouyè, V. Durrleman, A. Nikeghbali, G. Riboulet, and T. Roncalli, *Copulas for finance - a reading guide and some applications*. Group de Recherche Opérationnelle, Crédit Lyonnais, working paper, 2000.
- [99] P. Embrechts, F. Lindskog, and A. McNeil, “Modeling dependence with copulas and applications to risk management”, in *Handbook of Heavy Tailed Distributions in Finance*, ed. S. Rachev, Elsevier, Chapter 8, pp. 329-384, 2003.
- [100] S. Demarta, and A. J. McNeil, “The t copula and related copulas,” *International Statistical Review* **73**(1), pp. 111-129, 2005.
- [101] P. Capéraà, A. L. Fougères, and C. Genest, “Bivariate distributions with given extreme value attractor,” *Journal of Multivariate Analysis*, vol. 72, pp. 30-49, 2000.
- [102] G. Gudendorf, and J. Segers, “Extreme-value copulas,” In W. H. P. Jaworski, F. Durante and T. Rychlik (Eds.), *Proceedings of the Workshop on Copula Theory and its Applications*. Springer, pp. 127-146, 2010.
- [103] J. Segers, “Max-Stable Models for Multivariate Extremes,” *REVSTAT* vol. 10, no. 1, pp. 61-82, 2012.
- [104] R. A. Horn, C. R. Johnson, *Matrix analysis*, Cambridge University Press, 1990.
- [105] C. Wang, D. Sun, and K. C. Toh, “Solving log-determinant optimization problems by a Newton-CG primal point algorithm”, *Society for Industrial and Applied Mathematics*, vol. 20, pp. 2994-3013, 2009.

- [106] H. Liu, K. Roeder, and L. Wasserman, “Stability Approach to Regularization Selection (StARS) for High Dimensional Graphical Models,” *Advances in Neural Information Processing Systems*, 2010.
- [107] N. Meinshausen, P. Bühlmann, “Stability Selection,” *Journal of the Royal Statistical Society, Series B*, vol. 72, pp. 417-473, 2010.
- [108] K. Sachs, O. Perez, D. Peer, D. A. Lauffenburger, and G. P. Nolan “Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data,” *Science Magazine* vol. 308, pp. 523-529, 2005.
- [109] G. Henderson, E. Ifeachor, N. Hudson, C. Goh, N. Outram, S. Wimalaratna, C. Del Percio, and F. Vecchio, “Development and assessment of methods for detecting dementia using the human electroencephalogram,” *IEEE Transactions on Biomedical Engineering* 53, 2006.
- [110] J. Dauwels, F. Vialatte, and A. Cichocki, “Diagnosis of alzheimer’s disease from EEG signals: Where are we standing?” *Current Alzheimer Research*, vol. 7, pp. 487-505, 2010.
- [111] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society*, vol. 39, Series B, pp. 1-38, 1977.
- [112] G. Rodriguez-Yam, R. Davis, and L. Scharf, “Efficient Gibbs Sampling of Truncated Multivariate Normal with Application to Constrained Linear Regression,” Technical Report, Department of Statistics, Columbia University, 2004.
- [113] P. J. Green, “On Use of the EM for Penalized Likelihood Estimation,” *Journal of the Royal Statistical Society*, Series B, vol. 52, No. 3, pp. 443-452, 1990.
- [114] S. Li, L. Hsu, J. Peng and P. Wang, “Bootstrap inference for network construction with an application to a breast cancer microarray study,” *Annals of Applied Statistics*, vol. 7, no. 1, pp. 391-417, 2013.

- [115] <http://webapp.icpsr.umich.edu/GSS/>
- [116] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*, John Wiley and Sons, 1990.
- [117] A. Bhattacharya and D. B. Dunson, “Simplex Factor Models for Multivariate Unordered Categorical Data,” *Journal Amer. Stat. Assoc.* 107, pp. 362-377, 2012.
- [118] <http://voteview.com/senate109.htm>
- [119] M. Lavielle, and G. Teyssière, “Detection of Multiple Change-Points in Multivariate Time Series,” *Lithuanian Mathematical Journal*, vol. 46, pp. 287–306, 2006.
- [120] B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai, “An Algorithm for Optimal Partitioning of Data on an Interval,” *Signal Proces. Letter*, vol. 12, no. 2, pp. 105-108, 2005.
- [121] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,” *Circulation* vol. 101, pp. 215-220, 2000.
- [122] M. J. Choi, V. Chandrasekaran, D. M. Malioutov, J. K. Johnson, and A. S. Willsky, “Multiscale stochastic modeling for tractable inference and data assimilation,” *Comput. Methods Appl. Mech. Engrg.*, vol. 197, pp. 3492-3515, 2008.
- [123] P. L. Speckman and D. C. Sun, “Fully Bayesian spline smoothing and intrinsic autoregressive priors,” *Biometrika*, vol. 90, pp. 289-302, 2003.
- [124] G. Strang, *Computational Science and Engineering*, Wellesley-Cambridge Press, 2007.

- [125] J. R. M. Hosking, J. R. Wallis and E. F. Wood, “Estimation of the generalized extreme-value distribution by the method of probability-weighted moments”, *Technometrics*, vol. 27, pp. 251-261, 1985.
- [126] E. Castillo, and A. S. Hadi, “Parameter and quantile estimation for the generalized extreme-value distribution,” *Environmetrics*, vol. 5, pp. 417-432, 1994.
- [127] A. J. Laub, *Matrix Analysis for Scientists and Engineers*, SIAM: Society for Industrial and Applied Mathematics, 2004.
- [128] F. W. Scholz, “The Bootstrap Small Sample Properties,” Technical Report, Boeing Computer Services, Research and Technology, 2007.
- [129] C. van Loan, *Computational frameworks for the fast Fourier transform*, SIAM Press, Philadelphia, 1992.
- [130] X. D. Zhang, “The Laplacian eigenvalues of graphs: a survey,” *Preprint, arXiv:1111.2897v1*, 2011.
- [131] M. T. Heath, *Scientific Computing: An Introductory Survey*, McGraw Hill, New York, 2002.
- [132] T. J. Henderson, and R. J. Tibshirani, *Generalized Additive Models*, Chapman and Hall, London, 1990.
- [133] J. S. Hodges, and D. J. Sargent, “Counting degrees of freedom in hierarchical and other richly parameterized models,” *Biometrika*, vol. 88, pp. 367-379, 2001.
- [134] Oceanweather Inc. *GOMOS - USA Gulf of Mexico Oceanographic Study, Northern Gulf of Mexico Archive*, 2005.
- [135] M. Titsias, M. Lázaro-Gredilla, “Doubly Stochastic Variational Bayes for non-Conjugate Inference,” *Journal of Machine Learning Research*, W&CP, vol. 32, no. 1, 1971-1979, 2014.

- [136] M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley, “Stochastic variational inference,” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303-1347, 2013.
- [137] D. M. Malioutov, J. K. Johnson, M. J. Choi, and A. S. Willsky, “Low-rank variance approximation in GMRF models: Single and multiscale approaches,” *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4621-4634, 2008.
- [138] Y. Yue and P. L. Speckman, “Nonstationary Spatial Gaussian Markov Random Fields,” *Journal of Computational and Graphical Statistics*, vol. 19, no. 1, pp. 96-116, 2010.
- [139] C. J. Paciorek, “Spatial models for point and areal data using Markov random fields on a fine grid,” *Electronic Journal of Statistics*, vol. 7, pp. 946-972, 2013.
- [140] G. Casella, “An Introduction to Empirical Bayes Data Analysis,” *American Statistician*, vol. 39, no. 2, pp. 83-87, 1985.
- [141] M. J. Wainwright, and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends in Machine Learning*, vol. 1, nos. 1-2, pp. 1-305, 2008.
- [142] H. Robbins and S. Monro, “A stochastic approximation method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400-407, 1951.
- [143] R. Salakhutdinov, S. Roweis, and Z. Ghahramani, “Optimization with EM and Expectation-Conjugate-Gradient,” *Proc. ICML*, 2003.
- [144] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” *arXiv:1212.5701*, 2012.
- [145] T. Schaul, S. Zhang, Y. LeCun, “No more Pesky Learning Rates,” *Proc. ICML*, 2013.

- [146] R. Ranganath, C. Wang, D. M. Blei, and E. P. Xing, “An Adaptive Learning Rate for Stochastic Variational Inference,” *J. Mach. Learning Research*, W&CP vol. 28, 2013.
- [147] M. Schmidt, N. L. Roux, and F. Bach, “Minimizing Finite Sums with the Stochastic Average Gradient,” *arXiv:1309.2388*, 2013.
- [148] R. Johnson, and T. Zhang, “Accelerating Stochastic Gradient Descent using Predictive Variance Reduction,” *Proc. NIPS*, 2013.
- [149] S. Mandt, and D. Blei, “Smoothed Gradients for Stochastic Variational Inference,” *Proc. NIPS*, 2014.
- [150] J. Sheffield, G. Goteti, and E. F. Wood, “Development of a 50-yr high-resolution global dataset of meteorological forcings for land surface modeling,” *J. Climate*, vol. 19, no. 13, pp. 3088-3111, 2006.
- [151] K. Kamiguchi, O. Arakawa, A. Kitoh, A. Yatagai, A. Hamada, and N. Yasutomi, “Development of APHRO_JP, the first Japanese high-resolution daily precipitation product for more than 100 years,” *Hydrological Research Letters* vol. 4, pp. 60-64, 2010.
- [152] L. Mitas and H. Mitasova, “Spatial interpolation,” *In: Longley, P., Goodchild, M., Maguire, D., Rhind, D., (Eds.), Geographical Information Systems: Principles, Techniques, Management and Applications*, vol. 1, pp. 481-492, 1999.
- [153] Ø. Hjelle and M. Dæhlen, *Triangulations and Applications*, Berlin: Springer, 2006.
- [154] Y. Lin, S. Zhu, D. D. Lee, and B. Taskar, “Learning Sparse Markov Network Structure via Ensemble-of-Trees Models,” *12th International Conference on Artificial Intelligence and Statistics.*, 2009.

- [155] S. Kirshner, “Learning with tree-averaged densities and distributions,” *Advances in Neural Information Processing Systems (NIPS)*, 2008.
- [156] M. Meilă, and M. I. Jordan, “Learning with Mixtures of Trees,” *Journal of Machine Learning Research*, vol. 1, pp. 1-48, 2000.
- [157] S. Kirshner, and P. Smyth, “Infinite Mixtures of Trees,” *Int. Conf. Machine Learning*, pp. 417-423, 2007.
- [158] R. Silva, and R. B. Gramacy, “MCMC methods for Bayesian mixtures of copulas,” In *Proc. 25th Int. Conf. Artificial Intell. Stat.*, pp. 512-519, 2009.
- [159] A. W. Ledford, and J. A. Tawn, “Statistics for near independence in multivariate extreme values,” *Biometrika* **83**(1), pp. 169-187, 1996.
- [160] G. Draisma, H. Drees, A. Ferreira, and L. de Haan, “Bivariate tail estimation: dependence in asymptotic independence,” *Bernoulli* **10**, pp. 251-280, 2004.
- [161] M. Meilă, and T. Jaakkola, “Tractable Bayesian learning of tree belief networks,” *Statistics and Computing* **16**(1): 77-92, 2006.
- [162] L. W. Beineke, R. J. Wilson, and P. J. Cameron *Topics in Algebraic Graph Theory*. New York: Cambridge University Press, 2004.
- [163] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific., 2003.
- [164] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer, 1993.
- [165] M. Leordeanu and M. Hebert, “Smoothing-based Optimization,” *Proc. CVPR*, pp. 1-8, 2008.
- [166] C. Shen, M. J. Brooks, and A. van den Hengel, “Fast Global Kernel Density Mode Seeking: Applications to Localization and Tracking,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1457-1469, 2007.

- [167] R. M. Cooke, C. Kousky, and H. Joe, "Micro Correlations and Tail Dependence," in D. Kurowicka, H. Joe (Eds.), *Dependence Modeling: Vine Copula Handbook*. Singapore: World Scientific Publishing Co., pp. 89-113.
- [168] J. L. Wadsworth, J. A. Tawn, "Dependence modeling for spatial extremes," *Biometrika*, pp. 1-20, 2012.
- [169] A. C. Davison, R. Huser, and E. thibaud, "Geostatistics of Dependent and Asymptotically Independent Extremes," *Math. Geosci*, vol. 45, pp. 511-529, 2013.
- [170] B. Renard, and M. Lang, "Use of a Gaussian copula for multivariate extreme value analysis: Some case studies in hydrology," *Advances in Water Resources*. **30**(4), pp. 897-912.
- [171] A. S. Willsky, "Multiresolution Markov models for signal and image processing," *Proc. IEEE*, vol. 90, no. 8, pp. 1396-1458, Aug, 2002.
- [172] M. J. Choi, V. Chandrasekaran, and A. S. Willsky, "Gaussian Multiresolution Models: Exploring Sparse Markov and Covariance Structure," *IEEE transaction on signal processing*, vol. 58, No. 3, pp. 1012-1024, 2010.
- [173] R. Tolimieri, M. An, and C. Lu, *Mathematics of Multidimensional Fourier Transform Algorithms*, Springer, 1993.
- [174] L. Adams, " m -Step preconditioned conjugate gradient methods," *SIAM J. Sci. Stat. Comput.*, vol. 6, pp. 452-463, 1985.
- [175] O. Axelsson, "Bounds of eigenvalues of preconditioned matrices," *SIAM J. Matrix Anal. Applicat.*, vol. 13, no. 3, pp. 847-862, 1992.