Copula-based Analysis for Dependent Count Data

by Mingze Zhang

B.S. in Statistics, Jun 2017, University of Science and Technology of China
M.S. in Statistics, Dec 2021, The George Washington University

A Dissertation submitted to

The Faculty of
The Columbian College of Arts and Sciences
of The George Washington University
in partial satisfaction of the requirements
for the degree of Doctor of Philosophy

May 15, 2022

Dissertation directed by

Huixia Judy Wang
Professor of Statistics

The Columbian College of Arts and Sciences of The George Washington University certifies that Mingze Zhang has passed the Final Examination for the degree of Doctor of Philosophy as of March 2, 2022. This is the final and approved form of the dissertation.

Copula-based Analysis for Dependent Count Data

Mingze Zhang

Dissertation Research Committee:

Huixia Judy Wang, Professor of Statistics, Dissertation Director

Reza Modarres, Professor of Statistics, Committee Member

James Livsey, Adjunct Professor of Statistics, Committee Member

© Copyright 2022 by Mingze Zhang
All rights reserved

Acknowledgments

This work would not have been possible without the support from the fellowship and assistantship from George Washington University and National Science Foundation grant DMS-1712760.

I am grateful to all of those with whom I have had the pleasure to work during this and other related projects. Each of the members of my Dissertation Committee has provided me extensive personal and professional guidance and taught me a great deal about both scientific research and life in general. I would especially like to thank Dr. Huixia Judy Wang, the chairman of my committee. As my teacher and mentor, she has taught me more than I could ever give her credit for here. She has shown me, by her example, what a good scientist (and person) should be.

The most important support to me in the pursuit of this project came from my family. I would like to thank my parents, whose love and guidance are with me in whatever I pursue. They are my ultimate role models.

Abstract

Copula-based Analysis for Dependent Count Data

The analysis of count time series is challenging due to the nonuniqueness of dependence measure caused by the existence of ties. Various methods have been developed but most of them rely on restrictive parametric distributional assumptions. In this dissertation, we aim to develop copula-based methods for analyzing dependent count data, including count time series, multivariate count time series and spatio-temporal count data.

In the first project, we propose a copula-based Markov model for analyzing count time series, where the temporal dependence is captured by a parametric copula. One challenge in using copula to analyze count data is due to the identifiability issue, which arises from the discrepancy of using a continuous copula function to characterize discrete distributions. We show that the identifiability can be ensured in the regression setup under one sufficient condition. Resolving the identifiability issue allows us to develop a cross-validation procedure to select the appropriate copula to capture different types of temporal dependence, leading to more flexibility in modeling. We propose an estimation procedure and establish the asymptotic properties of the proposed estimator. The proposed method is data adaptive and computationally efficient to capture the temporal dependence. It also provides a convenient way to construct both point and interval predictions at a future time point. Through a simulation study and the analyses of a baseball data and a COVID-19 daily death data, we show that our method produces more stable point and interval predictions than existing methods based on Gaussian copula and autoregression.

In the second project, we focus on the analysis of multivariate count time series. We consider using the copula-linked Dvine (CDvine) to simultaneously capture the temporal and cross-sectional dependence. To reduce the computational burden, a sequential selection approach is used to select the order and bivariate copulas of each univariate Dvine, and a two-stage maximum likelihood estimator is proposed for estimation. We conduct simulation

studies to examine the performance of the sequential model selection procedure and demonstrate the flexibility of the CDvine. The application to a Covid-19 death data shows that the proposed method leads to better prediction over existing approaches.

The third project focuses on the analysis of spatio-temporal count data with zero-inflations. In the first two projects, the marginal distributions of the count data are estimated by fitting parametric regressions such as Poisson regression, which may be restrictive in some applications. To gain more flexibility, we extend the copula-based idea in the first two projects, and develop a new procedure based on nonparametric estimation of the marginal distribution through gradient boosting model (GBM). We consider zero-inflated generalized Poisson (ZIGP) distribution to account for zero-inflation and potential over-dispersion. Instead of specifying the link functions, we model the parameters non-parametrically using GBM. The spatial and temporal dependence is captured by CDvine and the Matérn correlation function. Simulation study is conducted to assess the performance of the proposed nonparametric estimation of the marginal distribution and the prediction accuracy of the proposed procedure over existing methods. This proposed method is applied to a Covid-19 mortality data from Northeastern US, and we observe significant improvement over the method based on parametric estimation of the marginal distribution.

Table of Contents

Acknowledgments . . . . . **iv**

Abstract . . . . . **v**

List of Figures . . . . . **ix**

List of Tables . . . . . **x**

1 Introduction **1**

1.1 Motivation and Objective 1
1.2 An Introduction to Copula 5

2 Copula-based Analysis for Count Time Series **7**

2.1 Introduction 7
2.2 Proposed Method 10
2.2.1 Copula-based Markov Model 10
2.2.2 Identifiability Issue 11
2.2.3 Parameter Estimation 12
2.2.4 Point and Interval Prediction 14
2.2.5 Asymptotic Properties 17
2.2.6 Copula Selection 18
2.3 Simulation Study 19
2.3.1 Parameter Estimation 20
2.3.2 Conditional Quantile Estimation with Selected Copula 21
2.3.3 Prediction at Future Time Points 22
2.4 Real Data Analysis 24
2.4.1 Baseball Data 24
2.4.2 COVID-19 Data 27
2.5 Discussion 31
2.6 Proof of Propositions and Theorems 31
2.6.1 Proof of Proposition 2.1 31
2.6.2 Proof of Theorem 2.1 33

3 Modeling Multivariate Time Series of Counts with Copula-linked D-vines **37**

3.1 Introduction 37
3.2 The D-vine-based Count Time Series Model 40
3.2.1 Univariate D-vine (uDvine) for Continuous Time Series 41
3.2.2 Univariate D-vine (uDvine) for Count Time Series 44
3.2.3 uDvine for Count Time Series Under the Regression Setup 47
3.2.4 Copula-linked D-vines (CDvine) Model for Multivariate Count Time Series 49
3.2.5 Two-stage Estimation Procedure 50
3.2.6 Point and Interval Prediction 53

3.2.7 Asymptotic Properties . . . 56
3.2.8 Selection of Pair-copulas for the uDvine . . . 58
3.3 Simulation Studies . . . 59
3.3.1 Copula Selection for the uDvine . . . 60
3.3.2 Comparison with Multivariate Poisson Autoregression Model (MPAR) . . . 61
3.3.3 Performance of Bootstrap Variance . . . 63
3.4 Real Data Analysis . . . 64
3.5 Discussion . . . 68
3.6 Proof of Propositions and Theorems . . . 69
3.6.1 Proof of Proposition 3.1 . . . 69
3.6.2 Proof of Theorem 3.1 . . . 70
3.6.3 Proof of Theorem 3.2 . . . 71

4 Copula-based Model for Spatio-temporal Data . . . 75

4.1 Introduction . . . 75
4.2 Proposed Model and Estimation Procedure . . . 77
4.2.1 Zero-inflated Generalized Poisson Linear Model (ZIGP-LM) . . . 79
4.2.2 Gradient Boosting Model (GBM) . . . 81
4.2.3 ZIGP Gradient Boosting Model (ZIGP-GBM) . . . 86
4.2.4 Matérn Correlation Class . . . 89
4.2.5 Three-stage Estimation Procedure . . . 90
4.2.6 Point and Interval Prediction . . . 91
4.3 Simulation Study . . . 93
4.3.1 Simulation for Univariate Time series . . . 93
4.3.2 Simulation for Spatio-temporal Data . . . 97
4.4 Real Data Analysis . . . 101
4.5 Discussion . . . 103

5 Conclusion and Discussion . . . 104

5.1 Concluding Remarks . . . 104
5.2 Future Research . . . 105

Bibliography . . . 107

List of Figures

1.1 Flow chart of the dissertation 4

2.1 Boxplots of $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\alpha}$ for $n = 200$ (white boxplots) and 500 (grey boxplots) across 500 simulations with α corresponding to $\tau = 0.3$. True values are shown as a black horizontal dashed line. COMC: copula-based Morkov model for count data; PF: particle filtering. 22

2.2 Boxplots of $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\alpha}$ for $n = 200$ (white boxplots) and 500 (grey boxplots) across 500 simulations with α corresponding to $\tau = 0.7$. True values are shown as a black horizontal dashed line. COMC: copula-based Morkov model for count data; PF: particle filtering. 23

2.3 Daily mortality in VA. Black line corresponds to the observation, while the red, blue and green shaded areas are the prediction intervals given by COMC, PF and GPA. 32

3.1 A 5-dimendion D-vine. 43

4.1 Scatter plot of Y versus X, where Y is from five unique uniform distributions in different regions of X. 86

4.2 Prediction and pseudo residuals after 1, 3, 5, 10 and 20 iterations of the GBM algorithm for analyzing a randomly generated data set from Example 4.1. . . 87

List of Tables

2.1 1000×bias and 1000×RMSE of different estimators of β_1, β_2 and α. 21
2.2 Percentages of copulas chosen by cross-validation across 500 simulations. . . 24
2.3 100×MSE of $\widehat{Q}_q(Y_{n+1}|Y_n)$ from different methods at $q = 0.5$ and $q = 0.9$. . . 25
2.4 Coverage percentages and 10×average lengths of 90% prediction intervals for predicting Y_{n+1}. 26
2.5 Coverage percentages and 10×average lengths of $[0, \widehat{Q}_{0.9}(Y_{n+1}|Y_n, \boldsymbol{X})]$ for predicting Y_{n+1}. 27
2.6 100× LOGS of prediction for Y_{n+1} given by COMC_0, COMC_{TC}, COMC_{SC}, PF and PA. 28
2.7 Estimated parameters in the COMC model of order 2 for the baseball data. . 28
2.8 One-step ahead prediction for Y_t from the baseball data produced from three methods. 29
2.9 Estimated parameters in the COMC model of order 1 for the COVID-19 data. 30
2.10 One-step ahead prediction for Y_t from COVID-19 data produced from three methods. 31

3.1 Percentages of times the uDvine order and pair-copulas are correctly chosen for simulation data generated from a uDvine model with $p = 1$. 61
3.2 Percentages of times the uDvine order and pair-copulas are correctly chosen for simulation data generated from a uDvine model with $p = 2$ and three combinations of pair-copulas in T_1 and T_2. 61
3.3 Coverage percentages and average lengths of 90% prediction intervals constructed by fitting the MPAR and CDvine models for Cases 1-3. 64
3.4 10× LOGS of joint prediction obtained by fitting the MPAR and CDvine models for Cases 1-3. Values in the parentheses are standard errors. 65
3.5 Monte Carlo variance and the average of bootstrap variances of the first-stage estimators $\{\hat{\beta}_i\}_{i=1}^2$ and $\{\hat{\alpha}_i\}_{i=1}^2$ in Cases 1-2 where data are generated from CDvine(1) and CDvine(2). Values in the parentheses are standard errors. . . 66
3.6 Monte Carlo variance and the average of bootstrap variance of the second-stage estimators $\hat{\gamma}$ in Cases 1-2 where data are generated from CDvine(1) and CDvine(2). Values in the parentheses are standard errors. 67
3.7 The estimated correlation matrix of the cross-sectional Gaussian Copula using COVID-19 data from North Carolina (NC), Virginia (VA) and Maryland (MD). Values in the parentheses are bootstrap standard errors. 67
3.8 Average LOGS for component-wise prediction of COVID-19 deaths in North Carolina (NC), Virginia (VA) and Maryland (MD), separately, from three methods.Values in the parenthesis are the standard errors. 68
3.9 Coverage percentages and average lengths of 90% component-wise prediction intervals for the COVID-19 data in NC, VA and MD from three methods. Values in the parenthesis are the standard errors. 68

4.1 The integrated mean squared error (IMSE) of ZIGP-GBM and ZIGP-LM for estimating the three parameter functions, $\mu(\cdot)$, $\varphi(\cdot)$, and $p(\cdot)$, in the ZIGP regression model for four cases, where data are generated from the ZIGP model with four different transformations for the parameter functions. Values in the parentheses are standard errors. 96
4.2 Comparison of ZIGP-GBM and ZIGP-LM for predicting Y_{n+1} assuming data generated from the trigonometric, quadratic, linear and tan-linear models. Values in the parentheses are standard errors. 98
4.3 Distance (in km) between states in Northeastern United States. 98
4.4 Coefficients in the ZIGP regression model with the trigonometric transformation at each location for the simulation study of spatio-temporal data. 99
4.5 The coverage percentages and average lengths of 95% prediction intervals and LOGS from two types of prediction methods for predicting Y_{n+1} at PA using simulated data mimicking the COVID-19 study. Values in the parentheses are standard errors. 100
4.6 Comparison of ZIGP-GBM and ZIGP-LM for predicting $\boldsymbol{Y}_{n+1}$ in nine states using simulated data. Values in the parentheses are standard errors. 101
4.7 Comparison of ZIGP-GBM and ZIGP-LM for predicting Y_{n+1} using the COVID-19 data in 9 states. Values in the parentheses are standard errors. 103

Chapter 1

Introduction

1.1 Motivation and Objective

The main goal of this dissertation is to develop flexible and computational convenient methods for dependent count data, including: (1) univariate count time series, (2) multivariate count time series and (3) spatio-temporal count data. Count time series is a sequence of data that has discrete marginal distributions. Often, the discreteness arises in the form of counts taking values in $\{0, 1, 2, \ldots\}$. Count series are widely used when describing storm numbers, accident tallies, wins by a sports team, disease cases, etc. For example, in Davis and Wu (2009), the monthly counts of poliomyelitis cases in the USA from 1970 to 1983 were studied. Heinen and Rengifo (2007) studied the number of trades within 5-minute intervals in five most important US department stores during the year of 1999. Chiogna and Gaetan (2010) studied the number of species from 1932 to 1972 in herbaceous plant communities in a mixed-grass prairie in Hays, Kansas (USA). To conduct valid inference and obtain accurate prediction, it is important to model and incorporate the temporal dependence, along with cross-sectional and spatial dependence for multivariate and spatio-temporal count data.

There have been many works for the analysis of dependent count data; see for example Davis and Wu (2009), Fokianos and Tjøstheim (2011)) and Fokianos et al. (2009) for count time series, and Aldor-Noiman et al. (2016) and Clark et al. (2018) for spatio-temporal count data. However, many of these works are either based on relatively strong model assumptions, such as the Poisson or Negative Binomial distribution conditioned on past observations

or an intensity process, or have limitations for capturing complex dependence. We aim to develop more flexible and computationally convenient procedures that rely on weaker model assumptions and can accommodate various types of dependence.

In the first project, we propose a copula-based Markov model for analyzing count time series, where the temporal dependence is captured by a parametric copula. One challenge in using copula to analyze count data is due to the identifiability issue, which arises from the discrepancy of using a continuous copula function to characterize the discrete distributions. We show that the identifiability can be ensured in the regression setup under one sufficient condition. Resolving the identifiability issue allows us to develop a cross-validation procedure to select the appropriate copula to capture different types of temporal dependence, leading to more flexibility in modeling. We propose an estimation procedure and establish the asymptotic properties of the proposed estimator. The proposed method is data adaptive and computationally efficient to capture the temporal dependence. It also provides a convenient way to construct both point and interval predictions at a future time point. Through a simulation study and the analyses of a baseball data and a COVID-19 daily death data, we show that our method produces more stable point and interval predictions than existing methods based on Gaussian copula and autoregression.

In the second project, we focus on the analysis of multivariate count time series. We consider using the copula-linked D-vine (CDvine) to simultaneously capture the temporal and cross-sectional dependence. To reduce the computational burden, a sequential selection approach is used to select the order and pair-copulas of each univariate D-vine, and a two-stage estimation procedure is proposed. We conduct simulation studies to examine the performance of the sequential model selection procedure and demonstrate the flexibility of the CDvine. The application to a COVID-19 death data shows that the proposed method leads to better prediction over existing approaches.

The third project focuses on the analysis of spatio-temporal count data with zero-inflation and over-dispersion. In the first two projects, the marginal distributions of the count data

are estimated by fitting parametric regressions such as Poisson regression, which may be restrictive in some applications. To gain more flexibility, we extend the copula-based idea in the first two projects, and develop a new procedure based on nonparametric estimation of the marginal distribution through gradient boosting model (GBM). We consider zero-inflated generalized Poisson (ZIGP) distribution to account for zero-inflation and potential over-dispersion. Instead of specifying the link functions, we estimate the parameter functions nonparametrically using GBM. The spatial and temporal dependence is still captured by CDvine, and the Matérn correlation is used to capture the spatial dependence. Simulation study is conducted to assess the semiparametric estimation of the marginal distribution and how it outperforms the traditional linear model. We also do simulation to compare the prediction generated from the CDvine with marginal distribution estimated by GBM and linear model with specific link functions, respectively, and show that CDvine with marginal distribution estimated by GBM can produce better prediction. This proposed method is applied to a COVID-19 mortality data from Northeastern US, and we observe significant improvement over the CDvine with marginal distribution estimated parametrically in the second project.

We give a brief introduction to copula in the next section. The rest of the dissertation is organized as follows. In Chapter 2, we propose a copula-based Markov model for count time series. In Chapter 3, we extend this model to multivariate count time series. In Chapter 4, we consider more flexible marginal distribution and incorporate spatial dependence. In Chapter 5, we conclude the thesis.

Figure 1.1 shows the flow chart of the dissertation and our main contributions. In the first project (Chapter 2), we propose a copula-based Markov model for analyzing count time series $(Y_1, \ldots, Y_n|\boldsymbol{X})$, where the temporal dependence is captured by a parametric copula and the marginal distribution of $(Y_t|\boldsymbol{X}_t)$ is modeled in a parametric form. In the second project (Chapter 3), we focus on the analysis of multivariate count time series. We consider using the copula-linked D-vine (CDvine) to simultaneously capture the temporal and cross-sectional

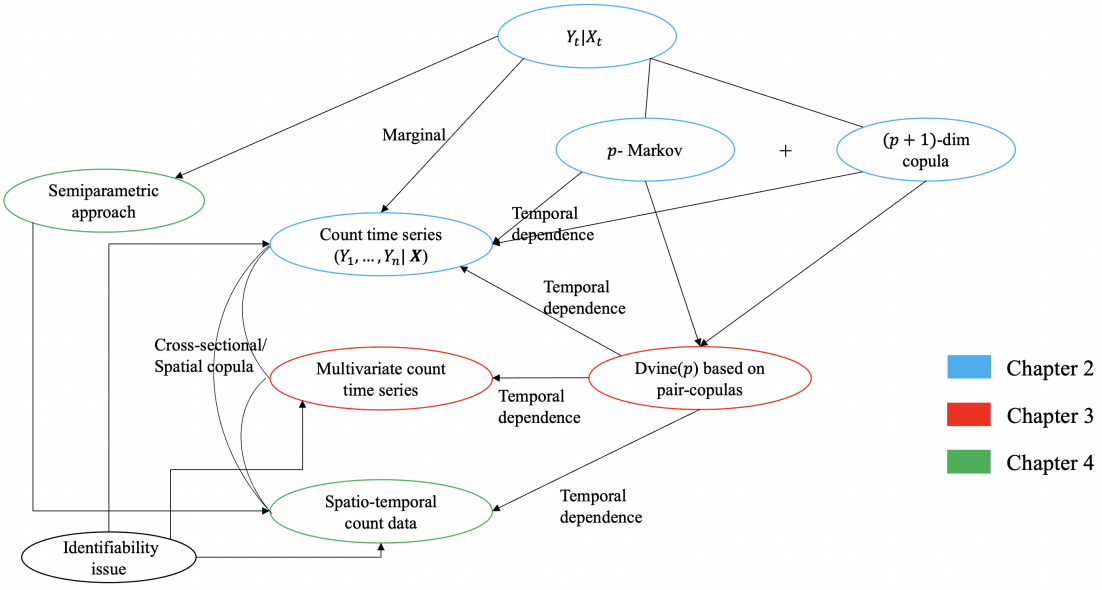

Figure 1.1: Flow chart of the dissertation

dependence. To reduce the computational burden, a sequential selection approach is used to select the order and bivariate copulas of each univariate D-vine truncated after order p (uDvine(p)). The third project focuses on the analysis of spatio-temporal count data with zero-inflation and over-dispersion. In the first two projects, the marginal distributions of the count data are estimated by fitting parametric regressions such as Poisson regression, which may be restrictive in some applications. To gain more flexibility, we extend the copula-based idea in the first two projects, and develop a new procedure based on semiparametric estimation of the marginal distribution through gradient boosting model (GBM). The spatial and temporal dependence is captured by CDvine and the Matérn correlation function. One challenge in using copula to analyse count data is due to the identifiability issue, which arises from the discrepancy of using a continuous copula function to characterize the discrete distribution. We find that the identifiability can be ensured in the regression setup under one sufficient condition. Throughout Chapters 2-4, we consider the regression setups to ensure the copula identifiability.

1.2 An Introduction to Copula

In this section we give a short review on copula, and we start from bivariate copulas. Let $X \sim F$, $Y \sim G$ be two dependent random variables with continuous distributions F and G. Denote $I = [0,1]$ as the unit interval. A bivariate copula can be used to obtain the joint distribution of X and Y. A bivariate copula is a function C from I^2 to I with the following properties:

1. For every u, v in I,

$$C(u,0) = C(0,v) = 0$$

and

$$C(u,1) = u \text{ and } C(1,v) = v.$$

2. For every u_1, u_2, v_1, v_2 in I such that $u_1 \leq u_2$ and $v_1 \leq v_2$,

$$C(u_2,v_2) - C(u_2,v_1) - C(u_1,v_2) + C(u_1,v_1) \geq 0.$$

The bivariate copula can be seen as a distribution function whose one-dimensional margins are uniform.

According to the Probability Integral Theorem (PIT), $F(X) \sim U(0,1)$ and $G(Y) \sim U(0,1)$. The following Sklar's theorem (Sklar (1959)) ensures that copula can join together one-dimensional distribution functions to form multivariate distribution function.

Sklar's theorem: Let H be a joint distribution function for X and Y with margins F and G. Then there exists a copula C such that for all x, y in $\bar{R}$,

$$H(x,y) = C(F(x), G(y)). \tag{1.1}$$

where $\bar{R}$ denote the extended real line $[-\infty, \infty]$. If F and G are continuous, then C is unique; otherwise, C is uniquely determined on Ran$F \times$ Ran$G$, where Ran$F$ and Ran$G$ denote the

ranges of F and G. Conversely, if C is a copula and F and G are distribution functions, then the function H defined above is a joint distribution function with margins F and G.

Sklar's Theorem also applies to n-dimensional copula, $n \geq 2$. Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be an n-dimensional random vector, and let H be its joint distribution function with margins F_1, F_2, ..., F_n. Then there exists a copula C such that for all x_1, x_2, ..., x_n in $\bar{R}$,

$$H(x_1, x_2, \ldots, x_n) = C(F_1(x_1), F_2(x_2), \ldots, F_n(x_n)). \tag{1.2}$$

Copula can be used to generate dependent uniform random variables. One procedure for generating such a pair (u, v) of uniform $(0, 1)$ variates is the conditional distribution method. Define conditional distribution function of V given $U = u$ as:

$$c_u(v) = P(V \leq v | U = u) = \lim_{\Delta u \to 0} \frac{C(u + \Delta u, v) - C(u, v)}{\Delta u} = \frac{\partial C(u, v)}{\partial u}.$$

The generation procedure is as follows:

1. Generate two independent uniform $(0, 1)$ variates u and t;
2. Set $v = c_u^{(-1)}(t)$, where $c_u^{(-1)}$ denotes a quasi-inverse of c_u .
3. The desired pair is (u, v).

To generate a bivariate (x, y) from copula in model (1.1), we can add another step in the end by letting: $x = F^{-1}(u)$ and $y = G^{-1}(v)$.

Chapter 2

Copula-based Analysis for Count Time Series

2.1 Introduction

With a surge in the applications from finance, environmental science and social science, there has been increasing interests in developing models and methods for analyzing time series of counts; see Davis and Liu (2016), Jung et al. (2006), Brandt et al. (2000), and Song et al. (2009). The analysis of count time series is challenging due to the nonuniqueness of dependence measure caused by the existence of ties. There exist some methods in the literature but many of them either rely on restrictive distributional assumptions, or are computationally intensive, or have limitations for capturing complex dependence. In this chapter, we aim to develop a flexible and computationally convenient procedure that reply on weaker assumptions and can accommodate various types of dependence.

Some existing methods assume that the data follows a Poisson or Negative Binomial distribution conditioned on past observations or an intensity process; see e.g. Davis and Wu (2009), Davis et al. (2003), Davis et al. (2000), Fokianos and Tjøstheim (2011)) and Fokianos et al. (2009). Ferland et al. (2006) proposed an integer-valued GARCH (INGARCH) model, and the observations $\{Y_t\}$ given the intensity process $\{\lambda_t\}$ follow a Poisson distribution and λ_t is a linear combination of its lagged values and lagged Y_t.

Other researchers have considered copula-based methods for analyzing time series of counts. One major advantage of copula-based methods is that it can separate the modeling of marginal and temporal dependence. Denuit and Lambert (2005) studied the concordance of integer-valued bivariate variables through the continuous extension (CE), and showed that the CE preserves Kendall's τ. The CE works through constructing continuous variables by adding a random perturbation taking values in [0,1] to the discrete variables. The constructed continuous variables are also referred to as jittered data in the literature. Madsen (2009) and Madsen and Fang (2011) applied the CE idea to analyze discrete longitudinal and spatial data by using Gaussian copula to capture the joint distribution of jittered data. Heinen and Rengifo (2007) proposed multivariate autoregressive models for count time series using CE and Gaussian copula. Rüschendorf (2009) proposed a distributional transform (DT) technique, so that standard uniform distribution can still be obtained from probability integral transformation even if the distribution is discrete. The CE-based methods use jitters to fill in the gap between discrete values, and then estimate parameters by maximizing the surrogate likelihood of the jittered data. Nikoloulopoulos (2013, 2016) showed that the maximum surrogate likelihood estimators based on CE may lead to biased estimations for multivariate discrete data, since they use information of jitters that are not in the observed data and the jittering is univariate without accounting for dependence. In addition, the CE-based methods are often time consuming since the surrogate likelihood needs to be approximated by averaging over multiple jitters. More recently, Jia et al. (2021) developed a general count time series model class using Gaussian copula with Hermite expansions and the particle filtering to approximate the likelihood.

The aforementioned copula-based methods all assume a latent Gaussian process, which may be restrictive in some applications. To gain more flexibility, it will be desirable to choose a copula from a class of families that better fits the data. However, this task faces a major challenge caused by the identifiability issue: the copula may not be unique when the marginal distribution is discrete. This issue arises from the discrepancy of using a continuous

copula function to characterize the discrete distribution. To overcome the limitations of existing methods, we propose a new copula-based Markov model for analyzing count time series data. While copula-based Markov models were also considered in Chen and Fan (2006), Chen et al. (2009), Rémillard et al. (2012) and Tang et al. (2019), they were all developed for continuous time series data. We formally address the identifiability issue and show that the identifiability can be ensured in the regression setup under one sufficient condition, that is, the dependence of the count response on one or more continuous covariates. Resolving the identifiability issue allows us to choose copula from a class of copula families to accommodate different types of temporal dependence, including, for instance, lower or upper tail dependence. We develop an estimation procedure based on appropriate handling of the likelihood of observed count data. Compared to existing approaches for analyzing count time series, the proposed method is more flexible as it does not restrict to latent Gaussian process, and it allows separate treatments of the univariate marginals and the temporal dependence. In addition, under the Markov assumption, we only need to model the dependence of data from adjacent time points. Therefore, the proposed procedure is computationally convenient as it avoids the challenge of high-dimensional integration as in Nikoloulopoulos (2013), Hughes (2015), Nikoloulopoulos (2016), to name a few.

The rest of this chapter is organized as follows. In Section 2.2, we introduce the copula-based Markov model and study the identifiability issue. In addition, we present the maximum likelihood estimator, establish its asymptotic properties, and propose a copula selection procedure. We assess the numerical performance of the proposed method through a simulation study in Section 2.3 and the analysis of a baseball data and a COVID-19 mortality data in Section 2.4. Finally we conclude the chapter in Section 2.5. All technical proofs are provided in Section 2.6.

2.2 Proposed Method

2.2.1 Copula-based Markov Model

Let $\boldsymbol{Y} = \{Y_t, t = 1,2,\ldots,n\}$ be the observed count time serie data at n time points, and $\boldsymbol{X} = (\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n)^T$ be the corresponding $n \times d$ covariate matrix, where $\boldsymbol{X}_t = (X_{t1},\ldots,X_{td})^T$ for $t = 1,\ldots,n$ and d is the dimension of the covariates. For instance, in the coastal hurricane study, researchers are interested in predicting the annual hurricane counts by using covariates information, such as North Atlantic oscillation (NAO), Northwest African rainfall, etc., from a few months earlier; see Elsner and Jagger (2006); Livsey et al. (2018).

We assume that the sequence of count $\{Y_t|\boldsymbol{X}, t = 1,2,\ldots,n\}$ is a Markov process of order p with discrete state space, that is, we assume

$$Y_t|Y_{t-1},\ldots,Y_1,\boldsymbol{X} \sim Y_t|Y_{t-1},\ldots,Y_{t-p},\boldsymbol{X}. \tag{2.1}$$

Under this assumption, the probabilistic properties of the sequence are fully determined by the joint distribution of neighboring time points $Y_{t-p},\ldots,Y_t$, denoted as $H(y_{t-p},\ldots,y_t|\boldsymbol{x})$. The Markov property in (2.1) assumes that the current outcomes only depend on their recent past and corresponding covariates.

In the following we first formally state the copula-based Markov model assumption for count time series. The Assumption A1 consists of three parts.

Assumption A1:

A1(a). At each time t, the marginal distribution of Y_t is characterized by $Y_t|\boldsymbol{X} = \boldsymbol{x} \sim G(y_t|\boldsymbol{x},\boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the unknown marginal parameter vector.

A1(b). The process is p-order Markovian in time satisfying $P(Y_t \leq y|Y_{t-1},\ldots,Y_1,\boldsymbol{X}) = P(Y_t \leq y|Y_{t-1},\ldots,Y_{t-p},\boldsymbol{X})$ for any $y \in \mathbb{R}$.

A1(c). There exists a $(p+1)$-dimensional copula C such that for any $y_t, \ldots, y_{t-p} \in \mathbb{R}$,

$$H(y_t, \ldots, y_{t-p}|\boldsymbol{x}) = C\{G(y_t|\boldsymbol{x};\boldsymbol{\beta}), \ldots, G(y_{t-p}|\boldsymbol{x};\boldsymbol{\beta});\boldsymbol{\alpha}\},$$

where $C(\cdot;\boldsymbol{\alpha})$ is the copula function that does not vary with covariates, and $\boldsymbol{\alpha}$ is the unknown copula parameter vector.

Throughout we use $\boldsymbol{\beta}_0$ and $\boldsymbol{\alpha}_0$ to denote the true parameter values. For continuous time series, by Sklar's theorem (Sklar, 1959; Nelsen, 2007), we can represent $H(y_t, \ldots, y_{t-p}|\boldsymbol{x})$ by the marginal conditional distribution function $G(\cdot|\boldsymbol{x},\boldsymbol{\beta})$ of Y_t and a unique copula function $C(\cdot)$. However, when the marginal distribution function $G(\cdot)$ is discrete for count data, the joint distribution $H(\cdot)$ is uniquely defined only on the support of the margins. The discrepancy between the continuity of copula functions and discreteness of count data leads to the identifiability issue, that is, there may exist more than one copula satisfying condition A1(c). For now, we assume that $C(\cdot)$ is a prespecified copula function that satisfies A1(c). We will discuss how to overcome this identifiability issue in Section 2.2.2. For the ease of presentation, we focus on $p = 1$ hereafter, but the proposed method can be adapted for general p with some modification and additional computation.

2.2.2 Identifiability Issue

Sklar (1959) showed that the uniqueness of copulas is only guaranteed at the range of marginal distribution functions. For continuous data, the marginal CDF takes value over the entire range of $[0,1]$. However, for count data, the marginal CDF can only take countable values on [0,1], so the copula can only be identified on the support and is not unique outside the support of the marginal CDF.

Several papers have discussed the identifiability issue of copula for discrete data. Genest and Nešlehová (2007) pointed out that the identifiability issue could be more damaging when the marginal distribution concentrates on a small number of values. This issue will

tend to diminish when the discrete variable puts its mass on more and more atoms. Trivedi and Zimmer (2017) conducted some simulation and concluded that copulas for discrete count outcomes fail to capture the dependence at extremely small means. Yang et al. (2020) discussed the identifiability issue and proposed a nonparametric estimation of copula functions under the regression setup when the outcome is discrete.

This chapter focuses on copula-based analysis for count data in a regression setup. Under condition A1, the marginal distributions vary with covariates, so the inclusion of continuous covariates in the marginal can expand the region of support for copula identifiability. For example, in the Poisson regression model, $G(0|\boldsymbol{x}) = \exp\{-\exp(\boldsymbol{x}_j'\boldsymbol{\beta})\}$. Hence, the inclusion of continuous covariates will widen the range of $G(\cdot|\boldsymbol{x})$ from a discrete number of points to an interval, which together with A1(c) ensures that the copula function can be uniquely determined by the population at the region composed of possible values of $G(\cdot|\boldsymbol{x})$. We summarize the conclusion as follows.

Proposition 2.1: Suppose that conditions in Assumption A1 hold and that at least one of the covariates $\boldsymbol{x}$ is continuous with nonzero coefficient in the marginal distribution $G(\cdot|\boldsymbol{x},\boldsymbol{\beta})$ in A1(a), then the copula function can be uniquely determined over $[0,1]^p$.

2.2.3 Parameter Estimation

We present a maximum likelihood estimator for both the marginal parameter $\boldsymbol{\beta}$ and the copula parameter $\boldsymbol{\alpha}$. We first derive the likelihood function for the sequence $\{Y_t\}$ under the Markov property with order $p = 1$. From Assumption A1(c) and Song (2000), we can obtain the joint probability mass function (PMF) of (Y_{t-1}, Y_t) for any $t > 1$ as

$$
\begin{aligned}
&P(Y_t = y_t, Y_{t-1} = y_{t-1}|\boldsymbol{X}) \\
=&C(G(y_t|\boldsymbol{x};\boldsymbol{\beta}), G(y_{t-1}|\boldsymbol{x};\boldsymbol{\beta});\boldsymbol{\alpha}) - C(G(y_t|\boldsymbol{x};\boldsymbol{\beta}), G(y_{t-1}-1|\boldsymbol{x};\boldsymbol{\beta});\boldsymbol{\alpha}) \\
&- C(G(y_t-1|\boldsymbol{x};\boldsymbol{\beta}), G(y_{t-1}|\boldsymbol{x};\boldsymbol{\beta});\boldsymbol{\alpha}) + C(G(y_t-1|\boldsymbol{x};\boldsymbol{\beta}), G(y_{t-1}-1|\boldsymbol{x};\boldsymbol{\beta});\boldsymbol{\alpha}) \\
=&C(u_t, u_{t-1};\boldsymbol{\alpha}) - C(u_t, v_{t-1};\boldsymbol{\alpha}) - C(u_{t-1}, v_t;\boldsymbol{\alpha}) + C(v_t, v_{t-1};\boldsymbol{\alpha}), \qquad (2.2)
\end{aligned}
$$

where $u_t = G(y_t|\boldsymbol{x};\boldsymbol{\beta})$ for $y_t = 0,1,2,\ldots$, and $v_t = G(y_t - 1|\boldsymbol{x};\boldsymbol{\beta})$ for $y_t = 1,2,\ldots$ and $v_t = 0$ for $y_t = 0$. That is, the joint PMF can be expressed as the finite difference of the copula function, which includes 4 terms of copula distribution functions. For general p-th order Markov processes, the PMF involves 2^{p+1} terms of copula functions. Then the conditional PMF of Y_t given Y_{t-1} is given by

$$\begin{aligned} & P(Y_t = y_t|Y_{t-1} = y_{t-1}, \boldsymbol{X}) \\ = & \frac{P(Y_t = y_t, Y_{t-1} = y_{t-1}|\boldsymbol{X})}{P(Y_{t-1} = y_{t-1}|\boldsymbol{X})} \\ = & \frac{C(u_t, u_{t-1};\boldsymbol{\alpha}) - C(u_t, v_{t-1};\boldsymbol{\alpha}) - C(u_{t-1}, v_t;\boldsymbol{\alpha}) + C(v_t, v_{t-1};\boldsymbol{\alpha})}{u_{t-1} - v_{t-1}}. \end{aligned} \tag{2.3}$$

Thus, the likelihood function of $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ can be derived as:

$$\begin{aligned} L(\boldsymbol{\beta},\boldsymbol{\alpha}) &= P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n|\boldsymbol{X};\boldsymbol{\beta},\boldsymbol{\alpha}) \\ &= P(Y_1 = y_1|\boldsymbol{X})P(Y_2 = y_2|Y_1 = y_1, \boldsymbol{X})\ldots P(Y_n = y_n|Y_{n-1} = y_{n-1}, \boldsymbol{X}) \\ &= \frac{\prod_{t=2}^{n}\{C(u_t, u_{t-1};\boldsymbol{\alpha}) - C(u_t, v_{t-1};\boldsymbol{\alpha}) - C(u_{t-1}, v_t;\boldsymbol{\alpha}) + C(v_t, v_{t-1};\boldsymbol{\alpha})\}}{\prod_{t=2}^{n-1}(u_t - v_t)}, \end{aligned}$$

and the log-likelihood function is given by

$$\begin{aligned} l(\boldsymbol{\beta},\boldsymbol{\alpha}) &= \log\{L(\boldsymbol{\beta},\boldsymbol{\alpha})\} \\ &= \sum_{t=2}^{n}\log\{C(u_t, u_{t-1};\boldsymbol{\alpha}) - C(u_t, v_{t-1};\boldsymbol{\alpha})C(u_{t-1}, v_t;\boldsymbol{\alpha}) + C(v_t, v_{t-1};\boldsymbol{\alpha})\} \\ &\quad - \sum_{t=2}^{n-1}\log(u_t - v_t), \end{aligned} \tag{2.4}$$

where $\boldsymbol{\beta}$ enters this expression through u and v. The maximum likelihood estimator of ($\boldsymbol{\beta}$, $\boldsymbol{\alpha}$) is obtained by

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\alpha}}) = \underset{\boldsymbol{\beta},\boldsymbol{\alpha}}{\operatorname{argmax}}\, l(\boldsymbol{\beta},\boldsymbol{\alpha}).$$

For many copulas such as Gaussian and t copula, the copula function $C(\cdot)$ does not

have a closed form, so numerical approximation is needed. If we use the finite difference likelihood (FDL) expression in (2.4), the estimator may be imprecise due to the accumulated approximation error involved in the four numerically computed orthant probabilities. The numerical error may even lead to negative values for the PMF in (2.2). The damage can be more severe when $p > 1$, for which the joint PMF involes 2^{p+1} numerically computed orthant probabilities. Moreover, this FDL may also slow down the computation for larger p. To overcome this issue, we adapt the simulated likelihood (SL) proposed in Nikoloulopoulos (2013) for Gaussian copula, which can be used to calculate the joint PMF with only one-time approximation:

$$\begin{aligned} l(\boldsymbol{\beta},\boldsymbol{\alpha}) &= \log\{P(Y_1 = y_1,\ldots,Y_n = y_n|\boldsymbol{X})\} \\ &= \log\{P(y_1-1 < Y_1 \le y_1,\ldots,y_n-1 < Y_n \le y_n|\boldsymbol{X})\} \\ &= \sum_{t=2}^{n} \log\left(\int_{\Phi^{-1}(v_{t-1})}^{\Phi^{-1}(u_{t-1})}\int_{\Phi^{-1}(v_t)}^{\Phi^{-1}(u_t)} \phi_R(z_1,z_2)\,dz_1 dz_2\right) - \sum_{t=2}^{n-1}\log(u_t - v_t), \end{aligned} \tag{2.5}$$

where Φ denotes the standard normal CDF and ϕ_R denotes the multivariate normal density with correlation matrix R. The same technique can also be applied to other elliptical copulas such as t copula. For such copulas, since the method involves one-time numerical approximation, it can improve the accuracy and reduce the computing time.

Throughout our implementation, we consider the simulated likelihood approach only for copula families without closed expressions, such as elliptical copulas, including Gaussian copula and t copula. For other copula families with closed expressions for the copula distribution functions, the finite difference likelihood can be calculated directly using (2.4).

2.2.4 Point and Interval Prediction

For time series data analysis, one common question of interest is prediction at future time points. We now discuss the one-step-ahead prediction, that is, prediction of Y_{n+1}, based

on the proposed model and estimator. The conditional distribution of Y_{n+1} given Y_n and $\boldsymbol{X}$ can be well characterized by the conditional quantiles. Through conditional quantiles, we can not only construct point prediction of Y_{n+1} by using, e.g. the conditional median, but also prediction intervals. To accommodate discrete distributions, we adopt the generalized definition of quantile and define the qth quantile of a random variable Y as

$$Q_q(Y) = \inf\{y : P(Y \leq y) \geq q\},$$

for any given quantile level $0 < q < 1$.

For $\{Y_t, t = 1, \cdots, n\}$ satisfying Assumption A1, the conditional quantile function of Y_{n+1} given $(Y_n, \boldsymbol{X})$ can be easily estimated. Define $G^{-1}(v|\boldsymbol{X};\boldsymbol{\beta}) = \inf\{y : G(y|\boldsymbol{X};\boldsymbol{\beta}) \geq v\}$. Note that $G^{-1}(v|\boldsymbol{X};\boldsymbol{\beta})$ is a monotonic transformation of v. Hence the q-th conditional quantile of Y_{n+1} is given by

$$\begin{aligned} Q_q\left(Y_{n+1}|Y_1,\ldots,Y_n,\boldsymbol{X}\right) &= Q_q\left(Y_{n+1}|Y_n,\boldsymbol{X}\right) \\ &= G^{-1}\left[Q_q\left\{U_{n+1}|U_n = G\left(Y_n|\boldsymbol{X};\boldsymbol{\beta}\right);\boldsymbol{\alpha}\right\}\Big|\boldsymbol{X};\boldsymbol{\beta}\right], \quad (2.6) \end{aligned}$$

where $Q_q\left(U_{n+1}|U_n;\boldsymbol{\alpha}\right)$ is the conditional quantile function of U_{n+1} given U_n, that is,

$$Q_q\left(U_{n+1}|U_n = u;\boldsymbol{\alpha}\right) = C_{2|1}^{-1}\left(q|u;\boldsymbol{\alpha}\right),$$

in which $C_{2|1}\left(\cdot|u;\boldsymbol{\alpha}\right) = \partial C\left(u,\cdot;\boldsymbol{\alpha}\right)/\partial u$ is the conditional distribution of U_{n+1} given $U_n = u$. It follows from (2.4) and (2.5) that the plug-in estimator of the conditional quantile of U_{n+1} given $U_n = u$ is

$$\widehat{Q}_q^U\left(u\right) = Q_q^U\left(u;\hat{\boldsymbol{\alpha}}\right) = C_{2|1}^{-1}\left(q|u;\hat{\boldsymbol{\alpha}}\right),$$

and the plug-in estimator of the qth conditional quantile Y_{n+1} given $(Y_n = y, \boldsymbol{X})$ is

$$\begin{aligned}\widehat{Q}_q^Y(Y|y,\boldsymbol{X}) &= G^{-1}\left[\widehat{Q}_q^U\left\{G\left(y\middle|\boldsymbol{X};\hat{\boldsymbol{\beta}}\right);\hat{\boldsymbol{\alpha}}\right\}\middle|\boldsymbol{X};\hat{\boldsymbol{\beta}}\right] \\ &= G^{-1}\left[C_{2|1}^{-1}\left\{q\middle|G\left(y\middle|\boldsymbol{X};\hat{\boldsymbol{\beta}}\right);\hat{\boldsymbol{\alpha}}\right\}\middle|\boldsymbol{X};\hat{\boldsymbol{\beta}}\right].\end{aligned}$$

After the conditional quantiles are estimated, we can contruct the $100(1-\nu)\%$ for the one-step ahead observation Y_{n+1}, which satisfies

$$P(L \leq Y_{n+1} \leq U|\boldsymbol{X};\hat{\boldsymbol{\beta}}) \geq 1-\nu. \tag{2.7}$$

For continuous distributions, a standard method to construct prediction intervals is to use appropriate pivotal-quantities. This is not possible for discrete distributions since such pivotal-quantities are not available. If we simply take $L = \widehat{Q}_{\nu/2}^Y(Y|y,\boldsymbol{X})$ and $U = \widehat{Q}_{1-\nu/2}^Y(Y|y,\boldsymbol{X})$, we have $P(Y \leq L;\boldsymbol{\theta}|\boldsymbol{X}) \geq \nu/2$ and $P(Y \geq U;\boldsymbol{\theta}|\boldsymbol{X}) \geq 1-\nu/2$, and we cannot ensure (2.7) hold. Bain and Patel (1993) proposed prediction intervals for some discrete distributions, while Kim et al. (2021) proposed prediction intervals for Poisson-based regression models. To ensure the coverage probability of the prediction interval is larger than $100(1-\nu)\%$, we define the lower and upper bound of the prediction interval as follows:

$$L = \sup\{y : G(y-1|\boldsymbol{X};\hat{\boldsymbol{\beta}}) \leq \nu/2\},\ U = \inf\{y : G(y|\boldsymbol{X};\hat{\boldsymbol{\beta}}) \geq 1-\nu/2\}. \tag{2.8}$$

By doing so, we can ensure that (2.7) holds and the prediction interval has over $100(1-\nu)\%$ coverage.

For count data with support $0,1,2,\ldots$, one-sided prediction interval can also be constructed with lower and upper bounds as

$$L = 0 \text{ and } U = \inf\{y : G(y|\boldsymbol{X};\hat{\boldsymbol{\beta}}) \geq 1-\nu\}. \tag{2.9}$$

2.2.5 Asymptotic Properties

Denote $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ as the parameter vector with dimension k, $\boldsymbol{\theta}_0 = (\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0)$ as the true value, and $\boldsymbol{\Theta}$ as the parameter space. Let $\mathscr{F}_t$ denote the σ-field generated by the first t observations $y_1, \ldots, y_t$ for $t \geq 1$, and $\mathscr{F}_0 = \{\varnothing, \Omega\}$. In addition, let $l_n(\boldsymbol{\theta})$, $s_n(\boldsymbol{\theta})$ and $-H_n(\boldsymbol{\theta})$ denote the log-likelihood and its first and second derivatives, respectively. Define $d_n(\boldsymbol{\theta}) = s_n(\boldsymbol{\theta}) - s_{n-1}(\boldsymbol{\theta})$. The following set of conditions are needed to establish to ensure the consistency and asymptotic normality of $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$.

A2. The parameter space $\boldsymbol{\Theta}$ is an open subset of $\mathbb{R}^m$, $m \in \mathbb{N}$.

A3. The probability measure P belongs to a parametric family $\{P_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$. For fixed n, the projections $\{P_{n,\boldsymbol{\theta}}, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ are mutually absolutely continuous.

A4. $l_n(\boldsymbol{\theta})$ is continuous and two times continuously differentiable.

A5. The score function $\{s_n(\boldsymbol{\theta}_0)\}$ is a square integrable zero mean martingale with respect to $\mathscr{F}_n$.

A6. The conditional information $\Gamma_n(\boldsymbol{\theta}_0) = \sum_{t=1}^{n} \mathrm{Cov}_{\theta}(d_t(\boldsymbol{\theta}_0))$ converages to a positive definite matrix, i.e.,

$$\frac{1}{n}\Gamma_n(\boldsymbol{\theta}_0) \xrightarrow{P} V(\boldsymbol{\theta}_0).$$

A7. For any $\varepsilon > 0$,

$$\frac{1}{n}\sum_{t=1}^{n} E\{d_t^T(\boldsymbol{\theta}_0) d_t(\boldsymbol{\theta}_0) I_{nt}(\varepsilon) | \mathscr{F}_{t-1}\} \xrightarrow{P} 0,$$

where $I_{nt}(\varepsilon) = I\{n^{-1} d_t^T(\boldsymbol{\theta}_0) d_t(\boldsymbol{\theta}_0) \geq \varepsilon^2\}$ with $I(\cdot)$ denoting the indicator function.

A8. The continuity condition

$$\sup_{\boldsymbol{\theta} \in N_n(\delta)} \frac{1}{n} \| (H_n(\boldsymbol{\theta}) - \Gamma_n(\boldsymbol{\theta}_0)) \| \xrightarrow{P} 0, \tag{2.10}$$

holds for any $\delta > 0$, where $N_n(\delta) = \{\boldsymbol{\theta} : \|\sqrt{n}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\| \leq \delta\}$.

Theorem 2.1: Suppose that Assumptions A1 - A8 hold, then we have (i) $\hat{\boldsymbol{\theta}} \xrightarrow{P} \boldsymbol{\theta}_0$; (ii) $\Gamma_n^{1/2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \xrightarrow{D} N(0, I)$.

Theorem 2.1 also holds for $p \geq 1$ but the proof will involve more complicated derivations.

2.2.6 Copula Selection

In Section 2.2.2, we discussed the identifiability issue and presented one sufficient condition to ensure the identifiability of copula under the proposed model. Resolving the identifiability issue allows us to select a parametric form for the copula from existing copula families.

Focusing on multivariate discrete responses, Yang et al. (2020) suggested a method to select copula by minimizing a L_2-norm distance between the fitted parametric copulas with their proposed nonparametric copula estimator. We considered this approach in our setup, but our numerical studies show that this method is time consuming with unstable performance. The main challenge is that the criterion is based on the integration of squared distance, which requires repeated kernel-type estimation at multiple points. The computation and performance are further complicated by the selection of bandwidth involved in the smoothing. To overcome these challenges, we propose a cross-validation procedure to select copula for count time series data. Let k be a pre-set minimum number of observations in the training data. We repeat the validation sequentially for a total of $n - k$ times. To assess the performance of the predictive models for count data based on a candidate copula, we focus on probabilistic forecasts, that is, evaluating the predictive distribution generated by the copula and parameters estimated by the training data. The reason we focus on probabilistic forecasts instead of using the commonly used mean squared error (MSE) for point prediction is that count data is often skewed, so that the MSE will be dominated by predictions at the tails. The detailed procedure is as follows.

For the ith cross validation, choose observations at time $1, 2, \ldots, k + i - 1$ as the training data and let the observation at time $k + i$ be the test data. Fit the proposed model with the

candidate copula by using the training data and let $\hat{\boldsymbol{\theta}}^*$ denote the estimated parameters. The predictive distribution for Y_t at $t = k+i$ can be obtained by $P(Y_t = y|Y_{t-1}, \boldsymbol{X}; \hat{\boldsymbol{\theta}}^*) \doteq \hat{P}^*$; see the expression in (2.3). To assess the performance of the resulting predictive distribution, we adapt the logarithmic score (LOGS), proposed in (Czado et al., 2009), at time $t = k+i$, defined as

$$\text{LOGS}(\hat{P}^*, y_t) = -\log\{P(Y_t = y_t|Y_{t-1} = y_{t-1}, \mathbf{X}); \hat{\boldsymbol{\theta}}^*\},$$

where y_t is the observed response at time t. The LOGS is a measure based on the predictive distribution and the observation at the testing time to evaluate probabilistic forecasts for count data. A smaller value of LOGS indicates that the predictive distribution fits the data better.

We repeat the above step for $i = 1, 2, \ldots, n-k$, and compute the average of LOGS across $n-k$ validations. The copula is then chosen to be the one that gives the smallest average LOGS.

Remark: For small sample sizes, this cross-validation approach may not perform well, and we can select the copula that gives the largest likelihood.

2.3 Simulation Study

We conduct a simulation to compare the finite sample performance of the proposed method with existing approaches for analyzing count time series data. We generate data $\{Y_t, t = 1, \ldots, n\}$ from the Markov process with bivariate copula $C(u, v; \alpha)$ and Poisson regression model $Y_t|\boldsymbol{X} \sim Poisson(\lambda_t)$, where $\log(\lambda_t) = \beta_1 + \beta_2 X_t$. Below are the steps used for data generation.

Step 1: Generate an $i.i.d.$ sequence from Unif$(0,1)$, denoted as $\{v_t\}_{t=1}^n$, and $\{X_t, t = 1, \ldots, n\}$ from $N(0,1)$.

Step 2: Set $U_1 = v_1$ and $U_t = C_{2|1}^{-1}(v_t|U = U_{t-1}; \alpha)$ for $t = 2, \ldots, n$.

Step 3: For $t = 1, \ldots, n$, set $Y_t = G^{-1}(U_t; \lambda_t)$, where $G(y; \lambda_t) = P(Y_t \leq y | \boldsymbol{X})$, $\log(\lambda_t) = \beta_1 + \beta_2 X_t$ and $G^{-1}(u; \lambda_t) = \inf\{y : G(y; \lambda_t) \geq u\}$.

To reach the stationarity of U_t, we generate sequences with a length of burn-in periods of 2,000. We consider two sample sizes, $n = 200$ and 500, and two different levels of temporal dependence with Kendall's $\tau = 0.3$ and 0.7 for Joe, Gaussian and t copulas, respectively. We consider two cases for data generating process: Case 1 with $\beta_1 = 2$, $\beta_2 = 1$ and Kendall's $\tau = 0.3$, and Case 2 with $\beta_1 = 0$, $\beta_2 = 1$ and Kendall's $\tau = 0.7$. For each scenario, the simulation is repeated 500 times. Hereafter, we refer to the proposed method as COMC since it is built on the copula-based Markov model for count data.

2.3.1 Parameter Estimation

We first assess the performance of the proposed estimator for the copula parameter α and the marginal parameters (β_1, β_2) assuming the correct specification of copula function. For data generated with Gaussian copula, the model assumption in Jia et al. (2021) also holds and parameters in two models coincide. Therefore, besides the proposed COMC estimator, we also include the Gaussian copula based particle filtering estimator (PF) in Jia et al. (2021) as a comparison. In the proposed method, we use the simulated likelihood (SL) method to calculate the likelihood for data generated with Gaussian copula.

Table 2.1 summarizes the averaged bias and root mean squared error (RMSE) of different estimators when $\alpha = 0.454$ in Case 1 and 0.891 in Case 2, corresponding to two different levels of temporal dependence with Kendall's $\tau = 0.3$ and 0.7. The boxplots of the estimators across 500 simulations for Case 1 are shown in Figure 2.1. The white plots correspond to $n = 200$ while grey boxplots correspond to $n = 500$. In the left plot, the true parameter value is $\beta_1 = 0$. In the middle plot, the true parameter value is $\beta_2 = 1$. In the right plot, the true parameter value is $\alpha = 0.454$. The boxplots for Case 2 are shown in Figure 2.2. In the left plot, the true parameter value is $\beta_1 = 2$. In the middle plot, the true parameter value is $\beta_2 = 1$. In the right plot, the true parameter value is $\alpha = 0.891$. Sample sizes of 200 and

Table 2.1: 1000×bias and 1000×RMSE of different estimators of β_1, β_2 and α.

			$\hat{\beta}_1$		$\hat{\beta}_2$		$\hat{\alpha}$	
Case	n	Method	Bias	RMSE	Bias	RMSE	Bias	RMSE
1	200	COMC	-1.2 (2.0)	43.7 (1.5)	0.6 (1.0)	21.6 (0.9)	-1.5 (2.4)	54.7 (1.6)
		PF	-1.4 (2.0)	44.3 (1.5)	0.5 (1.0)	21.3 (0.9)	-0.0 (2.5)	55.3 (1.6)
	500	COMC	-0.1 (1.2)	27.7 (0.8)	0.7 (0.6)	13.3 (0.4)	-2.3 (1.6)	35.8 (1.1)
		PF	-0.4 (1.3)	28.9 (0.9)	0.9 (0.6)	13.7 (0.5)	0.9 (1.7)	37.2 (1.1)
2	200	COMC	7.4 (10.2)	227.0 (7.4)	1.4 (3.3)	73.5 (2.4)	2.6 (1.0)	23.0 (0.7)
		PF	-39.5 (12.7)	286.6 (10.3)	12.2 (4.3)	96.2 (3.6)	-19.0 (1.2)	32.4 (1.1)
	500	COMC	0.9 (6.7)	150.2 (4.7)	-0.2 (2.2)	48.5 (1.5)	-0.4 (0.6)	14.0 (0.5)
		PF	-18.5 (9.1)	205.1 (6.9)	6.4 (3.0)	68.1 (2.3)	-34.5 (1.0)	40.7 (1.0)

COMC: copula-based Morkov model for count data; PF: particle filtering. Values in the parentheses are standard errors.

500 are given in the legend and estimation methods are indicated on the horizontal axis. Note that in Case 1, PF performs comparably to COMC due to weak temporal dependence. However, in Case 2, the bias and RMSE of PF are consistently larger than those of COMC, which shows that our proposed method can provide more accurate estimations.

2.3.2 Conditional Quantile Estimation with Selected Copula

We assess the performance of the proposed estimator of $Q_q^Y(Y_{n+1}|Y_n, X)$, the q-th conditional quantile of Y_{n+1} at time $n+1$ given the observed data, when the true copula is unknown and chosen from a class of candidate families, including Gaussian, t, Joe, Frank and Clayton copulas. Note that here we only consider these 5 candidate families, and in practice any copula family can be considered. We consider Cases 1-2 and two quantile levels $q = 0.5$ and 0.9. For comparison, we include two estimators from COMC: COMC_{TC}, COMC_{SC}, and two other methods: PF and PA. The method COMC_{TC} provides estimation with the true copula, while COMC_{SC} gives estimation with the selected copula. The method PF is the Gaussian copula based particle filtering approach proposed in Jia et al. (2021). The PA method is the Poisson autoregression method proposed in Fokianos and Tjøstheim (2011).

Table 2.2 summarizes the copula selection results based on the cross-validation method. Results show that the cross-validation method works well for selecting the correct copula,

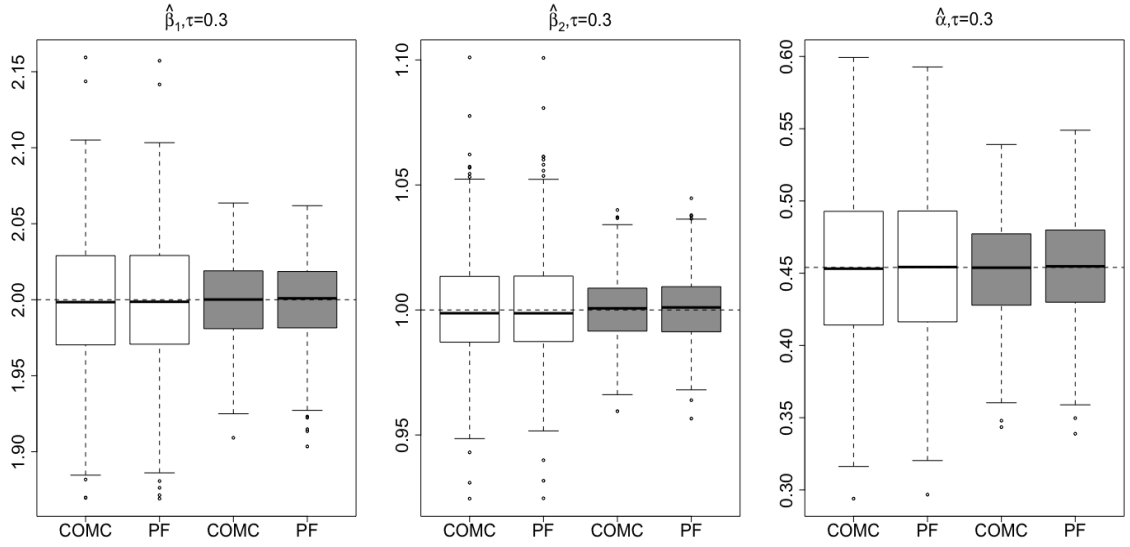

Figure 2.1: Boxplots of $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\alpha}$ for $n = 200$ (white boxplots) and 500 (grey boxplots) across 500 simulations with α corresponding to $\tau = 0.3$. True values are shown as a black horizontal dashed line. COMC: copula-based Morkov model for count data; PF: particle filtering.

and the selection accuracy increases with the sample size. Table 2.3 summarizes the MSE of $\widehat{Q}_q(Y_{n+1}|Y_n, X)$ from four methods. As expected, COMC_{TC} is the most efficient estimator as it is based on the correct copula function. In all cases considered, the COMC_{SC} estimator is competitive to COMC_{TC}, and it has smaller MSE than PF and PA, even when the copula was occasionally misspecified in COMC_{SC}.

2.3.3 Prediction at Future Time Points

We consider the following metrics to assess the prediction performance of different methods at time $t = n+1$: (i) the coverage percentages (CPs) and average lengths (ALs) of the 90% prediction intervals for predicting Y_{n+1}; (ii) logarithmic score (LOGS) of the predictive conditional distribution P and the realized count Y_{n+1}. For comparison, we include three methods from COMC: COMC_0, COMC_{TC}, COMC_{SC} and two other methods: PF and PA. The method COMC_0 is based on the true copula and true parameters. The method COMC_{TC} provides estimation with the true copula, while COMC_{SC} gives estimation with the selected

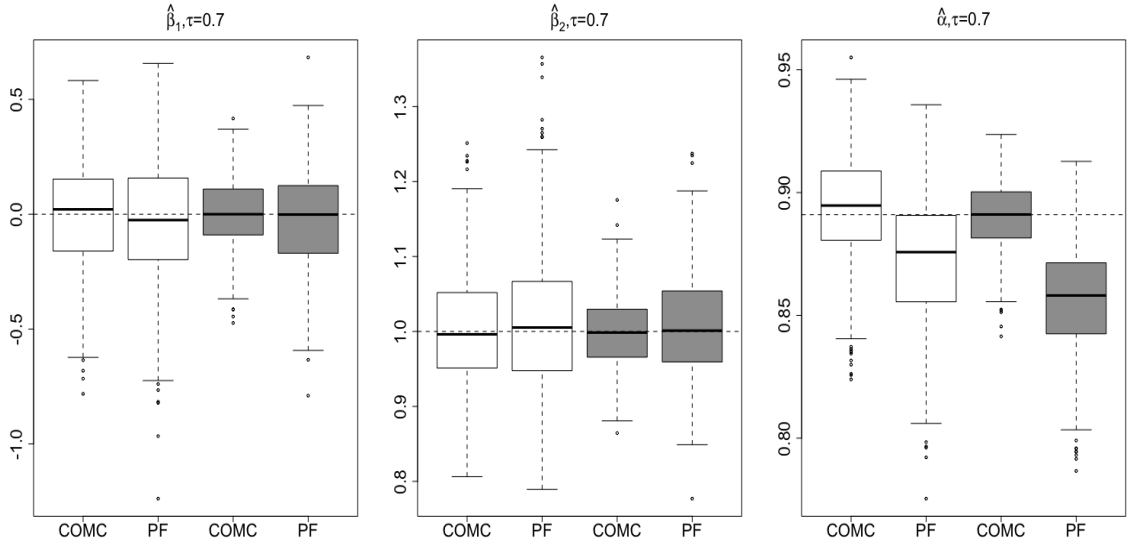

Figure 2.2: Boxplots of $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\alpha}$ for $n = 200$ (white boxplots) and 500 (grey boxplots) across 500 simulations with α corresponding to $\tau = 0.7$. True values are shown as a black horizontal dashed line. COMC: copula-based Morkov model for count data; PF: particle filtering.

copula. The method PF is the Gaussian copula based particle filtering approach proposed in Jia et al. (2021). The PA method is the Poisson autoregression method proposed in Fokianos and Tjøstheim (2011). Due to the discreteness of the data, it may be infeasible to get a prediction interval with exact 90% coverage. Therefore, we take the coverage percentage of the method COMC_0 as the baseline for comparing the other four methods.

Table 2.4 summarizes the coverage percentages (CPs) and the average lengths (ALs) of 90% prediction intervals from the five methods. For the CP, as expected, COMC_{TC} performs quite similar to COMC_0, since it is based on the correct copula. COMC_{SC} performs comparably to COMC_{TC} and COMC_0. However, for Joe copula, PF tends to have larger CP than COMC_0, while PA has the largest CP in all cases. For the AL, COMC_{TC} and COMC_{SC} always perform competitively to COMC_0, which produces the shortest prediction intervals. PF performs well when the ture copula is Gaussian since it is based on Gaussian copula, but produces wider intervals in other situations. PA always produces the widest prediction intervals. Table 2.5 summarizes the results of one-sided 90% prediction intervals, and the

Table 2.2: Percentages of copulas chosen by cross-validation across 500 simulations.

			Selected copula				
Case	True	n	Clayton	Gumbel	Joe	Gaussian	t
1	Joe	200	0.2	21	74.6	1.6	2.6
		500	0	15.6	83.8	0.2	0.4
	Gaussian	200	5.6	17	3.6	52	21.8
		500	0.2	10.2	0	67.2	22.4
	t_2	200	4.4	29	6.4	6	54.2
		500	0.8	17.6	1.2	1	79.4
2	Joe	200	0.2	19	77.6	1	2.2
		500	0	9.6	90.4	0	0
	Gaussian	200	12.2	15.8	3.6	48.2	20.2
		500	3	8	0	72.8	16.2
	t_2	200	3.6	8.6	4	1	82.8
		500	0.8	0.2	0	0	99

results are quite similar to those of two-sided intervals in Table 2.4. Table 2.6 summarizes the logarithmic score (LOGS) from the five methods. COMC_{TC} and COMC_{SC} always produce the similar LOGS as COMC_0, while the LOGS of PF is larger except for Gaussian copula. PA always produces the largest LOGS. These metrics give some evidence that our proposed COMC method can give good prediction in practice.

2.4 Real Data Analysis

In this section we apply our proposed COMC model to two datasets: the baseball data and the COVID-19 data.

2.4.1 Baseball Data

We apply the proposed method to analyze the number of major league baseball games where no hitters were pitched on an annual basis from $1893-2017$ ($n=125$).

In this dataset two covariates are available: the total number of games played in each year X_1 and the height of the pitching mound X_2. The total number of games played in a season has increased over the years as more teams have gradually joined the league. Of

Table 2.3: 100×MSE of $\widehat{Q}_q(Y_{n+1}|Y_n)$ from different methods at $q = 0.5$ and $q = 0.9$.

Model	Case	n	Quantile	COMC_{TC}	COMC_{SC}	PF	PA
Joe	1	200	0.5	6.0 (1.1)	6.6 (1.3)	15.6 (1.9)	252.8 (29.7)
			0.9	5.8 (1.0)	7.2 (1.2)	28.7 (2.5)	200.8 (37.5)
	1	500	0.5	2.6 (0.7)	2.6 (0.7)	20.0 (3.1)	291.0 (34.2)
			0.9	4.4 (1.2)	4.4 (1.2)	38.6 (4.9)	231.8 (36.8)
	2	200	0.5	27.6 (3.4)	40.4 (8.1)	181.2 (28.7)	749.4 (108.8)
			0.9	26.8 (4.2)	28.2 (4.3)	108.8 (11.0)	486.6 (51.4)
	2	500	0.5	19.0 (2.6)	20.4 (2.8)	159.6 (28.5)	605.0 (89.2)
			0.9	16.8 (2.5)	19.4 (2.6)	106.2 (14.8)	444.4 (49.8)
t_2	1	200	0.5	2.4 (0.7)	4.0 (0.9)	17.8 (2.6)	222.0 (18.3)
			0.9	6.4 (1.1)	10.4 (1.5)	27.2 (3.0)	157.8 (20.8)
	1	500	0.5	1.2 (0.5)	2.0 (0.6)	18.0 (2.1)	241.6 (23.0)
			0.9	4.4 (0.9)	6.8 (1.1)	29.6 (2.3)	186.0 (23.7)
	2	200	0.5	20.4 (7.5)	50.2 (18.1)	146.2 (73.1)	752.2 (324.4)
			0.9	29.4 (16.3)	54.4 (20.5)	612.2 (443.3)	392.0 (118.1)
	2	500	0.5	8.6 (1.3)	9.0 (1.5)	73.4 (18.9)	376.0 (65.3)
			0.9	11.2 (2.2)	11.8 (2.2)	118.2 (11.7)	298.2 (42.1)
Gaussian	1	200	0.5	8.0 (2.1)	7.2 (1.3)	11.0 (2.2)	246.8 (31.3)
			0.9	15.8 (3.6)	16.4 (2.0)	15.0 (2.5)	233.8 (34.5)
	1	500	0.5	5.2 (1.0)	4.8 (1.0)	7.6 (1.2)	183.2 (14.6)
			0.9	7.4 (1.2)	8.0 (1.2)	11.2 (1.4)	170.0 (18.2)
	2	200	0.5	69.8 (25.8)	70.4 (25.9)	65.8 (21.8)	400.0 (74.1)
			0.9	81.0 (26.0)	85.0 (26.1)	73.8 (21.9)	453.8 (90.2)
	2	500	0.5	70.8 (51.2)	74.2 (51.3)	79.6 (57.8)	350.0 (107.2)
			0.9	71.4 (51.2)	75.2 (51.3)	75.2 (51.2)	423.8 (146.9)

COMC_{TC}: COMC based on true copula; COMC_{SC}: COMC based on selected copula; PF: particle filtering method from Jia et al. (2021); PA: Poisson autoregression method from Fokianos and Tjøstheim (2011)). Values in the parentheses are standard errors.

course one expects more no hitters when more games are played. The height of the pitching mound was changed several times and could also be a significant factor. Higher pitching mounds are purported to give pitchers an advantage, so hitters tend to adjust quickly to such changes.

The data are over-dispersed, with a sample mean of 2.12 and a sample variance of 3.40. So we choose the marginal model to be Negative Binomial with parameter $\eta \in (0,1)$ fixed in time and r_t varing with time with the log link:

$$r_t = \exp(\beta_1 + \beta_2 X_1 + \beta_3 X_2).$$

Table 2.4: Coverage percentages and 10×average lengths of 90% prediction intervals for predicting Y_{n+1}.

Model	Case	n	Coverage Percentages					10×Average Lengths				
			COMC_0	COMC_{TC}	COMC_{SC}	PF	PA	COMC_0	COMC_{TC}	COMC_{SC}	PF	PA
Joe	1	200	97.4	97.2	97.0	98.2	98.4	18.4	18.3	18.7	24.1	34.2
		500	96.2	96.0	96.2	97.6	99.0	19.5	19.8	20.1	26.8	37.6
	2	200	96.4	96.4	96.2	96.6	93.8	93.1	93.0	93.4	97.7	107.4
		500	95.0	95.2	95.4	96.0	93.8	92.0	91.8	91.9	96.4	106.6
t_2	1	200	98.2	97.8	97.4	97.8	98.8	23.4	23.7	23.3	26.1	34.2
		500	97.0	96.2	96.6	97.4	98.8	23.2	23.3	23.1	27.3	34.6
	2	200	94.4	94.4	94.0	94.4	96.6	95.5	94.8	94.4	96.7	106.2
		500	96.4	96.6	96.6	96.2	96.2	93.0	93.2	93.2	98.2	106.7
Gaussian	1	200	98.0	98.2	97.8	98.4	99.0	24.8	24.7	24.6	26.1	36.1
		500	99.0	99.0	99.0	99.2	98.0	24.1	24.2	23.9	25.9	34.8
	2	200	95.8	96.2	96.8	96.2	95.8	99.3	99.4	99.6	99.1	108.6
		500	95.0	94.4	94.2	94.0	93.0	100.7	100.2	100.3	99.9	109.9

COMC_0: COMC based on the true copula and true parameters; COMC_{TC}: COMC based on the true copula; COMC_{SC}: COMC based on the selected copula; PF: particle filtering method from Jia et al. (2021); PA: Poisson autoregression method from Fokianos and Tjøstheim (2011). Values in the parentheses are standard errors.

In order to determine the order of Markov process and to specify the correct copula, we conduct a cross-validation using the procedure in Section 2.2.6. We choose the copula function among five copula families: Gaussian, t, Joe, Gumbel and Clayton. We take $k = 100$ for the cross-validation and choose t_7 copula and second order Markov process, as they produce the smallest LOGS among the last 25 observations. The parameters were estimated using the full data, and were shown in Table 2.7. Results show that most of the parameters are significantly different from zero.

For comparison, we also include the Negative Binomial autoregression (NBAR) method, which assumes the conditional distribution of Y_t is Negative Binomial (NB):

$$Y_t|(Y_{t-1},Y_{t-2},X_1,X_2) \sim NB(\exp(a+b_1\log(Y_{t-1}+1)+b_2\log(Y_{t-2}+1)+c_1X_{1,t}+c_2X_{2,t}),\eta)$$

and the particle filtering (PF) method proposed by Jia et al. (2021), which is a Gaussian copula based method. We assume the latent process follows an AR(2) model.

Table 2.8 summarizes the prediction performance of three methods. We consider the

Table 2.5: Coverage percentages and 10×average lengths of [0, $\widehat{Q}_{0.9}(Y_{n+1}|Y_n, \boldsymbol{X})$] for predicting Y_{n+1}.

Model	Case	n	Coverage Percentages					10×Average Lengths				
			COMC_0	COMC_{TC}	COMC_{SC}	PF	PA	COMC_0	COMC_{TC}	COMC_{SC}	PF	PA
Joe	1	200	99.4	99.2	99.2	99.4	96.8	26.6	26.6	26.8	29.1	30.9
		500	99.2	99.2	99.2	99.6	96.6	30.0	30.1	30.1	33.2	34.6
	2	200	95.0	95.4	95.2	94.0	90.2	153.9	153.9	153.9	157.0	156.7
		500	93.6	94.2	94.2	94.0	92.0	148.4	148.0	148.2	151.0	152.5
t_2	1	200	99.0	99.0	99.0	99.4	97.2	28.8	29.0	29.3	30.8	31.3
		500	97.6	97.6	97.6	98.0	95.0	27.4	27.6	27.7	30.3	30.5
	2	200	92.6	92.8	92.4	93.8	91.0	154.7	154.4	154.1	155.4	157.0
		500	94.2	94.2	94.2	95.4	94.2	152.5	152.5	152.5	155.7	157.2
Gaussian	1	200	99.6	99.4	99.6	99.2	95.2	31.6	31.3	31.2	31.7	33.1
		500	98.8	98.8	98.6	99.2	95.0	29.0	29.1	29.0	30.0	30.9
	2	200	94.2	93.4	93.2	93.4	92.6	164.9	165.1	164.9	165.0	166.0
		500	93.4	93.4	93.4	93.2	90.4	168.0	168.3	168.2	168.1	169.5

COMC_0: COMC based on the true copula and true parameters; COMC_{TC}: COMC based on the true copula; COMC_{SC}: COMC based on the selected copula; PF: particle filtering method from Jia et al. (2021); PA: Poisson autoregression method from Fokianos and Tjøstheim (2011). Values in the parentheses are standard errors.

coverage percentages (CPs) and the average lengths (ALs) of the prediction intervals for Y_t. For the point prediction, We construct the point predictions of Y_t using the conditional median $\hat{Y}_t$, and calculate the mean squared relative prediction error (MSRPE) by

$$MSRPE = \frac{1}{25}\sum_{t=101}^{125}\frac{(\hat{Y}_t - Y_t)^2}{Y_t^2}.$$

As we can see from Table 2.8, the LOGS of PF and NBAR are larger than that of COMC. For the interval prediction, COMC produces higher CP than PF and smaller AL than NBAR. Moreover, COMC produces the smallest MSRPE.

2.4.2 COVID-19 Data

In this section, we analyze the count time series of daily death from COVID-19 in Virginia. The dataset contains the daily deaths in Virginia due to COVID-19 from Jun 22, 2020 to Dec 31, 2021, which is maintained and archived by Centers for Disease Control and Prevention and is updated daily. This dataset contains 400 daily observations in Virginia, in which

Table 2.6: 100× LOGS of prediction for Y_{n+1} given by COMC_0, COMC_{TC}, COMC_{SC}, PF and PA.

Model	Case	n	COMC_0	COMC_{TC}	COMC_{SC}	PF	PA
Joe	1	200	80.4 (3.6)	79.8 (3.6)	79.7 (3.5)	85.2 (3.4)	125.0 (3.6)
		500	84.0 (3.4)	85.0 (3.5)	85.1 (3.5)	93.0 (3.3)	133.3 (3.5)
	2	200	221.6 (4.0)	221.0 (3.9)	221.6 (3.9)	229.8 (4.0)	243.0 (4.1)
		500	225.8 (3.9)	226.1 (3.9)	225.8 (3.9)	229.6 (3.9)	243.7 (4.0)
t_2	1	200	89.5 (3.6)	89.8 (3.7)	90.0 (3.7)	93.7 (3.7)	124.5 (3.4)
		500	91.4 (4.4)	92.1 (4.5)	93.8 (5.0)	100.1 (5.2)	129.2 (3.7)
	2	200	223.4 (4.2)	223.6 (4.2)	225.0 (4.3)	233.5 (5.7)	239.6 (4.1)
		500	217.0 (3.7)	217.0 (3.7)	217.1 (3.6)	222.9 (4.0)	233.1 (3.6)
Gaussian	1	200	90.8 (3.5)	92.6 (3.7)	93.1 (3.7)	93.0 (3.5)	127.6 (3.6)
		500	87.6 (3.6)	87.3 (3.5)	87.3 (3.6)	87.9 (3.3)	125.2 (3.7)
	2	200	230.4 (3.9)	231.0 (3.9)	230.3 (3.9)	231.1 (3.9)	239.7 (3.8)
		500	232.9 (3.9)	233.5 (4.0)	233.7 (4.0)	233.6 (4.0)	246.8 (4.3)

COMC_0: COMC based on the true copula and true parameters; COMC_{TC}: COMC based on the true copula; COMC_{SC}: COMC based on the selected copula; PF: particle filtering method from Jia et al. (2021); PA: Poisson autoregression method from Fokianos and Tjøstheim (2011). Values in the parentheses are standard errors.

Table 2.7: Estimated parameters in the COMC model of order 2 for the baseball data.

Model	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\eta}$
t_7	0.25	0.21	-1.08	0.0008	0.09	0.69
	(0.10)	(0.11)	(1.40)	(0.0003)	(0.08)	(0.08)

Values in the parentheses are standard errors.

the first 320 time points are taken as training set while the last 80 as testing set. For the covariates, we include the linear and quadratic terms of time and the number of confirmed cases 10 business days ago. Many works have been done for the prediction of COVID-19 mortality, e.g. ArunKumar et al. (2021) made prediction based on recurrent neural networks. There are two reasons why we do not use the deep learning method. One is due to the small sample size. With this small sample size, it may not be enough to train a deep learning model. The other one is the interpretability issue. In contrast to deep learning, our proposed model can provide better interpretation of the dependence on covariates and the temporal

Table 2.8: One-step ahead prediction for Y_t from the baseball data produced from three methods.

	90% prediction interval		95% prediction interval			
Model	CPs	ALs	CPs	ALs	LOGS	MSRPE
COMC	100	6.64 (0.15)	100	7.84 (0.12)	1.981 (0.09)	1.11 (0.50)
PF	96.0	7.74 (0.17)	100	9.20 (0.14)	2.018 (0.09)	2.23 (0.89)
NBAR	100	6.44 (0.20)	100	7.56 (0.14)	1.984 (0.09)	0.79 (0.38)

COMC: copula-based Morkov model for count data; PF: particle filtering method from Jia et al. (2021); NBA: Negative Binomial autoregression; CPs: coverage percentages; ALs: average lengths; MSPE: mean squared prediction error. Values in the parentheses are standard errors.

dependence.

The data are over-dispersed, with a sample mean of 29.3 and a sample variance of 1063.9. So we choose the marginal model to be generalized Poisson (GP) with parameter $\varphi > 1$ fixed in time and λ_t varing with time with the log link:

$$\log(\lambda_t) = \beta_1 + \beta_2 t + \beta_3 t^2 + \beta_4 \log(x_{t-10} + 1),$$

where x_{t-10} is the number of confirmed cases 10 days ago. In this application, t, t^2 and x_{t-10} can all be considered as continuous covariates due to their large ranges. In order to determine the order of Markov process and to specify the correct copula, we conduct a cross-validation using the procedure in Section 2.2.6. We choose the copula function among five copula families: Gaussian, t, Joe, Gumbel and Clayton and consider the first and second order Markov process. Through cross-validation with $k = 320$, for the first order Markov process, t_2 copula produces the smallest LOGS, and it is not worse than that of the second order Markov process. So for computational simplicity, we select t_2 copula and $p = 1$ corresponding to the first order Markov process.

The parameters were estimated using the full data, and the estimations are shown in Table 2.9. Results suggest that most of the parameters are significantly different from zero.

For comparison, we also include the generalized Poisson autoregression (GPA) method,

Table 2.9: Estimated parameters in the COMC model of order 1 for the COVID-19 data.

Model	$\hat{\alpha}$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\varphi}$
t_2	0.82	-0.88	-1.64e-3	3.07e-6	0.60	7.64
	(0.02)	(0.29)	(1.12e-3)	(2.55e-6)	(0.04)	(0.07)

Values in the parentheses are standard errors.

which assumes the conditional distribution of Y_t is generalized Poisson (GP):

$$Y_t|(Y_{t-1},X_t) \sim GP(\exp(a+b_1\log(Y_{t-1}+1)+c_1t+c_2t^2+c_3\log(X_{t-10}+1)),\varphi),$$

and the particle filtering (PF) method proposed by Jia et al. (2021), which is a Gaussian copula based method. We assume the latent process follows an AR(1) model.

The comparison of three methods are shown in Table 2.10. We consider the coverage percentages (CPs) and the average lengths (ALs) of the prediction intervals for Y_t. For the point prediction, We construct the point predictions of Y_t using the conditional median $\hat{Y}_t$, and calculate the mean squared relative prediction error (MSRPE) by

$$MSRPE = \frac{1}{80}\sum_{t=321}^{400}\frac{(\hat{Y}_t-Y_t)^2}{Y_t^2}.$$

As we can see from Table 2.10, the LOGS of COMC is smaller than those of PF and GPA. For the interval prediction, COMC produces higher CP than GPA and smaller AL than PF. Moreover, COMC gives the smallest MSRPE.

Figure 2.3 shows the observed daily mortality in VA along with the prediction intervals given by three methods. The black line is the observation, while the red, blue and green shaded areas are the prediction intervals given by COMC, PF and GPA. As Table 2.10 suggests, Figure 2.3 shows that COMC gives narrower prediction intervals than the PF method, while the prediction intervals from GPA are too narrow to cover the observed data at an acceptable level.

Table 2.10: One-step ahead prediction for Y_t from COVID-19 data produced from three methods.

	90% prediction interval		95% prediction interval			
Model	CPs	ALs	CPs	ALs	LOGS	MSRPE
COMC	97.5	69.5 (2.1)	97.5	89.3 (2.5)	4.12 (0.07)	0.14 (0.07)
PF	96.3	86.0 (2.7)	98.8	107.7 (3.3)	4.25 (0.06)	0.17 (0.03)
GPA	65.0	19.7 (0.3)	70.0	23.5 (0.4)	5.17 (0.49)	0.19 (0.07)

COMC: copula-based Morkov model for count data; PF: particle filtering method from Jia et al. (2021); GPA: generalized Poisson autoregression; CPs: coverage percentages; ALs: average lengths; MSRPE: mean squared relative prediction error. Values in the parentheses are standard errors.

2.5 Discussion

In this chapter, we propose an approach for analyzing time series of counts and copulas are used to capture the temporal dependence. The proposed method is more flexible than existing ones that rely on the specific parametric dependence. The developed cross-validation approach provides a convenient tool to select copula with discrete margins.

Our model assumes the parametric margin for the univariate time series of counts. In the future we will seek alternative ways to obtain more flexible marginal distributions, and we will try to extend our model to multivariate time series of counts.

2.6 Proof of Propositions and Theorems

2.6.1 Proof of Proposition 2.1

Proof. Without loss of generality, we assume that

$$X|Z=z \sim Poisson(\lambda_1(z)) = F(x|z), \;\; \log(\lambda_1(z)) = \alpha_1 + \beta_1 z, \tag{2.11}$$

$$Y|Z=z \sim Poisson(\lambda_2(z)) = G(y|z), \;\; \log(\lambda_2(z)) = \alpha_2 + \beta_2 z, \tag{2.12}$$

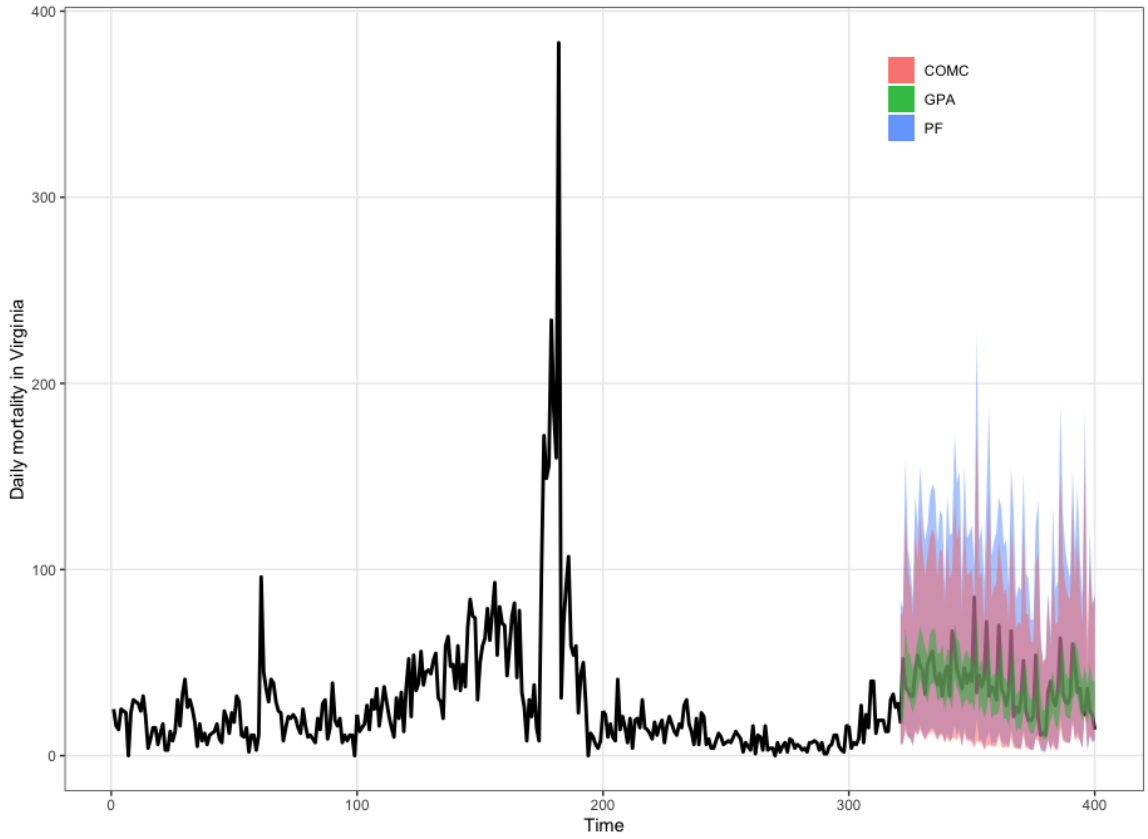

Figure 2.3: Daily mortality in VA. Black line corresponds to the observation, while the red, blue and green shaded areas are the prediction intervals given by COMC, PF and GPA.

where Z is a continuous random variable. Here X and Y are dependent. Let C_H be the set of copulas such that

$$H(x,y|z) = C(F(x|z), G(y|z); \theta), \quad x,y \in \mathbb{N} \tag{2.13}$$

holds. Therefore, for $\forall\, C, D \in C_H$, and $\forall u, v \in [0,1] \times [0,1]$, we have

$$\begin{aligned} &|C(u,v) - D(u,v)| \\ \leq &|C(u,v) - C(F(i|z), G(j|z))| + |D(u,v) - D(F(i|z), G(j|z))|. \end{aligned} \tag{2.14}$$

Since Z is a continuous random variable, $F(x|Z)$ and $G(y|Z)$ can take all the values in $(0,1)$. Therefore for $\forall\, u, v \in (0,1)$, there exist z, i, j, where i and j depend on z, such that $F(i|z) = u$

and $G(j|z) = v$, and consequently

$$\begin{aligned} |C(u,v) - D(u,v)| &\leq |C(u,v) - C(F(i|z), G(j|z)| + |D(u,v) - D(F(i|z), G(j|z)| \\ &= 0, \end{aligned} \tag{2.15}$$

and we get $C(u,v) = D(u,v)$. □

2.6.2 Proof of Theorem 2.1

Proof. We first show that the score function $\{s_t\}$, evaluated at θ, is a square integrable zero mean martingale with respect to $\mathscr{F}_t$.

Let $\{y_t, t = 1,2,\cdots\}$ be such a process generated by the copula $C(\cdot,\cdot;\alpha)$, whose marginal distribution is on a probability space $(\Omega, \mathscr{F}, P)$. The σ-field generated by the first t observations $y_1,\cdots,y_t$ is denoted by $\mathscr{F}_t$, $\mathscr{F}_0 = \{\varnothing, \Omega\}$. The probability measure P is assumed to belong to a parametric family $\{P_\beta, \beta \in B\}$, where the parameter space B is an open subset of $\mathbb{R}^m$, $m \in \mathbb{N}$. For a fixed t, let the projections $\{P_{t,\beta}, \beta \in B\}$ on the first t observations be mutually absolutely continuous. Then the corresponding likelihood exists. If the likelihood is continuous, it is nonzero for all $\beta \in B$ and unique up to a factor which does not depend on β a.s. Beyond the continuity, the likelihood is assumed to be two times continuously differentiable. Let $l_t(\alpha,\beta)$, $s_t(\alpha,\beta)$ and $-H_t(\alpha,\beta)$ denote the log-likelihood and its first and second derivatives, respectively; in addition, define $d_t(\alpha,\beta) = s_t(\alpha,\beta) - s_{t-1}(\alpha,\beta)$.

We would like to first prove the following lemma.

Lemma 2.1: The score function $\{s_t\}$, evaluated at (α,β), is a square integrable zero mean martingale with respect to $\mathscr{F}_t$.

Proof. The log-likelihood function has been shown to be

$$l_t(\beta,\alpha;y) = \sum_{i=2}^{t} \log[C(u_i,u_{i-1};\alpha) - C(u_i,v_{i-1};\alpha) - C(u_{i-1},v_i;\alpha) + C(v_i,v_{i-1};\alpha)] - \sum_{i=2}^{t-1} \log(u_i - v_i),$$

where $u_i = G(y_i|x_i;\beta)$ and $v_i = G(y_i - 1|x_i;\beta)$. We consider the scoring function s_t, which is the derivative of l_t. Let $s_t^{(1)} = \partial l_t/\partial\alpha$ and $s_t^{(2)} = \partial l_t/\partial\beta$, and we have

$$\begin{aligned} d_t^{(1)} \doteq & s_t^{(1)} - s_{t-1}^{(1)} \\ = & \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\alpha}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \end{aligned}$$

and

$$\begin{aligned} & E[d_t^{(1)}|\mathscr{F}_{t-1}] \\ = & E\left[\frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\alpha}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)}\middle|\mathscr{F}_{t-1}\right] \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\alpha}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \\ & \cdot P(Y_t = y_t|Y_{t-1} = y_{t-1}) \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\alpha}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \\ & \cdot \frac{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)}{P(Y_{t-1} = y_{t-1})} \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\alpha}{P(Y_{t-1} = y_{t-1})} \\ = & \frac{\partial}{\partial\alpha} \sum_{y_t=0}^{\infty} \frac{[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]}{P(Y_{t-1} = y_{t-1})} \\ = & \frac{\partial}{\partial\alpha} \sum_{y_t=0}^{\infty} P(Y_t = y_t|Y_{t-1} = y_{t-1}) \\ = & 0, \end{aligned}$$

so we have $\{s_t^{(1)}, \mathscr{F}_t\}$ is a martingale. Now we consider $s_t^{(2)}$. Let

$$\begin{aligned} d_t^{(2)} \doteq & s_t^{(2)} - s_{t-1}^{(2)} \\ = & \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\beta}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \\ & - \frac{\partial(u_{t-1}-v_{t-1})/\partial\beta}{u_{t-1}-v_{t-1}}. \end{aligned}$$

Taking the expectation gives

$$\begin{aligned} & E[d_t^{(2)}|\mathscr{F}_{t-1}] \\ = & E\left[\frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\beta}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)}\middle|\mathscr{F}_{t-1}\right] \\ & - \frac{\partial(u_{t-1}-v_{t-1})/\partial\beta}{u_{t-1}-v_{t-1}} \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\beta}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \\ & \cdot P(Y_t = y_t|Y_{t-1} = y_{t-1}) - \frac{\partial(u_{t-1}-v_{t-1})/\partial\beta}{u_{t-1}-v_{t-1}} \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\beta}{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)} \\ & \cdot \frac{C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)}{P(Y_{t-1} = y_{t-1};\beta)} \\ & - \frac{\partial(P(Y_{t-1} = y_{t-1};\beta))/\partial\beta}{P(Y_{t-1} = y_{t-1};\beta)} \\ = & \sum_{y_t=0}^{\infty} \frac{\partial[C(u_t,u_{t-1};\alpha) - C(u_t,v_{t-1};\alpha) - C(u_{t-1},v_t;\alpha) + C(v_t,v_{t-1};\alpha)]/\partial\beta}{P(Y_{t-1} = y_{t-1};\beta)} \\ & - \frac{\partial(P(Y_{t-1} = y_{t-1};\beta))/\partial\beta}{P(Y_{t-1} = y_{t-1};\beta)} \\ = & \frac{1}{P(Y_{t-1} = y_{t-1};\beta)} \left\{ \frac{\partial}{\partial\beta} \left[\sum_{y_t=0}^{\infty} P(Y_t = y_t, Y_{t-1} = y_{t-1};\alpha,\beta) - P(Y_{t-1} = y_{t-1};\beta) \right] \right\} \\ = & \frac{1}{P(Y_{t-1} = y_{t-1};\beta)} \left\{ \frac{\partial}{\partial\beta} \left[P(Y_{t-1} = y_{t-1};\beta) - P(Y_{t-1} = y_{t-1};\beta) \right] \right\} \\ = & 0. \end{aligned}$$

So $\{s_t^{(2)}, \mathscr{F}_t\}$ is also a martingale. □

The rest of the proof of Theorem 2.1 follows from Theorem 3 in Kaufmann (1987).

□

Chapter 3

Modeling Multivariate Time Series of Counts with Copula-linked D-vines

3.1 Introduction

Modeling dependence for multivariate count time series is essential to statistical applications in various fields, e.g. in finance and economics (Heinen and Rengifo, 2007; Heinen, 2003; Fokianos et al., 2020), and in climatology (Livsey et al., 2018; Elsner and Jagger, 2006). There are two types of dependence embedded in multivariate time series of counts. One is the temporal dependence within each time series. The other is the cross-sectional dependence across the multiple time series. Multivariate time series often present complicated dependence structures, such as nonlinear dependence, tail dependence and asymmetric dependence, which makes dependence modeling a challenging yet crucial task. It's desirable to construct a convenient modeling of multivariate time series to accommodate the complex dependence in both temporal and cross-sectional dimensions.

There have been some work for multivariate continuous time series, among which Gaussian processes are commonly used; see Storlie et al. (2017) and Datta et al. (2016). However, these models are not directly applicable to count data. For multivariate count time series, most of the existing works assume an autoregressive process; see, for instance Aldor-Noiman et al. (2016), Clark et al. (2018), and Fokianos et al. (2020). Chiogna and Gaetan (2010) suggested an interchangeable approach for modeling spatio-temporal count data.

All these works are based on relatively strong model assumptions with specific parametric forms or Markov property. We aim to build a flexible copula-based model with weaker model assumptions to accommodate different types of dependence for multivariate count time series.

Researchers have considered using copulas to model the temporal dependence of count time series. Heinen and Rengifo (2007, 2008) proposed multivariate autoregressive models for count data using Gaussian copulas based on the continuous extension (CE) proposed in Denuit and Lambert (2005). Jia et al. (2021) developed the particle filtering (PF) method to approximate the likelihood of count series. The CE based and PF methods both assume latent Gaussian process, which could be restrictive to some applications. Moreover, most of the literature focus on first-order Markov chains, possibly due to the mathematical tractability in the low dimensional setting.

To extend the copula-based univariate time series model to higher-order Markov chains, a framework to generate flexible yet tractable multivariate copulas is desirable. One of the promising directions is the vine copula, which generates multivariate copulas based on iterative pairwise construction of pair-copulas. More details of the vine-copula can be found in Joe (2014), Bedford and Cooke (2002). Aas et al. (2009) proposed regular vine (R-vine) to decompose multivariate copulas based on recursive pairwise construction of pair-copulas. The D-vine, which is a specially structured R-vine, is of particular interest due to its simplicity and natural interpretation under the time series setting. Smith et al. (2010) and Shi and Yang (2018) employed D-vine to account for the temporal dependence in longitudinal data, and Loaiza-Maya et al. (2018) used D-vine to capture the temporal dependence in stationary heteroskedastic time series. All of these methods were developed for continuous time series. Panagiotelis et al. (2012) proposed a vine copula construction for discrete data.

A few researchers have proposed copula-based models for continuous time series data to account for both temporal and cross-sectional dependence. For discrete data, Panagiotelis

et al. (2012) proposed a vine copula construction method and the corresponding algorithm. Tang et al. (2019) proposed a copula-based model for spatio-temporal data assuming Markov property, which can simultaneously model the temporal and spatial dependence. One potential drawback is that it involves a direct copula-based joint distribution of a high-dimensional vector, which is challenging both analytically and computationally, especially when the number of spatial locations is large. More recently, Zhao et al. (2021) suggested modeling the temporal dependence with D-vine and joining multiple D-vines via a parametric copula to capture cross-sectional dependence, which allows for some flexibility for modeling multivariate time series.

In this chapter we propose the copula-linked D-vine (CDvine) model for multivariate count time series. The temporal dependence of the count time series is captured by the D-vine. The D-vine structure is similar to that in the continuous case, but there is no closed expression for the likelihood under the discrete case. Based on the pair-copula construction, we design a univariate D-vine (uDvine) that generalizes the first-order copula-based Markov process in Chapter 2 to an arbitrary-order Markov process. Moreover, this pair-copula construction using D-vine decomposes a high-dimensional copula into pair-copulas, which leads to computational efficiency. In addition, this provides more flexibility in modeling since a wide class of pair-copulas can be used as building blocks for the D-vine. To capture the dependence across multiple time series, we link each of the univariate time series using a cross-sectional copula.

The rest of the chapter is organized as follows. In Section 3.2, we first introduce the uDvine and the CDvine, and then present the proposed model, a two-stage estimator and its asymptotic properties and the sequential selection procedure of pair-copula building blocks. In Section 3.3, we conduct simulation studies to assess the performance of the proposed method and demonstrate the flexibility of the CDvine. An application to a COVID-19 mortality data is considered in Section 3.4. Some concluding remarks are given in Section 3.5. All technical proofs are provided in Section 3.6.

3.2 The D-vine-based Count Time Series Model

We start from introducing the D-vine-based model for univariate count time series. After that we will extend it to the multivariate setup. Let $\boldsymbol{Y} = \{Y_t, t = 1,2,\ldots,n\}$ be the observed count time serie data at n time points, and $\boldsymbol{X} = (\boldsymbol{X}_1,\ldots,\boldsymbol{X}_n)^T$ be the corresponding $n \times d$ covariate matrix, where $\boldsymbol{X}_t = (X_{t1},\ldots,X_{td})^T$ for $t = 1,\ldots,n$ and d is the dimension of the covariates.

We first state the assumed data generating process for the univariate count time series, specified in Assumption B1.

Assumption B1:

B1(a). At each time t, the marginal distribution of Y_t is characterized by $Y_t|\boldsymbol{X} = \boldsymbol{x} \sim F(y_t|\boldsymbol{x},\boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the unknown marginal parameter vector.

B1(b). Suppose that there exists an n-dimensional copula C_n such that

$$P(Y_1 \leq y_1,\ldots,Y_n \leq y_n|\boldsymbol{X}) = C_n\{F(y_1|\boldsymbol{X}),\ldots,F(y_n|\boldsymbol{X});\boldsymbol{\alpha}\}$$

for any $y_1,\ldots,y_n \in \mathbb{R}$, where $\boldsymbol{\alpha}$ is the copula parameter vector.

Unlike the assumed Markov property in Chapter 2, the assumption B1(b) considers a general case and uses an n-dimensional copula to capture the temporal dependence. As is well known, the n-dimensional copula C_n has limitations regarding both the choices of copula families and computation, especially for large n. To overcome these challenges, we consider using D-vine to decompose C_n into pair-copula building blocks to make it more flexible and computationally more efficient. For easier presentation, we first introduce the D-vine for continuous time series and then discuss how to construct D-vine for count time series.

3.2.1 Univariate D-vine (uDvine) for Continuous Time Series

For notational simplicity, we omit the covariates $\boldsymbol{X}$ in the marginal distribution. From Sklar's Theorem as reviewed in Chapter 1, for every multivariate random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^T \sim F$ with marginal distribution functions $F_1, \ldots, F_n$, there exists a copula C_n associated with $\boldsymbol{Y}$, such that $F(y_1, \ldots, y_n) = C_n\{F_1(y_1) \ldots, F_n(y_n)\}$. The density of $\boldsymbol{Y}$ can be decomposed similarly:

$$f(y_1, \ldots, y_n) = c_n\{F_1(y_1), \ldots, F_n(y_n)\} f_1(y_1) \ldots . f_n(y_n),$$

where $c_n(u_1, \ldots, u_n) := \partial^n C_n(u_1, \ldots, u_n) / \partial u_1 \cdots \partial u_n$ is the copula density and $f_1, \ldots, f_n$ are the marginal densities. Joe (2014) and Nelsen (2007) introduced many examples of parametric copulas, especially pair-copulas. Those are of special interest to us since they are the building blocks used for the pair-copula construction of D-vines.

We apply the probability integral transform (PIT) to its marginals: $U_j := F_j\left(Y_j\right), j = 1, \ldots, n$, and $U_j's$ are then uniformly distributed and their joint distribution function is the copula C_n. For a random vector $\boldsymbol{Y}$, a set $D \subset \{1, \ldots, d\}$ and $i, j \in \{1, \ldots, d\} \backslash D$. We define the following notations:

(a) $C_{Y_i, Y_j; \mathbf{Y}_D(\cdot,\cdot; \mathbf{y}_D)}$ denotes the copula associated with the conditional distribution of $(Y_i, Y_j)'$ given $\boldsymbol{Y}_D = \boldsymbol{y}_D$. We abbreviate this by $C_{ij;D}(\cdot, \cdot; \mathbf{y}_D)$. Further, $c_{ij;D}(\cdot, \cdot; \mathbf{x}_D)$ is the copula density corresponding to $C_{ij;D}(\cdot, \cdot; \mathbf{x}_D)$.

(b) $F_{Y_i|\mathbf{y}_D}(\cdot, \cdot \mid \mathbf{y}_D)$ denotes the conditional distribution of the random variable Y_i given $\boldsymbol{Y}_D = \boldsymbol{y}_D$. We use $F_{i|D}(\cdot, \cdot \mid \mathrm{x}_D)$ as an abbreviation.

(c) $C_{U_i|\mathbf{U}_D}(\cdot, \cdot \mid \mathbf{u}_D)$ denotes the conditional distribution of the PIT random variable U_i given $\boldsymbol{U}_D = \boldsymbol{u}_D$. We abbreviate this by $C_{i|D}(\cdot, \cdot \mid \mathbf{u}_D)$.

Following Aas et al. (2009), the joint density of the continuously distributed random vector $\boldsymbol{Y}$ can be written in terms of the (conditional) pair-copula densities and its marginal

densities as

$$f(y_1,\ldots,y_n)=\prod_{k=1}^{n} f_k(y_k)\prod_{i=1}^{n-1}\prod_{j=i+1}^{n} c_{ij;i+1,\ldots,j-1}\left\{F_{i|i+1,\ldots,j-1}\left(y_i \mid y_{i+1},\ldots,y_{j-1}\right),\right.$$
$$\left. F_{j|i+1,\ldots,j-1}\left(y_j \mid y_{i+1},\ldots,y_{j-1}\right); y_{i+1},\ldots,y_{j-1}\right\}. \tag{3.1}$$

We call (3.1) that results from the pair-copula construction (PCC) a D-vine density.

Example 3.1. Figure 3.1 shows an exemplary 5-dimensional D-vine corresponding with the joint density

$$\begin{aligned} f(y_1,y_2,y_3,y_4,y_5) =& f_1(y_1) f_2(y_2) f_3(y_3) f_4(y_4) f_5(y_5) \\ & \cdot c_{12}\cdot c_{23}\cdot c_{34}\cdot c_{45} && (T_1) \\ & \cdot c_{13;2}\cdot c_{24;3}\cdot c_{35;4} && (T_2) \\ & \cdot c_{14;2,3}\cdot c_{25;3,4} && (T_3) \\ & \cdot c_{15;2,3,4}, && (T_4) \end{aligned}$$

where for brevity we omitted the arguments of the pair-copulas. Taking $c_{14;2,3}$ as an example, the full expression of $c_{14;2,3}$ should be

$$c_{14;2,3}\left\{F_{1|2,3}(y_1 \mid y_2,y_3), F_{4|2,3}(y_4 \mid y_2,y_3); y_2,y_3\right\}.$$

We see that all pair-copulas used in the decomposition appear as edges in the corresponding nested set of trees displayed in Figure 3.1. In order to decompose an n-dimensional copula, we need $n-1$ trees, denoted as $T_1,\ldots,T_{n-1}$, and $n-1$ is the order of the D-vine copula.

In order to fit a D-vine copula with a fixed order to a given data, all pair-copulas appearing in (3.1) are chosen from existing parametric pair-copula families. The conditional

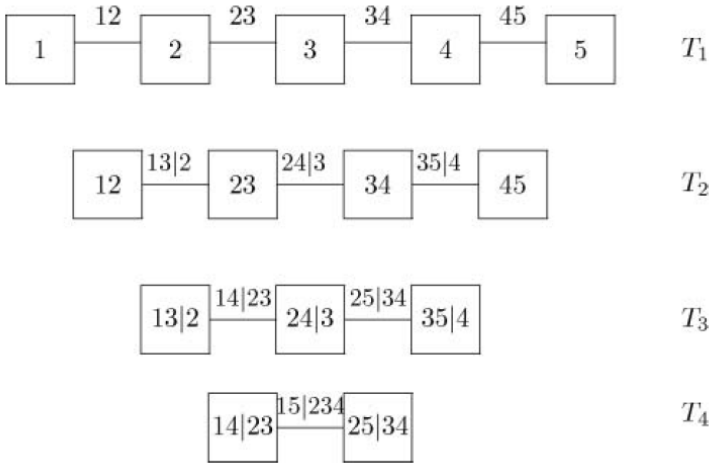

Figure 3.1: A 5-dimendion D-vine.

distributions $F_{i|D}(y_i \mid \mathbf{y}_D)$, which are the arguments of the pair-copulas we omitted in Example 3.1, can be evaluated using only the pair-copulas specified for the D-vine from lower trees by applying the following recursion, which was stated by Joe (2014) as follows. Let $l \in D$ and $D_{-l} := D \backslash \{l\}$. Then,

$$F_{i|D}(y_i \mid \mathbf{y}_D) = h_{i|l;D_{-l}}\left\{F_{i|D_{-l}}\left(y_i \mid \mathbf{y}_{D_{-l}}\right) \mid F_{l|D_{-l}}\left(y_l \mid \mathbf{y}_{D_{-l}}\right)\right\}, \tag{3.2}$$

where for $i, j \notin D, i < j$,

$$h_{i|j;D}(u \mid v) = \partial C_{ij;D}(u,v)/\partial v = C_{i|j;D}(u \mid v),$$

and analogously

$$h_{j|i;D}(u \mid v) = \partial C_{ij;D}(u,v)/\partial u = C_{j|i;D}(u \mid v)$$

are the h-functions associated with the pair-copula $C_{ij;D}$. In Example 3.1, the first argument of $c_{14;2,3}$ from T_3, namely $F_{1|2,3}(y_1|y_2,y_3)$, can be evaluated using the h-functions associated

with $C_{13,2}, C_{12}$ and C_{23} from the first two trees:

$$\begin{aligned} F_{1|2,3}(x_1 \mid x_2, x_3) &= h_{1|3;2}\left\{F_{1|2}(x_1 \mid x_2) \mid F_{3|2}(x_3 \mid x_2)\right\} \\ &= h_{1|3;2}\left[h_{1|2}\left\{F_1(x_1) \mid F_2(x_2)\right\} \mid h_{3|2}\left\{F_3(x_3) \mid F_2(x_2)\right\}\right]. \end{aligned}$$

3.2.2 Univariate D-vine (uDvine) for Count Time Series

In this section, we introduce vine pair-copula constructions (PCCs) for discrete margins. Using similar notation as in Section 3.2.1, for a count vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)' \sim F$ with marginal distribution functions $F_1, \ldots, F_n$, we assume that the exists an n-dimensional copula C_n such that

$$F(y_1, \ldots, y_n) = C_n\{F_1(y_1) \ldots, F_n(y_n)\}.$$

We will decompose the n-dimensional copula using D-vine. For continuous case, the joint density of $\boldsymbol{Y}$ constructed under the D-vine framework has a closed expression. However, for discrete distributions, there is no closed form for the joint CDF resulting from the D-vine model, and consequently for the associated joint PMF of $\boldsymbol{Y}$.

Below we discuss how to decompose the n-dimensional copula using D-vine for count data. Let $l \in D$ and $D_{-l} := D \backslash \{l\}$. We derive the conditional distribution as follows:

$$\begin{aligned} & P(Y_i = y_i \mid \mathbf{Y}_D = \boldsymbol{y}_D) \\ =& \frac{P\left(Y_i = y_i, Y_l = y_l \mid \mathbf{Y}_{D_{-l}} = \mathbf{y}_{D_{-l}}\right)}{P\left(Y_l = y_l \mid \mathbf{Y}_{D_{-l}} = \mathbf{y}_{D_{-l}}\right)} \\ =& \frac{\sum_{j_i=0,1}\sum_{j_l=0,1}(-1)^{j_i+j_l} P\left(Y_i \leq y_i - j_i, Y_l \leq y_l - i_l \mid \mathbf{Y}_{D_{-l}} = \mathbf{y}_{D_{-l}}\right)}{P\left(Y_l = y_l \mid \mathbf{Y}_{D_{-l}} = \mathbf{y}_{D_{-l}}\right)} \\ =& \frac{\sum_{j_i=0,1}\sum_{j_l=0,1}(-1)^{j_i+j_l} C_{il|D_{-l}}\{F_{i|D_{-l}}(y_i - j_i|\mathbf{y}_{D_{-l}}), F_{l|D_{-l}}(y_l - j_l|\mathbf{y}_{D_{-l}})\}}{P\left(Y_l = y_l \mid \mathbf{Y}_{D_{-l}} = \mathbf{y}_{D_{-l}}\right)}. \end{aligned} \tag{3.3}$$

Equation (3.3) is a discrete analogue to (3.2) since it can be applied recursively to decompose a multivariate PMF into pair-copula building blocks. The decomposition procedure is similar to that of the continuous setup. For illustration, we now outline such a decomposition

in detail for the three-dimensional case.

Example 3.2: For $n = 3$, we decompose the joint PMF as

$$P(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) = P(Y_1 = y_1 | Y_2 = y_2, Y_3 = y_3) P(Y_3 = y_3 | Y_2 = y_2) \cdot P(Y_2 = y_2). \tag{3.4}$$

Applying (3.3) to $P(Y_1 = y_1 | Y_2 = y_2, Y_3 = y_3)$ gives

$$\begin{aligned} & P(Y_1 = y_1 | Y_2 = y_2, Y_3 = y_3) \\ = & \frac{\sum_{j_1=0,1} \sum_{j_3=0,1} (-1)^{j_1+j_3} C_{13|2} \{F_{1|2}(y_1 - j_1 | y_2), F_{3|2}(y_3 - j_3 | y_2)\}}{P(Y_3 = y_3 \mid Y_2 = y_2)}, \end{aligned} \tag{3.5}$$

where the first argument of the copula function in the numerator is

$$F_{1|2}(y_1 - j_1 | y_2) = \frac{C_{12}\{F_1(y_1 - j_1), F_2(y_2)\} - C_{12}\{F_1(y_1 - j_1), F_2(y_2 - 1)\}}{P(Y_2 = y_2)}, \tag{3.6}$$

and the second argument can be expressed as

$$F_{3|2}(y_3 - j_3 | y_2) = \frac{C_{32}\{F_3(y_3 - j_3), F_2(y_2)\} - C_{32}\{F_3(y_3 - j_3), F_2(y_2 - 1)\}}{P(Y_2 = y_2)}. \tag{3.7}$$

The CDF of negative values are defined as 0 above. Since the denominator of (3.5) cancels with the second term on the right-hand side of (3.4), the full expression for the PMF of this three-dimensional D-vine is given by

$$\begin{aligned} P(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3) = & \left[\sum_{j_1=0,1} \sum_{j_3=0,1} (-1)^{j_1+j_3} C_{13|2} \right. \\ & \left\{ \frac{C_{12}\{F_1(y_1 - j_1), F_2(y_2)\} - C_{12}\{F_1(y_1 - j_1), F_2(y_2 - 1)\}}{P(Y_2 = y_2)}, \right. \\ & \left. \left. \frac{C_{32}\{F_3(y_3 - j_3), F_2(y_2)\} - C_{32}\{F_3(y_3 - j_3), F_2(y_2 - 1)\}}{P(Y_2 = y_2)} \right\} \right] \\ & \{F_2(y_2) - F_2(y_2 - 1)\}. \end{aligned} \tag{3.8}$$

Panagiotelis et al. (2012) proposed the following algorithm to compute the PMF of a D-vine. For notation used in this algorithm, for a vector of integers $\boldsymbol{i}$, let $\boldsymbol{Y_i} = (Y_i, : i \in \boldsymbol{i})$. $F^+_{g|\boldsymbol{i}} := \mathrm{P}(Y_g \leq y_g | \boldsymbol{Y_i} = \boldsymbol{y_i})$, $F^-_{g|\boldsymbol{i}} := \mathrm{P}(Y_g \leq y_g - 1 | \boldsymbol{Y_i} = \boldsymbol{y_i})$, and $f_{g|\boldsymbol{i}} := \mathrm{P}(Y_g = y_g | \boldsymbol{Y_i} = \boldsymbol{y_i})$. For the copula function evaluated at four differernt values, we let

$$C^{++}_{gj|\boldsymbol{i}} := C_{gj|\boldsymbol{i}}\left(F^+_{g|\boldsymbol{i}}, F^+_{j|\boldsymbol{i}}\right), \quad C^{+-}_{gj|\boldsymbol{i}} := C_{gj|\boldsymbol{i}}\left(F^+_{g|\boldsymbol{i}}, F^-_{j|\boldsymbol{i}}\right),$$
$$C^{-+}_{gj|\boldsymbol{i}} := C_{gj|\boldsymbol{i}}\left(F^-_{g|\boldsymbol{i}}, F^+_{j|\boldsymbol{i}}\right), \quad C^{--}_{gj|\boldsymbol{i}} := C_{gj|\boldsymbol{i}}\left(F^-_{g|\boldsymbol{i}}, F^-_{j|\boldsymbol{i}}\right).$$

The algorithm in Panagiotelis et al. (2012) is as follows:

Algorithm 3.1

1. For $d = 1, \cdots, n$, evaluate F^+_d, F^-_d, and $f_d = F^+_d - F^-_d$.

2. For $d = 1, \cdots, n-1$, evaluate $C^{++}_{d,d+1} = C_{d,d+1}\left(F^+_d, F^+_{d+1}\right)$, $C^{+-}_{d,d+1} = C_{d,d+1}\left(F^+_d, F^-_{d+1}\right)$, $C^{-+}_{d,d+1} = C_{d,d+1}\left(F^-_d, F^+_{d+1}\right)$, and $C^{--}_{d,d+1} = C_{d,d+1}\left(F^-_d, F^-_{d+1}\right)$.

3. For $d = 1, n-2$:

   (a) Evaluate $F^+_{d|d+1} = \left(C^{++}_{d,d+1} - C^{+-}_{d,d+1}\right) / (f_{d+1})$, $F^-_{d|d+1} = \left(C^{-+}_{d,d+1} - C^{--}_{d,d+1}\right) / (f_{d+1})$, and $f_{d|d+1} = F^+_{d|d+1} - F^-_{d|d+1}$. These are used as the first argument for $C_{d,d+2|d-1}$ in step (c) below.

   (b) Evaluate $F^+_{d+2|d+1} = \left(C^{++}_{d+2,d+1} - C^{+-}_{d+2,d+1}\right) / (f_{d+1})$, $F^-_{d+2|d+1} = \left(C^{-+}_{d+2,d+1} - C^{--}_{d+2,d+1}\right) / (f_{d+1})$, and $f_{d+2|d+1} = F^+_{d+2|d+1} - F^-_{d+2|d+1}$. These are used as the first argument for $C_{d,d+2|d-1}$ in step (c) below.

   (c) Evaluate $C^{ab}_{d,d+2|d+1}\left(F^a_{d|d+1}, F^b_{d+2|d+1}\right)$ for $a, b \in \{+, -\}$.

4. For $t = 3, \cdots, n-1$ and $d = 1, \cdots, n-t$:

   (a) For the first argument of $C_{d,d+t|d+1,\cdots,d+t-1}$ for step (c) below:

      (i) Evaluate $F^+_{d|d+1,\cdots,d+t-1} = \frac{C^{++}_{d,d+t-1|d+1,\ldots,d+t-2} - C^{+-}_{d,d+t-1|d+1,\ldots,d+t-2}}{f_{d+t-1|d+1,\ldots,d+t-2}}$.

      (ii) Evaluate $F^-_{d|d+1,\cdots,d+t-1} = \frac{C^{-+}_{d,d+t-1|d+1,\ldots,d+t-2} - C^{--}_{d,d+t-1|d+1,\ldots,d+t-2}}{f_{d+t-1|d+1,\ldots,d+t-2}}$.

(iii) Evaluate $f_{d|d+1,\cdots,d+t-1} = F^{+}_{d|d+1,\cdots,d+t-1} - F^{-}_{d|d+1,\cdots,d+t-1}$.

(b) For the second argument of $C_{d,d+t|d+1,\cdots,d+t-1}$ for step (c) below:

(i) Evaluate $F^{+}_{d+t|d+1,\cdots,d+t-1} = \frac{C^{++}_{d+1,d+t|d+2,\ldots,d+t-1} - C^{+-}_{d+1,d+t|d+2,\ldots,d+t-1}}{f_{d+1|d+2,\ldots,d+t-1}}$.

(ii) Evaluate $F^{-}_{d+t|d+1,\cdots,d+t-1} = \frac{C^{-+}_{d+1,d+t|d+2,\ldots,d+t-1} - C^{--}_{d+1,d+t|d+2,\ldots,d+t-1}}{f_{d+1|d+2,\ldots,d+t-1}}$.

(iii) Evaluate $f_{d+t|d+1,\cdots,d+t-1} = F^{+}_{d+t|d+1,\cdots,d+t-1} - F^{-}_{d+t|d+1,\cdots,d+t-1}$.

(c) Evaluate $C^{ab}_{d,d+t|d+1,\ldots,d+t-1}\left(F^{a}_{d|d+1,\ldots,d+t-1}, F^{b}_{d+t|d+1,\ldots,d+t-1}\right)$.

5. Evaluate: $F^{+}_{3|1,2} = \left(C^{++}_{1,3|2} - C^{-+}_{1,3|2}\right) / \left(f_{3|2}\right)$, $F^{-}_{3|1,2} = \left(C^{+-}_{1,3|2} - C^{--}_{1,3|2}\right) / \left(f_{3|2}\right)$.

and for $d = 4, \cdots, n$,

$$F^{+}_{d|1,\cdots,d-1} = \left(C^{++}_{1,d|2,\ldots,d-1} - C^{-+}_{1,d|2,\ldots,d-1}\right) / \left(f_{1|2,\ldots,d-1}\right).$$

$$F^{-}_{d|1,\cdots,d-1} = \left(C^{+-}_{1,d|2,\ldots,d-1} - C^{--}_{1,d|2,\ldots,d-1}\right) / \left(f_{1|2,\ldots,d-1}\right).$$

and

$$f_{d|1,\cdots,d-1} = F^{+}_{d|1,\cdots,d-1} - F^{-}_{d|1,\cdots,d-1} \quad \text{for } d = 3,4,\cdots,n.$$

6. The PMF is given by $f_{12}\prod_{i=3}^{m} f_{i|1,\ldots,i-1}$, where f_{12} is the joint PMF of Y_1 and Y_2.

3.2.3 uDvine for Count Time Series Under the Regression Setup

We first state formally the model assumption under the regression setup:

Assumption B1:

B1(a). At each time t, the marginal distribution of Y_t is characterized by $Y_t|\boldsymbol{X} = \boldsymbol{x} \sim F(y_t|\boldsymbol{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the unknown marginal parameter vector.

B1(b). Suppose that there exists an n-dimensional copula C_n such that

$$P(Y_1 \leq y_1, \ldots, Y_n \leq y_n|\boldsymbol{X}) = C_n\{F(y_1), \ldots, F(y_n); \boldsymbol{\alpha}\}$$

for any $y_1, \ldots, y_n \in \mathbb{R}$, where $\boldsymbol{\alpha}$ is the copula parameter vector, and we assume that C_n is the same across covariates $\boldsymbol{X}$.

As we mentioned before, D-vine can be used to decompose C_n to pair-copula building blocks. The decomposition of an n-dimensional copula involves a total of $n(n-1)/2$ number of pair-copula. Selecting appropriate pair-copulas is a challenging task when n is large. For simplicity, we assume all pair-copulas in the same tree are the same, i.e. $C_{i,j}(\cdot,\cdot;\boldsymbol{\alpha}) = C_{i',j'}(\cdot,\cdot;\boldsymbol{\alpha})$ when $i-j = i'-j'$ for computational convenience, but they are allowed to vary across trees. Zhao et al. (2021) showed that this assumption can lead to stationary time series under the non-regression setup. Thus, under this assumption, in order to specify the joint distribution of $\boldsymbol{Y}|\boldsymbol{X}$, we need to specify a marginal distribution $F(\cdot|\boldsymbol{X};\beta)$ and $(n-1)$ pair-copulas instead of $(n-1)n/2$ pair-copulas. This could still be challenging especially when n is large.

One solution to overcome the challenge is to truncate the uDvine after a certain level (say T_p) and set all pair-copulas beyond T_p, i.e. $\{c_{s,t}, t-s > p\}$, to be independent copulas, where $p \ll n$ (p is far less than n). We refer to the truncated uDvine at level p with pair-copula parameters $\boldsymbol{\alpha}$ as uDvine(p,$\boldsymbol{\alpha}$). The formal definition is as follows:

Definition 3.1: Define $\boldsymbol{Y}|\boldsymbol{X} \sim$ uDvine$(p, \boldsymbol{\alpha})$ if the distribution of $\boldsymbol{Y}|\boldsymbol{X}$ is uDvine truncated at tree p with pair-copula parameters $\boldsymbol{\alpha}$.

Joe (2014) also shows a p-th truncated D-vine only has p-th order dependence if no covariates is included. Here we show that under the regression setup, p-th order dependence of the p-th truncated D-vine still holds, which then implies the Markov property as assumed in Chapter 2.

Proposition 3.1: The univariate time series $\boldsymbol{Y}$ generated by uDvine(p, $\boldsymbol{\alpha}$) is a p-order conditional Markov process, i.e.,

$$P(Y_t \leq y | Y_{t-1}, \ldots, Y_1, \boldsymbol{X}) = P(Y_t \leq y | Y_{t-1}, \ldots, Y_{t-p}, \boldsymbol{X})$$

for any $y \in \mathbb{R}$.

In Example 3.1, a 2-nd truncated 5-dimensional D-vine has the density

$$\begin{aligned} f(y_1,y_2,y_3,y_4,y_5) =& f_1(y_1) f_2(y_2) f_3(y_3) f_4(y_4) f_5(y_5) \\ & \cdot c_{12} \cdot c_{23} \cdot c_{34} \cdot c_{45} \qquad (T_1) \\ & \cdot c_{13;2} \cdot c_{24;3} \cdot c_{35;4}. \qquad (T_2) \end{aligned}$$

3.2.4 Copula-linked D-vines (CDvine) Model for Multivariate Count Time Series

Assume we have m-dimentional multivariate time series data $\boldsymbol{Y} = \{\boldsymbol{Y_t} = \{Y_{ti}\}_{i=1}^m\}_{t=1}^n$ with covariates matrix $\boldsymbol{X} = \{\boldsymbol{X_t} = \{\boldsymbol{X}_{ti}\}_{i=1}^m\}_{t=1}^n$, where $\boldsymbol{X}_{ti} = (X_{ti1}, \ldots, X_{tid})$ for $i = 1, \ldots, m$, $t = 1, \ldots, n$ and d is the dimension of the covariates at each time point for each component time series. Let $\mathscr{F}_{t-1} = \sigma(\boldsymbol{Y}_{t-1}, \boldsymbol{Y}_{t-2}, \ldots)$ denote the sigma field of all past information and $\mathscr{F}_{t-1}^i = \sigma(\boldsymbol{Y}_{t-1,i}, \boldsymbol{Y}_{t-2,i}, \ldots)$ denote the σ-field of the past information from the i-th component univariate time series. $\mathscr{F}_0, \mathscr{F}_0^1, \ldots, \mathscr{F}_0^m$ are all defined to be $= \{\varnothing, \Omega\}$. We have the following definition for CDvine:

Definition 3.2: The time series $\boldsymbol{Y}|\boldsymbol{X}$ is defined as a copula-linked D-vines (CDvine) if its component univariate time series $\{Y_{ti}|\boldsymbol{X}\}_{t=1}^n$ follows uDvine(p_i, $\boldsymbol{\alpha}_i$), for $i = 1, \ldots, m$, where p_i is the order of the i-th uDvine.

The conditional joint distribution can be written as:

$$\begin{aligned} & G(\boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{\gamma}) \\ =& P(Y_{t1} \le y_{t1}, \ldots, Y_{tm} \le y_{tm}|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{\gamma}) \\ =& C_m\left\{F_1\left(y_{t1}|\mathscr{F}_{t-1}^1, \boldsymbol{X}; \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1\right), \ldots, F_m\left(y_{tm}|\mathscr{F}_{t-1}^m, \boldsymbol{X}; \boldsymbol{\alpha}_m, \boldsymbol{\beta}_m\right)|\mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma}\right\}, \qquad (3.9) \end{aligned}$$

where $C_m(\cdot|\mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma})$ is an m-dimensional copula with parameter $\boldsymbol{\gamma}$ that captures the conditional cross-sectional dependence given history $\mathscr{F}_{t-1}$ and $\boldsymbol{X}$, and $F_i(\cdot|\mathscr{F}_{t-1}^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ is the conditional marginal distribution of Y_{ti} given its own history $\mathscr{F}_{t-1}^i$ and $\boldsymbol{X}$. The C_m

and $\boldsymbol{\gamma}$ are both assumed to be time-invariant.

Now we formally state the model assumption for the assumed data generating process:

Assumption B:

B1 For each univariate time series, at each time t, the marginal distribution of Y_{ti} is characterized by $Y_{ti}|\boldsymbol{X} = \boldsymbol{x} \sim F(y_{ti}|\boldsymbol{x}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the unknown marginal parameter vector, $i = 1, \ldots, m$.

B2 For each univariate time series $\{Y_{ti}|\boldsymbol{X}\}_{t=1}^{n}$, the temporal dependence is captured by uDvine(p_i, $\boldsymbol{\alpha}_i$), that is,

$$\{Y_{ti}\}_{t=1}^{n}|\boldsymbol{X} \sim \text{uDvine}(p_i, \boldsymbol{\alpha}_i), i = 1, \ldots, m.$$

B3 There exists an m-dimensional copula C_m such that the conditional joint distribution given history $\mathscr{F}_{t-1}$ and $\boldsymbol{X}$ is given by:

$$\begin{aligned} & G(\boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^{m}, \{\boldsymbol{\beta}_i\}_{i=1}^{m}, \boldsymbol{\gamma}) \\ =& C_m\left\{F_1\left(y_{t1}|\mathscr{F}_{t-1}^{1}, \boldsymbol{X}; \boldsymbol{\alpha}_1, \boldsymbol{\beta}_1\right), \ldots, F_m\left(y_{tm}|\mathscr{F}_{t-1}^{m}, \boldsymbol{X}; \boldsymbol{\alpha}_m, \boldsymbol{\beta}_m\right)|\mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma}\right\}, \end{aligned} \tag{3.10}$$

where C_m and $\boldsymbol{\gamma}$ are both assumed to be time-invariant.

3.2.5 Two-stage Estimation Procedure

Given the model assumption B, there are three components to be estimated: (a) the marginal parameters $\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m\}$ of the m component uDvines, (b) the parameters $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$ of pair-copulas in the m component uDvines, (c) the parameter $\boldsymbol{\gamma}$ of the cross-sectional copula $C_m(\cdot)$. Let

$$g(\boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^{m}, \{\boldsymbol{\beta}_i\}_{i=1}^{m}, \boldsymbol{\gamma}) = P(\boldsymbol{Y}_t = \boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^{m}, \{\boldsymbol{\beta}_i\}_{i=1}^{m}, \boldsymbol{\gamma})$$

denote the conditional joint distribution of $\boldsymbol{Y}_t$. Similar to Chapter 2, if C_m in Assumption B3 is selected to be copula families without closed expressions, such as elliptical copulas, including Gaussian copula or t copula, then the conditional joint distribution of $\boldsymbol{Y}_t$ can be obtained as

$$
\begin{aligned}
& g(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{\boldsymbol{\beta}_i\}_{i=1}^m,\boldsymbol{\gamma}) \\
=& P(\boldsymbol{Y}_t=\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{\boldsymbol{\beta}_i\}_{i=1}^m,\boldsymbol{\gamma}) \\
=& P(Y_{t1}=y_{t1},\ldots,Y_{tm}=y_{tm}|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{\boldsymbol{\beta}_i\}_{i=1}^m,\boldsymbol{\gamma}) \\
=& \int_{\Phi_1^{-1}(F_{t1}^2)}^{\Phi_1^{-1}(F_{t1}^1)} \cdots \int_{\Phi_1^{-1}(F_{tm}^2)}^{\Phi_1^{-1}(F_{tm}^1)} \phi_m\left(\psi_1,\ldots,\psi_m|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\gamma}\right) d\psi_1\cdots d\psi_m. \qquad (3.11)
\end{aligned}
$$

Here, $\phi_m(\cdot;\mathscr{F}_{t-1},\boldsymbol{X},\boldsymbol{\gamma})$ denotes the probability density function (PDF) of an m-dimensional elliptical distribution with location 0 and correlation given by the parameters $\boldsymbol{\gamma}$, Φ_1 denotes the CDF of the univariate margin of the same elliptical distribution, $F_{tj}^1 := \mathrm{P}\left(Y_{tj}\le y_{tj}|\mathscr{F}_{t-1}^j,\boldsymbol{X};\boldsymbol{\alpha}_j,\boldsymbol{\beta}_j\right)$ and $F_{tj}^2 := \mathrm{P}\left(Y_{tj}\le y_{tj}-1|\mathscr{F}_{t-1}^j,\boldsymbol{X};\boldsymbol{\alpha}_j,\boldsymbol{\beta}_j\right)$ for $t=1,\ldots,n,j=1,\ldots,m$. Otherwise, if the m-dimensional copula has a closed expression for the CDF, we can then use the finite difference form as in Chapter 2 to calculate the conditional joint distribution, that is,

$$
\begin{aligned}
& g(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{\boldsymbol{\beta}_i\}_{i=1}^m,\boldsymbol{\gamma}) \\
=& P(\boldsymbol{Y}_t=\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{\boldsymbol{\beta}_i\}_{i=1}^m,\boldsymbol{\gamma}) \\
=& \sum_{j_1=1}^{2}\cdots\sum_{j_m=1}^{2}(-1)^{j_1+\cdots+j_m} C_m\left(F_{t1}^{j_1},\ldots,F_{tm}^{j_m}|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\gamma}\right). \qquad (3.12)
\end{aligned}
$$

The log-likelihood function of $\boldsymbol{Y}|\boldsymbol{X}$ can be derived as follows based on the conditional

joint distribution:

$$
\begin{aligned}
& L(\{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{\gamma} | \{\boldsymbol{y}_t\}_{t=1}^n, \boldsymbol{X}) \\
= & \sum_{t=1}^n \log g\left(\boldsymbol{y}_t | \mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{\gamma}\right) \\
= & \begin{cases} \sum_{t=1}^n \log \left\{ \int_{\Phi_1^{-1}\left(F_{t1}^2\right)}^{\Phi_1^{-1}\left(F_{t1}^1\right)} \cdots \int_{\Phi_1^{-1}\left(F_{tm}^2\right)}^{\Phi_1^{-1}\left(F_{tm}^1\right)} \phi_m\left(\psi_1, \ldots, \psi_m | \mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma}\right) d\psi_1 \cdots d\psi_m \right\} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad C_m\text{: without closed expression} \\ \sum_{t=1}^n \log \left\{ \sum_{j_1=1}^2 \cdots \sum_{j_m=1}^2 (-1)^{j_1+\cdots+j_m} C_m\left(F_{t1}^{j_1}, \ldots, F_{tm}^{j_m} | \mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma}\right) \right\} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad C_m\text{: with closed expression} \end{cases}
\end{aligned}
\tag{3.13}
$$

For copula families without closed expressions, the likelihood involves integration, and the simulated likelihood proposed in Nikoloulopoulos (2013) can be used to assist the calculation. Note that even if we only see parameter $\boldsymbol{\gamma}$ from the likelihood function, $\{\boldsymbol{\alpha}_i\}_{i=1}^m$ and $\{\boldsymbol{\beta}_i\}_{i=1}^m$ enters the likelihood function through $F_{t1}^1, F_{t1}^2, \ldots, F_{tm}^1, F_{tm}^2$. The joint estimation of all parameters $\{\boldsymbol{\alpha}_i\}_{i=1}^m$, $\{\boldsymbol{\beta}_i\}_{i=1}^m$ and $\boldsymbol{\gamma}$ can be computationally expensive due to the large number of unknown parameters, even though it can lead to more efficient estimation. To improve computational efficiency, we adapt the two-stage estimation procedure. The basic idea is to decompose (3.13) into several components and optimize each component separately.

In the first stage, for $i = 1, \ldots, m$, the marginal parameter $\boldsymbol{\beta}_i$ and the parameter $\boldsymbol{\alpha_i}$ in the uDvine(p_i, $\boldsymbol{\alpha_i}$) are estimated using the i-th component univariate time series $\{Y_{ti}\}_{t=1}^n$. The likelihood of uDvine(p_i, $\boldsymbol{\alpha_i}$) can be obtained using Algorithm 3.1. That is, we first obtain the estimators $\hat{\boldsymbol{\alpha}}_i$ and $\hat{\boldsymbol{\beta}}_i$ for each $i = 1, \ldots, m$ separately by maximizing the log-likelihood:

$$
l_{1i}(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i | \{y_{ti}\}_{t=1}^n, \boldsymbol{X}) = \sum_{t=1}^n \log f_i(y_{ti} | \mathscr{F}_{t-1}^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i), \tag{3.14}
$$

where $f_i(y_{ti} | \mathscr{F}_{t-1}^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ denotes the conditional PMF of Y_{ti} given $\mathscr{F}_{t-1}^i$ and $\boldsymbol{X}$.

In the second stage, given estimators $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^m$ and $\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^m$, the estimator of $\boldsymbol{\gamma}$, denoted as $\hat{\boldsymbol{\gamma}}$, can be obtained by maximizing

$$l_2\left(\boldsymbol{\gamma}|\left\{\boldsymbol{y}_t\right\}_{t=1}^n\right)=\sum_{t=1}^n \log g\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^m,\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^m,\boldsymbol{\gamma}\right). \tag{3.15}$$

3.2.6 Point and Interval Prediction

For time series data analysis, one common question of interest is prediction at future time points. We now discuss the one-step-ahead prediction, that is, prediction of $\boldsymbol{Y}_{n+1}$, based on the proposed model and estimator. Different from Chapter 2, here our interest is the prediction of an m-dimensional response vector at time $n+1$. As stated earlier, based on our CDvine model framework, the predictive distribution of $\boldsymbol{Y}_{n+1}$ does not have a closed expression. We use numerical simulation to characterize the predictive distribution and conduct prediction. Specifically, we generate 1000 samples from the predictive distribution with parameters estimated using the observed data. The samples can then be used to construct predictions, e.g. component-wise point predictions based on the sample means or sample medians, or prediction regions; see e.g. Turkkan and Pham-Gia (1997) and Park et al. (2001) for the discussion of credible regions.

Below we describe the algorithm to generate data from a fitted CDvine. Let $\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^n$, $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^m$ and $\hat{\boldsymbol{\gamma}}$ be the two-stage estimators of marginal parameters, uDvine parameters and cross-sectional copula parameters obtained by fitting the model using the observed data. We denote an m-dimensional multivariate time series by $\{\mathbf{Y}_t=(Y_{t1},\ldots,Y_{tm})\}_{t=1}^n$. Recall that

the predictive distribution (PMF) of $\boldsymbol{Y}_{n+1}$, given $\mathscr{F}_n$ and $\boldsymbol{X}$ is

$$
\begin{aligned}
&P(\boldsymbol{Y}_t = \boldsymbol{y}_t | \mathscr{F}_{t-1}, \boldsymbol{X}; \{\boldsymbol{\alpha}_i\}_{i=1}^m, \{\boldsymbol{\beta}_i\}_{i=1}^m, \boldsymbol{\gamma}) \\
&= \begin{cases}
\sum_{t=1}^n \log \left\{ \int_{\Phi_1^{-1}(F_{t1}^2)}^{\Phi_1^{-1}(F_{tm}^1)} \cdots \int_{\Phi_1^{-1}(F_{tm}^2)}^{\Phi_1^{-1}(F_{tm}^1)} \phi_m(\psi_1, \ldots, \psi_m | \mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma}) \, d\psi_1 \cdots d\psi_m \right\} \\
\qquad\qquad\qquad\qquad\qquad\qquad C_m \text{: without closed expression} \\
\sum_{t=1}^n \log \left\{ \sum_{j_1=1}^2 \cdots \sum_{j_m=1}^2 (-1)^{j_1+\cdots+j_m} C_m \left(F_{t1}^{j_1}, \ldots, F_{tm}^{j_m} | \mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\gamma} \right) \right\} \\
\qquad\qquad\qquad\qquad\qquad\qquad C_m \text{: with closed expression}
\end{cases}.
\end{aligned}
\tag{3.16}
$$

The prediction is based on 1000 random samples drawn from the predictive distribution. For b from 1 to 1000, we repeat the following steps:

Step 1: Simulate $(V_1^b, \ldots, V_m^b)$ from the cross-sectional copula $C_m(\cdot; \hat{\boldsymbol{\gamma}})$.

Step 2: Generate a one-day ahead sample $\boldsymbol{Y}_{n+1}^b = (Y_1^b, \ldots, Y_m^b)^T$ by $Y_i^b = F_i^{-1}\left(V_i^b \mid \mathscr{F}_n^i, \boldsymbol{X}; \hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i\right)$, where $F_i^{-1}, i = 1, \ldots, m$ is defined as follows:

$$
F_i^{-1}\left(v_i \mid \mathscr{F}_n^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\right) = \inf\{y_i \geq 0 : F_i\left(y_i \mid \mathscr{F}_n^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\right) \geq v_i\}. \tag{3.17}
$$

In (3.17), $F_i\left(\cdot \mid \mathscr{F}_n^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\right)$ for $i = 1, \ldots, m$ could be obtained from using Steps 3-5 outlined in Algorithm 3.1, in which we have parameters $\boldsymbol{\beta}_i$ from the marginal distribution and parameters $\boldsymbol{\alpha}_i$ from the pair-copulas in uDvine(p_i, $\boldsymbol{\alpha}_i$).

We can use the sample component-wise mean or median of the 1000 random samples from the predictive distribution to construct pointwise prediction of $Y_{n+1,i}$ for the i-th component, $i = 1, \ldots, m$. We can also use the random draws to construct the $100(1-\nu)\%$ prediction interval $[L, U]$ for the one-step ahead observation $Y_{n+1,i}$, which satisfies

$$
P(L \leq Y_{n+1,i} \leq U | \mathscr{F}_n^i, \boldsymbol{X}; \hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i) \geq 1 - \nu. \tag{3.18}
$$

Similar to the construction of prediction intervals in Chapter 2, we take

$$L = \sup\{y : G_i(y-1) \leq \nu/2\},\ U = \inf\{y : G_i(y) \geq 1-\nu/2\}. \tag{3.19}$$

where G_i is the empirical distribution of the random samples $\{Y_i^b\}_{b=1}^{1000}$.

We focus on probabilistic forecasts to evaluate the predictive performance of a candidate model, e.g., the CDvine based on a candidate copula, for multivariate count data. The multivariate rank is an assessment tool for probabilistic forecasts of vector-valued continuous quantities, but it's difficult to get for multivariate count data. Similar to Chapter 2, we use the LOGS (defined as in Chapter 2, Section 2.2.6) to evaluate the predictive distribution. The reason we focus on probabilistic forecasts instead of using the commonly used mean squared error (MSE) for point prediction is that count data is often skewed, so that the MSE will be dominated by predictions at the tails. What's more, LOGS can be used to assess the performance of both component-wise prediction and joint prediction. We first consider the LOGS for component-wise prediction. Suppose that the training data consists of observations from time 1 to n, and the testing time point is $t = n+1$. Let $\boldsymbol{y}_{n+1} = (y_{n+1,i}, i = 1,\ldots,m)^T$ denote the observed response vector at the testing time point. The predictive distribution for $Y_{n+1,i}$ can be obtained by

$$P(Y_{n+1,i} = y|\mathscr{F}_n^i, \mathbf{X}; \hat{\boldsymbol{\alpha}}_i^*, \hat{\boldsymbol{\beta}}_i^*) \doteq P_i^*;$$

and the LOGS for the component-wise prediction is defined as

$$\text{LOGS}(P_i^*, y_{n+1,i}) = -\log\{P(Y_{n+1,i} = y_{n+1,i}|\mathscr{F}_n^i, \mathbf{X}; \hat{\boldsymbol{\alpha}}_i^*, \hat{\boldsymbol{\beta}}_i^*)\}, \tag{3.20}$$

where $y_{n+1,i}$ is the observed response of the i-th component at time $n+1$. See Steps 3-5 outlined in Algorithm 3.1 for the calculation of the predictive distribution.

Similarly, the joint predictive distribution for the m-dimensional vector $\mathbf{Y}_{n+1}$ at $t = n+1$

can be obtained by

$$P(\boldsymbol{Y}_{n+1} = \boldsymbol{y} | \mathscr{F}_n, \mathbf{X}; \{\hat{\boldsymbol{\alpha}}^*\}_{i=1}^n, \{\hat{\boldsymbol{\beta}}^*\}_{i=1}^n, \hat{\boldsymbol{\gamma}}^*) \doteq P^*;$$

and the LOGS for the joint prediction is defined as

$$\text{LOGS}(\hat{P}^*, \boldsymbol{y}_{n+1}) = -\log\{P(\boldsymbol{Y}_{n+1} = \boldsymbol{y}_{n+1} | \mathscr{F}_n, \mathbf{X}; \{\hat{\boldsymbol{\alpha}}^*\}_{i=1}^n, \{\hat{\boldsymbol{\beta}}^*\}_{i=1}^n, \hat{\boldsymbol{\gamma}}^*)\}. \tag{3.21}$$

3.2.7 Asymptotic Properties

In this section we establish the asymptotic properties of the proposed estimator when the uDvine(p_i, $\boldsymbol{\alpha}_i$), $i = 1, \ldots, m$ and C_m are all correctly specified. Given the two-stage estimator proposed in Section 3.2.5, we first establish the asymptotic properties of $\{\hat{\boldsymbol{\alpha}}_i\}$ and $\{\hat{\boldsymbol{\beta}}_i\}$ for each $i = 1, \ldots, m$ separately and then obtain the asymptotic properties of $\hat{\boldsymbol{\gamma}}$.

3.2.7.1 Asymptotic Properties of $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^m$ and $\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^m$

Let $\boldsymbol{\theta}_i = (\boldsymbol{\alpha}_i^T, \boldsymbol{\beta}_i^T)^T$ denote the marginal parameter vector with dimension k, which is the sum of dimension of $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$, and $\boldsymbol{\theta}_{i0}$ denote the true value, $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}_i$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \cdots; \boldsymbol{\theta}_m)$ is the vector of marginal parameters with dimension km. Let $f_i(y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)$ denote the conditional PMF of Y_{ti} given $\boldsymbol{X}$ and $\mathscr{F}_{t-1}^i$. We assume the following conditions to establish the asymptotic properties of $\hat{\boldsymbol{\theta}}_i = (\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i)$, for any $i = 1, \ldots, m$:

C1. (i) $\boldsymbol{\Theta_i}$ is a compact subset of $\mathscr{R}^k$; (ii) $f_i(y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)$ is continuous on $\boldsymbol{\Theta_i}$ a.s.; (iii) $E[l_{1i}(\boldsymbol{\theta_i}; |\{y_{ti}\}_{t=1}^n, \boldsymbol{X})]$ has identifiably unique maximizer interior to $\boldsymbol{\Theta_i}$.

C2. $E(\log f_i(Y_{ti}; \boldsymbol{\theta_i}))$ is continuous on $\boldsymbol{\Theta_i}$.

C3. $\{Y_{ti}\}_{t=1}^n$ is β-mixing with order $O(t^{-b})$ for $b > \frac{r-1}{r}$, $r > 1$.

C4. $l_{1i}(\boldsymbol{\theta_i}; |\{y_{ti}\}_{t=1}^n, \boldsymbol{X})$ is continuously differentiable of order 2.

C5. (i) $\nabla E[f_i(Y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)] = E[\nabla f_i(Y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)] < \infty$; (ii) $\nabla^2 E[f_i(Y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)] = E[\nabla^2 f_i(Y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)] < \infty$; (iii) $E[\nabla^2 f_i(Y_{ti} | \boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)]$

is continuous on $\boldsymbol{\Theta_i}$.

C6. Let $A = \nabla^2 f_i(Y_{ti}|\boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)$, then A is $O(1)$ and negative definite uniformly in n.

C7. $E[\sup_{\boldsymbol{\theta_i} \in \boldsymbol{\Theta_i}} f_i(Y_{ti}|\boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)^2] < \infty$; $E[\sup_{\boldsymbol{\theta_i} \in \Theta_i} |\nabla^2 f_i(Y_{ti}|\boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)|^2] < \infty$.

Theorem 3.1: Suppose that Assumption B1 and conditions C1 - C7 hold for the ith component time series, we have: (1) $\hat{\boldsymbol{\theta}}_{\boldsymbol{i}} \xrightarrow{P} \boldsymbol{\theta}_{i0}$; (2) $\sqrt{n}(\hat{\boldsymbol{\theta}}_{\boldsymbol{i}} - \boldsymbol{\theta}_{i0}) \xrightarrow{D} N(0, A_{i0}^{-1} B_{i0} A_{i0}^{-1})$, where $A_{i0} = E[\nabla^2 \log f_i(Y_{ti}; \boldsymbol{\theta}_{i0})]$ and $B_{i0} = Var(\nabla \log f_i(Y_{ti}; \boldsymbol{\theta}_{i0}))$.

In fact, Condition C3 is easy to be satisfied. Beare (2012, 2010) proved that series generated from most of the Archimedean copulas and t copula satisfy this property. Chen et al. (2009) established this property for t Copula while Chen and Fan (2006) proved this property for Gaussian Copula.

3.2.7.2 Asymptotic Properties of $\hat{\boldsymbol{\gamma}}$

After showing the asymptotic property of $\hat{\boldsymbol{\theta}}_i$ for $i = 1, \ldots, n$, we focus on the asymptotic property of the second stage estimator $\hat{\boldsymbol{\gamma}}$. Compared to $\hat{\boldsymbol{\theta}}$, $\hat{\boldsymbol{\gamma}}$ is obtained based on a log-likelihood function that depends on the extra finite-dimensional estimators $\{\hat{\boldsymbol{\theta}}_i\}_{i=1}^m$. The presence of the extra $\{\hat{\boldsymbol{\theta}}_i\}_{i=1}^m$ is the main difference between the setting of $\hat{\boldsymbol{\gamma}}$ and the setting of $\hat{\boldsymbol{\theta}}_i$, and this will make the variance of $\hat{\boldsymbol{\gamma}}$ larger than that from joint estimation.

Here we list the conditions for the asymptotic properties of the second-stage estimator $\hat{\boldsymbol{\gamma}}$:

A1. (i) C_m is absolutely continuous with respect to Lebesgue measure on $[0,1]^m$; (ii) $E[\sup_{\boldsymbol{\theta}} |\log f_i(Y_{ti}|\boldsymbol{X}, \mathscr{F}_{t-1}; \boldsymbol{\theta}_i)|] < \infty$ and $\mathrm{E}\left[\sup_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}, \boldsymbol{\theta} \in \boldsymbol{\Theta}_\delta} |\log g(\boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\theta}, \boldsymbol{\gamma})|\right] < \infty$.

A2. $\boldsymbol{\Theta}$ and $\boldsymbol{\Gamma}$ are compact subsets of $\mathscr{R}^{km}$ and $\mathscr{R}^{k^*}$.

A3. $\{\boldsymbol{Y}_t\}_{t=1}^n$ is β-mixing with order $O(t^{-b})$ for $b > \frac{r-1}{r}$, $r > 1$.

A4. $E[l_2(\boldsymbol{\gamma}|\{\boldsymbol{Y}_t\}_{t=1}^n)]$ has identifiably unique maximizer interior to $\boldsymbol{\Gamma}$.

A5. f_i is continuous in $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}_i$ and $E[g(\boldsymbol{Y_t}|\mathscr{F}_{t-1}; \boldsymbol{\theta}, \boldsymbol{\gamma})]$ is Lipschitz continuous in $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$.

A6. $g(\boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \boldsymbol{\theta}, \boldsymbol{\gamma})$ is uniformly continuous in $\boldsymbol{\theta}$, uniformly over $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$.

A7. $\partial \log g\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\theta},\boldsymbol{\gamma}\right)/\partial\boldsymbol{\gamma}$ is Lipschitz continuous in $\boldsymbol{\gamma}$.

A8. (i) Let $H_{\boldsymbol{\gamma}} = -\mathrm{E}\left[\partial^2 \log g\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\theta},\boldsymbol{\gamma}\right)/\partial\boldsymbol{\gamma}^2\right]$, then $H_{\boldsymbol{\gamma}}$ is positive definite; (ii) Let $H_{i\boldsymbol{\theta}} = -\mathrm{E}\left[\partial^2 \log f_i(Y_{ti}|\boldsymbol{X},\mathscr{F}_{t-1};\boldsymbol{\theta}_i)/\partial\boldsymbol{\theta}_i\partial\boldsymbol{\theta}_i'\right]$ and $H_{\boldsymbol{\theta}} = \mathrm{diag}(H_{1\boldsymbol{\theta}},\ldots,H_{m\boldsymbol{\theta}})$, then $H_{\boldsymbol{\theta}}$ is positive definite; (iii) $f^{\#}(\boldsymbol{Y}_t) = (f_1(Y_{t1}|\boldsymbol{X},\mathscr{F}_{t-1};\boldsymbol{\theta}_1),\ldots,f_m(Y_{tm}|\boldsymbol{X},\mathscr{F}_{t-1};\boldsymbol{\theta}_m))$ is differentiable in $\boldsymbol{\theta}^*$.

A9. (i) The derivatives of $\partial \log g\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\theta},\boldsymbol{\gamma}\right)/\partial\boldsymbol{\gamma}$ are uniformly continuous in $(\boldsymbol{\theta},\boldsymbol{\gamma})$; (ii) The following limit hold in probability:

$$P_{n1} = n^{-1}\sum_{t=p+1}^{n}\left.\frac{\partial^2 \log g\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{\theta},\boldsymbol{\gamma}\right)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\gamma}}\right|_{(\boldsymbol{\theta},\boldsymbol{\gamma})=(\boldsymbol{\theta}_0,\boldsymbol{\gamma}_0)} = P_1 + o_p(1).$$

Theorem 3.2: Suppose that Assumption B and conditions A1 - A9 hold, we have: (1) $\hat{\boldsymbol{\gamma}} \xrightarrow{P} \boldsymbol{\gamma}_0$; (2) $\sqrt{n}(\hat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0) \xrightarrow{D} N\left(0, H_{\boldsymbol{\gamma}_0}^{-1}\Omega_{\boldsymbol{\gamma}}^{\#}H_{\boldsymbol{\gamma}_0}^{-1}\right)$, where

$$\Omega_{\boldsymbol{\gamma}}^{\#} = \lim_{n\to\infty}\mathrm{Var}\left(\frac{1}{\sqrt{n}}\sum_{t=1}^{n}\left(g_{\boldsymbol{\gamma}}\left(\boldsymbol{\theta}_0,\boldsymbol{\gamma}_0,\boldsymbol{Y_t}|\mathscr{F}_{t-1}\right) + P_1 H_{\boldsymbol{\theta}}^{-1}\frac{\partial \log f^{\#}\left(Y_t,\boldsymbol{\theta}_0\right)}{\partial\boldsymbol{\theta}}\right)\right).$$

Here $\hat{\boldsymbol{\gamma}}$ depends on the first-stage estimators of $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^m$ and $\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^m$, whose variations are reflected in the second term of $\Omega_{\boldsymbol{\gamma}}^{\#}$.

The asymptotic variances in Theorems 3.1 and 3.2 have complicated forms, which are difficult to calculate directly. For uncertainty quantification, one practical solution is to adopt the parametric bootstrap. The procedure is as follows. Based on the estimated $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\beta}}$ for each univariate component and $\hat{\boldsymbol{\gamma}}$ for the cross-sectional copula, we simulate the multivariate time series of length n by B times. For each of the B parametric bootstrap sample, we obtain a bootstrapped estimator of parameters and we use the sample variance of the B bootstrapped estimators as an estimator for the variance of $\hat{\boldsymbol{\alpha}}$, $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\gamma}}$.

3.2.8 Selection of Pair-copulas for the uDvine

To implement a CDvine, we need to specify the order p and the pair-copulas $\{C_{s,t}\}$ for each component uDvine and the cross-sectional copula $C_m(\cdot)$. In our implementation, we assume

that the pair-copulas are the same within each tree, but they are allowed to vary across trees.

For computational feasibility, we adapt the tree-by-tree sequential selection procedure as described in Shi and Yang (2018) for each of the component time series separately. We first fit a rough model. We assume it is a 1-st order D-vine, and we select the appropriate copula from a set of candidates, including e.g., Gaussian, t, Clayton, Gumbel and Joe copulas. For each component $i = 1, \ldots, m$, we estimate the parameters of the candidate copulas and select the one that gives the smallest BIC, defined as

$$\text{BIC}_i = k\log(n) - 2l_{1i}(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i | \{y_{ti}\}_{t=1}^n, \boldsymbol{X}) \text{ for } i = 1, \ldots, m,$$

where k is the dimensions of pair-copula parameters $\boldsymbol{\alpha}_i$ and marginal parameters $\boldsymbol{\beta}_i$, and l_{1i} is defined in (3.14). Then we fix our selected copula and the estimated parameters in the first tree, and we select the appropriate copula from candidates and estimate its parameters for the second tree. We continue this process until an independent copula is selected for the p-th tree, then we truncate the uDvine after the p-th tree. Through this process, we select the pair-copulas and the order for uDvine at the same time for each component of the time series separately.

After selecting the pair-copulas for each uDvine, we then select the cross-sectional copula. Similarly, we consider a set of candidates, e.g., Gaussian, t, Clayton, Gumbel and Joe copulas, and we select the one that can produce the smallest BIC, defined as:

$$\text{BIC} = k^* \log(n) - 2l_2\left(\hat{\boldsymbol{\gamma}} | \{\boldsymbol{y}_t\}_{t=1}^n\right),$$

where k^* is the dimension of $\boldsymbol{\gamma}$, and l_2 is defined in (3.15).

3.3 Simulation Studies

In this section we first use simulation to assess the performance of the sequential selection procedure described in Section 3.2.8. Then we compare the finite sample performance of the

proposed method with existing approaches for analyzing count time series data. Finally we assess the performance of the bootstrap variance estimator. We generate bivariate time series $\boldsymbol{Y} = \{\boldsymbol{Y_t} = \{Y_{ti}\}_{i=1}^{2}\}_{t=1}^{n}$ from CDvine and Poisson regression model $Y_{ti}|\boldsymbol{X} \sim Poisson(\lambda_{ti})$, where $\log(\lambda_{ti}) = \beta_1 + \beta_2 X_{ti}$. The CDvine is constructed with uDvine(p_1, $\boldsymbol{\alpha}_1$) and uDvine(p_2, $\boldsymbol{\alpha}_2$) and the cross-sectional copula is $C_2(\cdot;\gamma)$. Below are the steps used for data generation. Each time we generate $n+1$ observations and the last one is taken as the test point.

Step 1: Generate an $i.i.d.$ bivariate sequence from $C_2(\cdot,\cdot;\gamma)$, denoted as $\{v_{t1}, v_{t2}\}_{t=1}^{n}$, and $\{X_t, t = 1,\ldots,n+1\}$ from $N(0,1)$.

Step 2: For $i = 1,2$, set $y_{ti} = F_i^{-1}\left(v_{ti} \mid \mathscr{F}_{t-1}^{i}, \boldsymbol{X}; \boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\right)$ for $t = 1,\ldots,n+1$, where F_i^{-1} is defined in (3.17).

3.3.1 Copula Selection for the uDvine

In this section we consider a univariate time series. We assess the performance of the proposed sequential selection procedure for selecting p and pair-copulas involved in uDvine. We simulate data from two models: uDvine with order $p = 1$ and $p = 2$. When $p = 1$, we set T_1 as one of the following four pair-copulas: Gaussian ($\alpha_1 = 0.7$), t_2 ($\alpha_2 = 0.7$), Clayton ($\alpha_3 = 2$) and Joe ($\alpha_4 = 2.9$). These copula parameters are chosen to make Kendall's τ equal 0.5. When $p = 2$, we consider three cases. For the first uDvine(2), we set T_1 as Joe ($\alpha_1^1 = 2.9$) copula and T_2 as Frank ($\alpha_2^1 = 2.9$) copula. For the second uDvine(2), we set T_1 as Gaussian ($\alpha_2^2 = 0.7$) copula and T_2 as Clayton ($\alpha_2^2 = 0.9$) copula. For the third uDvine, we set T_1 as t_2 ($\alpha_1^3 = 0.7$) copula and T_2 as Gumbel ($\alpha_2^3 = 1.4$) copula. The parameters of all the pair-copulas are specified to make the Kendall's τ of T_1 equal 0.5 and that of T_2 equal 0.3.

We consider the candidate pool of pair-copulas as Gaussian, t, Clayton, Gumbel, Joe and independent copulas. The same candidate pool is considered in the entire simulation section. Note that we only consider these six candidate families here, but in practice any

Table 3.1: Percentages of times the uDvine order and pair-copulas are correctly chosen for simulation data generated from a uDvine model with $p = 1$.

		Percentages of correct selection	
True pair-copula	n	Order p	Pair-copula
Gaussian	500	88.8	85.2
	1000	97.6	91.4
t_2	500	91.4	99.4
	1000	97.4	100.0
Clayton	500	88.8	100.0
	1000	98.0	100.0
Joe	500	90.0	98.8
	1000	97.0	99.8

Table 3.2: Percentages of times the uDvine order and pair-copulas are correctly chosen for simulation data generated from a uDvine model with $p = 2$ and three combinations of pair-copulas in T_1 and T_2.

		Percentages of correct selection		
True pair-copula in T_1, T_2	n	Order p	Pair-copula in T_1	Pair-copula in T_2
Joe, Frank	500	90.8	74.2	98.0
	1000	98.0	84.4	100.0
Gaussian, Clayton	500	85.8	97.0	77.8
	1000	96.0	100.0	90.2
t_2, Gumbel	500	87.6	98.8	74.2
	1000	96.0	100.0	94.2

copula family can be considered. For each uDvine(2), we perform the sequential selection procedure to select the pair-copulas. We consider two sample sizes $n = 500$ and 1000. For each scenario, we repeat the simulation 500 times. Tables 3.1 and 3.2 summarize the percentages of times that the pair-copulas are correctly chosen for data simulated from uDvine models with $p = 1$ and $p = 2$, respectively. Results suggest that the sequential selection procedure performs well for the selection of both order p and the pair-copula, and the performance is improving with the increase of sample size n.

3.3.2 Comparison with Multivariate Poisson Autoregression Model (MPAR)

Fokianos et al. (2020) proposed the following MPAR model for multivariate count time

series. Assume that $\{\boldsymbol{Y}_t = (Y_{ti}), t = 1,2,\ldots,n, i = 1,2,\ldots,m\}$ denotes an m-dimensional count time series. Let $\{\boldsymbol{\lambda}_t = (\lambda_{ti}), t = 1,2,\ldots,n, i = 1,2,\ldots,m\}$ be the corresponding m-dimensional intensity process and $\mathscr{F}_t^{\boldsymbol{Y},\boldsymbol{\lambda}}$ be the $\sigma-$field generated by $\{\boldsymbol{Y}_0,\ldots,\boldsymbol{Y}_t,\boldsymbol{\lambda}_0\}$ with $\boldsymbol{\lambda}_0$ being an m-dimensional vector denoting the starting value of $\{\boldsymbol{\lambda}_t\}$. The following log-linear model is used to capture the dependence:

$$Y_{ti}|\mathscr{F}_t^{\boldsymbol{Y},\boldsymbol{\lambda}} \quad \text{is marginally Poisson}\,(\lambda_{ti}),$$

$$\boldsymbol{v}_t = \boldsymbol{d} + \boldsymbol{A}\boldsymbol{v}_{t-1} + \boldsymbol{B}\log(\boldsymbol{Y}_{t-1} + \boldsymbol{1}_m) + \boldsymbol{C}\boldsymbol{X}_{t-1},$$

where $\boldsymbol{v}_t \equiv \log\boldsymbol{\lambda}_t$ is defined componentwise and $\boldsymbol{1}_m$ denotes the m-dimensional vector consisting of ones.

We consider three different cases for generating bivariate time series data. In Cases 1-2, the data are generated from the CDvine(1) and CDvine(2) models. In Case 3, the data are generated from the MPAR model with

$$\boldsymbol{A} = \begin{pmatrix} 0.3 & 0.05 \\ 0.1 & 0.25 \end{pmatrix}, \quad \boldsymbol{B} = \begin{pmatrix} 0.1 & 0.05 \\ 0.1 & 0.4 \end{pmatrix}, \quad \boldsymbol{C} = \begin{pmatrix} 0.01 & 0.05 \\ 0.01 & 0.04 \end{pmatrix}, \quad \boldsymbol{d} = \begin{pmatrix} 0.3 \\ 0.1 \end{pmatrix}.$$

For Case 1, we set T_1 as Clayton ($\alpha_1 = 2$) copula for the first component uDvine(1) and Joe ($\alpha_1 = 2.86$) copula for the second component uDvine(1). For Case 2, we set T_1 as Gaussian ($\alpha_1 = 0.7$) copula and T_2 as Joe ($\alpha_2 = 1.77$) copula for the first component uDvine(2), and T_1 as Frank ($\alpha_1 = 5.74$) copula and T_2 as Clayton ($\alpha_2 = 0.86$) copula for the second component uDvine(2). The parameters of all the pair-copulas are specified to make the Kendall's τ of T_1 equal 0.5 and that of T_2 equal 0.3. Finally, we set the cross-sectional copula $C_2(\cdot)$ as Gumbel copula with $\gamma = 2$ for both Case 1 and Case 2. For three cases we analyze the data by fitting the proposed CDvine model and the MPAR model. The pair-copulas and order of each uDvine in CDvine are chosen by the sequential selection procedure, and the cross-sectional copula is chosen as the one giving the smallest BIC. To compare the prediction accuracy of

two methods, we construct 90% prediction intervals for each component time series and report the average LOGS for both component-wise and joint predictions.

Table 3.3 summarizes the coverage percentages and average lengths of 90% component-wise prediction intervals from three methods: CDvine, MPAR and True. Here the True method refers to the predictions based on random draws from the true model used to generate the data with the true parameters. We take the first n observations as training data and the last observation as the testing data. The prediction intervals are constructed by using numerical simulation as described in Section 3.2.6. In all three Cases, the nominal level of 90% is preserved by the predictive intervals from all three methods. We note that all coverage probabilities are above 90%, and this is due to the discreteness of the data, which makes it impossible to construct intervals with exact coverage. Results show that in Cases 1-2 where the data are generated from CDvine, MPAR has more severe over coverage, which is associated with much wider prediction intervals. In Case 3 where the data are generated from the MPAR model, prediction intervals from CDvine have comparable coverage and average lengths to those from MPAR.

Table 3.4 shows the average LOGS for joint prediction given by MPAR and CDvine in different cases. Results show that the CDvine gives significantly lower LOGS than MPAR when data come from CDvine, while the two give similar LOGS in Case 3 where the data are generated from the MPAR model. Results in Tables 3.3 and 3.4 both demonstrate the flexibility of the proposed CDvine model for analyzing multivariate count data.

3.3.3 Performance of Bootstrap Variance

We assess the performance of the parametric bootstrap method for estimating the variance of the two-stage estimator. The data are generated in the same way as in Cases 1-2 (based on CDvine(1) and CDvine(2) models) in Section 3.3.2. We set $B = 500$ and set all the parameters to be the same as the ones in Section 3.2. Tables 3.5 summarizes the average bootstrap variance and Monte Carlo variance of the first-stage marginal parameters estimators $\{\hat{\beta}_i\}_{i=1}^{2}$

Table 3.3: Coverage percentages and average lengths of 90% prediction intervals constructed by fitting the MPAR and CDvine models for Cases 1-3.

			Coverage percentages			Average lengths		
Case	n	Component	True	MPAR	CDvine	True	MPAR	CDvine
1	500	1	94.6	98.0	94.2	32.3	43.2	32.6
		2	97.4	99.4	97.6	28.9	43.0	29.0
	1000	1	96.0	97.8	95.6	33.2	44.5	33.4
		2	96.0	98.0	96.0	29.0	44.2	29.1
2	500	1	96.6	99.4	96.0	27.4	44.2	26.9
		2	96.0	98.8	95.8	30.1	44.4	30.1
	1000	1	97.0	99.8	96.6	27.2	43.7	26.7
		2	96.0	98.4	95.8	29.7	43.9	30.1
3	500	1	98.4	98.0	97.2	49.8	47.9	46.1
		2	93.0	95.0	92.6	52.3	54.8	50.7
	1000	1	98.0	97.8	96.6	49.5	47.1	46.4
		2	94.6	96.4	93.0	50.9	53.9	50.0

True: the coverage percentage produced by the true model with true parameters; MPAR: multivariate Poisson autoregression; CDvine: copula-linked D-vine.

and uDvine parameters estimators $\{\hat{\alpha}_i\}_{i=1}^{2}$. Tables 3.6 summarizes the average bootstrap variance and Monte Carlo variance of the estimator $\hat{\gamma}$ in cross-sectional copula C_2. Results show that the average bootstrap variances are quite close to Monte Carlo variances for all three estimators.

3.4 Real Data Analysis

In this section, we analyze a COVID-19 mortality data from the Centers for Disease Control and Prevention: https://data.cdc.gov/Case-Surveillance/United-States-COVID-19-Cases-and-Deaths-by-State-o/9mfq-cb36. The dataset contains the daily deaths from each state of United States due to COVID-19 from January 22, 2020 to Dec 31, 2021. The data are maintained and archived by Centers for Disease Control and Prevention and are updated daily. The dataset contains 509 daily observations for each state. Since the data may not be accurate from January 2021 to May 2021 and on weekends, as many of the death might not

Table 3.4: 10× LOGS of joint prediction obtained by fitting the MPAR and CDvine models for Cases 1-3. Values in the parentheses are standard errors.

Case	n	True	MPAR	CDvine
1	500	24.8 (0.4)	28.8 (0.4)	25.1 (0.7)
	1000	25.3 (0.4)	29.1 (0.4)	25.3 (0.6)
2	500	23.4 (0.4)	27.8 (0.4)	23.7 (0.6)
	1000	23.2 (0.4)	27.7 (0.4)	23.4 (0.7)
3	500	33.9 (0.2)	34.0 (0.3)	35.0 (0.2)
	1000	34.0 (0.2)	33.9 (0.4)	34.4 (0.2)

True: the LOGS produced by the true model with true parameters; MPAR: multivariate Poisson autoregression; CDvine: Copula-linked D-vine.

have been reported, we only use the data after June 22, 2020 from weekdays, resulting in 400 observations. We study the COVID-19 daily death data from three states that are near each other: North Carolina (NC), Virginia (VA) and Maryland (MD).

Before applying our proposed CDvine method, we need to model the marginal distribution. The data contains the number of confirmed cases, and we treat it as one of the covariates. To capture the time trend, we also include the time t and its quadratic term t^2 as the covariates, and we arrive at the following Poisson regression model:

$$Y_{ti} \sim Poisson(\lambda_{ti}), \tag{3.22}$$

where

$$\log(\lambda_{ti}) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 \log(X_{t-10,i} + 1). \tag{3.23}$$

In (3.23), $X_{t-10,i}$ is the number of confirmed cases in the same location 10 business days ago, as studies show that deaths often occur one to eight weeks after the onset of COVID-19 symptoms; see e.g., Jin (2021) and Donovan et al. (2021). In this application, t, t^2 and $X_{t-10,i}$ can all be considered as continuous covariates due to their large ranges. The reason we use $\log(X_{t-10,i}+1)$ instead of $X_{t-10,i}$ is to transform both λ_t and $X_{t-10,i}$ to the same

Table 3.5: Monte Carlo variance and the average of bootstrap variances of the first-stage estimators $\{\hat{\beta}_i\}_{i=1}^2$ and $\{\hat{\alpha}_i\}_{i=1}^2$ in Cases 1-2 where data are generated from CDvine(1) and CDvine(2). Values in the parentheses are standard errors.

			Monte Carlo variance				Average bootstrap variance			
Case	n	Component	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\alpha}_1$	$\hat{\alpha}_2$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\alpha}_1$	$\hat{\alpha}_2$
1	500	1	0.20	0.31	0.15		0.16	0.27	0.21	
							(0.03)	(0.03)	(0.05)	
		2	0.22	0.35	0.19		0.24	0.38	0.22	
							(0.04)	(0.02)	(0.02)	
	1000	1	0.14	0.23	0.12		0.17	0.16	0.19	
							(0.04)	(0.04)	(0.05)	
		2	0.16	0.24	0.16		0.22	0.28	0.20	
							(0.04)	(0.03)	(0.03)	
2	500	1	0.17	0.18	0.08	0.15	0.15	0.21	0.09	0.13
							(0.02)	(0.02)	(0.02)	(0.01)
		2	0.18	0.18	0.54	0.13	0.20	0.17	0.60	0.11
							(0.02)	(0.02)	(0.05)	(0.02)
	1000	1	0.12	0.13	0.06	0.12	0.13	0.15	0.07	0.10
							(0.02)	(0.02)	(0.01)	(0.01)
		2	0.15	0.14	0.46	0.11	0.17	0.15	0.49	0.09
							(0.02)	(0.02)	(0.04)	(0.03)

scale.

Besides the proposed method CDvine, we also consider two other methods: the Poisson regression (PR) that ignores dependence, and the MPAR model. To capture the cross-sectional dependence of CDvine, we select a 3-dimentional Gaussian copula. Table 3.7 reports the estimated correlation matrix of the cross-sectional copula, and the results suggest significant correlation between three states.

We use a rolling-window validation method to assess the performance of different methods for prediction. For all methods, we update the parameter estimation progressively in a rolling-window format: we incorporate the last 40 observations as they become observed, with a window length of 160 days. We update the parameters in each method in a daily manner. We first consider component-wise prediction, that is, prediction for each state separately. Table 3.8 summarizes the average LOGS of different methods for one-day-ahead forecasts, and Table 3.9 summarizes the coverage percentages and average lengths of

Table 3.6: Monte Carlo variance and the average of bootstrap variance of the second-stage estimators $\hat{\gamma}$ in Cases 1-2 where data are generated from CDvine(1) and CDvine(2). Values in the parentheses are standard errors.

Model	n	Monte Carlo variance	Average bootstrap variance
Case 1	500	0.11	0.11
			(0.03)
	1000	0.07	0.08
			(0.03)
Case 2	500	0.10	0.11
			(0.02)
	1000	0.08	0.08
			(0.01)

Table 3.7: The estimated correlation matrix of the cross-sectional Gaussian Copula using COVID-19 data from North Carolina (NC), Virginia (VA) and Maryland (MD). Values in the parentheses are bootstrap standard errors.

	VA	MD	NC
VA	1	0.18 (0.01)	0.08 (0.01)
MD		1	0.08 (0.01)
NC			1

90% prediction intervals for one-day-ahead forecasts, for each state separately. Table 3.8 shows that the CDvine gives smaller LOGS than the other two methods in all three states, suggesting that the former leads to a more accurate predictive distribution for this data. Table 3.9 shows that CDvine gives prediction intervals with higher coverage percentages than PR and smaller lengths than MPAR.

We also obtain the joint LOGS, which evaluates the performance of the predictive distribution of three-dimensional responses for one-day ahead prediction. The average joint LOGS across testing points from PR, MPAR and CDvine are 13.85 (1.91), 11.32 (1.58) and 10.21 (1.49), respectively. Among three methods, the CDvine gives the smallest joint LOGS, suggesting that it leads to more accurate joint predictive distributions of COVID-19 deaths from the three states jointly.

Table 3.8: Average LOGS for component-wise prediction of COVID-19 deaths in North Carolina (NC), Virginia (VA) and Maryland (MD), separately, from three methods.Values in the parenthesis are the standard errors.

State	PR	MPAR	CDvine
VA	5.24 (0.43)	4.20 (0.31)	3.71 (0.23)
MD	4.48 (0.37)	3.83 (0.45)	3.16 (0.34)
NC	3.92 (0.33)	3.91 (0.19)	3.92 (0.36)

PR: Poisson regression ignoring the dependence; MPAR: multivariate Poisson autoregression; CDvine: Copula-linked D-vine.

Table 3.9: Coverage percentages and average lengths of 90% component-wise prediction intervals for the COVID-19 data in NC, VA and MD from three methods. Values in the parenthesis are the standard errors.

	Coverage percentages			Average lengths		
State	PR	MPAR	CDvine	PR	MPAR	CDvine
VA	62.50	98.75	87.50	15.67 (0.75)	24.32 (1.66)	20.70 (1.30)
MD	63.75	98.75	93.75	13.18 (0.20)	19.23 (1.39)	12.60 (0.75)
NC	77.50	98.75	90.00	16.85 (1.03)	28.00 (2.08)	22.07 (1.92)

PR: Poisson regression ignoring the dependence; MPAR: multivariate Poisson autoregression; CDvine: Copula-linked D-vine,

3.5 Discussion

In this section we proposed a copula-linked D-vine model and estimation procedure for analyzing multivariate time series of counts. The framework allows us to simultaneously capture temporal and cross-sectional dependence using copula, and the vine copula leads to more flexibility and convenience in computation. We first studied a univariate time series model: uDvine, that extends the first-order copula-based Markov property to Markov process of an arbitrary-order. Through the pair-copula construction, the uDvine provides flexible specifications for the marginal distribution and temporal dependence of univariate time series. To generalize uDvine to the multivariate context, we designed the CDvine by linking multiple uDvines via a cross-sectional copula. Compared to existing multivariate time series models, the CDvine shows more flexibility in the numerical studies. The application

to a COVID-19 data showed that the CDvine method leads to more accurate predictions than existing methods in terms of both component-wise and joint predictions at future time points.

One limitation of the CDvine model framework is that the marginal distribution of the response is modeled through a parametric regression model. In the next chapter we will investigate how to relax this assumption and extend the CDvine framework to analyze spatio-temporal count data.

3.6 Proof of Propositions and Theorems

3.6.1 Proof of Proposition 3.1

Proof. We show the Markov property for continuous case under the regression setup as the notation is simpler. The proof for discrete case is quite the same. By Aas et al. (2009), we have

$$\begin{aligned} & f\left(y_t \mid y_{t-1}, \ldots, y_1, \boldsymbol{x}\right) \\ =& f\left(y_t\right) \prod_{s=1}^{t-1} c_{s,t}\left(F_{s \mid (s+1):(t-1)}\left(y_s \mid y_{s+1}, \ldots, y_{t-1}, \boldsymbol{x}\right), F_{t \mid (s+1):(t-1)}\left(y_t \mid y_{s+1}, \ldots, y_{t-1}, \boldsymbol{x}\right)\right), \end{aligned} \tag{3.24}$$

where $c_{s,t}(\cdot)$ denotes the bivariate copula in T_{t-s}.

By the definition of the uDvine(p), all bivariate copulas beyond tree p are independent copulas. Thus, we have for $t > p$,

$$\begin{aligned} & f\left(y_t \mid y_{t-1}, \ldots, y_1, \boldsymbol{x}\right) \\ =& f\left(y_t\right) \prod_{s=t-p}^{t-1} c_{s,t}\left(F_{s \mid (s+1):(t-1)}\left(y_s \mid y_{s+1}, \ldots, y_{t-1}, \boldsymbol{x}\right), F_{t \mid (s+1):(t-1)}\left(y_t \mid y_{s+1}, \ldots, y_{t-1}, \boldsymbol{x}\right)\right). \end{aligned} \tag{3.25}$$

That is, the conditional distribution of $Y_t|\boldsymbol{X}$ given the history only depends on the past p

observations $(Y_{t-1},\ldots,Y_{t-p}|\boldsymbol{X})$, thus $\{Y_t|\boldsymbol{X}\}$ is a p-order Markov process.

□

3.6.2 Proof of Theorem 3.1

Proof. We first introduce a statement in Longla and Peligrad (2012).

Proposition: The mixing coefficients of a Markov chain $\{X_i\}_{i\in Z}$ generated by a given copula C and marginal distribution uniform on $[0,1]$, are larger than or equal to those of a Markov chain generated by the same copula and another marginal distribution F, not necessarily continuous.

Proof. According to the definition of the $\beta-$mixing coefficients, we have the $\beta-$mixing coefficients between 2 $\sigma-$fields $\mathscr{A}$, $\mathscr{B}$ is

$$\beta(\mathscr{A},\mathscr{B}) = \frac{1}{2}\sup_{\{A_i\},\{B_j\}}\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\Pr\left(A_i\cap B_j\right)-\Pr\left(A_i\right)\Pr\left(B_j\right)\right|,$$

where the supremum is taken over all positive integers n and m, and all finite partitions $\{A_i\}$, $\{B_j\}$ with $A_i\in\mathscr{A}$ and $B_j\in\mathscr{B}$.

Now we consider a stationary sequence $\{X_n\}_{n\in Z}$. Let $\mathscr{P}=\sigma\{X_k,K\leq 0\}$ be the information provided by the past of the process and $\mathscr{F}_n=\sigma\{X_k,k\geq n\}$ describe the future after n steps. Then define $\beta_n=\beta\{\mathscr{P},\mathscr{F}_n\}$ be the $\beta-$mixing coefficients of $\{X_n\}_{n\in Z}$.

Let $\{U_i\}_{i\in Z}$ be the Markov chain generated by a given copula C and marginal distribution uniform on $[0,1]$, $\mathscr{P}_1=\sigma\{U_k,K\leq 0\}$, $\mathscr{F}_{1n}=\sigma\{U_k,k\geq n\}$, and $\{X_i\}_{i\in Z}$ be the Markov chain generated by the same copula and another marginal distribution F, $\mathscr{P}_2=\sigma\{X_k,K\leq 0\}$, $\mathscr{F}_{2n}=\sigma\{X_k,k\geq n\}$. So the $\beta-$mixing coefficients of $\{U_n\}_{n\in Z}$ and $\{X_n\}_{n\in Z}$ can be expressed as

$$\beta_{1n}=\beta(\mathscr{P}_1,\mathscr{F}_{1n}) = \frac{1}{2}\sup_{\{A_{1i}\},\{B_{1j}\}}\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\Pr\left(A_{1i}\cap B_{1j}\right)-\Pr\left(A_{1i}\right)\Pr\left(B_{1j}\right)\right|,$$

where the supremum is taken over all positive integers n and m, and all finite partitions $\{A_{1i}\}$, $\{B_{1j}\}$ with $A_{1i} \in \mathscr{P}_1$ and $B_{1j} \in \mathscr{F}_{1n}$ and

$$\beta_{2n} = \beta(\mathscr{P}_2, \mathscr{F}_{2n}) = \frac{1}{2} \sup_{\{A_{2i}\},\{B_{2j}\}} \sum_{i=1}^{n} \sum_{j=1}^{m} \left| \Pr\left(A_{2i} \cap B_{2j}\right) - \Pr\left(A_{2i}\right) \Pr\left(B_{2j}\right) \right|,$$

where the supremum is taken over all positive integers n and m, and all finite partitions $\{A_{2i}\}$, $\{B_{2j}\}$ with $A_{2i} \in \mathscr{P}_2$ and $B_{2j} \in \mathscr{F}_{2n}$.

Note that $\mathscr{P}_1 \supset \mathscr{P}_2$ and $\mathscr{F}_{1n} \supset \mathscr{F}_{2n}$, so we get $\beta_{1n} \geq \beta_{2n}$, and condition C3 is satistied. Beare (2010) proved that series generated from most of the Archimedean copulas satisfy this property. Chen et al. (2009) established this property for t Copula while Chen and Fan (2006) proved this property for Gaussian Copula. □

Using Corollary 1 in Pötscher and Prucha (1989), $\{\log f_t(X_t;\theta)\}$ obeys the strong ULLN, which allows for the interchange of expectation and differentiation, which is condition C5. By Theorem 2.8.1 in Lehmann (2004), the double array $\{n^{-1/2}s_t = n^{-1/2} \bigtriangledown \log f_t(X^t;\boldsymbol{\theta}_0)\}$ obeys the central limit theorem with covariance matrix $\{B_{i0}\}$. The proof then follows directly from White (1996) using the standard maximum likelihood theory. □

3.6.3 Proof of Theorem 3.2

Proof. (i) For this second step estimation, we need to verify that $\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}} \left\| \widehat{Q}_n(\boldsymbol{\theta}) - Q(\boldsymbol{\theta}) \right\| = o_p(1)$, where

$$\widehat{Q}_n(\boldsymbol{\theta}) = \frac{1}{n} \sum_{t=2}^{n} g(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}, \boldsymbol{Y_t} | \mathscr{F}_{t-1}), \quad Q(\boldsymbol{\theta}) = \mathrm{E}\left[g(\boldsymbol{\theta}_0, \boldsymbol{\gamma}, \boldsymbol{Y_t} | \mathscr{F}_{t-1}\right),$$

and $g(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{Y_t} | \mathscr{F}_{t-1}) = \log f\left(\boldsymbol{Y_t} | \mathscr{F}_{t-1}, \boldsymbol{\theta}, \boldsymbol{\gamma}\right)$.

Denote

$$Q_n(\boldsymbol{\gamma}) = \frac{1}{n} \sum_{t=2}^{n} g(\boldsymbol{\theta}_0, \boldsymbol{\gamma}, \boldsymbol{Y_t} | \mathscr{F}_{t-1}).$$

By (1) Assumption A2: compactness of $\boldsymbol{\Gamma}$; (2) Assumption A3: weak dependence of $\boldsymbol{Y_t}$;

(3) Assumption A7: $g(\cdot)$ is continuous in $\boldsymbol{\theta}$; (4) Assumption A1(ii): E[$\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma},\boldsymbol{\theta}\in\boldsymbol{\Theta}_\delta}|g(\boldsymbol{\theta},\boldsymbol{\gamma}$, $\boldsymbol{Y_t}|\mathscr{F}_{t-1})|]<\infty$, we have

$$\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}|Q_n(\boldsymbol{\theta})-Q(\boldsymbol{\theta})|=o_p(1).$$

Thus, it suffices to show that

$$\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}\left|\widehat{Q}_n(\boldsymbol{\theta})-Q_n(\boldsymbol{\theta})\right|=o_p(1).$$

Under the assumption A6 that $g(\boldsymbol{\theta},\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1})$ is uniformly continuous in $\boldsymbol{\theta}$, uniformly over $\boldsymbol{\gamma}\in\boldsymbol{\Gamma}$, we can show that, for all sequences $\{\varepsilon_n\}$ with $\varepsilon_n=o(1)$,

$$\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma},|\theta-\theta_0\|\leq\varepsilon_n}\left|\frac{1}{n}\sum_{t=2}^{n}[g(\boldsymbol{\theta},\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1})-g(\boldsymbol{\theta}_0,\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1})]\right|=o_p(1).$$

Notice that

$$\begin{aligned}&\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}\left|\widehat{Q}_n(\boldsymbol{\theta})-Q_n(\boldsymbol{\theta})\right|\\ \leq&\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma},|\theta-\theta_0\|\leq\varepsilon_n}\left|\frac{1}{n}\sum_{t=2}^{n}[g(\boldsymbol{\theta},\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1})-g(\boldsymbol{\theta}_0,\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1})]\right|\\ =&o_p(1).\end{aligned}$$

Thus $\sup_{\boldsymbol{\gamma}\in\boldsymbol{\Gamma}}\left\|\widehat{Q}_n(\boldsymbol{\theta})-Q(\boldsymbol{\theta})\right\|=o_p(1)$, and the consistency is proven.

(ii) The likelihood function is given by

$$\widehat{Q}_n(\boldsymbol{\gamma})=\frac{1}{n}\sum_{t=p+1}^{n}g(\hat{\boldsymbol{\theta}},\boldsymbol{\gamma},\boldsymbol{Y_t}|\mathscr{F}_{t-1}).$$

Let $\sqrt{n}(\boldsymbol{\gamma}-\boldsymbol{\gamma}_0)=\Delta_2$, $\sqrt{n}\left(\hat{\boldsymbol{\theta}}-\boldsymbol{\theta}_0\right)=\Delta_{1n}$ and $\sqrt{n}(\hat{\boldsymbol{\gamma}}-\boldsymbol{\gamma}_0)=\Delta_{2n}$, then we may re-write

the criterion function $\widehat{Q}_n(\boldsymbol{\theta})$ as

$$V_n(\Delta_2) = \frac{1}{n}\sum_{t=p+1}^{n} g(\boldsymbol{\theta}_0 + n^{-1/2}\Delta_{1n}, \boldsymbol{\gamma}_0 + n^{-1/2}\Delta_2, \boldsymbol{Y_t}|\mathscr{F}_{t-1}),$$

and $\min_{\boldsymbol{\gamma}} \widehat{Q}_n(\boldsymbol{\gamma})$ is equivalent to $\min_{\Delta_2} V_n(\Delta_2)$.

The first order condition corresponding to the minimization of $V_n(\Delta_2)$ w.r.t. Δ_2 is given by

$$\left.\frac{\partial V_n(\Delta_2)}{\partial \Delta_2}\right|_{\Delta_2=\Delta_{2n}} = 0.$$

Expanding $\left.\frac{\partial V_n(\Delta_2)}{\partial \Delta_2}\right|_{\Delta_2=\Delta_{2n}}$ around $\Delta_2 = 0$, we have

$$\begin{aligned} 0 &= \left.\frac{\partial V_n(\Delta_2)}{\partial \Delta_2}\right|_{\Delta_2=\Delta_{2n}} \\ &= \frac{1}{n}\sum_{t=2}^{n} g_{\boldsymbol{\gamma}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}_0, \boldsymbol{Y_t}|\mathscr{F}_{t-1}) + n^{-1/2}\left[\frac{1}{n}\sum_{t=2}^{n} g_{\boldsymbol{\gamma}\boldsymbol{\gamma}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}^{\#}, \boldsymbol{Y_t}|\mathscr{F}_{t-1})\right]\Delta_{2n}, \end{aligned}$$

where $\boldsymbol{\gamma}^{\#}$ is the middle value between $\boldsymbol{\gamma}_0$ and $\hat{\boldsymbol{\gamma}}$, $g_{\boldsymbol{\gamma}} = \partial g/\partial \boldsymbol{\gamma}$ and $g_{\boldsymbol{\gamma}\boldsymbol{\gamma}} = \partial^2 g/\partial \boldsymbol{\gamma}^2$.

Let $\widehat{H}_{n\boldsymbol{\gamma}} = -n^{-1}\sum_{t=p+1}^{n} g_{\boldsymbol{\gamma}\boldsymbol{\gamma}}\left(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}^{\#}, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right), \widehat{S}_{n\boldsymbol{\gamma}} = n^{-1/2}\sum_{t=2}^{n} g_{\boldsymbol{\gamma}}\left(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}_0, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right)$. Denote $\eta = (\eta_1, \eta_2)$. By the consistency of $\hat{\boldsymbol{\gamma}}$ and Assumption A9, we can show that, for any sequence $\{\varepsilon_n\}$ with $\varepsilon_n = o(1)$,

$$\sup_{\|\eta\|\le\varepsilon_n} \frac{1}{n}\sum_{t=2}^{n} \left\| g_{\boldsymbol{\gamma}\boldsymbol{\gamma}}\left(\hat{\boldsymbol{\theta}} + \eta_1, \boldsymbol{\gamma}_0 + \eta_2, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right) - g_{\boldsymbol{\gamma}\boldsymbol{\gamma}}\left(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right)\right\| = o_p(1),$$

$$\sup_{\|\eta\|\le\varepsilon_n} \frac{1}{n}\sum_{t=2}^{n} \left\| g_{\boldsymbol{\gamma}\boldsymbol{\theta}}\left(\hat{\boldsymbol{\theta}} + \eta_1, \boldsymbol{\gamma}_0 + \eta_2, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right) - g_{\boldsymbol{\gamma}\boldsymbol{\theta}}\left(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right)\right\| = o_p(1),$$

we have $\widehat{H}_{n\boldsymbol{\gamma}} = H_{n\boldsymbol{\gamma}} + o_p(1)$, where

$$H_{n\boldsymbol{\gamma}} = -\frac{1}{n}\sum_{t-2}^{n} g_{\boldsymbol{\gamma}\boldsymbol{\gamma}}\left(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0, \boldsymbol{Y_t}|\mathscr{F}_{t-1}\right).$$

Denote

$$S_{n\gamma} = \frac{1}{\sqrt{n}} \sum_{t=p+1}^{n} g_{\gamma}\left(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0, \boldsymbol{Y_t} | \mathscr{F}_{t-1}\right),$$

and expand $g_{\gamma}\left(\hat{\boldsymbol{\theta}}, \boldsymbol{\gamma}_0, \boldsymbol{Y_t} | \mathscr{F}_{t-1}\right)$ around $\boldsymbol{\theta}_0$, we can show that

$$\widehat{S}_{n\gamma} = S_{n\gamma} + n^{-1} \sum_{i=p+1}^{n} g_{\gamma\boldsymbol{\theta}}\left(\boldsymbol{\theta}_0, \boldsymbol{\gamma}_0, \boldsymbol{Y_t} | \mathscr{F}_{t-1}\right) \Delta_{1n} + o_p(1).$$

Thus

$$\begin{aligned} \sqrt{n}(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0) &= H_{n\gamma}^{-1} S_{n\gamma} + H_{n\gamma}^{-1} P_{n1} \sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) + o_p(1) \\ &= H_{\gamma}^{-1} N(0, H_{\gamma}) + H_{\gamma}^{-1} P_1 \sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) + o_p(1). \end{aligned} \tag{3.26}$$

Notice that $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) = H_{n\boldsymbol{\theta}}^{-1} S_{n\boldsymbol{\theta}} + o_p(1)$, where

$$H_{n\boldsymbol{\theta}} = -\frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 \log f\left(Y_t, \boldsymbol{\theta}_0\right)}{\partial \boldsymbol{\theta}_0 \partial \boldsymbol{\theta}_0'} \quad \text{and} \quad S_{n\boldsymbol{\theta}} = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} \frac{\partial \log f\left(Y_t, \boldsymbol{\theta}_0\right)}{\partial \boldsymbol{\theta}_0}.$$

Therefore, we have

$$\sqrt{n}(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}_0) = H_{n\gamma}^{-1}[S_{n\gamma} + P_{n1} H_{n\boldsymbol{\theta}}^{-1} S_{n\boldsymbol{\theta}}].$$

□

Chapter 4

Copula-based Model for Spatio-temporal Data

4.1 Introduction

In Chapter 3, we proposed a procedure to model multivariate count time series using CDvine. The method is based on a parametric modeling of the marginal distribution. Specifically, in the numerical studies we considered the Poisson regression to model the marginal distribution, which has two limitations. First, the model cannot accommodate over-dispersion and zero-inflation, which is commonly seen in many applications, e.g. the COVID-19 mortality data. Second, the model is based on a parametric specification of the coefficients, assuming that the coefficients, after some prespecified link transformations, are linear functions of the covariates. In this chapter, we focus on spatio-temporal count data. We build on and extend the method in Chapter 3 to address three issues: (1) accommodate possible over-dispersion and zero-inflation; (2) develop a semiparametric approach to model the marginal distribution; (3) accomodate spatial-temporal dependence using copula.

To deal with data with over-dispersion, the generalized Poisson (GP) distribution was first introduced by Consul and Jain (1970). Zero-inflated Generalized Poisson (ZIGP) models have recently been found useful for the analysis of count data with zero-inflation; see Czado et al. (2007) and Gschlößl and Czado (2008). Some other researchers have considered zero-inflated Possion (ZIP) regression; see Lambert (1992). We adopt the ZIGP

model to accommodate both over-dispersion and zero-inflation. Like in generalized linear models, the ZIGP model assumes that the regression coefficients, after some transformation, are linear functions of the covariates. Climov et al. (2002) proposed a semiparametric estimator for a single index Poisson regression model. We also try to relax this linear assumption and provide a more flexible estimation algorithm using a tree-based ensemble method. In machine learning, tree-based models are among the most popular methods used for approximating target functions, in which a function can be learned by splitting the dataset into subsets based on an response-attribute value test. This process is repeated on each derived subset in a recursive manner and is represented by a tree model. Each terminal region is assigned a response value. A popular method of tree-based regression and classification is called CART (Classification and Regression Tree), which was introduced in Breiman et al. (2017) and Lehmann (2004). Lee and Jin (2006) modified the classification and regression tree (CART) algorithm to use the negative ZIP likelihood as an impurity measure in a region. Borisov et al. (2009) took two ensembles to sequentially approximate the transformed ZIP regression parameters. The ZIGP model includes three different types of parameters corresponding to the location, dispersion and zero-inflation proportion. We model the three parameters as nonparametric functions of covariates, and approximate the transformed functions through three sequentially built ensembles. The functions are then estimated using gradient boosting.

We extend the CDvine method in Chapter 3 to capture the spatial-temporal dependence. Specifically, we model the temporal dependence at each location by using uDvine, and capture the spatial dependence by using Gaussian copula with Matérn correlation. In our numerical studies, we choose the Matérn correlation as it allows for a continuous parameterization of the smoothness of the underlying Gaussian random field; see Bevilacqua et al. (2022). However, our proposed method can also adopt other spatial correlation including non-separable spatial correlation matrix discussed in Gneiting (2002), and other copulas such as the elliptic copula (Rémillard et al., 2012) and v-transformed multivariate

normal (VTMN) copula (Bárdossy and Li, 2008); see some related discussions in Tang et al. (2019).

The rest of the chapter is organized as follows. In Section 4.2, we present the proposed model and estimation procedure. In Section 4.3, we assess the numerical performance of the proposed method through a simulation study. In Section 4.4 we analyze the COVID-19 data from nine states in northeastern region of U.S. Some concluding remarks are proposed in Section 3.5.

4.2 Proposed Model and Estimation Procedure

Let $Y(\boldsymbol{s},t)$ be a count spatio-temporal process of the location $\boldsymbol{s} \in \mathscr{S} \subset \mathbb{R}^2$ and time $t \in \mathscr{T} \subset \mathbb{N}$, and $\{Y(\boldsymbol{s}_i,t), i = 1,\ldots,m, t = 1,\ldots,n\}$ be the observed data generated from this process at m locations and n time points with corresponding covariates $\{\boldsymbol{X}(\boldsymbol{s}_i,t), i = 1,\ldots,m, t = 1,\ldots,n\}$. For notational simplicity, we denote $Y_{ti} = Y(\boldsymbol{s}_i,t)$, $\boldsymbol{Y}_t = (Y_{t1},\ldots,Y_{tm})^T$, $\boldsymbol{X}_{ti} = \boldsymbol{X}(\boldsymbol{s}_i,t)$ and $\boldsymbol{X}_t = (\boldsymbol{X}_{t1},\ldots,\boldsymbol{X}_{tm})^T$, where $\boldsymbol{X}_{ti} = (X_{ti1},\ldots,X_{tid})^T$ for $i = 1,\ldots,m$, $t = 1,\ldots,n$ and d is the dimension of the covariates at each time point for each location. Let $\mathscr{F}_{t-1} = \sigma(\boldsymbol{Y}_{t-1},\boldsymbol{Y}_{t-2},\ldots)$ denote the σ-field of all past information and $\mathscr{F}^i_{t-1} = \sigma(\boldsymbol{Y}_{t-1,i},\boldsymbol{Y}_{t-2,i},\ldots)$ denote the σ-field of the past information from location $\boldsymbol{s}_i$.

We assume that the data is generated from a process that satisfies assumptions specified as below.

Assumption C:

C1 For each univariate time series, at each time t, the marginal distribution of Y_{ti} is characterized by $Y_{ti}|\boldsymbol{X} = \boldsymbol{x} \sim F^i(y_{ti}|\boldsymbol{x})$.

C2 For each univariate time series $\{Y_{ti}|\boldsymbol{X}\}_{t=1}^n$, uDvine($p_i$, $\boldsymbol{\alpha}_i$) is used for temporal dependence:

$$\{Y_{ti}\}_{t=1}^n|\boldsymbol{X} \sim \text{uDvine}(p_i,\boldsymbol{\alpha}_i),$$

where uDvine$(p_i,\boldsymbol{\alpha}_i)$ is defined in Definition 3.1.

C3 There exists an m-dimensional copula C_S such that the conditional joint distribution given history $\mathscr{F}_{t-1}$ and $\boldsymbol{X}$ is given by:

$$\begin{aligned}
&G\left(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{F^i\}_{i=1}^m,\boldsymbol{R}\right)\\
=&P(Y_{t1}\le y_{t1},\ldots,Y_{tn}\le y_{tn}|\mathscr{F}_{t-1},\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{F^i\}_{i=1}^m,\boldsymbol{R})\\
=&C_S\left\{F_1\left(y_{t1}|,\mathscr{F}_{t-1}^1,\boldsymbol{X};\boldsymbol{\alpha}_1,F^1\right),\ldots,F_m\left(y_{tm}|\mathscr{F}_{t-1}^m,\boldsymbol{X};\boldsymbol{\alpha}_m,F^m\right)|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{R}\right\},
\end{aligned}$$

where $F_i(\cdot|\mathscr{F}_{t-1}^i,\boldsymbol{X};\boldsymbol{\alpha}_i,F^i)$ for $i=1,\ldots,m$ is the conditional marginal distribution of Y_{ti} given its own history $\mathscr{F}_{t-1}^i$ and $\boldsymbol{X}$ at location $\boldsymbol{s}_i$.

From Assumption C, the conditional joint distribution of $\boldsymbol{Y}_t$ can be obtained as

$$\begin{aligned}
&g(\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{F^i\}_{i=1}^m,\{\boldsymbol{\alpha}_i\}_{i=1}^m,\boldsymbol{R})\\
=&P(\boldsymbol{Y}_t=\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{F^i\}_{i=1}^m,\{\boldsymbol{\alpha}_i\}_{i=1}^m,\boldsymbol{R})\\
=&P(Y_{t1}=y_{t1},\ldots,Y_{tm}=y_{tm}|\mathscr{F}_{t-1},\boldsymbol{X};\{F^i\}_{i=1}^m,\{\boldsymbol{\alpha}_i\}_{i=1}^m,\boldsymbol{R})\\
=&\int_{\Phi_1^{-1}(F_{t1}^2)}^{\Phi_1^{-1}(F_{t1}^1)}\cdots\int_{\Phi_1^{-1}(F_{tm}^2)}^{\Phi_1^{-1}(F_{tm}^1)}\phi_m(\psi_1,\ldots,\psi_m|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{R})\,d\psi_1\cdots d\psi_m.
\end{aligned}$$

Consequently the joint PMF of $\boldsymbol{Y}|\boldsymbol{X}$ is

$$\begin{aligned}
&P(\boldsymbol{Y}=\boldsymbol{y}|\boldsymbol{X}=\boldsymbol{x};\{F^i\}_{i=1}^m,\{\boldsymbol{\alpha}_i\}_{i=1}^m,\boldsymbol{R})\\
=&\prod_{t=1}^n P(\boldsymbol{Y}_t=\boldsymbol{y}_t|\mathscr{F}_{t-1},\boldsymbol{X};\{F^i\}_{i=1}^m,\{\boldsymbol{\alpha}_i\}_{i=1}^m,\boldsymbol{R})\\
=&\prod_{t=1}^n\left\{\int_{\Phi_1^{-1}(F_{t1}^2)}^{\Phi_1^{-1}(F_{t1}^1)}\cdots\int_{\Phi_1^{-1}(F_{tm}^2)}^{\Phi_1^{-1}(F_{tm}^1)}\phi_m(\psi_1,\ldots,\psi_m|\mathscr{F}_{t-1},\boldsymbol{X};\boldsymbol{R})\,d\psi_1\cdots d\psi_m\right\},
\end{aligned}\tag{4.1}$$

where $\phi_m(\cdot;\mathscr{F}_{t-1},\boldsymbol{X},\boldsymbol{R})$ denotes the probability density function (PDF) of an m-dimensional normal distribution with location parameter 0 and scale given by the matrix $\boldsymbol{R}$, Φ_1 denotes the CDF of the univariate margins of the normal distribution, $F_{tj}^1:=\mathrm{P}(Y_{tj}\le y_{tj}|\mathscr{F}_{t-1}^j,\boldsymbol{X};F^j,\boldsymbol{\alpha}_j)$ and $F_{tj}^2:=\mathrm{P}(Y_{tj}\le y_{tj}-1|\mathscr{F}_{t-1}^j,\boldsymbol{X};F^j,\boldsymbol{\alpha}_j)$ for $t=1,\ldots,n,\ j=1,\ldots,m$, can be ob-

tained from Algorithm 3.1.

4.2.1 Zero-inflated Generalized Poisson Linear Model (ZIGP-LM)

In this section we drop the subscript i for notational simplicity. At a given time t, suppose that Y_t follows the generalized Possion (GP) distribution with the location and dispersion parameters μ_t and φ_t. The probability mass function (PMF) of the GP distribution is:

$$P(Y_t = y_t; \mu_t, \varphi_t) = \frac{\mu_t\{\mu_t + (\varphi_t - 1)y_t\}^{y_t - 1}}{y_t!}\varphi_t^{-y_t} e^{-\frac{1}{\varphi_t}\{\mu_t + (\varphi_t - 1)y_t\}}, \tag{4.2}$$

where μ_t is the location parameter and φ_t is the dispersion parameter satisfying $\varphi_t > \max(1/2, 1 - \mu_t/M)$ and M is the largest natural number with $\mu_t + M(\varphi_t - 1) > 0$. When $\varphi_t = 1$, the GP distribution reduces to the Poisson distribution and this is a case of equi-dispersion. When $\varphi_t > 1$, the GP distribution represents count data with over-dispersion. When $\varphi_t < 1$, the GP distribution represents count data with under-dispersion. In this section, we only consider over-dispersion data, so we assume $\varphi_t > 1$.

The GP regression model is given as follows:

$$\begin{aligned} &P(Y_t = y_t|\boldsymbol{x}; \mu, \varphi) \\ =&\frac{\mu(\boldsymbol{x}_t)[\mu(\boldsymbol{x}_t) + \{\varphi(\boldsymbol{x}_t) - 1\}y_t]^{y_t - 1}}{y_t!}\varphi^{-y_t}(\boldsymbol{x}_t)e^{-\frac{1}{\varphi(\boldsymbol{x}_t)}[\mu(\boldsymbol{x}_t) + \{\varphi(\boldsymbol{x}_t) - 1\}y_t]}, \end{aligned} \tag{4.3}$$

where

$$\log\{\mu(\boldsymbol{X}_t)\} = \beta_{t0}^{\mu} + \sum_{j=1}^{d} \beta_{tj}^{\mu} X_{tj}, \tag{4.4}$$

$$\log\{\varphi(\boldsymbol{X}_t) - 1\} = \beta_{t0}^{\varphi} + \sum_{j=1}^{d} \beta_{tj}^{\varphi} X_{tj}. \tag{4.5}$$

To account for zero-inflated count data, Famoye and Singh (2006) proposed the following

ZIGP regression model, $Y_t \sim ZIGP(\mu_t, \varphi_t, p_t)$, for which the PMF is defined as:

$$
\begin{aligned}
&P(Y_t = y_t|\boldsymbol{x};\mu,\varphi,p) \\
&= \begin{cases} p(\boldsymbol{x}_t) + \{1-p(\boldsymbol{x}_t)\}\, e^{\frac{-\mu(\boldsymbol{x}_t)}{\varphi(\boldsymbol{x}_t)}}, & y_t = 0, \\ \{1-p(\boldsymbol{x}_t)\}\, \frac{\mu(\boldsymbol{x}_t)[\mu(\boldsymbol{x}_t)+\{\varphi(\boldsymbol{x}_t)-1\}y_t]^{y_t-1}}{y_t!}\varphi^{-y_t}(\boldsymbol{x}_t)\times & \\ \qquad e^{-\frac{1}{\varphi(\boldsymbol{x}_t)}[\mu(\boldsymbol{x}_t)+\{\varphi(\boldsymbol{x}_t)-1\}y_t]}, & y_t = 1,2,\ldots. \end{cases}
\end{aligned} \tag{4.6}
$$

The traditional ZIGP regression model assumes that $\mu(\boldsymbol{x}_t)$, $\varphi(\boldsymbol{x}_t)$ and $p(\boldsymbol{x}_t)$ are linear functions of covariates via log- and logit-link functions, which will be referred to as Zero-inflated Generalized Poisson Linear Model (ZIGP-LM) hereafter. That is, the ZIGP-LM assumes

$$\log\{\mu(\boldsymbol{X}_t)\} = \beta^{\mu}_{t0} + \sum_{j=1}^{d} \beta^{\mu}_{tj} X_{tj}, \tag{4.7}$$

$$\log\{\varphi(\boldsymbol{X}_t)\} = \beta^{\varphi}_{t0} + \sum_{j=1}^{d} \beta^{\varphi}_{tj} X_{tj}, \tag{4.8}$$

$$\log\left\{\frac{p(\boldsymbol{X_t})}{1-p(\boldsymbol{X}_t)}\right\} = \beta^{p}_{t0} + \sum_{j=1}^{d} \beta^{p}_{tj} X_{tj}, \tag{4.9}$$

where X_{tj} is the j-th covariate at time t. Let $\beta^{\mu} = \{\{\beta^{\mu}_{tj}\}^{d}_{j=1}\}^{n}_{t=1}$, $\beta^{\varphi} = \{\{\beta^{\varphi}_{tj}\}^{d}_{j=1}\}^{n}_{t=1}$ and $\beta^{p} = \{\{\beta^{p}_{tj}\}^{d}_{j=1}\}^{n}_{t=1}$. This ZIGP regression model is usually fit using the maximum likelihood method. Ignoring the temporal dependence, the log-likelihood function is:

$$l(\beta^{\mu},\beta^{\varphi},\beta^{p};\{y_t\}^{n}_{t=1},\boldsymbol{x}) = \sum_{t=1}^{n} \log P(Y_t = y_t|\boldsymbol{x};\beta^{\mu},\beta^{\varphi},\beta^{p}). \tag{4.10}$$

The log-likelihood (4.10) can be maximized by the Newton-Raphson method or the expectation-maximization (EM) algorithm (Hastie et al. (2009)).

4.2.2 Gradient Boosting Model (GBM)

In this section, we provide a brief review to tree-based regression and gradient boosting model. We use some simple notation in this section, and the notations in this section do not relate to the notations elsewhere.

Suppose that the data consists of $(\boldsymbol{x}_i, y_i)$ for $i = 1, 2, \ldots, n$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$, $i = 1, 2, \ldots, n$ are inputs and $\{y_i, i = 1, 2, \ldots, n\}$ are the responses. Suppose first that we have a partition into L regions $R_1, R_2, \ldots, R_L$. We would like to find a target function $T(\boldsymbol{x}; \theta)$ taking the same value in the same region:

$$T(\boldsymbol{x}; \theta) = \sum_{l=1}^{L} c_l I(\boldsymbol{x} \in R_l),$$

where c_l is the value that $T(\boldsymbol{x}; \theta)$ takes in region R_l and $\theta = \{R_l, c_l\}_{l=1}^{L}$, such that $T(\boldsymbol{x}_i; \theta)$ can be used to approximate $E(Y_i | \boldsymbol{X}_i = \boldsymbol{x}_i)$. If the goal is to minimize the sum of squares $\sum_{i=1}^{n} \{y_i - T(\boldsymbol{x}_i; \theta)\}^2$, it is easy to see that the best $\hat{c}_l$ is just the average of y_i in the region R_l:

$$\hat{c}_l = \text{ave}(y_i \mid \boldsymbol{x}_i \in R_l).$$

Now finding the best binary partition by minimizing the sum of squares is generally computationally infeasible, especially when the dimension of X_i is large. Hence we proceed with a greedy algorithm. Denote $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ with X_j being the j-th covariate, $j = 1, \ldots, p$. Starting with all of the data, consider a splitting variable X_j and split point s, where $j = 1, \ldots, p$, and define the pair of half-planes

$$R_1(j, s) = \left\{ \boldsymbol{X} \mid X_j \leq s \right\} \text{ and } R_2(j, s) = \left\{ \boldsymbol{X} \mid X_j > s \right\}.$$

Then we seek the splitting variable X_j and split point s that solve

$$\min_{j,s}\left\{\min_{c_1}\sum_{\boldsymbol{x}_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{\boldsymbol{x}_i\in R_2(j,s)}(y_i-c_2)^2\right\}. \tag{4.11}$$

For any choice of X_j and s, the inner minimization in (4.11) is solved by

$$\hat{c}_1=\text{ave}\left\{y_i\mid \boldsymbol{x}_i\in R_1(j,s)\right\} \text{ and } \hat{c}_2=\text{ave}\left\{y_i\mid \boldsymbol{x}_i\in R_2(j,s)\right\}.$$

For each splitting variable, the determination of the split point s can be done very quickly and hence it is feasible to determine the best pair $(j;s)$ by scanning through all of p input variables. After finding the best split, we partition the data into the two resulting regions and repeat the splitting process on each of the two regions. Then this process is repeated on all of the resulting regions.

The size of the tree is a tuning parameter governing the model's complexity, and the optimal tree size should be adaptively chosen from the data. Clearly a very large tree might over-fit the data, while a small tree might not capture the important structure. A preferred strategy is to grow a large tree, and stop the splitting process only when some minimum tree size is reached.

However, a single regression tree often has low prediction accuracy and is not stable to small fluctuations in the data. So researchers proposed using multiple trees (an ensemble of trees) for regression problems; see Friedman (2001) for boosting ensembles. Ensembles have much higher prediction accuracy and generalization ability while keeping all the advantages of the single tree by using multiple trees. We focus on the boosting tree model below, which gives prediction in the form of an ensemble of trees.

The boosting tree model is a sum of trees,

$$f_K(\boldsymbol{x})=\sum_{k=1}^{K}T\left(\boldsymbol{x};\theta_k\right),$$

which are induced in a forward stagewise manner. At each step in the forward stagewise procedure one must solve

$$\hat{\theta}_k = \arg\min_{\theta_k} \sum_{i=1}^{n} L\left\{y_i, f_{k-1}\left(\boldsymbol{x}_i\right) + T\left(\boldsymbol{x}_i; \theta_k\right)\right\} \tag{4.12}$$

for the region set and constants $\theta_k = \{R_{lk}, c_{lk}\}_{l=1}^{L_k}$ of the next tree given the current model $f_{k-1}(\boldsymbol{x})$, and L in (4.12) a pre-specified loss function, e.g., the squared loss. For regression problems with squared loss, the solution to (4.12) is the same as that for a single tree. It is simply the regression tree that best predicts the current residuals $y_i - f_{k-1}(\boldsymbol{x}_i)$, and $\hat{c}_{lk}$ is the mean of the these residuals in each corresponding region.

One fast approximate algorithms for solving (4.12) with any differentiable loss criterion can be derived by analogy to numerical optimization. The loss in using $f(\boldsymbol{x})$ to predict y on the training data is

$$L(f) = \sum_{i=1}^{n} L\left\{y_i, f\left(\boldsymbol{x}_i\right)\right\}. \tag{4.13}$$

The goal is to minimize $L(f)$ with respect to f, where $f(\boldsymbol{x})$ is constrained to be a sum of trees. Ignoring this constraint, minimizing (4.13) can be viewed as a numerical optimization

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} L(\boldsymbol{f}), \tag{4.14}$$

where the "parameters" $\boldsymbol{f} \in \mathbb{R}^n$ are the values of the approximating function $f(\boldsymbol{x}_i)$ at each of the n data points $\boldsymbol{x}_i$:

$$\boldsymbol{f} = \left\{f\left(\boldsymbol{x}_1\right), f\left(\boldsymbol{x}_2\right), \ldots, f\left(\boldsymbol{x}_n\right)\right\}.$$

The solution to (4.13) is assumed to be a sum of component vectors

$$\boldsymbol{f}_K = \sum_{k=0}^{K} \boldsymbol{h}_k, \quad \boldsymbol{h}_k \in \mathbb{R}^n,$$

where $\boldsymbol{f}_0 = \boldsymbol{h}_0$ is an initial guess, and each successive $\boldsymbol{f}_k$ is induced based on the current parameter vector $\boldsymbol{f}_{k-1}$, which is the sum of the previously induced updates. Numerical optimization methods differ in their prescriptions for computing each increment vector $\boldsymbol{h}_k$ ("step"). The steepest descent chooses $\boldsymbol{h}_k = \nu \boldsymbol{g}_k$, where ν is a scalar and $\boldsymbol{g}_k \in \mathbb{R}$ is the gradient of $L(\boldsymbol{f})$ evaluated at $\boldsymbol{f} = \boldsymbol{f}_{k-1}$. The components of the gradient $\boldsymbol{g}_k$ are

$$g_{ik} = \left[\frac{\partial L\{y_i, f(\boldsymbol{x}_i)\}}{\partial f(\boldsymbol{x}_i)}\right]_{f(\boldsymbol{x}_i)=f_{k-1}(\boldsymbol{x}_i)}. \tag{4.15}$$

The current solution is then updated by

$$\boldsymbol{f}_k = \boldsymbol{f}_{k-1} - \nu \boldsymbol{g}_k,$$

and the process is repeated at the next iteration. The steepest descent can be viewed as a very greedy strategy, since $-\boldsymbol{g}_k$ is the local direction in $\mathbb{R}$ for which $L(\boldsymbol{f})$ is most rapidly decreasing at $\boldsymbol{f} = \boldsymbol{f}_{k-1}$. Unfortunately the gradient (4.15) is defined only at the training data points, whereas the ultimate goal is to generalize $f_K(\boldsymbol{x})$ to new data not represented in the training set. A possible resolution to this dilemma is to induce a tree $T(\boldsymbol{x}; \theta_k)$ at the k-th iteration whose predictions are as close as possible to the negative gradient. Using the squared error to measure closeness leads us to

$$\tilde{\theta}_k = \arg\min_{\theta} \sum_{i=1}^{n} \{-g_{ik} - T(\boldsymbol{x}_i; \theta)\}^2. \tag{4.16}$$

That is, one fits the tree T to the negative gradient values (4.15) by the least squares estimation.

Algorithm 4.1 presents the generic gradient boosting model (GBM) algorithm for regression. Specific algorithms are obtained by inserting different loss criteria $L\{y, f(\boldsymbol{x})\}$. The first line of the algorithm initializes to the optimal constant model, which is just a single terminal region tree. The components of the negative gradient computed at step 2(a) are

referred to as pseudo residuals.

Algorithm 4.1: Algorithm for gradient boosting model (GBM).

1. Initialize $f_0(\boldsymbol{x}) = \arg\min_\gamma \sum_{i=1}^n L(y_i, \gamma)$.

2. For $k = 1$ to K:

   (a) For $i = 1, 2, \ldots, n$, compute

   $$r_{ik} = -\left[\frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)}\right]_{f=f_{k-1}}.$$

   (b) Fit a regression tree to the targets $\{r_{ik}, i = 1, \ldots, n\}$ and obtain a partition into J_k regions $R_{jk}, j = 1, 2, \ldots, J_k$.

   (c) For $j = 1, 2, \ldots, J_k$, compute

   $$c_{jk} = \arg\min_c \sum_{\boldsymbol{x}_i \in R_{jk}} L\{y_i, f_{k-1}(\boldsymbol{x}_i) + c\}.$$

   (d) Update $f_k(\boldsymbol{x}) = f_{k-1}(\boldsymbol{x}) + \nu \sum_{j=1}^{J_k} c_{jk} I\left(\boldsymbol{x} \in R_{jk}\right)$, where ν is the scalar that controls the learning rate of the algorithm.

3. Output $\hat{f}(\boldsymbol{x}) = f_K(\boldsymbol{x})$.

For simplicity, we restrict all trees to be of the same size J. Thus J becomes a hyper-parameter and needs to be adjusted to seek a good balance between bias and variance.

Example 4.1. We give a simple example to illustrate GBM. The sample size is 50 and the response Y are from Uniform distribution on [10, 15], [20, 25], [0, 5], [30, 32] and [13, 17] in five different regions of X, which is simply 1 to 50, as shown in Figure 4.1 below.

Then we start to fit the data using Algorithm 4.1. The loss function is the squared error loss. Left plots in Figure 4.2 show the predictions after 1, 3, 5, 10 and 20 iterations, and right plots show the pseudo residuals. The red lines in the left plots represent the predicted value

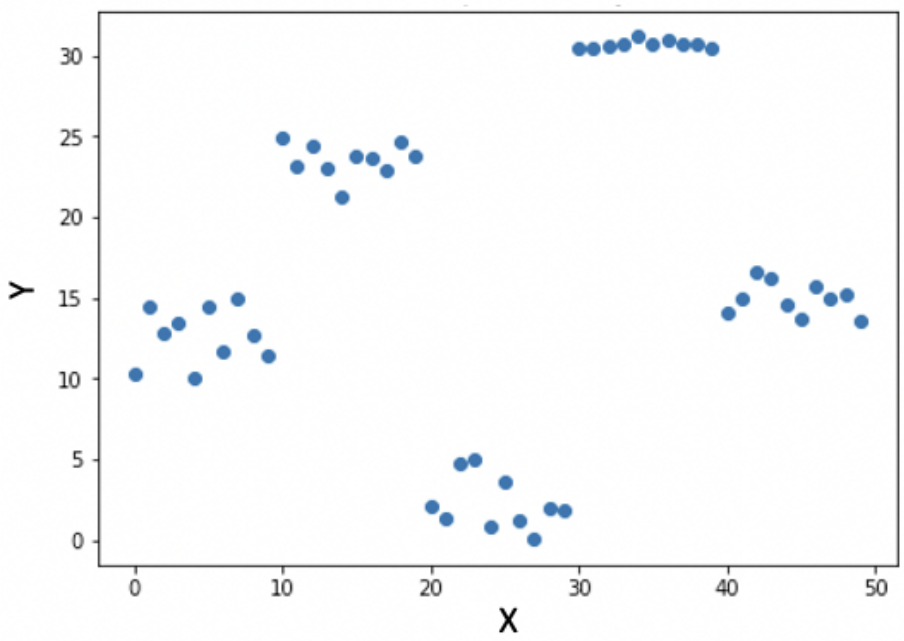

Figure 4.1: Scatter plot of Y versus X, where Y is from five unique uniform distributions in different regions of X.

of y. As the iteration goes on, the fitted value can gradually capture the trend of the response variable and the values of residuals become smaller and smaller. After 20 iterations, the residuals are distributed around 0.

4.2.3 ZIGP Gradient Boosting Model (ZIGP-GBM)

In this section we drop the subscript i for notational simplicity. We derive our own GBM algorithm to fit the ZIGP regression using negative ZIGP likelihood as the loss function, and regression tree as the base learner. We use three ensembles of trees to approximate the transformed ZIGP parameter functions. We use the same link function as used in ZIGP-LM

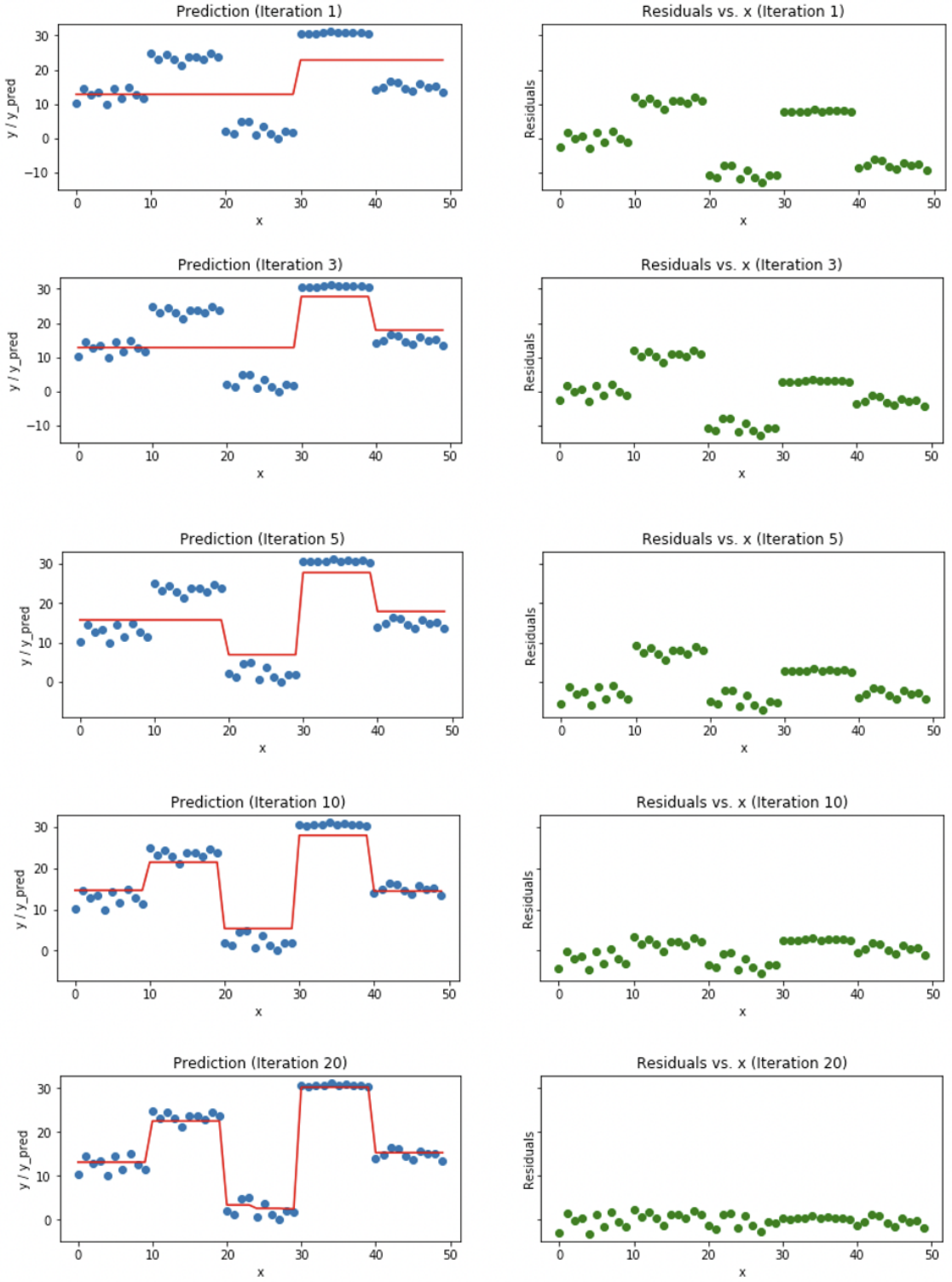

Figure 4.2: Prediction and pseudo residuals after 1, 3, 5, 10 and 20 iterations of the GBM algorithm for analyzing a randomly generated data set from Example 4.1.

to convert the support of the parameters to be on the real line:

$$\mu^*(\boldsymbol{x}_t) = \log\{\mu(\boldsymbol{x}_t)\},\ \mu(\boldsymbol{x}_t) = e^{\mu_t^*(\boldsymbol{x}_t)}, \tag{4.17}$$

$$\varphi^*(\boldsymbol{x}_t) = \log\{\phi(\boldsymbol{x}_t) - 1\},\ \varphi(\boldsymbol{x}_t) = e^{\varphi^*(\boldsymbol{x}_t)} + 1, \tag{4.18}$$

$$p^*(\boldsymbol{x}_t) = \log[p(\boldsymbol{x}_t)/\{1 - p(\boldsymbol{x}_t)\}],\ p(\boldsymbol{x}_t) = e^{p^*(\boldsymbol{x}_t)} / \left\{1 + e^{p^*(\boldsymbol{x}_t)}\right\}. \tag{4.19}$$

So the first ensemble fits model for $\mu^*(\boldsymbol{x}_t)$, the second one for $\varphi^*(\boldsymbol{x}_t)$ and the third one for $p^*(\boldsymbol{x}_t)$. Initial values for them are the maximum likelihood estimator without considering covariates. For notational simplicity, denote $\mu_t^* = \mu^*(\boldsymbol{x}_t)$, $\varphi_t^* = \varphi^*(\boldsymbol{x}_t)$, $p_t^* = p^*(\boldsymbol{x}_t)$, $\mu_t = \mu(\boldsymbol{x}_t)$, $\varphi_t = \varphi(\boldsymbol{x}_t)$ and $p_t = p(\boldsymbol{x}_t)$.

The loss function to be minimized takes the form

$$\begin{aligned} L(\mu_t^*, \varphi_t^*, p_t^*) &= \sum_{t=1}^{n} -\log P(Y_t = y_t | \boldsymbol{x}; \mu_t, \varphi_t, p_t) \\ &= -\sum_{y_t=0} \log\left[p_t + (1-p_t)\exp\left\{-\frac{\mu_t}{\varphi_t}\right\}\right] - \\ &\quad \sum_{y_t>0} [\log(1-p_t) + \log(\mu_t) + (y_t - 1)\{\mu_t + (\varphi_t - 1)y_t\} \\ &\quad + y_t \log(\varphi_t) - \frac{1}{\varphi_t}\{\mu_t + (\varphi_t - 1)y_t\}], \end{aligned}$$

where $\mu_t = e^{\mu_t^*}$, $\varphi_t = e^{\varphi_t^*} + 1$ and $p_t = e^{p_t^*} / \left(1 + e^{p_t^*}\right)$.

The pseudo residuals are calculated as follows. The pseudo-responses for the μ^*-ensemble are

$$\begin{aligned} \tilde{\mu}_t^{*k} &= -\left\{\frac{\partial L(\mu_t^*, \varphi_t^*, p_t^*)}{\partial \mu_t^*}\right\}_{\mu_t^* = \mu_t^{*k-1}(\boldsymbol{x}_t), \varphi_t^* = \varphi_t^{*k-1}(\boldsymbol{x}_t), p_t^* = p_t^{*k-1}(\boldsymbol{x}_t)} \\ &= -\left\{\frac{\partial L(\mu_t^*, \varphi_t^*, p_t^*)}{\partial \mu_t}\right\} \cdot \left(\frac{\partial \mu_t}{\partial \mu_t^*}\right)_{\mu_t^* = \mu_t^{*k-1}(\boldsymbol{x}_t), \varphi_t^* = \varphi_t^{*k-1}(\boldsymbol{x}_t), p_t^* = p_t^{*k-1}(\boldsymbol{x}_t)} \\ &= \begin{cases} \frac{-(1-p_t)\mu_t}{\varphi_t[p_t \exp\{\mu_t/\varphi_t\} + (1-p_t)]}, & y_t = 0 \\ 1 + \frac{\mu_t(y_t - 1)}{\mu_t + (\varphi_t - 1)y_t} - \frac{\mu_t}{\varphi_t}, & y_t > 0 \end{cases}. \end{aligned}$$

where μ_t^{*k}, φ_t^{*k} and p_t^{*k} are the estimations of μ_t, φ_t and p_t after the k-th iteration.

The pseudo residuals for the φ^*-ensemble are

$$\begin{aligned}
\tilde{\varphi}_t^{*k} &= -\left\{\frac{\partial L(\mu_t^*,\varphi_t^*,p_t^*)}{\partial \varphi_t^*}\right\}_{\mu_t^*=\mu_t^{*k-1}(\boldsymbol{x}_t),\varphi_t^*=\varphi_t^{*k-1}(\boldsymbol{x}_t),p_t^*=p_t^{*k-1}(\boldsymbol{x}_t)} \\
&= -\left\{\frac{\partial L(\mu_t^*,\varphi_t^*,p_t^*)}{\partial \varphi_t}\right\}\cdot\left(\frac{\partial \varphi_t}{\partial \varphi_t^*}\right)_{\mu_t^*=\mu_t^{*k-1}(\boldsymbol{x}_t),\varphi_t^*=\varphi_t^{*k-1}(\boldsymbol{x}_t),p_t^*=p_t^{*k-1}(\boldsymbol{x}_t)} \\
&= \begin{cases} \frac{(1-p_t)\mu_t(\varphi_t-1)}{[p_t\exp\{\mu_t/\varphi_t\}+(1-p_t)]\varphi_t^2}, & y_t=0 \\ \left\{\frac{(y_t-1)y_t}{\mu_t+(\varphi_t-1)y_t}-\frac{y_t}{\varphi_t}+\frac{\mu_t-y_t}{\varphi_t^2}\right\}(\varphi_t-1), & y_t>0 \end{cases}.
\end{aligned}$$

The pseudo residuals for the p^*-ensemble are

$$\begin{aligned}
\tilde{p}_t^{*k} &= -\left\{\frac{\partial L(\mu_t^*,\varphi_t^*,p_t^*)}{\partial p_i^*}\right\}_{\mu_t^*=\mu_t^{*k-1}(\boldsymbol{x}_t),\varphi_t^*=\varphi_t^{*k-1}(\boldsymbol{x}_t),p_t^*=p_t^{*k-1}(\boldsymbol{x}_t)} \\
&= -\left\{\frac{\partial L(\mu_t^*,\varphi_t^*,p_t^*)}{\partial p_t}\right\}\cdot\left(\frac{\partial p_t}{\partial p_t^*}\right)_{\mu_t^*=\mu_t^{*k-1}(\boldsymbol{x}_t),\varphi_t^*=\varphi_t^{*k-1}(\boldsymbol{x}_t),p_t^*=p_t^{*k-1}(\boldsymbol{x}_t)} \\
&= \begin{cases} \frac{\exp\{\mu_t/\varphi_t\}-1}{p_t\exp\{\mu_t/\varphi_t\}+(1-p_t)}p_t(1-p_t), & y_t=0 \\ p_t, & y_t>0 \end{cases}.
\end{aligned}$$

Then Algorithm 4.1 can be used to fit gradient boosting model for ZIGP, and this estimation method will be referred to as ZIGP-GBM hereafter. Thus we have the estimations of the marginal distributions $\{\hat{F}^i\}_{i=1}^m$ at m locations. We also note that for computational convenience, we estimate the marginal distribution at each location by ignoring the temporal dependence. To improve the estimation efficiency, one may consider an alternative loss function to accommodate temporal dependence but this will require a prespecification of the temporal dependence and more complicated calculation.

4.2.4 Matérn Correlation Class

For the copula function to appropriately capture the spatio-temporal dependence, we must specify the structure of the matrix $\boldsymbol{R}$ of the spatial copula C_S in Assumption C3. In this

chapter, we assume the correlation between two locations only depends on the Euclidean distance of their locations, not on the locations themselves, i.e., the correlation structure is isotropic. We adopt the Matérn covariance class, and let $U_{ti} = \Phi_1^{-1}\{F_i(y_{ti}|\mathscr{F}_{t-1}^i, \boldsymbol{X}; \boldsymbol{\alpha}_i, F^i)\}$ and assume that the covariance between U_{ti} at location $\boldsymbol{s}_i$ and U_{tj} at location $\boldsymbol{s}_j$ is

$$C_\nu(D) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{D}{\rho}\right)^\nu K_\nu \left(\sqrt{2\nu}\frac{D}{\rho}\right), \tag{4.20}$$

where $D = ||\boldsymbol{s}_i - \boldsymbol{s}_j||$ is the Euclidean distance between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ν and ρ are parameter and range parameters, respectively. Other correlation matrix may also be used, e.g. the non-separable spatial correlation matrix incorporating possible anisotropy in space in Tang et al. (2019) and Gneiting (2002).

Note that the Matérn correlation is a function of the distance D. As D increases, the correlation decreases. It is challenging to determine an appropriate smoothness parameter, and in practice, researchers often fit the model with a few known smoothing parameters and choose the one that yields the largest likelihood. We consider a grid of ν from 0.25 to 4 with step 0.25. For each of the grid, we estimate the range parameter ρ and get the corresponding likelihood. Finally we select the pair producing the maximum likelihood.

4.2.5 Three-stage Estimation Procedure

Given the parametric form of the CDvine, there are three components to be estimated: (a) the marginal distribution $\{F^i\}_{i=1}^m$ of the m component uDvines in Assumption C1, (b) the parameters $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m\}$ of pair-copulas in the m component uDvines in Assumption C2, (c) the matrix $\boldsymbol{R}$ of the spatial copula $C_s(\cdot)$ in Assumption C3. Throughout this section, we assume that the parametric form of the CDvine is known. In the first stage, we assume no temporal dependence for each of the component uDvine for each location, and the marginal distribution can be obtained by the ZIGP-GBM that was introduced in Section 4.2.3. In the

second stage, for $i = 1, \ldots, m$, the parameters $\boldsymbol{\alpha}_i$ in the uDvine(p_i, $\boldsymbol{\alpha}_i$) are estimated using the i-th component univariate time series $\{Y_{it}\}_{t=1}^{n}$ by maximizing the likelihood:

$$l_{1i}(\boldsymbol{\alpha}_i|\{y_{ti}\}_{t=1}^{n}, \boldsymbol{X}) = \sum_{t=1}^{n} \log P(Y_{ti} = y_{ti}|\mathscr{F}_{t-1}^{i}, \boldsymbol{X}; \hat{F}^{i}, \boldsymbol{\alpha}_i), \tag{4.21}$$

where $P(Y_{ti} = y_{ti}|\mathscr{F}_{t-1}^{i}, \boldsymbol{X}; \hat{F}^{i}, \boldsymbol{\alpha}_i)$ can be calculated from Steps 3-5 outlined in Algorithm 3.1. In the third stage, given marginal distributions $\{\hat{F}^{i}\}_{i=1}^{m}$ estimated from ZIGP-GBM from the first stage and estimators $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^{m}$ from the second stage, the parameters ν and ρ involved in the Matérn covariance matrix $\boldsymbol{R}$ can be calculated by maximizing

$$l_2(\nu, \rho|\{\boldsymbol{y}_t, \boldsymbol{X}\}_{t=1}^{n}) = \sum_{t=1}^{n} \log P(\boldsymbol{Y}_t = \boldsymbol{y}_t|\mathscr{F}_{t-1}, \boldsymbol{X}; \{\hat{F}^{i}\}_{i=1}^{m}, \{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^{m}, \boldsymbol{R}).$$

4.2.6 Point and Interval Prediction

Based on the CDvine model with estimated parameters, we can obtain a one-step ahead prediction at time $n+1$. We can predict $\boldsymbol{Y}_{n+1}$ by generating random draws from the conditional distribution of $\boldsymbol{Y}_{n+1}|\mathscr{F}$. Specifically, we generate 1000 samples from the predictive distribution with marginal distributions and CDvine parameters estimated using the observed data. The samples can then be used to construct predictions, e.g. component-wise point predictions based on the sample means or sample medians, or prediction regions.

Below we describe the algorithm to generate data from a fitted CDvine model. Let $\{\hat{\boldsymbol{F}}^{i}\}_{i=1}^{n}$, $\{\hat{\boldsymbol{\alpha}}_i\}_{i=1}^{n}$ and $\hat{\boldsymbol{R}}$ be the estimators of the marginal distributions, uDvine parameters and cross-sectional copula parameters obtained by fitting the model using the observed data. We denote the spatio-temporal data by $\{\boldsymbol{Y}_t = (Y_{t1}, \ldots, Y_{tm})\}_{t=1}^{n}$. Recall that the predictive

distribution (PMF) of $\boldsymbol{Y}_{n+1}$, given $\mathscr{F}_n$ and $\boldsymbol{X}$ is

$$\begin{aligned}
&P\left(\boldsymbol{Y}_{n+1}=\boldsymbol{y}_{n+1}|\mathscr{F}_n,\boldsymbol{X};\{\boldsymbol{\alpha}_i\}_{i=1}^m,\{F^i\}_{i=1}^m,\boldsymbol{R}\right)\\
&=\int_{\Phi_1^{-1}\left(F^2_{n+1,1}\right)}^{\Phi_1^{-1}\left(F^1_{n+1,1}\right)}\cdots\int_{\Phi_1^{-1}\left(F^2_{n+1,m}\right)}^{\Phi_1^{-1}\left(F^1_{n+1,m}\right)}\phi_m\left(\psi_1,\ldots,\psi_m|\mathscr{F}_n,\boldsymbol{X};\boldsymbol{R}\right)d\psi_1\cdots d\psi_m. \qquad (4.22)
\end{aligned}$$

The prediction is based on 1000 randomly drawn samples from the predictive distribution. For b from 1 to 1000, we repeat the following steps.

Step 1: Generate the m dimensional vector $(V_1^b,\ldots,V_m^b)$ from the spatial copula $C_m(\cdot;\hat{\boldsymbol{R}})$.

Step 2: Generate a one-day ahead sample $\boldsymbol{Y}_{n+1}^b=(Y_1^b,\ldots,Y_m^b)^T$ by $Y_i^b=F_i^{-1}\left(U_i^b\mid\mathscr{F}_n^i,\boldsymbol{X};\hat{\boldsymbol{\alpha}}_i,\hat{F}^i\right)$, where for $i=1,\ldots,n$, $F_i\left(\cdot\mid\mathscr{F}_n^i,\boldsymbol{X};\hat{\boldsymbol{\alpha}}_i,\hat{F}^i\right)$ for $i=1,\ldots,n$ can be obtained by using Steps 3-5 outlined in Algorithm 3.1, and F_i^{-1} is defined as follows:

$$F_i^{-1}\left(u_i\mid\mathscr{F}_n^i,\boldsymbol{X};\boldsymbol{\alpha}_i,F^i\right)=\inf\{y_i\geq 0:F_i\left(y_i\mid\mathscr{F}_n^i,\boldsymbol{X};\boldsymbol{\alpha}_i,F^i\right)\geq u_i\}. \qquad (4.23)$$

Thus, at each location $\boldsymbol{s}_i$, we can construct a point forecast of $\boldsymbol{Y}_{n+1,i}$ by using either the sample mean of the random draws, or the sample median, which is more suitable for skewed distributions. We can also use the random draws to construct the $100(1-\nu)\%$ prediction interval $[L,U]$ for the one-step ahead observation $Y_{n+1,i}$, which satisfies

$$P(L\leq Y_{n+1,i}\leq U|\mathscr{F}_n^i,\boldsymbol{X};\hat{\boldsymbol{\alpha}}_i,\hat{F}^i)\geq 1-\nu. \qquad (4.24)$$

Similar to the prediction interval in Chapter 2, we take

$$L=\sup\{y:G_i(y-1)\leq\nu/2\},\ U=\inf\{y:G_i(y)\geq 1-\nu/2\}, \qquad (4.25)$$

where G_i is the empirical distribution of the random samples $\{V_i^b\}_{b=1}^{1000}$.

To evaluate the performance of predictive models at time $n+1$, we adapt the logarithmic

score (LOGS), proposed in (Czado et al., 2009), defined as

$$\text{LOGS}(\hat{P}^*, \boldsymbol{y}_{n+1}) = -\log\{P(\boldsymbol{Y}_{n+1} = \boldsymbol{y}_{n+1} | \mathscr{F}_n, \boldsymbol{X}; \{\hat{\boldsymbol{\alpha}}^*\}_{i=1}^m, \{\hat{F}^{i*}\}_{i=1}^m, \hat{\boldsymbol{R}}^*)\},$$

where $P^* \doteq P(\boldsymbol{Y}_{n+1} = \boldsymbol{y} | \mathscr{F}_n, \boldsymbol{X}; \{\hat{\boldsymbol{\alpha}}^*\}_{i=1}^m, \{\hat{F}^{i*}\}_{i=1}^m, \hat{\boldsymbol{R}}^*)$ and $\boldsymbol{y}_{n+1}$ is the observed response vector at time $n+1$. The LOGS is an assessment tool for probabilistic prediction of both component-wise predictive distribution and joint predictive distribution. It is better than the mean squared error (MSE) for point prediction because MSE can be dominated by predictions at the tails if the distribution is skewed. In addition, the LOGS can also assess how well the spatial dependence in the data is captured by the predictive distribution.

4.3 Simulation Study

4.3.1 Simulation for Univariate Time series

4.3.1.1 Estimation of the Marginal Distribution

We first conduct a simple simulation study to assess the performance of the proposed ZIGP-GBM method for fitting the ZIGP regression with no assumptions on the forms of the parameter functions by considering independent data. We generate independent random variables $Y_1, \ldots, Y_n$ from the ZIGP regression model with the following four types of transformations for the parameters. We consider two sample sizes $n = 500$ and 1000, and repeat the simulation 500 times for each scenario. Each time we generate $n+1$ observations and the last one is taken as the test point.

(1) Trigonometric function:

$$\log(\mu_t) = 2 + \sin(2X_t),$$

$$\log(\varphi_t - 1) = -2 + 3\sin(2X_t),$$

$$\log(\frac{p_t}{1-p_t}) = -2+2\sin(2X_t),$$

where $X_1,\ldots,X_n \sim_{iid} U(0,3)$.

(2) Linear function:

$$\log(\mu_t) = 1+X_t,$$

$$\log(\varphi_t - 1) = -1+X_t,$$

$$\log(\frac{p_t}{1-p_t}) = -2+X_t,$$

where $X_1,\ldots,X_n \sim_{iid} U(0,2)$.

(3) Quadratic function:

$$\log(\mu_t) = 1+X_t^2,$$

$$\log(\varphi_t - 1) = -2+X_t^2,$$

$$\log(\frac{p_t}{1-p_t}) = -X_t^2,$$

where $X_1,\ldots,X_n \sim_{iid} U(-1,2)$.

(4) Tan-linear function:

$$\tan[\pi * \{1/(\mu_t+1) - 0.5\}] = -3-3X_t,$$

$$\tan\{\pi * (1/\varphi_t - 0.5))\} = 1-X_t,$$

$$\tan\{\pi * (p_t - 0.5))\} = -7-7X_t,$$

where $X_1,\ldots,X_n \sim_{iid} U(-5,5)$. We fit the ZIGP regression model by using two methods: the proposed ZIGP-GBM method and the ZIGP-LM method. We compare the performance of the two methods for estimating three parameter functions: $\mu(\cdot)$, $\varphi(\cdot)$, and $p(\cdot)$, based on

the integrated mean squared error (IMSE), defined as:

$$IMSE(\hat{\mu}) = \frac{1}{n}\sum_{t=1}^{n}(\mu_t - \hat{\mu}_t)^2,$$

$$IMSE(\hat{\varphi}) = \frac{1}{n}\sum_{t=1}^{n}(\varphi_t - \hat{\varphi}_t)^2,$$

$$IMSE(\hat{p}) = \frac{1}{n}\sum_{t=1}^{n}(p_t - \hat{p}_t)^2.$$

Table 4.1 summarizes the IMSE of two methods for estimating three parameter functions. In Case (2) where the transformed parameters are linear functions of X and the link functions are correctly specified, the ZIGP-LM gives more accurate estimation. However, the ZIGP-GBM outperforms ZIGP-LM in the other three cases when the parameter functions depend on X in a more complex way. Results suggest that the ZIGP-GBM has good potentials for modeling the distribution of zero-inflated count data with over-dispersion without assuming specific forms of the parameter functions.

Table 4.1: The integrated mean squared error (IMSE) of ZIGP-GBM and ZIGP-LM for estimating the three parameter functions, $\mu(\cdot)$, $\varphi(\cdot)$, and $p(\cdot)$, in the ZIGP regression model for four cases, where data are generated from the ZIGP model with four different transformations for the parameter functions. Values in the parentheses are standard errors.

Case	n	Method	IMSE($\hat{\mu}$)	IMSE($\hat{\varphi}$)	IMSE($\hat{p}$)
Trigonometric	500	ZIGP-GBM	4.44 (0.22)	1.31 (0.031)	0.006 (0.0001)
		ZIGP-LM	26.34 (0.26)	1.58 (0.003)	0.017 (0.0001)
	1000	ZIGP-GBM	2.55 (0.09)	0.30 (0.10)	0.004 (0.0001)
		ZIGP-LM	25.17 (0.14)	1.46 (0.08)	0.017 (0.00005)
Quadratic	500	ZIGP-GBM	125.08 (4.94)	1.03 (0.37)	0.006 (0.0002)
		ZIGP-LM	180.21 (1.54)	2.94 (0.03)	0.030 (0.0004)
	1000	ZIGP-GBM	63.20 (4.17)	0.90 (0.48)	0.004 (0.00009)
		ZIGP-LM	178.32 (1.09)	2.96 (0.02)	0.030 (0.00026)
Linear	500	ZIGP-GBM	2.86 (0.10)	1.59 (1.150)	0.004 (0.00008)
		ZIGP-LM	0.43 (0.02)	0.04 (0.002)	0.001 (0.00005)
	1000	ZIGP-GBM	1.82 (0.05)	0.18 (0.0489)	0.0024 (0.00005)
		ZIGP-LM	0.24 (0.01)	0.02 (0.0009)	0.0005 (0.00002)
Tan-linear	500	ZIGP-GBM	75.26 (3.57)	45.79 (7.42)	0.001 (0.0009)
		ZIGP-LM	1001.48 (40.53)	90.67 (4.45)	0.003 (0.0008)
	1000	ZIGP-GBM	43.98 (2.45)	50.54 (8.83)	0.001 (0.0009)
		ZIGP-LM	839.59 (21.24)	66.91 (1.99)	0.003 (0.0005)

ZIGP-GBM: zero-inflated generalized Poisson by gradient boosting model; ZIGP-LM: zero-inflated generalized Poisson by linear model.

4.3.1.2 Prediction at Future Time Points

In this section we apply the proposed method to analyze univariate time series by using the ZIGP-GBM to estimate the marginal distribution. We generate the univariate count time

series data from the uDvine model as in Section 3.3 but with the ZIGP marginal distribution. We consider the same four cases for the marginal distributions as in Section 4.3.1.1. For the trigonometric model, we use uDvine(2) and set tree 1 as Gaussian ($\theta_1 = 0.7$) copula and tree 2 as Joe ($\theta_2 = 1.8$) copula. For the linear model, we use uDvine(1) and set tree 1 as Gumbel ($\theta_1 = 2$). For the quadratic model, we use uDvine(2) and set tree 1 as Clayton ($\theta_1 = 2$) copula and tree 2 as Frank ($\theta_2 = 2.9$) copula. For the tan-linear model, we use uDvine(1) and set tree 1 as t ($\theta_1 = 0.7$) copula and degree of freedom as 2. The parameters of all the bivariate copulas are specified to make the Kendall's τ of tree 1 equal 0.5 and that of tree 2 equal 0.3.

We consider the following metrics to assess the prediction performance of ZIGP-GBM and ZIGP-LM at time $t = n+1$: (i) the coverage percentages (CPs) and average lengths (ALs) of 95% prediction intervals; (ii) logarithmic score (LOGS) of the predictive conditional distribution at each location.

Table 4.2 summarizes the coverage percentages (CPs) and average lengths (ALs) of 95% prediction intervals and the average LOGS when the marginal distribution is estimated by using the ZIGP-GBM and ZIGP-LM methods, respectively. Results show that the two methods give prediction intervals with similar coverage, but the intervals from ZIGP-GBM tend to be shorter, indicating higher accuracy. Besides, ZIGP-GBM can almost always give smaller LOGS for prediction than ZIGP-LM except when the true model is ZIGP-LM.

4.3.2 Simulation for Spatio-temporal Data

In this section, we conduct simulation for analyzing spatio-temporal data. We would like to mimic the COVID-19 mortality data in Northeastern United States, so we consider the nine states (Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont, New Jersey, New York, Pennsylvania) in Northeastern United States and use their distance matrix calculated based on the longitudes and latitudes of the 9 states. The distance matrix can be found in Table 4.3.

Table 4.2: Comparison of ZIGP-GBM and ZIGP-LM for predicting Y_{n+1} assuming data generated from the trigonometric, quadratic, linear and tan-linear models. Values in the parentheses are standard errors.

Model	n	ZIGP-GBM			ZIGP-LM		
		CPs	ALs	LOGS	CPs	ALs	LOGS
Trigonometric	500	95.0	12.43 (0.53)	2.48 (0.06)	96.2	19.68 (0.88)	2.76 (0.05)
	1000	94.2	13.09 (0.50)	2.60 (0.06)	94.6	21.01 (0.95)	2.91 (0.06)
Quadratic	500	93.5	33.32 (2.57)	3.26 (0.06)	95.1	53.09 (1.90)	3.66 (0.06)
	1000	94.2	38.99 (2.56)	3.43 (0.08)	93.0	54.29 (2.13)	3.80 (0.06)
Linear	500	96.2	18.62 (0.54)	3.08 (0.05)	97.6	20.52 (0.66)	3.06 (0.04)
	1000	97.8	18.88 (0.47)	3.09 (0.05)	98.0	19.84 (0.56)	3.05 (0.04)
Misspecified Link	500	95.4	24.41 (1.34)	3.50 (0.05)	94.3	33.87 (2.12)	4.13 (0.06)
	1000	94.7	23.21 (1.72)	3.43 (0.05)	95.0	31.44 (2.01)	3.96 (0.04)

ZIGP-GBM: zero-inflated generalized Poisson by gradient boosting model; ZIGP-LM: zero-inflated generalized Poisson by linear model.

Table 4.3: Distance (in km) between states in Northeastern United States.

	CT	ME	MA	NH	RI	VT	NJ	NY
ME	503.09							
MA	104.07	402.83						
NH	247.63	264.16	159.25					
RI	97.60	451.50	68.21	223.13				
VT	272.42	307.01	213.08	97.18	281.29			
NJ	228.41	727.08	332.47	465.18	309.88	462.36		
NY	274.08	569.11	314.29	329.62	356.40	262.19	315.70	
PA	432.45	855.58	522.01	599.45	529.70	551.02	275.37	298.03

We assume the ZIGP distribution with trigonometric transformation for each location:

$$\log(\mu) = \beta_{\mu_0} + \beta_{\mu_1} \sin(2X),$$

$$\log(\varphi - 1) = \beta_{\varphi_0} + \beta_{\varphi_1} \sin(2X),$$

$$\log(\frac{p}{1-p}) = \beta_{p0} + \beta_{p1} \sin(2X),$$

where $X \sim U(0,3)$. To generate the coefficients for nine locations, we first take the initial values $\beta_{\mu_0} = 2$, $\beta_{\mu_1} = 1$, $\beta_{\varphi_0} = -2$, $\beta_{\varphi_1} = 3$, $\beta_{p_0} = -2$, $\beta_{p_1} = 2$, and add some random

Table 4.4: Coefficients in the ZIGP regression model with the trigonometric transformation at each location for the simulation study of spatio-temporal data.

Location	β_{μ_0}	$\beta_{\mu 1}$	β_{φ_0}	β_{φ_1}	β_{p0}	β_{p1}
CT	1.60	0.95	-1.08	2.38	-1.66	1.09
ME	1.32	1.32	-2.09	2.89	-2.81	2.57
MA	1.92	0.29	-2.16	2.98	-2.29	2.63
NH	1.97	1.04	-2.14	3.18	-0.93	1.83
RI	1.99	1.68	-2.15	3.60	-2.07	2.26
VT	1.02	1.32	-2.20	4.00	-1.40	2.00
NJ	2.46	1.02	-1.78	3.01	-2.31	1.90
NY	2.19	1.83	-1.66	3.95	-1.73	2.84
PA	2.65	0.49	-1.07	2.73	-2.24	1.78

numbers generated from $N(0, 0.5)$ to the initial values. Then we get the coefficients for nine locations, which are shown in Table 4.4. For the temporal dependence, a uDvine(1) is used for four locations: CT, ME, MA and NH, and a uDvine(2) is used for five locations: RI, VT, NJ, NY and PA. For uDvine(1), we set tree 1 as Gumbel ($\theta_1 = 2$) copula. For uDvine(2), we set tree 1 as Clayton ($\theta_1 = 2$) copula and tree 2 as Frank ($\theta_2 = 2.9$) copula. The parameters of all the bivariate copulas are specified to make the Kendall's τ of tree 1 equal 0.5 and that of tree 2 equal 0.3. The parameters in the Matérn covariance matrix are set as $\nu = 2$ and $\rho = 100$.

We consider the following metrics to assess the prediction performance of ZIGP-GBM and ZIGP-LM at time $t = n+1$: (i) the coverage percentages (CPs) and average lengths (ALs) of the component-wise 95% prediction intervals at each location; (ii) logarithmic score (LOGS) of the predictive conditional distribution for joint prediction.

4.3.2.1 Comparison of Two Types of Component-wise Prediction

To demonstrate the value of accommodating spatial correlation and temporal dependence using CDvine, we compare two types of component-wise prediction at future time points:

1. Type I: We only use PA's own history data to compute its one-day ahead prediction.

2. Type II: We use both the PA's own history and the other eight states' contemporaneous

Table 4.5: The coverage percentages and average lengths of 95% prediction intervals and LOGS from two types of prediction methods for predicting Y_{n+1} at PA using simulated data mimicking the COVID-19 study. Values in the parentheses are standard errors.

	Type I prediction			Type II prediction		
n	CPs	ALs	LOGS	CPs	ALs	LOGS
500	97.19	11.02 (0.28)	2.31 (0.04)	97.60	9.56 (0.24)	2.19 (0.04)
1000	95.98	10.87 (0.31)	2.31 (0.04)	97.19	9.34 (0.26)	2.14 (0.04)

observations to perform the one-day ahead prediction for PA.

As a demonstration, we select PA as the testing state and the aim is to predict the outcome at PA at future time point $t = n+1$. Type I predictions are actually the same as what we did for the univariate count time series in Section 4.3.1.2. Type II predictions are the same as the procedure in Section 4.2.6.

Table 4.5 summarizes the coverage percentages (CPs) and average lengths (ALs) of the 95% prediction intervals and the average LOGS produced by Type I and Type II predictions, respectively. Type II prediction gives consistently smaller LOGS and narrower prediction intervals than Type I prediction. This verifies that incorporating spatial information using the proposed CDvine model framework can improve the prediction accuracy over predictions based on information from the location alone.

4.3.2.2 Prediction at Future Time Points

In this section we consider the joint prediction at nine states, and compare the results from ZIGP-GBM and ZIGP-LM. Table 4.6 summarizes the coverage percentages (CPs) and average lengths (ALs) of the 95% prediction intervals at each location and the average LOGS produced by ZIGP-GBM and ZIGP-LM, respectively. For both $n = 500$ and 1000, ZIGP-GBM gives narrower component-wise prediction intervals and smaller LOGS than ZIGP-LM. The joint LOGS produced from ZIGP-GBM, 15.64 (0.31), is also significantly smaller than that of ZIGP-LM, 17.12 (0.25).

Table 4.6: Comparison of ZIGP-GBM and ZIGP-LM for predicting $\boldsymbol{Y}_{n+1}$ in nine states using simulated data. Values in the parentheses are standard errors.

Location	n	ZIGP-GBM			ZIGP-LM		
		CPs	ALs	LOGS	CPs	ALs	LOGS
CT	500	95.2	12.46 (0.53)	2.60 (0.06)	96.0	19.63 (0.86)	2.94 (0.05)
	1000	94.4	13.19 (0.51)	2.60 (0.05)	94.6	21.23 (1.01)	2.80(0.06)
ME	500	94.3	13.27 (0.54)	2.56 (0.05)	95.2	20.88 (0.86)	2.95 (0.06)
	1000	95.2	14.21 (0.50)	2.54 (0.06)	94.1	22.42 (0.91)	2.89 (0.05)
MA	500	94.7	11.86 (0.53)	2.62 (0.06)	96.2	18.71 (0.87)	2.94 (0.05)
	1000	95.2	12.45 (0.52)	2.51 (0.06)	96.6	19.19 (0.96)	2.89 (0.06)
NH	500	95.5	12.56 (0.52)	2.56 (0.05)	95.2	18.75 (0.86)	2.87 (0.06)
	1000	94.8	13.68 (0.52)	2.52 (0.06)	95.8	20.99 (0.95)	2.89 (0.05)
RI	500	94.9	13.95 (0.55)	2.56 (0.05)	95.8	20.77 (0.88)	2.77 (0.06)
	1000	95.1	14.32 (0.52)	2.57 (0.05)	94.5	21.82 (0.95)	2.84 (0.06)
VT	500	95.1	11.92 (0.53)	2.49 (0.06)	95.7	22.76 (0.86)	2.82 (0.06)
	1000	96.0	12.64 (0.50)	2.53 (0.05)	94.6	23.83 (0.96)	2.92 (0.05)
NJ	500	94.7	14.52 (0.54)	2.48 (0.05)	95.9	21.64 (0.88)	2.81 (0.06)
	1000	95.0	14.85 (0.49)	2.50 (0.06)	94.7	22.49 (0.97)	2.82 (0.05)
NY	500	95.3	11.79 (0.56)	2.62 (0.05)	95.8	18.74 (0.88)	2.92 (0.06)
	1000	94.2	12.01 (0.49)	2.57 (0.05)	95.1	19.18(0.95)	2.87 (0.06)
PA	500	94.8	12.43 (0.55)	2.45 (0.06)	96.2	19.68 (0.87)	2.77 (0.05)
	1000	95.3	13.09 (0.50)	2.44 (0.06)	94.4	21.01 (0.96)	2.69 (0.05)

ZIGP-GBM: zero-inflated generalized Poisson by gradient boosting model; ZIGP-LM: zero-inflated generalized Poisson by linear model.

4.4 Real Data Analysis

In this section, we reanalyze the COVID-19 data studied in Chapters 2 and 3, but focused on nine locations, including 9 states in Northeastern United States: CT, ME, MA, NH, RI, VT, NJ, NY and PA. The data set contains the number of daily deaths in these states from Jun 22, 2020 to Dec 31, 2021, corresponding to 400 time points. The first 320 time points are taken as training set while the last 80 are taken as the testing set.

Similar to Chapter 3, we first need to figure out the marginal distribution. We take the number of confirmed cases as one of the covariates, along with the time t and its quadratic

term t^2 to capture the time trend. We arrive at the following ZIGP regression model:

$$Y_{ti} \sim ZIGP(\mu_{ti}, \varphi_{ti}, p_{ti}), \tag{4.26}$$

where

$$\begin{aligned}
\log(\mu_{ti}) &= f_\mu(t, t^2, x_{t-10,i}), \\
\log(\varphi_{ti} - 1) &= f_\varphi(t, t^2, x_{t-10,i}), \\
\log(\frac{p_{ti}}{1-p_{ti}}) &= f_p(t, t^2, x_{t-10,i}).
\end{aligned}$$

In this application, t, t^2 and $X_{t-10,i}$ can all be considered as continuous covariates due to their large ranges. Besides the proposed marginal estimation method ZIGP-GBM, we also include ZIGP-LM for comparison. For both of these methods, CDvine is used to capture spatial and temporal dependence, and Matérn correlation class is used in the 9-dimensional spatial copula. Similar to Chapter 3, we use a rolling-window validation method to assess the performance of different methods for prediction. For all methods, we update the parameter estimation progressively in a rolling-window format with a window length of 320 days, and update the parameters in each method in a daily base. We first consider component-wise prediction for each location. Table 4.7 summarizes the average LOGS, the coverage percentages (CPs) and average lengths (ALs) of 95% prediction intervals for one-day-ahead forecasts from each method, for each location separately. As we can see, ZIGP-GBM can always give smaller LOGS than ZIGP-LM, which means ZIGP-GBM leads to more accurate predictive distributions at each location. Also, ZIGP-GBM gives similar coverage percentages but shorter prediction intervals than ZIGP-LM for all states except PA, and this also implies that ZIGP-GBM gives more accurate component-wise predictive distributions.

We also obtain the joint LOGS, which evaluates the performance of the predictive distribution of nine locations for one-day ahead prediction. The average joint LOGS across testing points from ZIGP-GBM and ZIGP-LM are 17.30 (0.34) and 17.82 (0.39),

Table 4.7: Comparison of ZIGP-GBM and ZIGP-LM for predicting Y_{n+1} using the COVID-19 data in 9 states. Values in the parentheses are standard errors.

Location	ZIGP-GBM			ZIGP-LM		
	CPs	ALs	LOGS	CPs	ALs	LOGS
CT	94.5	24.64 (1.02)	2.66 (0.12)	92.5	26.20 (3.18)	2.75 (0.13)
ME	98.0	16.22 (0.61)	2.17 (0.08)	97.5	17.44 (0.77)	2.27 (0.09)
MA	96.5	29.42 (0.87)	3.29 (0.07)	94.0	42.68 (4.32)	3.44 (0.09)
NH	98.5	11.82 (0.57)	1.99 (0.07)	99.5	85.58 (31.7)	2.15 (0.09)
RI	99.5	7.44 (0.22)	1.58 (0.06)	100.0	7.85 (0.48)	1.61 (0.05)
VT	96.5	4.85 (0.13)	1.29 (0.09)	96.0	6.30 (0.17)	1.41 (0.11)
NJ	98.5	56.03 (1.73)	3.55 (0.05)	99.0	63.97 (2.35)	3.67 (0.05)
NY	96.5	30.23 (1.66)	3.15 (0.07)	96.5	72.96 (13.37)	3.33 (0.08)
PA	99.0	148.98 (5.33)	4.40 (0.07)	97.0	145.60 (9.18)	4.54 (0.08)

ZIGP-GBM: zero-inflated generalized Poisson by gradient boosting model; ZIGP-LM: zero-inflated generalized Poisson by linear model.

respectively, which implies ZIGP-GBM leads to more accurate joint predictive distributions for this spatio-temporal data.

4.5 Discussion

In this section we proposed a more flexible estimation of marginal distribution using ZIGP-GBM. Moreover, to capture the spatial dependence, we apply the Matérn correlation class in Gaussian copula so that the correlation is a function of distance. Similar to Chapter 3, D-vine is used to model the temporal dependence and a high-dimensional copula is used to model the spatial dependence. Compared to ZIGP-LM, ZIGP-GBM shows greater flexibility. The application to the COVID-19 data demonstrated the promises of ZIGP-GBM and CDvine for modeling and analyzing spatio-temporal count data.

Chapter 5

Conclusion and Discussion

5.1 Concluding Remarks

The study of copulas and their applications in statistics and probability have attracted increasing attention from researchers. In this work, we explore copula-based estimation for dependent count data.

For a univariate time series with Markov property, bivariate copula functions are utilized to capture temporal dependence, while leaving the marginal distribution to be estimated separately. However, due to the identifiability issue of copula functions for count data, the copula function cannot be uniquely determined. We resolve this issue by including continuous regressors. This allows us to perform the copula selection. We develop a cross-validation strategy and select the copula that can produce the smallest LOGS. MLE is used for estimation, and based on the estimator, we further investigate the conditional quantile estimation and asymptotic property. Throughout our simulation, we show that our proposed copula-selection method is comparable to the correctly specified copula, in terms of prediction. We also demonstrate the improvement achieved by the proposed method in the real data analysis with baseball data and COVID-19 data, compared to existing methods for count time series.

In the second project, we propose a copula-linked D-vine model to analyze multivariate count time series. Our main goal is to make prediction for these sequences. We use a D-vine to capture the temporal dependence of each component time series and a cross-

sectional copula to capture the cross-sectional dependence. This copula linked D-vine allows us to model temporal and cross-sectional dependence simultaneously. To reduce the computational burden of D-vine, we truncate it at some order and show that it is a Markov process after truncation. A two-stage estimation procedure is proposed. Simulation study is conducted to verify the sequential selection approach and show the flexibility of our proposed model. We also analyze the COVID-19 data from three states and show that our proposed method outperforms other existing methods.

In the third project, we consider an extension to the CDvine method to make it more flexible and accommodate spatio-temporal count data. First we extend the Poisson distribution to zero-inflated generalized Poisson distribution, so that this marginal distribution can deal with zero-inflation and over-dispersion. Second, we use the gradient boosting approach to obtain a semiparametric estimation for the marginal distribution. We conduct simulation study and demonstrate that GBM performs better than linear model for the estimation of marginal distribution. Moreover, this improvement can lead to better prediction at future time points.

5.2 Future Research

We propose copula-based methods for analyzing dependent count data. The utilization of copula functions provides a flexible way to incorporate different potential marginal distributions. Below we identify two future research directions for potential improvement of our proposed methods.

In Chapter 4, we consider Gaussian copula to model the spatial dependence. Inspired by Sun and Stein (2015), we may also consider t-copula to model the spatial dependence structure. In addition, C-vine (Erhardt et al., 2015) is also a convenient tool to capture the spatial dependence.

While we adopt a machine learning approach to relax the assumption on estimating the marginal distribution in Chapter 4, the method still relies on the parametric distribution

assumption. Future research is needed to estimate the marginal distribution under more relaxed assumptions.

Bibliography

Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198.

Aldor-Noiman, S., Brown, L. D., Fox, E. B., and Stine, R. A. (2016). Spatio-temporal low count processes with application to violent crime events. *Statistica Sinica*, pages 1587–1610.

ArunKumar, K., Kalaga, D. V., Kumar, C. M. S., Kawaji, M., and Brenza, T. M. (2021). Forecasting of covid-19 using deep layer recurrent neural networks (rnns) with gated recurrent units (grus) and long short-term memory (lstm) cells. *Chaos, Solitons & Fractals*, 146:110861.

Bain, L. J. and Patel, J. K. (1993). Prediction intervals based on partial observations for some discrete distributions. *IEEE Transactions on Reliability*, 42(3):459–463.

Bárdossy, A. and Li, J. (2008). Geostatistical interpolation using copulas. *Water Resources Research*, 44(7).

Beare, B. K. (2010). Copulas and temporal dependence. *Econometrica*, 78(1):395–410.

Beare, B. K. (2012). Archimedean copulas and temporal dependence. *Econometric Theory*, pages 1165–1185.

Bedford, T. and Cooke, R. M. (2002). Vines: A new graphical model for dependent random variables. *Annals of Statistics*, pages 1031–1068.

Bevilacqua, M., Caamaño-Carrillo, C., and Porcu, E. (2022). Unifying compactly supported and matern covariance functions in spatial statistics. *Journal of Multivariate Analysis*, page 104949.

Borisov, A., Runger, G., Tuv, E., and Lurponglukana-Strand, N. (2009). Zero-inflated boosted ensembles for rare event counts. In *International Symposium on Intelligent Data Analysis*, pages 225–236. Springer.

Brandt, P. T., Williams, J. T., Fordham, B. O., and Pollins, B. (2000). Dynamic modeling for persistent event-count time series. *American Journal of Political Science*, pages 823–843.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Chen, X. and Fan, Y. (2006). Estimation of copula-based semiparametric time series models. *Journal of Econometrics*, 130(2):307–335.

Chen, X., Wu, W. B., Yi, Y., et al. (2009). Efficient estimation of copula-based semiparametric markov models. *The Annals of Statistics*, 37(6B):4214–4253.

Chiogna, M. and Gaetan, C. (2010). An interchangeable approach for modelling spatio-temporal count data. *Environmetrics*, 21(7-8):849–867.

Clark, N. J., Kaiser, M. S., and Dixon, P. M. (2018). A spatially correlated auto-regressive model for count data. *arXiv preprint arXiv:1805.08323*.

Climov, D., Delecroix, M., and Simar, L. (2002). Semiparametric estimation in single index poisson regression: a practical approach. *Journal of Applied Statistics*, 29(7):1047–1070.

Consul, P. and Jain, G. (1970). On generalization of poisson distribution. In *Annals of Mathematical Statistics*, volume 41, page 1387. INST MATHEMATICAL STATISTICS IMS BUSINESS OFFICE-SUITE 7, 3401 INVESTMENT ....

Czado, C., Erhardt, V., Min, A., and Wagner, S. (2007). Zero-inflated generalized poisson models with regression effects on the mean, dispersion and zero-inflation level applied to patent outsourcing rates. *Statistical Modelling*, 7(2):125–153.

Czado, C., Gneiting, T., and Held, L. (2009). Predictive model assessment for count data. *Biometrics*, 65(4):1254–1261.

Datta, A., Banerjee, S., Finley, A. O., Hamm, N. A., and Schaap, M. (2016). Nonseparable dynamic nearest neighbor gaussian process models for large spatio-temporal data with an application to particulate matter analysis. *The Annals of Applied Statistics*, 10(3):1286.

Davis, R. A., Dunsmuir, W. T., and Streett, S. B. (2003). Observation-driven models for poisson counts. *Biometrika*, 90(4):777–790.

Davis, R. A., Dunsmuir, W. T., and Wang, Y. (2000). On autocorrelation in a poisson regression model. *Biometrika*, 87(3):491–505.

Davis, R. A. and Liu, H. (2016). Theory and inference for a class of nonlinear models with application to time series of counts. *Statistica Sinica*, pages 1673–1707.

Davis, R. A. and Wu, R. (2009). A negative binomial model for time series of counts. *Biometrika*, 96(3):735–749.

Denuit, M. and Lambert, P. (2005). Constraints on concordance measures in bivariate discrete data. *Journal of Multivariate Analysis*, 93(1):40–57.

Donovan, S., Brett-Major, D., and Lawler, J. V. (2021). Lag between covid-19 cases and subsequent deaths.

Elsner, J. B. and Jagger, T. H. (2006). Prediction models for annual us hurricane counts. *Journal of Climate*, 19(12):2935–2952.

Erhardt, T. M., Czado, C., and Schepsmeier, U. (2015). Spatial composite likelihood inference using local c-vines. *Journal of Multivariate Analysis*, 138:74–88.

Famoye, F. and Singh, K. P. (2006). Zero-inflated generalized poisson regression model with an application to domestic violence data. *Journal of Data Science*, 4(1):117–130.

Ferland, R., Latour, A., and Oraichi, D. (2006). Integer-valued garch process. *Journal of Time Series Analysis*, 27(6):923–942.

Fokianos, K., Rahbek, A., and Tjøstheim, D. (2009). Poisson autoregression. *Journal of the American Statistical Association*, 104(488):1430–1439.

Fokianos, K., Støve, B., Tjøstheim, D., Doukhan, P., et al. (2020). Multivariate count autoregression. *Bernoulli*, 26(1):471–499.

Fokianos, K. and Tjøstheim, D. (2011). Log-linear poisson autoregression. *Journal of Multivariate Analysis*, 102(3):563–578.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Genest, C. and Nešlehová, J. (2007). A primer on copulas for count data. *ASTIN Bulletin: The Journal of the IAA*, 37(2):475–515.

Gneiting, T. (2002). Nonseparable, stationary covariance functions for space–time data. *Journal of the American Statistical Association*, 97(458):590–600.

Gschlößl, S. and Czado, C. (2008). Modelling count data with overdispersion and spatial effects. *Statistical Papers*, 49(3):531–552.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Heinen, A. (2003). Modelling time series count data: an autoregressive conditional poisson model. *Available at SSRN 1117187*.

Heinen, A. and Rengifo, E. (2007). Multivariate autoregressive modeling of time series count data using copulas. *Journal of Empirical Finance*, 14(4):564–583.

Heinen, A. and Rengifo, E. (2008). Multivariate reduced rank regression in non-gaussian contexts, using copulas. *Computational Statistics & Data Analysis*, 52(6):2931–2944.

Hughes, J. (2015). copcar: A flexible regression model for areal data. *Journal of Computational and Graphical Statistics*, 24(3):733–755.

Jia, Y., Kechagias, S., Livsey, J., Lund, R., and Pipiras, V. (2021). Latent gaussian count time series. *Journal of the American Statistical Association*, (just-accepted):1–28.

Jin, R. (2021). The lag between daily reported covid-19 cases and deaths and its relationship to age. *Journal of Public Health Research*, 10(3).

Joe, H. (2014). *Dependence modeling with copulas*. CRC press.

Jung, R. C., Kukuk, M., and Liesenfeld, R. (2006). Time series of count data: modeling, estimation and diagnostics. *Computational Statistics & Data Analysis*, 51(4):2350–2364.

Kaufmann, H. (1987). Regression models for nonstationary categorical time series: asymptotic estimation theory. *The Annals of Statistics*, pages 79–98.

Kim, T., Lieberman, B., Luta, G., and Peña, E. A. (2021). Prediction intervals for poisson-based regression models. *Wiley Interdisciplinary Reviews: Computational Statistics*, page e1568.

Lambert, D. (1992). Zero-inflated poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14.

Lee, S.-K. and Jin, S. (2006). Decision tree approaches for zero-inflated count data. *Journal of Applied Statistics*, 33(8):853–865.

Lehmann, E. L. (2004). *Elements of large-sample theory*. Springer Science & Business Media.

Livsey, J., Lund, R., Kechagias, S., and Pipiras, V. (2018). Multivariate integer-valued time series with flexible autocovariances and their application to major hurricane counts. *The Annals of Applied Statistics*, 12(1):408–431.

Loaiza-Maya, R., Smith, M. S., and Maneesoonthorn, W. (2018). Time series copulas for heteroskedastic data. *Journal of Applied Econometrics*, 33(3):332–354.

Longla, M. and Peligrad, M. (2012). Some aspects of modeling dependence in copula-based markov chains. *Journal of Multivariate Analysis*, 111:234–240.

Madsen, L. (2009). Maximum likelihood estimation of regression parameters with spatially dependent discrete data. *Journal of Agricultural, Biological, and Environmental Statistics*, 14(4):375–391.

Madsen, L. and Fang, Y. (2011). Joint regression analysis for discrete longitudinal data. *Biometrics*, 67(3):1171–1175.

Nelsen, R. B. (2007). *An introduction to copulas*. Springer Science & Business Media.

Nikoloulopoulos, A. K. (2013). On the estimation of normal copula discrete regression models using the continuous extension and simulated likelihood. *Journal of Statistical Planning and Inference*, 143(11):1923–1937.

Nikoloulopoulos, A. K. (2016). Efficient estimation of high-dimensional multivariate normal copula models with discrete spatial responses. *Stochastic Environmental Research and Risk Assessment*, 30(2):493–505.

Panagiotelis, A., Czado, C., and Joe, H. (2012). Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association*, 107(499):1063–1072.

Park, E. S., Guttorp, P., and Henry, R. C. (2001). Multivariate receptor modeling for temporally correlated data by using mcmc. *Journal of the American Statistical Association*, 96(456):1171–1183.

Pötscher, B. M. and Prucha, I. R. (1989). A uniform law of large numbers for dependent and heterogeneous data processes. *Econometrica: Journal of the Econometric Society*, pages 675–683.

Rémillard, B., Papageorgiou, N., and Soustra, F. (2012). Copula-based semiparametric models for multivariate time series. *Journal of Multivariate Analysis*, 110:30–42.

Rüschendorf, L. (2009). On the distributional transform, sklar's theorem, and the empirical copula process. *Journal of Statistical Planning and Inference*, 139(11):3921–3927.

Shi, P. and Yang, L. (2018). Pair copula constructions for insurance experience rating. *Journal of the American Statistical Association*, 113(521):122–133.

Sklar, M. (1959). Fonctions de repartition an dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8:229–231.

Smith, M., Min, A., Almeida, C., and Czado, C. (2010). Modeling longitudinal data using a pair-copula decomposition of serial dependence. *Journal of the American Statistical Association*, 105(492):1467–1479.

Song, P. X.-K. (2000). Multivariate dispersion models generated from gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320.

Song, P. X.-K., Li, M., and Yuan, Y. (2009). Joint regression analysis of correlated data using gaussian copulas. *Biometrics*, 65(1):60–68.

Storlie, C. B., Reich, B. J., Rust, W. N., Ticknor, L. O., Bonnie, A. M., Montoya, A. J., and Michalak, S. E. (2017). Spatiotemporal modeling of node temperatures in supercomputers. *Journal of the American Statistical Association*, 112(517):92–108.

Sun, Y. and Stein, M. L. (2015). A stochastic space-time model for intermittent precipitation occurrences. *The Annals of Applied Statistics*, 9(4):2110–2132.

Tang, Y., Wang, H. J., Sun, Y., and Hering, A. S. (2019). Copula-based semiparametric models for spatiotemporal data. *Biometrics*, 75(4):1156–1167.

Trivedi, P. and Zimmer, D. (2017). A note on identification of bivariate copulas for discrete count data. *Econometrics*, 5(1):10.

Turkkan, N. and Pham-Gia, T. (1997). Algorithm as 308: Highest posterior density credible region and minimum area confidence region: the bivariate case. *Applied Statistics*, pages 131–140.

White, H. (1996). *Estimation, inference and specification analysis*. Number 22. Cambridge university press.

Yang, L., Frees, E. W., and Zhang, Z. (2020). Nonparametric estimation of copula regression models with discrete outcomes. *Journal of the American Statistical Association*, 115(530):707–720.

Zhao, Z., Shi, P., and Zhang, Z. (2021). Modeling multivariate time series with copula-linked univariate d-vines. *Journal of Business & Economic Statistics*, pages 1–15.