GAUSSIAN COPULA MODELLING FOR INTEGER-VALUED TIME SERIES

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Engineering and Physical Sciences

2016

Hannah Lennon
School of Mathematics

Contents

Word count 62155

List of Tables

List of Figures

The University of Manchester

Hannah Lennon
Doctor of Philosophy
Gaussian Copula Modelling for Integer-Valued Time Series
May 31, 2016

This thesis is concerned with the modelling of integer-valued time series. The data naturally occurs in various areas whenever a number of events are observed over time. The model considered in this study consists of a Gaussian copula with autoregressive-moving average (ARMA) dependence and discrete margins that can be specified, unspecified, with or without covariates. It can be interpreted as a '*digitised*' ARMA model. An ARMA model is used for the latent process so that well-established methods in time series analysis can be used.

Still the computation of the log-likelihood poses many problems because it is the sum of 2^N terms involving the Gaussian cumulative distribution function when N is the length of the time series. We consider an Monte Carlo Expectation-Maximisation (MCEM) algorithm for the maximum likelihood estimation of the model which works well for small to moderate N. Then an Approximate Bayesian Computation (ABC) method is developed to take advantage of the fact that data can be simulated easily from an ARMA model and digitised. A spectral comparison method is used in the rejection-acceptance step. This is shown to work well for large N. Finally we write the model in an R-vine copula representation and use a sequential algorithm for the computation of the log-likelihood. We evaluate the score and Hessian of the log-likelihood and give analytic solutions for the standard errors. The proposed methodologies are illustrated using simulation studies and highlight the advantages of incorporating classic ideas from time series analysis into modern methods of model fitting. For illustration we compare the three methods on US polio incidence data (Zeger, 1988) and we discuss their relative merits.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Policy on Presentation of Theses.

Acknowledgements

I would like to thank my supervisor Dr. Jingsong Yuan for suggesting a highly interesting topic which has gone much further than we ever anticipated, and for giving help and guidance throughout. Thank you to the many members of staff, PhD students and friends from the School of Mathematics for everything along the way.

Thank you to my family, the Lennons and the Powers, for their endless support. And thank you to Sam.

Dedicated to Christopher J. Power.

Chapter 1

Introduction

Dependence modelling of non-normal multivariate data is essential for many applications such as longitudinal studies, spatial statistics and time series analysis. Models are required with unknown parameters of a probability distribution to describe the data for the purpose of prediction and forecasting. The focus of this thesis is the modelling of temporal dependence for integer-valued time series data. Integer-valued time series appear naturally in various areas whenever a number of events are observed over time. Observed values tend to be low, highly-skewed and zero values not being uncommon because typically the data of interest occurs as the number of failures or uncommon events. Examples of such data are the number of:

a) hospital admissions each day (Herwartz et al., 2015);

b) insurance claims each year (Shi and Valdez, 2014);

c) away goals scored per match (Karlis and Ntzoufras, 2003), *and*

d) radiation detection alarms per security session.

Despite integer-valued time series being common in practice, methods for statistical analysis are not well known (McKenzie, 2003; Fokianos, 2009). Interesting and complex models have been proposed (Jacobs and Lewis, 1978a; Lewis, 1980; Al-Osh and Alzaid, 1987), yet currently there is not a dominant or well established model for discrete-valued time series. This may be due to the computational cost of fitting complex models to data or may be that the available models are too restrictive or fail to capture the true data generating process. The assumption of independence is commonly used by necessity yet it is not appropriate in many modelling scenarios. Creating the dependence component of a model is a difficult task even when neglecting the marginal specification. If the marginal distributions are specified correctly then desired dependence may be impossible to achieve. Models which incorporate both marginal and dependence structures are typically computationally intensive. The advancement of computing architecture in the last few decades has impacted on the kind of models that can be considered for applications. The increase in collecting and accessing discrete data sets are driving the desire to analyse and the need for suitable forecast and prediction models.

Copula Modelling

Copulas have been used in univariate continuous-valued time series analysis to characterise the dependence in a sequence of observations (Joe, 1997). The advantage of the copula approach is that we are able to specify the marginal distributions of random variables separately from the temporal dependence. Copulas can be used to construct multivariate models for dependent responses of any type; continuous, discrete or mixed. This thesis uses the copula of a Gaussian autoregressive-moving average (ARMA) model with discrete margins to construct a model for integer-valued time series. The model can be seen as a '*digitised*' ARMA model. We explore the challenges of modelling integer-valued time series using this Gaussian copula model.

In copula modelling, high-dimensional discrete distributions pose challenges for maximum likelihood estimation (Genest and Nešlehová, 2007; Nikoloulopoulos and Karlis, 2009; Danaher and Smith, 2011; Smith and Khaled, 2012; Panagiotelis et al., 2012; Joe, 2015). The major challenges in copula modelling, as summarised by Mikosch (2006), Embrechts (2009) and Erhardt (2010), are:

a) copulas with discrete margins;

b) the difficulties arising in high dimensions;

c) the choice of suitable copula class, *and*

d) the difficulties in applying copulas in time series analysis.

The Gaussian copula is natural choice for dimensions greater than $d = 2$ with several desirable properties while providing an interpretation of the model parameters. Other high dimensional copulas do not facilitate complex dependence structures; specifically they do not lend easily to the classical time series dependence.

The likelihood for a copula model with N discrete margins is a sum of 2^N copula evaluations. The Gaussian copula does not have a closed analytical form and requires N-dimensional integration to evaluate. In high-dimensions, typical of time series, this has a large computational cost and likelihood inference is difficult. To simplify the estimation we incorporate ideas from time series analysis.

We develop three methods for fitting the Gaussian copula model to integer-valued time series data. The selection of approaches have developed naturally to overcome the computational aspects of parameter inference and their merits are showcased for different lengths of time series. First, we consider a Monte Carlo Expectation-Maximisation (MCEM) algorithm (Dempster et al., 1977; Wei and Tanner, 1990) for maximum likelihood estimation of the model where we take advantage of the '*digitised*' interpretation and consider the latent ARMA as missing data to use an EM algorithm. The idea is to iteratively estimate an underlying ARMA parameters by conditioning on the observed integer-valued time series using current parameter estimates, then maximise the log-likelihood of the estimated ARMA time series to update the parameter estimates.

A second approach for estimation involves Approximate Bayesian Computation (ABC), which is a likelihood-free simulation approach based on a rejection algorithm

(Tavaré et al., 1997; Beaumont et al., 2002). Data are simulated from the copula model using many different parameter values. We accept these parameters if the simulated data are 'similar', in some sense, to the observed data and the set of accepted parameters are then used for the basis of Monte Carlo inference. The key to this approach is quantifying the similarity of the observed and simulated integer-valued data. A method for comparing two integer-valued time series using summary statistics is introduced and an ABC algorithm is derived for parameter estimation.

The final estimation method uses vine copula methods (Bedford and Cooke, 2001; Panagiotelis et al., 2012) to write the intractable likelihood in terms of bivariate Gaussian copulas to reduce the computational burden. The Gaussian copula model fits neatly into this methodology and this approach facilitates joint maximum likelihood estimation as well as analytic solutions to the standard errors.

1.1 Thesis Structure

This thesis addresses the deficiencies in the theory and estimation methods for the class of Gaussian copula models with discrete margins for modelling integer-valued time series data. Chapter 2 reviews the background material and literature for time series analysis in the continuous and discrete case. In Chapter 3, copula theory is introduced and the probability mass function for the copula model is given. We extend the theory concerning a conditional distribution of the underlying process and the computational difficulties of the Gaussian copula model for integer-valued time series data are posed alongside a literature review of current methods. Chapters 4 to 6 develop three independent estimation techniques, namely MCEM, ABC and vine copula methods, to enable copula modelling for discrete time series data. Simulated series will be used within in these chapters to demonstrate how the methodology and estimation applies in situations where the model is known before dealing with real world data in Chapter 7. We finish with an empirical comparison between our three techniques as well as current methods.

Chapter 2

Time Series

The necessary preliminary material for time series analysis is introduced in this chapter. For continuous-valued variables, the standard models are well established and a review of linear time series models is given in Section 2.1. This material is necessary to subsequent chapters because the Gaussian copula model for integer-valued time series uses the copula of the continuous ARMA time series. For discrete-valued variables, models and methods are not as well established and we give a summary of existing models in Section 2.2.

2.1 Time series analysis

A time series is a series of observations $\{x_t\}$ observed over a period of time. The term 'time series' could sensibly be applied to a record of any time-varying process, either deterministic or random. When we observe a time series $\{x_t\}$, usually we assume that $\{x_t\}$ is a realisation of a random process $\{X_t\}$. Formally, a random process $\{X_t, t \in T\}$ is a family of random variables indexed by t belonging to some set T. Accordingly, for each t the random variable X_t has properties that will be described by some probability distribution, $F(x_t)$, later known as the marginal distribution or margin. We observe a time series only at discrete time points $t = 1, \ldots, N$.

Let us first introduce the simplest statistical time series, a purely random process which corresponds to a 'no-memory' process.

Definition 1 (White noise)**.** *A sequence $\{X_t, t \in \mathbb{Z}\}$ of uncorrelated random variables with zero mean and constant variance is called a white noise.*

These are building blocks for time series models with dependence.

2.1.1 Stationary time series

A fundamental task in time series analysis is to find out from the data how x_t depends on past values $x_{t-1}, x_{t-2}, \ldots$ and use this information for the purpose of prediction or forecasting. We consider stationary processes whose statistical properties do not change over time.

Definition 2 (Stationary Process)**.** *A process $\{X_t\}$ is said to be stationary if for any admissible $t_1, \ldots, t_N$ and any k, the joint distribution of $(X_{t_1}, \ldots, X_{t_N})$ is identical with that of $(X_{t_1+k}, \ldots, X_{t_N+k})$. This is called strict stationarity.*

We use a less severe requirement that only the moments up to order 2 are identical and this is called weakly stationarity. The process $\{X_t\}$ is weak stationary if for all $t \in \mathbb{Z}$, $\mathrm{E}(X_t)^2 < \infty$,

a) the mean $\mathrm{E}(X_t) = \mu$ is constant;

b) the variance $\mathrm{var}(X_t)$ is constant, *and*

c) the covariance $\mathrm{cov}(X_t, X_{t-k})$ is a function of k only, for all $k \in \mathbb{Z}$.

The covariance matrix can provide insight into the dependence between the finite number of random variables $(X_1, \ldots, X_N)$. For a time series the concept of the covariance matrix is extended to deal with an infinite collection of random variables.

Definition 3 (Autocovariance Function)**.** *For stationary process $\{X_t\}$ with $\mathrm{var}(X_t) < \infty$, the autocovariance function is*

$$R(k) = \mathrm{E}\left[(X_t - \mu)(X_{t-k} - \mu)\right], \qquad \textit{for } k = 0, \pm 1, \pm 2, \ldots,$$

where k is the lag.

The autocorrelation function can be defined as a function of the lag k.

Definition 4 (Autocorrelation Function)**.** *The autocorrelation function $r(k)$ of $\{X_t\}$ defined for $k = 0, \pm 1, \pm 2, \ldots$ is given by*

$$r(k) = R(k)/R(0).$$

Linear time series models are designed to model the covariance structure.

Definition 5 (General linear process)**.** *A time series $\{X_t\}$ is said to be a linear process if it can be expressed in the form*

$$X_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}, \tag{2.1}$$

where $\{\varepsilon_t\}$ is white noise and $\{\psi_i\}$ is a given sequence of constants satisfying $\sum_{i=0}^{\infty} \psi_i^2 < \infty$. This condition is required so that X_t has a finite variance (Priestley, 1981).

The most popular general linear process is the ARMA(p, q) process.

Definition 6 (Autoregressive-Moving Average Process of order (p,q))**.** *The process $\{X_t\}$ is a mixed autoregressive-moving average process, denoted ARMA(p, q), if it is a stationary process satisfying*

$$X_t - a_1 X_{t-1} - \cdots - a_p X_{t-p} = \varepsilon_t + b_1 \varepsilon_{t-1} + \cdots + b_q \varepsilon_{t-q}, \tag{2.2}$$

where $a_1, \ldots, a_p, b_1, \ldots, b_q$ are constants and $\{\varepsilon_t\}$ is white noise.

The AR(p) and MA(q) processes are special cases of an ARMA(p, q) process when $q = 0$ or $p = 0$ respectively.

ARMA(p,q) models are typically used to capture the serial linear dependence. It is convenient to write the ARMA model in terms of the backward shift operator B, which is defined for any time series $\{X_t\}$ as $BX_t = X_{t-1}$. It has the following properties: $B^2 X_t = B(BX_t) = BX_{t-1} = X_{t-2}$ and $B^0 X_t = X_t$. Thus

$$B^k X_t = X_{t-k}, \qquad k = 0, \pm 1, \pm 2, \ldots.$$

Letting

$$\alpha(z) = 1 - a_1 z - \cdots - a_p z^p \quad \text{and} \quad \beta(z) = 1 + b_1 z + \cdots + b_q z^q,$$

the ARMA(p, q) model (2.2) can be expressed as

$$(1 - a_1 B - \cdots - a_p B^p) X_t = (1 + b_1 B + \cdots + b_q B^q) \varepsilon_t.$$

or

$$\alpha(B) X_t = \beta(B) \varepsilon_t.$$

The class of autoregressive-moving average processes are linear processes since the polynomial $\alpha(B)$ can be inverted to give the coefficients $\{\psi_i\}$ in (2.1) as $[\alpha(B)]^{-1} \beta(B) \varepsilon_t$.

A process is said to be causal if it can be written as a MA(∞) model, in the form

$$X_t = \sum_{i=0}^{\infty} \psi_i \varepsilon_{t-i}, \tag{2.3}$$

where ε_t is white noise. An ARMA process is causal if the roots of $\alpha(B)$ lie outside the unit circle.

An ARMA model is said to be invertible if it can be written as an AR(∞) model, in the form

$$X_t = \sum_{j=1}^{\infty} \phi_j X_{t-j} + \varepsilon_t.$$

This is true if the roots of $\beta(B)$ lie outside the unit circle.

Typically with time series data only a single realisation of the stochastic process is available. Without further conditions the ARMA process (2.2) is not identifiable since there are many sets of coefficients $(a_1, \ldots, a_p, b_1, \ldots, b_q)$ all of which give rise to the same autocovariance function for $\{X_t\}$. The representation of an ARMA process is unique if:

a) all the roots of $\alpha(B)$ and $\beta(B)$ lie outside the unit circle, *and*

b) the characteristic polynomials $\alpha(B)$ and $\beta(B)$ have no common factors.

The reciprocal polynomials as defined as

$$\alpha^*(B) = B^p\alpha(1/B), \qquad \text{and} \qquad \beta^*(B) = B^q\beta(1/B). \tag{2.4}$$

For example if $f(B) = 1 - a_1B - a_2B^2 - a_3B^3$ then $f^*(B) = B^3 - a_1B^2 - a_2B - a_3$ with coefficients written in reverse order. If B_1 is a root of $f(B)$ then $1/B_1$ is a root of $f^*(B)$ (Pless, 2011, p. 59). Satisfying the condition a) above is equivalent to satisfying the condition that the roots of the reciprocal polynomials $\alpha^*(B)$ and $\beta^*(B)$ lie inside the unit circle. This is the approach we take, as does Yao and Brockwell (2006). We can test directly and indirectly that these conditions are satisfied.

Direct: This involves computing the roots by solving the p-degree polynomial and directly checking the absolute values do not exceed one.

For example for AR(1), The root of the polynomial $1 - a_1B = 0$ is $1/a_1$ which lies outside the unit circle when $-1 < a_1 < 1$. For example for an AR(2) model, the stationarity conditions are

$$a_1 + a_2 < 1, \qquad a_2 - a_1 < 1, \qquad |a_2| < 2,$$

while the stationarity conditions of an MA(2) process are

$$b_1 + b_2 > -1, \qquad b_1 - b_2 < 1, \qquad |b_2| < 2.$$

Indirect: Farebrother (1973) gives the sets of inequalities, which can check are satisfied, for the roots of polynomials of up to order 4 to have absolute value less than one as

$$\begin{aligned} &a_4 > 1, \\ &3 - 3a_4 > -a_2, \\ &1 - a_2 - a_4 > |a_1 + a_3|. \end{aligned}$$

For a more general approach Shor's algorithm can be used.

By transforming parameters Alternatively, the parameters can be transformed in such a way that the conditions are automatically satisfied as suggested by Jones (1980). That is, by re-parameterising in terms of the partial autoregressive and moving average regression coefficients that are constrained to the open interval $(-1, 1)$ and we can use the transformations

$$\alpha_i = \frac{1 - \exp(-u_i)}{1 + \exp(-u_i)},$$
$$\beta_j = \frac{1 - \exp(-w_j)}{1 + \exp(-w_j)}.$$

For maximum likelihood estimation, a nonlinear optimisation can then be carried out with respect to u_i and w_j, $i = 1, \ldots, p$, $j = 1, \ldots, q$. This is the procedure used by the R function `arima()` to fit an ARMA(p, q) model in R.

Computing the autocovariances of an ARMA model

The theoretical autocovariances $R(k)$ of an ARMA(p, q) model can be obtained by solving the first p equations of (2.5) which are found by taking covariances with X_{t-k} on both sides of (2.2), then recursively solving for $k > p + 1$. This is the standard method in Brockwell and Davis (1987).

For $k = 0, 1, \ldots, p$

$$\begin{aligned}
R(0) - a_1R(1) - a_2R(2) - \cdots - a_pR(p) &= \sigma_\varepsilon^2(1 + b_1\psi_1 + b_2\psi_2 + \cdots + b_q\psi_q),\\
R(1) - a_1R(0) - a_2R(1) - \cdots - a_pR(p-1) &= \sigma_\varepsilon^2(b_1 + b_2\psi_1 + b_3\psi_2 + \cdots + b_q\psi_{q-1}),\\
R(2) - a_1R(1) - a_2R(0) - \cdots - a_pR(p-2) &= \sigma_\varepsilon^2(b_2 + b_3\psi_1 + b_4\psi_2 + \cdots + b_q\psi_{q-2}),\\
\vdots \qquad\qquad & \qquad \vdots\\
R(p) - a_1R(p-1) - \cdots - a_pR(0) &= \sigma_\varepsilon^2(b_p + b_{p+1}\psi_1 + b_{p+2}\psi_2 + \ldots\ldots).
\end{aligned}$$

For $k > p$

$$R(k) - a_1R(k-1) - \cdots - a_pR(k-p) = \sigma_\varepsilon^2(b_k + b_{k+1}\psi_1 + b_{k+2}\psi_2 + \ldots\ldots). \tag{2.5}$$

where the coefficients ψ_k are found recursively using

$$\psi_k = b_k + a_k + a_{k-1}\psi_1 + a_{k-1}\psi_2 + \cdots + a_1\psi_{k-1}.$$

We let $a_k = 0$ for $k > p$ and $b_k = 0$ for $k > q$.

The partial autocorrelations

For any $k \geq 1$ let

$$\widehat{X}_t^{(k)} = \rho_{k1}X_{t-1} + \rho_{k2}X_{t-2} + \rho_{k3}X_{t-3} + \cdots + \rho_{kk}X_{t-k}$$

be the best linear predictor of X_t in terms of $X_{t-1}, \ldots, X_{t-k}$, i.e., it has the smallest mean square error. Then ρ_{kk} is the partial autocorrelation at lag k. It can be interpreted as the correlation of X_{t-k} on X_t after removing the influence of the intermediate variables, $X_{t-1}, \ldots, X_{t-k+1}$. The variance of the prediction error $(X_t - \widehat{X}_t^{(k)})$ is denoted by τ_k^2.

The partial autocorrelations ρ_{kk} of an ARMA model satisfy the Yule-Walker equations (Brockwell and Davis, 1987, pg. 239)

$$\begin{bmatrix} 1 & r\left(1\right) & r\left(2\right) & \ldots & r\left(k-1\right) \\ r\left(1\right) & 1 & r\left(1\right) & & \vdots \\ r\left(2\right) & r\left(1\right) & 1 & \ddots & \vdots \\ \vdots & & \vdots & \vdots & r\left(1\right) \\ r\left(k-1\right) & r\left(k-2\right) & \ldots & r\left(1\right) & 1 \end{bmatrix} \begin{bmatrix} \rho_{k1} \\ \rho_{k2} \\ \vdots \\ \vdots \\ \rho_{kk} \end{bmatrix} = \begin{bmatrix} r\left(1\right) \\ r\left(2\right) \\ \vdots \\ \vdots \\ r\left(k\right) \end{bmatrix}. \tag{2.6}$$

These equations can be solved recursively for $k = 1, 2, \ldots, N$ using the Levinson-Durbin algorithm (Brockwell and Davis, 1987, pg. 169) without inverting the autocorrelation matrix $R_{k\times k} = \left(r\left(i-j\right)\right)_{i,j=1,\ldots,k}$.

The Levinson Durbin algorithm is recursively given for $k = 1, 2, \ldots,$

$$\text{For } \ k = 1 \qquad \rho_{11} = R(1)/R(0) = r(1), \qquad \tau_1^2 = R(0)(1 - r(1)^2),$$

$$\text{For } \ k > 1 \qquad \rho_{kk} = \frac{R(k) - \sum_{j=1}^{k-1} \rho_{k-1,j} R(k-j)}{\tau_{k-1}^2}, \quad \tau_k^2 = \tau_{k-1}^2(1 - \rho_{kk}^2),$$

and

$$\begin{bmatrix} \rho_{k1} \\ \rho_{k2} \\ \vdots \\ \rho_{k,k-1} \end{bmatrix} = \begin{bmatrix} \rho_{k-1,1} \\ \rho_{k-1,2} \\ \vdots \\ \rho_{k-1,k-1} \end{bmatrix} - \rho_{kk} \begin{bmatrix} \rho_{k-1,k-1} \\ \rho_{k-1,k-2} \\ \vdots \\ \rho_{k-1,1} \end{bmatrix}, \tag{2.7}$$

where $\tau_k^2 = \mathrm{E}[(X_t - \widehat{X}_t^{(k)})^2]$ is the mean square prediction error. The output of the algorithm consists of ρ_{kk} and τ_k^2 for $k = 1, \ldots, N$.

2.1.2 Gaussian time series

The ARMA model is very generally specified as the linear, additive structure determining the correlation structure of the stationary sequence $\{X_t\}$, under well-known restrictions on the parameters and a linear combination of random variables $\{\varepsilon_t\}$ of unspecified distribution. If we require $\{X_t\}$ to be Gaussian, we take the ε_t's to be normally distributed as $N(0, \sigma_\varepsilon^2)$, i.e. Gaussian white noise. Then $\{X_t, t = 1, \ldots, N\}$ has an N-dimensional Gaussian distribution $N_N(\mathbf{0}, \Sigma)$ and for any $r \geq 1$ and any admissible subset $t_1, t_2, \ldots, t_N$, the joint probability distribution of $(X_{t_1}, X_{t_2}, \ldots, X_{t_r})$ is multivariate Gaussian.

Thus for a Gaussian ARMA model, the joint density of $\mathbf{X} = (X_1, \ldots, X_N)$ can be written as

$$f(x_1, \ldots, x_N) = (2\pi)^{-N/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{X} - \boldsymbol{\mu})\right),$$

where Σ is an $N \times N$ Toeplitz matrix of the autocovariances of $\{X_t\}$ i.e., $\Sigma(i,j) = \sigma_{i-j} = \mathrm{E}\left[X_{t+i-j} X_t\right]$, $0 \leq i, j \leq N - 1$. The elements of Σ^{-1} are the partial autocovariances so that the Levinson recursions provide an efficient way of evaluating the likelihood for a continuous time series.

2.2 Discrete-valued time series analysis

The ARMA model is well developed, flexible and easy to implement. The model is completely specified and fully described by the parameters $(a_1, \ldots, a_p, b_1, \ldots, b_q)$ and σ_ε^2. However, it does not guarantee X_t to be integer valued. The integer nature of our data means the assumptions above are not met. When considering linear models such as (2.2) the white noise terms $\{\varepsilon_t\}$ cannot follow a Gaussian distribution, if $\{X_t\}$ must take integer values. In practical applications when discrete data is high-valued, the Gaussian ARMA model can be used and can work well. However, it is not uncommon for integer-valued data to have low mean values, appear highly skewed and have a peak around zero so overall the standard ARMA model has limited usefulness for low-valued integer times series. Lewis (1980) stress the importance of developing a class of useful time series models for non-Gaussian data. McKenzie (2003) describes modelling of discrete variate as the most challenging.

Discrete-valued time series occur naturally as counting processes. In this section we discuss the existing literature of discrete-valued time series models, in particular integer-valued time series models. There are various ways of addressing dependency over time between discrete random variables. Natural approaches are

a) to allow for some type of autocorrelation in the error (extending the Gaussian ARMA models to discrete random variables), *or*
b) to introduce lagged dependent variables and trend terms to regression models (extending discrete models for i.i.d. random variables to include time dependency).

Diverse models for discrete variate time series have been developed along these lines including Markov chain models, models based on linear models with recognised correlation structures, models based on the notion of thinning operators, models based on state-space models, and regression models to name a few. McKenzie (2003) gives a detailed overview of many of these models and a short summary is presented here.

There have been several attempts to propose models that resemble the structure and properties of the ARMA models. The most popular class of models are Markov chain models, Discrete ARMA (DARMA), and INteger-valued ARMA (INARMA) models (Raftery, 1985; Jacobs and Lewis, 1978a; McKenzie, 1985).

Markov chains are very similar to AR(p) processes, but they can have discrete values. A discrete time series $\{X_t\}$ is called a Markov chain of order-p if the Markov dependence property is satisfied, that is

$$\begin{aligned}&\Pr(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \dots)\\&= \Pr(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_{t-p} = x_{t-p}).\end{aligned}$$

The AR(p) process has the order-p Markov property that given all the past values, X_t only depends on the last p values. However they tend to be over-parameterised and their correlation structure is often too limited for application (MacDonald and Zucchini, 1997, p. 13). Further, the data to be modelled can often be shown to be non-Markovian, or at least not first-order Markovian (Jacobs and Lewis, 1983). Higher order Markov chains can be used but this only increases the problem of over-parameterisation. Raftery (1985) deals with the over-parameterisation by introducing Mixture Transition Models (MTD) to model discrete time series with particular marginal distributions if necessary.

A series of papers by Jacobs and Lewis (1978a,b,c, 1983) are considered the first attempt at a general class of simple models for discrete time series. The Discrete AutoRegressive Moving Average (DARMA) class of models is very general and can support any desired marginal distribution, say π. Table 2.1 gives examples from the class of DARMA models where $\{X_t\}$ is stationary with any particular distribution π. The DARMA models are formed by taking mixtures of i.i.d. discrete random variables Z_t. The $\{Z_t\}$ can be considered as the discrete analogue to $\{\varepsilon_t\}$ in the ARMA model and the correlation structure of the processes mimic that of the ARMA(p, q) processes.

The class of DARMA models have been used in a range of applications: hydrology, biomedical engineering and broadcast video traffic (Chang et al., 1984; Bernotas et al., 1986; Heyman et al., 1992; Heyman and Lakshman, 1996). They seem to work better in theory than in applications. Limitations of the use of DARMA models for discrete variate time series include the following:

a) DARMA models only allow for positive correlations,

b) DARMA processes are extremely general, and derive no benefits from the structure of particular distributions,

c) McKenzie (1985, 2003) noted that these series tend to have sample paths that are constant for long runs which is a trait which is not generally seen in data,

d) Finally, dependence is created by means of having runs of specific values. This is unlikely to be a reasonable assumption, except perhaps in the case of binary variates (Azzalini, 1994).

Table 2.1: The DARMA time series models. Let $\{Z_t\}$ be a sequence of i.i.d. random variables with desired distribution given by π. If X_0 is sampled from the desired distribution π then the model generates a stationary process $\{X_t\}$ whose marginal distribution is π.

Name	Model	Comments
DAR(1)	$X_t = V_t\, X_{t-1} + (1 - V_t)\, Z_t,$ $\{V_t\}$ are i.i.d. Bernoulli variables with $\Pr(V_t = 1) = \alpha$, $0 \leq \alpha \leq 1$.	· The model defines the current observation to be the last observation with probability α or another independent random variate from the same distribution π. · The autocorrelation function is $r(k) = \alpha^k$.
DAR(p)	$X_t = V_t\, X_{t-A_t} + (1 - V_t)\, Z_t,$ $\{A_t\}$ are i.i.d. random variables with $\Pr(A_t = i) = a_i$, for $i = 1, \dots, p$.	· With probability α_i, the current observation is one of the past values, $X_{t-1}, \dots, X_{t-p}$, (chosen stochastically), or another random variable Z_t from the same distribution π. · The model has the same autocorrelation function as an AR(p) model, $r(k) = \alpha \sum_{i=1}^{p} a_i\, r(k-i)$.
DMA(q)	$X_t = Z_{t-S_t},$ $\{S_t\}$ are i.i.d. random variables with $\Pr(S_t = j) = b_j$, for $j = 0, 1, \dots, q$.	· The moving average behaviour of an ARMA process can be produced for discrete random variables by a random index model. · The autocorrelation function is $r(k) = \sum\limits_{j=0}^{q-k} b_j\, b_{j-k}$ for $k = 1, \dots, q$.

Name	Model	Comments
DARMA$(p, q+1)$	$X_t = U_t\ Y_{t-S_t} + (1-U_t)\ Z_{t-q-1}$, $Z_t = V_t\ Z_{t-A_t} + (1-V_t)\ Y_t$, $\{U_t\}$ are i.i.d. Bernoulli variables with $\Pr(U_t = 1) = \beta$, $0 \leq \beta \leq 1$	To summarise · $\{Z_t\}$ are i.i.d. with distribution π. · $\{A_t\}$ are i.i.d. with $\Pr(A_t = i) = a_i$ · $\{S_t\}$ are i.i.d. random variables defined on the set $\{0, 1, \ldots q\}$ with $\Pr(S_t = j) = b_j$ · $\{V_t\}$ are i.i.d. Bernoulli variables with $\Pr(V_t = 1) = \alpha$, $0 \leq \alpha \leq 1$,
NDARMA(p, q)	$X_t = V_t\ X_{t-A_t} + (1-V_t)\ Z_{t-S_t}$.	· Jacobs and Lewis (1983) later simplified the two equations in one and named it the New DARMA model. · The model is a mixture of dependent variables with distribution π; $Z_{t-S_t} \sim \pi$ and $X_{t-A_t} \sim \pi$ therefore a mixture of these $X_t \sim \pi$. · Weiß (2011) further extend this class to generalised choice models, for modelling categorical process with serial dependence similar to an ARMA process, using a backshift mechanism.

Another class of general models that have been proposed for stationary discrete variate time series arises from the notion of a thinning operator by Steutel and Van Harn (1979). These are the most innovative time series models in the sense that they adapt the standard ARMA model to the count data by using the probabilistic operation of binomial thinning as an alternative to multiplication.

Definition 7 (Binomial Thinning Operator)**.** *Suppose X is a non-negative discrete random variable and consider $\alpha \in [0, 1]$. Then the binomial thinning operator $\circ$ is defined by*

$$\alpha \circ X = \sum_{i=1}^{X} Y_i,$$

where Y_i are i.i.d. Bernoulli random variables, independent of X, with $\Pr(Y_i = 1) = \alpha$ *and* $\Pr(Y_i = 0) = 1 - \alpha$*.*

In other words, $\alpha \circ X$ is the realised value of a binomial random variable with X trials and probability α of a success in each trial.

Stationary integer-valued series that imitate autoregressive moving-average methods and a variety of marginal distributions can be generated using the binomial thinning operation; binomial, Poisson, geometric, negative binomial, etc., (McKenzie, 1985, 1986, 1988; Al-Osh and Alzaid, 1987). Table 2.2 details the class of INARMA models where X_t in each of these models is the sum of a count random variable whose value depends on past outcomes and the realisation of an i.i.d. count random variable ε_t whose value does not depend on past outcomes.

Given a specified distribution for $\{\varepsilon_t\}$, the unconditional stationary distribution for $\{X_t\}$ can be found by probability generating function techniques, for example an INAR(1) model is found if

$$\{\varepsilon_t\} \sim \text{Poisson}\left(\frac{\lambda}{1-\alpha}\right) \quad \text{then} \quad \{X_t\} \sim \text{Poisson}\left(\lambda\right). \tag{2.8}$$

Higher-order integer-valued autoregressive processes of order-p, INAR(p), were proposed by Alzaid and Al-Osh (1990) and Jin-Guan and Yuan (1991) and they

Table 2.2: The INARMA time series models, where $\circ$ and $*$ denote the binomial and generalised thinning operator. Let $\{\varepsilon_t\}$ be a sequence of i.i.d. random variables with any arbitrary distribution given by π, then the distribution of $\{X_t\}$ can be found using probability generating function techniques. However they are not as simple as one would hope.

Name	Model
INAR(1)	$X_t = a_1 \circ X_{t-1} + \varepsilon_t,$
INAR(p)	$X_t = a_1 \circ X_{t-1} + \cdots + a_p \circ X_{t-p} + \varepsilon_t,$
INMA(q)	$X_t = \varepsilon_t + b_1 \circ \varepsilon_{t-1} + \cdots + b_q \circ \varepsilon_{t-q},$
INARMA(p, q)	$X_t = a_1 \circ X_{t-1} + \cdots + a_p \circ X_{t-p} + \varepsilon_t$ $+b_1 \circ \varepsilon_{t-1} + \cdots + b_q \circ \varepsilon_{t-q},$
Generalised INAR(p)	$X_t = a_1 \circ X_{t-1} * \cdots + a_p * X_{t-p} + \varepsilon_t.$

have been extended to generalised-INAR(p) (GINAR(p)) models using a generalised thinning operation (Latour, 1997, 1998). The realisations of thinned models seem to be more realistic than those of the DARMA models, however unlike DARMA models, thinning techniques cannot produce arbitrary unconditional marginal distributions.

2.2.1 Inference for discrete-valued time series

The class of integer-valued autoregressive-moving average, INARMA, processes can presently be considered as the major model for discrete valued time series, particularly INAR models. However, estimation of INARMA models is difficult due to the contribution of the thinning operator and the innovation sequence. The important distinction for INARMA models is that both parts of the model are stochastic. This results in complicated forms of likelihood functions involving convolutions even for the relatively simple INAR(1) process.

Al-Osh and Alzaid (1987) addressed the problem of estimating the parameter α and the mean parameter λ of a Poisson INAR(1) (2.8) process using three different types of estimators. Two of these methods use moment based methods and are asymptotically equivalent whilst the third method uses a maximum likelihood approach. Karlis and Xekalaki (2001) proposes an EM algorithm for maximum likelihood estimation for INAR models where the random components are unobserved and considered as missing data. Neal and Subba Rao (2007) outline an efficient MCMC algorithm for fitting INARMA processes. Drost et al. (2009) consider a semi-parametric maximum likelihood estimation for INAR(p) models and McCabe et al. (2011) present a non-parametric maximum likelihood estimation for INAR models for efficient probabilistic forecasts.

Zeger (1988), Zeger and Qaqish (1988), Brännäs and Johansson (1994), Davis et al. (2000) and Davis and Wu (2009) proposed regression models for time series count data based on the concepts of generalized linear models (Nelder and Wedderburn, 1972; McCullagh and Nelder, 1989) and attempted to incorporate both trend and serial correlation, by incorporating into the conditional mean function a latent autoregressive process which evolves independently of the past observed counts. This process introduces autocorrelation as well as over-dispersion into the model. Zeger (1988) gave an illustration for US Polio incidences, which has subsequently become a well known dataset and we will use for illustration in Chapter 7. Brännäs and Johansson (1994) extended Zeger's model to panel data. Closely related to regression models are the autoregressive conditional Poisson models of Heinen (2003). Kedem and Fokianos (2005), Fokianos (2009) and Cameron and Trivedi (2013, Ch. 7) review recent developments for regression models for time series of counts.

Models for multivariate count data, although not time series, have been considered in the literature. Chib and Winkelmann (2001) introduced a set of correlated latent effects to model the correlation among count data. Conditional on these latent effects, the counts are assumed independent Poisson, with a conditional mean function that depends on the latent effect. The result is a Poisson log-normal model and is a multivariate mixing model which can be fit by a MCMC algorithm.

The focus of the time series models described above and in the previous section has been modelling the dependence as the primary interest. The marginal distribution of a stationary time series contains interesting information. It is regarded to be of primary interest with the dependence as second interest by some authors (Francq and Zakoïan, 2013; Masarotto et al., 2012). For example, Francq and Zakoïan (2013) estimate the parameterised marginal distribution of a stationary time series without specifying the dependence structure which is treated as a nuisance. Masarotto et al. (2012) consider a Gaussian copula marginal regression model for modelling marginal distributions with covariates where dependence is significant but treated as secondary interest. There is a lack of models with equal interest and importance on marginal and dependence modelling due to the absence of high-dimensional distributions available to researchers to model random vectors.[1] Copulas have been shown to be useful for high-dimensional modelling and also successful in applications of continuous time series data (Chen and Fan, 2006). As we will go on to show, copulas have the ability to create high-dimensional distributions which model the dependence and the marginal distributions with equal importance, thus increasing the range of models available for integer-valued time series. The copula model does not lead to simple expressions of conditional expectation.

[1] Each random variable $\{X_t\}$ follows some probability distribution where the random vector $\mathbf{X} = (X_1, \dots X_N)$ follows an N-dimensional probability distribution, i.e., a time series can be viewed as a single drawing from a multivariate distribution.

Chapter 3

Copulas

The central mathematical objects studied in this thesis are copulas and the necessary preliminary material for the Gaussian copula model is introduced in this chapter. Copulas and Sklar's Theorem are established in Section 3.1 and Section 3.2 introduced on copula modelling. The Gaussian copula model is introduced in Section 3.3 and its 'digitised' interpretation. Section 3.4 looks at the uniqueness of the Gaussian copula for discrete data. Inference is discussed in Section 3.5 and the likelihood given in Section 3.6. We finish this chapter with a discussion on the computational aspects of fitting the Gaussian copula model to discrete data in Section 3.7.

Definition 8 (Copula)**.** *An* N*-dimensional* copula *is a cumulative distribution function* C *of* N *variables such that the marginal distributions are uniform on* $[0,1]$*. Thus*

$$C(u_1, \dots, u_N) = \Pr(U_1 \le u_1, \dots, U_N \le u_N), \quad 0 \le u_1, \dots, u_N \le 1,$$

where each U_i *is uniformly distributed on* $[0,1]$.

The dependence information for the random variables $(U_1, \dots, U_N)$ is contained in C. When C is parametrised by a parameter vector $\boldsymbol{\theta}$, we call $\boldsymbol{\theta}$ the *dependence* parameter. We write $\boldsymbol{\theta}_{\text{copula}}$ to distinguish from the parameters of the marginal distributions.

For example the Clayton copula (Kimeldorf and Sampson, 1975) which is an asymmetric two-dimensional copula that allows for fat tails and exhibits greater dependence in the negative tail than in the positive is given by

$$C(u_1, u_2; \delta) = \left(u_1^{-\delta} + u_2^{-\delta} - 1\right)^{-(\frac{1}{\delta})},$$

where $0 < \delta < \infty$ is a parameter controlling the dependence. Perfect dependence is obtained when $\delta \to \infty$ while $\delta \to 0$ implies independence.

There are many other parametric families of copulas such as the Gaussian, Students t, Plackett, Frank, Gumbel etc, see Nelsen (1999) for a comprehensive list. Much of the work on copulas is for the purpose of modelling extreme events and long tails.

The condition that C is a distribution function with uniform marginals leads to the following properties (Nelsen, 1999):

a) $C : [0,1]^N \to [0,1]$;

b) $C(u_1, \dots, u_N)$ is increasing in each component u_i;

c) $C(1, \dots, 1, u_i, 1, \dots, 1) = u_i$ for all $i = 1, \dots, N$, $u_i \in [0,1]$, *and*

d) For any $(a_1, \dots, a_N), (b_1, \dots, b_N) \in [0,1]^N$ with $a_i \leq b_i$ we have

$$\sum_{i_1=0}^{1} \cdots \sum_{i_N=0}^{1} (-1)^{i_1+\cdots+i_N} C(u_{1 i_1}, \dots, u_{N i_N}) \geq 0$$

where $u_{i0} = a_i$ and $u_{i1} = b_i$, $i = 1, \dots, N$. The reason is that the left hand side is the probability that $(U_1, \dots, U_N)$ falls in $(a_1, b_1] \times \cdots \times (a_N, b_N]$.

Conversely every function C with these properties is a copula.

The copula density allows maximum likelihood estimation in the continuous case and will be looked at in Chapter 6. It can be interpreted as the strength of dependence between its uniform marginals.

Definition 9 (Copula density)**.** *A copula density is a multivariate probability density on* $[0,1]^N$ *having uniform marginals. The copula density is given by*

$$c(u_1, u_2, \dots, u_N) = \frac{\partial^N C(u_1, u_2, \dots, u_N)}{\partial u_1 \dots \partial u_N}, \quad 0 < u_1, \dots, u_N < 1, \tag{3.1}$$

when the copula C is absolutely continuous.

For example, the copula density for the Clayton copula is (Venter, 2001)

$$c(u_1, u_2; \delta) = (1+\delta)(u_1 u_2)^{-\delta-1}(u_1^{-\delta} + u_2^{-\delta} - 1)^{-2-1/\delta}.$$

3.1 Sklar's Theorem

The importance of copulas is rooted in Sklar's Theorem which states that any multivariate distribution can be represented as a copula function of its marginals.

Theorem 1 (Sklar's Theorem)**.** *Let F be a joint distribution function with marginals $F_1, \ldots, F_N$. Then there exists a copula $C : [0,1]^N \to [0,1]$ such that*

$$F(y_1, \ldots, y_N) = C(F_1(y_1), \ldots, F_N(y_N)), \quad y_1, \ldots, y_N \in \mathbb{R}. \tag{3.2}$$

If each F_i is continuous for $i = 1, \ldots, N$ then C is unique; otherwise C is uniquely determined only on Range $(F_1) \times \cdots \times$ Range (F_N), where Range (F_i) denotes $\{u_i : u_i = F(y), y \in \mathbb{Z}\}$.

Sklar's Theorem gives the existence of a copula C but it does not tell us how to find it. When the marginals F_i, $i = 1, \ldots, N$ are continuous $F_i(Y) \sim U(0,1)$.
If F_i^{-1} is the inverse of F_i and $U \sim U(0,1)$ then $F_i^{-1}(U) \sim F_i$ by the quantile transform (Joe, 2015, p. 8). Letting $u_i = F(y_i)$ in (3.2) we can write

$$C(u_1, \ldots, u_N) = F(F_1^{-1}(u_1), \ldots, F_N^{-1}(u_N)), \quad 0 \le u_1, \ldots, u_N \le 1. \tag{3.3}$$

This is the unique copula of F.

The Gaussian copula is the copula associated with the Gaussian distribution according to Sklar's Theorem. Let $\mathbf{Y} \sim N_N(\mathbf{0}, \Sigma)$ with zero mean, unit variance and cdf

$$F(y_1, \ldots, y_N) = \Phi_\Sigma(y_1, y_2, \ldots, y_N),$$

where Σ is the correlation/covariance matrix. The marginals $F_i(y_i)$ are standard normal so $F_i = \Phi$. Since the inverse of Φ exists we can write $y_i = \Phi^{-1}(\Phi(y_i))$ and

$$F(y_1, \ldots, y_N) = \Phi_\Sigma(\Phi^{-1}(\Phi(y_1)), \Phi^{-1}(\Phi(y_2)), \ldots, \Phi^{-1}(\Phi(y_N))). \tag{3.4}$$

Let $U_i = \Phi(Y_i)$, $i = 1, \ldots, N$ then $U_i \sim U[0,1]$. Therefore we can write $Y_i = \Phi^{-1}(U_i)$ and the cdf of $(U_1, \ldots, U_N)$ is

$$\begin{aligned} &\Pr(U_1 \leq u_1, \ldots, U_N \leq u_N) \\ =&\Pr(Y_1 \leq \Phi^{-1}(u_1), \ldots, Y_N \leq \Phi^{-1}(u_N)), \\ =&\Phi_\Sigma(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_N)), \end{aligned}$$

where $0 \leq u_1, \ldots, u_N \leq 1$. Thus

$$C(u_1, \ldots, u_N) = \Phi_\Sigma(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_N)), \tag{3.5}$$

with $0 \leq u_1, \ldots, u_N \leq 1$ is a cdf with uniform marginals. From (3.4) we can see that it is the Gaussian copula corresponding to $N_N(\mathbf{0}, \Sigma)$. Then by Sklar's Theorem, the Gaussian copula is (3.5). The density of the Gaussian copula is

$$c(u_1, \ldots, u_N) = \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(u_1, \ldots, u_N)^\top \left(\Sigma^{-1} - \mathbf{I}_N\right)(u_1, \ldots, u_N)\right\}, \tag{3.6}$$

where $0 \leq u_1, \ldots, u_N \leq 1$ (Song, 2000).

We use the Gaussian copula because of its special properties that it inherits from the Gaussian distribution such as the ability to accommodate complicated dependence structures (Embrechts, 2009, p. 191). The Gaussian copula has a very nice property that the form of a conditional Gaussian copula is unaffected by the value of the conditioning variable. Stöber (2013) calls this 'a copula of the simplified form'. This is an important property in vine copulas methodology as we will see in Chapter 6.

3.2 Copula models

Each marginal distribution F_i contains all the information on the individual variable Y_i, while the joint distribution F contains all the joint as well as marginal information. By Sklar's Theorem, the multivariate distribution function can be decomposed into two parts: a set of marginal distribution functions and the dependence structure which is specified in terms of its copula. This is instructive from a modelling perspective as it suggests a natural way to model multivariate data, in particular we can construct multivariate discrete models based on copulas with discrete margins; an area which currently presents challenges, as discussed in Section 2.2.

Copulas are flexible in use and have been applied in many subject areas. Fang et al. (2002) discuss the class of continuous copulas models and their properties which can be constructed using elliptical copulas with continuous marginals. In practice choosing and estimating a useful form of the marginal distribution of each variable is often a straightforward task because there is an extensive library of univariate distributions to choose from and the marginal distribution can be estimated non-parametrically to get an idea of what the true distribution might be. Then copulas can be used for dependence modelling between the variables.

The parameters in a copula reflect the degree of dependence among variables; for example, the dependence for the multivariate Gaussian distribution is summarised in the correlation matrix Σ of which the elements are the pairwise correlation coefficients, with large correlation coefficients indicating strong dependence among variables.

The Gaussian copula is parameterised by its variance/ correlation matrix Σ. An unstructured correlation matrix can become unmanageable for estimation and common choices for a structured parametrisation of Σ include compound symmetry, Toeplitz, autoregressive of order one and autoregressive-moving average, two of which are shown in (3.7) in dimension N=5,

$$\Sigma^{\mathrm{CS}} = \begin{pmatrix} 1 & a & a & a & a \\ a & 1 & a & a & a \\ a & a & 1 & a & a \\ a & a & a & 1 & a \\ a & a & a & a & 1 \end{pmatrix}, \qquad \Sigma^{\mathrm{AR(1)}} = \begin{pmatrix} 1 & a & a^2 & a^3 & a^4 \\ a & 1 & a & a^2 & a^3 \\ a^2 & a & 1 & a & a^2 \\ a^3 & a^2 & a & 1 & a \\ a^4 & a^3 & a^2 & a & 1 \end{pmatrix}. \tag{3.7}$$

3.3 The Gaussian copula model

Let $\mathbf{Y} = (Y_1, \ldots, Y_N)$ be a random vector with a Gaussian copula (3.5) and discrete marginal distribution functions F_i, $i = 1, \ldots, N$. The joint distribution F of $\mathbf{Y}$ is

$$F(y_1, \ldots, y_N) = \Phi_\Sigma\left(\Phi^{-1}(F_1(y_1)), \ldots, \Phi^{-1}(F_N(y_N))\right), \tag{3.8}$$

where Σ is the correlation matrix associated with the Gaussian copula. This is the Gaussian copula model and the main subject of this study. We are interested in this model for a single time series of counts.

Gaussian copulas are a natural choice when moving beyond the bivariate case of copula modelling because they correspond to classical multivariate normal methods after variables have been transformed to $N(0, 1)$ and therefore the methodology has developed faster than other copula models.

The model has been used by many authors including Song (2000), Frey et al. (2001), Renard and Lang (2007) for continuous data to name a few. It has been applied to discrete data by Pitt et al. (2006), Hoff (2007), Song et al. (2009), Danaher and Smith (2011), Smith and Khaled (2012), Panagiotelis et al. (2012), Masarotto et al. (2012), Nikoloulopoulos (2013) and Shi and Valdez (2014).

The Gaussian copula model (3.8) will be satisfied if $\mathbf{Y} = (Y_1, \ldots, Y_N)$ can be written as

$$Y_i = F_i^{-1}(\Phi(X_i)) \qquad i = 1, \ldots, N, \tag{3.9}$$

where $\{X_i\}$ are standard normal with variance/correlation matrix Σ and F_i^{-1} denotes the generalised inverse of the cdf F_i of Y_i. The generalised inverse F^{-1} of F, $F^{-1} : [0, 1] \to \mathbb{R}$, is defined by

$$F^{-1}(p) = \inf\{z \in \mathbb{R} : F(z) \geq p\}, \qquad p \in (0, 1), \tag{3.10}$$

where we use the convention $\inf \emptyset = \infty$.

Proof of (3.9) $\implies$ (3.8).

Under (3.9), the cdf of $(Y_1, \ldots, Y_N)$ *is given by*

$$\begin{aligned}
&\Pr(Y_1 \leq y_1, \ldots, Y_N \leq y_N)\\
=&\Pr(F^{-1}(\Phi(X_1)) \leq y_1, \ldots, F^{-1}(\Phi(X_N)) \leq y_N),\\
=&\Pr(\Phi(X_1) \leq F(y_1), \ldots, \Phi(X_N) \leq F(y_N)),\\
=&\Pr(X_1 \leq \Phi^{-1}(F(y_1)), \ldots, X_N \leq \Phi^{-1}(F(y_N))),\\
=&\Phi_\Sigma(\Phi^{-1}(F(y_1)), \ldots, \Phi^{-1}(F(y_N))).
\end{aligned}$$

Therefore the cdf of $(Y_1, \ldots, Y_N)$ *is (3.8).* □

Thus is $\mathbf{X}$ has a multivariate normal distribution and Y_i is defined via (3.9) then $\mathbf{Y}$ has joint distribution given by (3.8). Therefore the Gaussian copula model is a latent Gaussian model where the dependence in $\mathbf{Y}$ is specified indirectly through $\mathbf{X}$.

3.3.1 Conditional distribution of the latent Gaussian

For discrete marginals, the unobserved latent X_i cannot be directly recovered from the observed Y_i by inverting (3.9) because F_i is a step function with generalised inverse defined in (3.10). In this section, we prove a key result from a modelling perspective; that the observed discrete data translates to intervals for the latent Gaussian variables, consequently facilitating maximum likelihood estimation using an EM algorithm in Chapter 4.

Proposition 1. *Let X be a continuous random variable, Φ the standard normal cdf and let F be a cdf on $\mathbb{Z}$. Suppose $Y = F^{-1}(\Phi(X))$ then*

$$F(Y-1) \quad < \quad \Phi(X) \quad \leq \quad F(Y).$$

Proof.

Consider the set $A = \{y \in \mathbb{R} : F(y) \geq p\}$ *for some* $0 < p < 1$. *For any* $n > 1$, $F^{-1}(p) + \frac{1}{n}$ *is not a lower bound of* A. *Thus there exists* $y_n \in A$ *such that*

$$y_n \; < \; F^{-1}(p) + \frac{1}{n},$$

which implies

$$F\left(F^{-1}(p) + \frac{1}{n}\right) \;\geq\; F(y_n) \geq p.$$

Let $n \to \infty$, *then* $F(F^{-1}(p)) \geq p$ *by the right continuity of* F. *Let* $p = \Phi(X)$ *then*

$$F(F^{-1}(\Phi(X))) \;\geq\; \Phi(X).$$

Thus

$$F(Y) \;\geq\; \Phi(X).$$

If, for contradiction, $F(Y-1) \;\geq\; \Phi(X)$ *then* $Y-1$ *belongs to the set* A. *But* Y *is the infimum of the set* A, *it follows that* $Y-1$ *is not in* A. *Therefore* $F(Y-1) \;<\; \Phi(X)$. *Hence*

$$F(Y-1) < \Phi(X) \leq F(Y).$$

□ Because $F(Y-a), a > 0$ only changes at integer-valued a, we take $a = 1$ so that $Y - a$ is closest to Y.

Lemma 1. *Let* $Y = F^{-1}(\Phi(X))$, *where* X *is a continuous random variable and* F *is the distribution function of an integer-valued random variable. Then*

$$\begin{aligned} Y = y &\iff F(y-1) < \Phi(X) \leq F(y), \\ &\iff \Phi^{-1}(F(y-1)) < X \leq \Phi^{-1}(F(y)). \end{aligned}$$

This lemma follows immediately from Proposition 1 by the definition of the infimum and since Φ^{-1} is a continuous, monotonically increasing function.

We have shown that $Y_i = F^{-1}(\Phi(X_i))$ (3.9) is proportional to

$$\Phi^{-1}(F(Y_i - 1)) < X_i \leq \Phi^{-1}(F(Y_i)), \quad i = 1, \ldots, \textsc{n},$$

and hence when Y_i takes the value y_i we have an interval for the unobserved X_i given by

$$\Phi^{-1}(F(y_i - 1)) < X_i \leq \Phi^{-1}(F(y_i)). \tag{3.11}$$

Essentially by conditioning on the given data, the range of values that X can take is truncated. A truncated distribution is a conditional distribution when the domain of a parent distribution is restricted to a smaller region. For example, a truncated normal distribution is a normal distribution that is restricted to lie within a finite range (Tallis, 1961). If X is $N(\mu, \sigma^2)$ truncated on $(k, l]$, $X \sim TN(\mu, \sigma^2, k, l)$, then X has pdf

$$f(x) = \begin{cases} \dfrac{\phi(\frac{x-\mu}{\sigma})}{\Phi(\frac{l-\mu}{\sigma}) - \Phi(\frac{k-\mu}{\sigma})} & k < x \leq l, \\ 0 & \text{otherwise,} \end{cases}$$

where the denominator is the scaling factor to account for the probability of falling in the region $(k, l]$ and Φ and ϕ are the standard normal cdf and pdf respectively. Therefore when X has a Gaussian distribution $X \mid Y = y$ has a truncated normal distribution on the interval

$$\Phi^{-1}(F(y - 1)) < X \leq \Phi^{-1}(F(y)). \tag{3.12}$$

Therefore for our model (3.9) and observed $\mathbf{y} = (y_1, \ldots, y_N)$ then $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$ has a truncated normal distribution on the interval $(\Phi^{-1}(F(\mathbf{y} - 1)), \Phi^{-1}(F(\mathbf{y}))]$. The pdf of the multivariate truncated normal distribution $TN_N(\boldsymbol{\mu}, \Sigma, \mathbf{k}, \mathbf{l})$ is given by

$$f(\mathbf{x}) = \begin{cases} \dfrac{\exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\}}{\int_{\mathbf{k}}^{\mathbf{l}} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\} \, \mathrm{d}\mathbf{x}} & \mathbf{k} < \mathbf{x} \leq \mathbf{l}, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} \mathbf{k} = (k_1, \ldots, k_N) &= \left(\Phi^{-1}(F(y_1 - 1)), \ldots, \Phi^{-1}(F(y_N - 1))\right), \\ \mathbf{l} = (l_1, \ldots, l_N) &= \left(\Phi^{-1}(F(y_1)), \ldots, \Phi^{-1}(F(y_N))\right). \end{aligned}$$

are vectors of truncation points for $\mathbf{X}$.

The most instructive amount of information we can infer about the random variables $\mathbf{X}$ is that they are within the interval specified by

$$\left(\Phi^{-1}(F(\mathbf{y}-1)), \Phi^{-1}(F(\mathbf{y}))\right].$$

This is relevant for the E-step of the EM algorithm of Chapter 4 since the conditional distribution of $\mathbf{X}$ given our observed discrete data $\mathbf{y}$, that is $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$, has a truncated normal distribution on the interval

$$\left(\Phi^{-1}(F(\mathbf{y}-1)), \Phi^{-1}(F(\mathbf{y}))\right] \in \mathbb{R}^{N}. \tag{3.13}$$

The truncated multivariate distribution can be sampled from directly, as we will describe in Section 4.3.1 and has been well studied.

As a side note, the result is similar when we consider the conditional distribution of $\Phi(X)$ given the observed data y, $\Phi(X) \mid Y = y$. The variable $\Phi(X)$ is uniformly distributed on $[0, 1]$ and $\Phi(X) \mid Y = y \sim U\left[F(y-1), F(y)\right]$, i.e, by conditioning we truncate the range of values that $\Phi(X)$ can take to $[F(y-1), F(y)]$.

3.3.2 The Gaussian copula model for time series

By specifying an ARMA(p, q) process for $\{X_i\}$, the correlation matrix Σ is completely determined by the parameters $a_1, \ldots, a_p, b_1, \ldots, b_q$. We interpret the Gaussian copula model as a transformation such that $(X_1, \ldots, X_N)$ is a latent ARMA process, $\mathbf{X} \sim N_N(\mathbf{0}, \Sigma)$ where Σ is an $N \times N$ Toeplitz matrix of autocorrelations, that is with (i, j) elements $r(i-j)$ (Definition 4). Thus the model (3.8) can be considered as a *'digitised'* ARMA(p, q) model.

Copulas have been used in univariate continuous time series analysis to model the serial dependence with AR(1) and AR(2) models (Joe, 1997, Ch. 8). We take the dependence structure of the well-established ARMA model and we apply it to discrete marginals to glue them together creating a discrete model with time series dependence, for modelling integer-valued time series data. This is the rationale for the Gaussian copula with time series dependence and discrete margins studied in this thesis. We make use of the underlying ARMA structure for interpretablity and to develop methods for parameter estimation.

3.3.3 Examples of models

Any discrete univariate distribution can be specified for F_i, including univariate distributions with explanatory variables, making the model highly flexible. An example with explanatory variables is given in Chapter 7 on a well known polio data set (Zeger, 1988). The marginal distributions of integer-valued time series are often highly skewed with low means and zeros are not uncommon. Common choices for count data are the Poisson, geometric, negative binomial, zero-inflated Poisson and the zero-inflated negative binomial (Johnson et al., 2005; Mullahy, 1986; Greene, 1994).

In this thesis, we use the negative binomial distribution throughout to account for over/under-dispersion within the time series. Due to the many different models that give rise to the negative binomial distribution, there are a variety of definitions in the literature. We use the following definition which allows for $s \in \mathbb{R}_+^*$.

Negative Binomial Family A random variable Y is said to have a negative binomial distribution with parameters $0 < \pi < 1$ and $s \in \mathbb{R}_+^*$ if

$$\Pr(Y = y) = \frac{\Gamma(s+y)}{y!\,\Gamma(s)}\pi^s(1-\pi)^y, \quad y = 0, 1, 2, \ldots.$$

To avoid misspecification of the marginal distributions, the semi-parametric Gaussian copula model can be used. A semi-parametric copula model can be created by estimating either the dependence structure or the marginals non-parametrically. That is, without placing any assumptions on their parametric form. An example is provided for the polio data set in Sections 7.1.2 and 7.3.

3.4 Uniqueness of copula representation

There is a general lack of uniqueness of a copula representation (3.3) for discrete distributions. In the discrete case, Sklar's Theorem guarantees that there exists a copula for F in (3.3) but there are several functions that satisfy (3.3). These functions are not all guaranteed to be a copula, or even a distribution and an identifiability issue arises. For discrete F_i, the distribution of $U_i = F_i(y_i)$ does not follow a uniform distribution on $(0, 1)$. The identifiability affects the inference because of the ties introduced from the discrete cdfs.

We still have the same equation given in (3.3) but it is only for $\{u_i : u_i = F(y), y \in \mathbb{Z}\}$ and other copulas exist for the same distribution. The copula C is not unique outside the set

$$\text{Range}(F_1) \times \cdots \times \text{Range}(F_{\scriptscriptstyle N}). \tag{3.14}$$

To assess the extent of unidentifiability, Carley (2002) derived sharp bounds that can be applied to any copula compatible with F. In the continuous case the copula C characterises the dependence structure but in the discrete case this cannot be said. However, F often inherits dependence properties from C and Genest & Nesklova state copulas are still valid and the copula parameters continues to govern association between discrete margins. Trivedi and Zimmer (2007) suggests that non-uniqueness is a theoretical issue to be confronted in analytical proofs but it does not inhibit empirical applications. Copulas are still used for discrete distributions (Nikoloulopoulos and Karlis, 2009). We discuss the use of continuous copulas for discrete distributions in the context of vine copulas in Section 6.4.1 where we choose an interpolated version of the Gaussian copula.

Due to the inferential issues of applying copulas to discrete data, methods for indirect applications of copula models to discrete data have been proposed. Denuit and Lambert (2005) suggest a 'continuising' technique to make the data appear continuous. The discrete variable Y is 'extended' to a continuous variable X defined by

$$X = Y + (U - 1), \tag{3.15}$$

where U is a continuous random variable in $(0, 1)$ independent of Y. Shi and Valdez (2014) adopt this method to apply the Gaussian copula model with negative binomial margins to insurance claims data in a longitudinal context with the name 'jittering'. Wu (2013) also follows this continuising approach when modelling binary data. Wu proposes a joint model for the mixed outcomes (that is discrete Y and the continuised X in (3.15)) using a Gaussian copula. We do not take the indirect approach. We apply the Gaussian copula model directly to the integer-valued time series as do many others including Danaher and Smith (2011), Panagiotelis et al. (2012) and Masarotto et al. (2012).

3.5 Inference from copula models

Maximum likelihood is the preferred method of estimation so that the standard tools for hypothesis testing and model selection can be used, such as likelihood ratio statistics and information criteria. The MLE is generally found by numerical maximisation of the log-likelihood since analytical solutions are rare for complex likelihood functions. However, typically copula models are high-dimensional and require integration when the copula does not have a closed form making MLE difficult if not impossible.

Two-Stage Estimation

To construct a multivariate model, the copula C and the marginal distributions F_i can be separately chosen. The separation of the dependence and marginal part of the model is a very useful consequence because different estimation methods can be used for the components.

A common approach to estimating copula models is to perform a two stage procedure named inference functions for margins (IFM) (Xu, 1996). First the marginal parameters are separately estimated. Then the dependence structure is estimated by optimising the likelihood, with the marginal estimates plugged in, as a function of dependence parameters only.

This inference method is known more generally as multi-stage maximum likelihood estimation. The two-stage estimation is used only for computation and not for theoretical analysis. There is a trade-off between the efficiency of the IFM estimates and the numerical instabilities encountered in maximum likelihood (Min and Czado, 2010). The asymptotic efficiency of IFM has been studied by Joe (2005) and comparisons suggest that the IFM method is highly efficient when compared to standard MLE. However, this two-stage estimation method remains difficult for the Gaussian copula model because the dependence structure is still high dimensional and the estimates remain hard to compute. Zhao and Joe (2005) suggest composite likelihood methods of which IFM is a special case. Method of moment estimators in low dimensions have also been proposed (Genest and Rivest, 1993), however for dimensions typical of time series we require other estimation methods to be developed.

Semi-parametric Estimation

Continuing along similar lines using two stage estimation, a semi-parametric model can be created by estimating either the dependence structure or the marginals non-parametrically. That is, without placing any assumptions on their parametric form. To estimate the copula C non-parametrically, Deheuvels (1979) introduced the empirical copula and Chen and Huang (2007) proposed kernel based estimators. However, in this work we estimate the Gaussian copula parametrically since the interest is in the dependence structure of the Gaussian copula and characterising the dependence of the integer values over time.

To estimate the marginal distributions F_i non-parametrically, the empirical distribution function can be used

$$\widehat{F}_i(y) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y_i \leq y). \tag{3.16}$$

When non-parametric estimates of the distributions are used then the IFM method is known as *canonical maximum likelihood* or *pseudo-maximum likelihood* (Genest et al., 1995; Tsukahara, 2005).

3.6 The likelihood function for discrete data

Zimmer and Trivedi (2006) highlight that the challenges that high-dimensional discrete distributions pose in copula modelling are because the likelihood function differs substantially from that in the continuous case.

Consider discrete random variables $Y_i, i = 1, 2, \ldots, N$, with cdfs $F_1, \ldots, F_N$. In the case of variables with integer values the left-hand limit of F at y_i is $F(y_i - 1)$. The joint distribution of the N-dimensional random vector $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_N)$ completely defines the dependence and marginal properties of Y_i. The joint distribution can be written as $F = C(F_1, \ldots, F_N)$ by Sklar's Theorem and the multivariate probability mass function (pmf) is obtained from rectangle probabilities. For example, in the bivariate case the joint distribution of Y_1 and Y_2 where F_1, F_2 have support on the integers can be expressed as $F(y_1, y_2) = C_{12}(F_1(y_1), F_2(y_2))$. Writing $C(F(y_i - \ell_1), F(y_j - \ell_2))$ as $C_{ij}^{\ell_1\ell_2}$, the probability mass function is

$$\begin{aligned}
&\Pr(Y_1 = y_1, Y_2 = y_2)\\
=&\Pr(y_1 - 1 < Y_1 \leq y_1,\, y_2 - 1 < Y_2 \leq y_2),\\
=&F(y_1, y_2) - F(y_1, y_2 - 1) - F(y_1 - 1, y_2) + F(y_1 - 1, y_2 - 1),\\
=&C(F(y_1), F(y_2)) - C(F(y_1), F(y_2 - 1)) - C(F(y_1 - 1), F(y_2)) + C(F(y_1 - 1), F(y_2 - 1)),\\
=&C_{12}^{00} - C_{12}^{01} - C_{12}^{10} + C_{12}^{11}.
\end{aligned}$$

By induction, the joint probability mass function for N variables consists of 2^N finite differences of the copula (Nelsen, 1999),

$$\begin{aligned}
&\Pr(Y_1 = y_1, \ldots, Y_N = y_N) \\
=&\sum_{\ell_1=0}^{1}\sum_{\ell_2=0}^{1}\cdots\sum_{\ell_N=0}^{1}(-1)^{\ell_1+\cdots+\ell_N}\Pr(Y_1 \le y_1-\ell_1, \ldots, Y_N \le y_N-\ell_N), \\
=&\sum_{\ell_1=0}^{1}\sum_{\ell_2=0}^{1}\ \ldots\ \sum_{\ell_N=0}^{1}(-1)^{\ell_1+\cdots+\ell_N}\, C(F_1(y_1-\ell_1), \ldots, F_N(y_N-\ell_N)). \qquad (3.17)
\end{aligned}$$

This becomes costly for large N, especially when the N-dimensional copula is difficult to compute. For example in N=3 dimensions, the Gaussian copula pmf is

$$\begin{aligned}
\Pr(Y_1 = y_1, \ldots, Y_N = y_N) =&\Phi_\Sigma(\Phi^{-1}(F_1(y_1^-)), \Phi^{-1}(F_2(y_2^-)), \Phi^{-1}(F_3(y_3^-))) \\
&+ \Phi_\Sigma(\Phi^{-1}(F_1(y_1^-)), \Phi^{-1}(F_2(y_2^-)), \Phi^{-1}(F_3(y_3))) \\
&+ \Phi_\Sigma(\Phi^{-1}(F_1(y_1^-)), \Phi^{-1}(F_2(y_2)), \Phi^{-1}(F_3(y_3^-))) \\
&- \Phi_\Sigma(\Phi^{-1}(F_1(y_1^-)), \Phi^{-1}(F_2(y_2)), \Phi^{-1}(F_3(y_3))) \\
&+ \Phi_\Sigma(\Phi^{-1}(F_1(y_1)), \Phi^{-1}(F_2(y_2^-)), \Phi^{-1}(F_3(y_3^-))) \\
&- \Phi_\Sigma(\Phi^{-1}(F_1(y_1)), \Phi^{-1}(F_2(y_2^-)), \Phi^{-1}(F_3(y_3))) \\
&- \Phi_\Sigma(\Phi^{-1}(F_1(y_1)), \Phi^{-1}(F_2(y_2)), \Phi^{-1}(F_3(y_3^-))) \\
&+ \Phi_\Sigma(\Phi^{-1}(F_1(y_1)), \Phi^{-1}(F_2(y_2)), \Phi^{-1}(F_3(y_3))).
\end{aligned}$$

3.7 Computational aspects of the Gaussian copula

The Gaussian copula lacks a closed form cdf which means that high-dimensional integration is required to evaluate multivariate probabilities. Likelihood based estimation evaluates the integral either numerically or by simulation techniques, as we will discuss further in Section 6.4.2, resulting in significant computational demand. One likelihood evaluation for the model (3.8) requires finite differencing of 2^N evaluations of the N-dimensional multivariate Gaussian cdf meaning maximum likelihood estimation is not feasible for large N. Direct evaluation of the model for integer-valued time series data of length N=20 on a 1.4 GHz Intel Core i5 CPU takes over 6 days for one evaluation. Even so, there have been significant efforts to apply elliptical copulas and in particular the Gaussian copula to discrete data (Pitt et al., 2006; Hoff, 2007; Song et al., 2009; Danaher and Smith, 2011; Smith and Khaled, 2012; Panagiotelis et al., 2012; Masarotto et al., 2012; Nikoloulopoulos, 2013; Shi and Valdez, 2014).

In the longitudinal setting, with short time series, the Gaussian copula has been successfully fitted using maximum likelihood to count data to model the number of insurance claims over 8 years (Shi and Valdez, 2014).

Using an importance sampling approach, Masarotto et al. (2012) approximate the likelihood of the Gaussian copula model in a simulation based approach. The Gaussian copula model can be seen as a multivariate probit (Ashford and Sowden, 1970), which has been known for some time with estimation methods developed. The authors generalise methods for probit regression models to the Gaussian copula model where they consider the discrete and continuous cases separately. For discrete margins, the authors approximate the likelihood using a sequential sampling algorithm resulting in a fast computational algorithm, typically in the seconds and they give examples of applications for time series, spatial data and survival analysis. Masarotto et al.'s method uses the truncated normal as the importance density, which we discussed in Section 3.3.1.

The conditional distribution of X_t given X_{t-1} e.t.c depends on both the discrete marginals and the dependence between the random variables. The copula approach is to separate these two things so that they can be specified separately but the copula model does not lead to simple expressions of conditional expectation.

From a Bayesian modelling approach, the latent Gaussian representation (3.12) lends easily to Bayesian data augmentation and MCMC methods are widely used for latent variable/missing data problems. In a semi-parametric approach, Hoff (2007) uses a MCMC algorithm based on Gibbs sampling to fit a Gaussian copula model using an extended rank likelihood, where there are no parametric assumptions made on the marginal distribution and their focus is only on the dependence structure. Pitt et al. (2006) introduce a general Bayesian Markov chain Monte Carlo (MCMC) estimation of a Gaussian copula model for continuous, discrete and mixed margins. The latent variable approach (Section 3.3) is used with the likelihood for the augmented observed and unobserved variables written as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{X}) &= f(\mathbf{y}, \mathbf{X}|\boldsymbol{\theta}),\\ &= \prod_{i=1}^{N} \mathbb{1}(\Phi^{-1}(F_i(y_i - 1)) < X_i \leq \Phi^{-1}(F_i(y_i))\phi_{\Sigma}(X_1, \ldots, X_{N}|\boldsymbol{\theta}),\end{aligned}$$

where $\mathbb{1}$ is the indicator function. The latent variables are integrated out using Monte Carlo. The realisations x_i of the latent variables are generated explicitly in a Bayesian MCMC simulation algorithm. The priors for the correlation matrix of the Gaussian copula, and also the Bayesian selection framework, are unaffected by whether the data is discrete or continuous (Smith, 2011). The Bayesian MCMC proposed by Pitt et al. (2006) involves a complex prior for the correlation matrix which is not easily applied to high-dimensions. Danaher and Smith (2011) extend the algorithm to allow higher dimensions using a cholesky decomposition of the correlation matrix in which they demonstrate with N=45 in an example of joint modelling of website views. Smith and Khaled (2012) extend further outside the elliptical family of copula models. Smith (2011) bridge the gap between the Bayesian and copula communities by providing a detailed review of Bayesian methods for copula modelling, with a focus on the Gaussian and D-vine copulas. We do not pursue MCMC methods as our primary interest in this work is maximum likelihood estimation. However, we undertook preliminary work which suggested MCMC methods were computationally intensive for large $N$and indicated the estimates were poor. Panagiotelis et al. (2012) support this attitude saying that these methods are still computationally intensive and may not scale easily to higher dimensions.

3.7.1 Summary

As seen in Section 2, models for integer-valued time series have so far failed to achieve equal importance for the marginal and dependence structure. The key advantage of copula modelling for discrete data is the ability to combine completely different univariate distributions into a well-defined multivariate model where no approximations are required in high dimensions. The Gaussian copula model lends easily as a *'digitised'* ARMA model with flexible marginal specification and interpretable parameters.

The high-dimensionality of time series are not typically seen in the multivariate models and by incorporating classical time series ideas into modern copula estimation we open up the avenues for methods of parameter estimation that cannot be achieved through direct maximum likelihood. The subsequent three chapter aims to simplify the computation by using copula theory combined with well-studied time series analysis. Each approach proposed has its own merits, which can be seen when using integer-valued time series of different lengths.

Chapter 4

Estimation I: A Monte Carlo EM Algorithm

In this chapter a Monte Carlo Expectation Maximisation (MCEM) algorithm is considered for parameter estimation of the Gaussian copula model considered in Chapter 3.

The Gaussian copula model for discrete margins has the ability to accommodate complicated correlation structures. However, the log-likelihood

$$\ell(\boldsymbol{\theta};\mathbf{y}) = \sum_{\ell_1=0}^{1} \cdots \sum_{\ell_N=0}^{1} (-1)^{\ell_1+\cdots+\ell_N}\, \Phi_\Sigma(\Phi^{-1}(F_1(y_1-\ell_1)),\ldots,\Phi^{-1}(F_N(y_N-\ell_N))), \quad (4.1)$$

is difficult to maximise when N is moderately large because one evaluation of the function involves 2^N evaluations of the N-dimensional Gaussian distribution function Φ_Σ, where Σ is the $N \times N$ correlation matrix of the Gaussian copula and N is the length of the observed time series $\mathbf{y} = (y_1,\ldots,y_N)^\top$. We seek a computationally efficient method for finding the maximum likelihood estimates. The Expectation Maximisation (EM) algorithm can be used for this purpose because $\{y_t\}$ can be thought of as a digitised version of a real-valued time series $\{x_t\}$, the missing data as discussed in Section 3.3.

4.1 The EM algorithm

The EM algorithm (Dempster et al., 1977) is an iterative algorithm designed to compute maximum likelihood estimates, for situations where the observed data can be augmented to create a simpler likelihood function to maximise. McLachlan and Krishnan (2007) summarise the large variety of uses and applications since the first 1700 publications involving the EM algorithm.

Let the incomplete data log-likelihood be denoted as $\ell(\boldsymbol{\theta};\mathbf{y})$ and the complete data log-likelihood as $\ell(\boldsymbol{\theta};\mathbf{y},\mathbf{x})$. The EM algorithm works on the complete data log-likelihood $\ell(\boldsymbol{\theta};\mathbf{y},\mathbf{x})$, yet it can be shown (Dempster et al., 1977 and Mclachlan and Krishnan, 2007, p. 78) that the log-likelihood of the incomplete data (observed data) improves through the iterations, i.e.,

$$\ell(\boldsymbol{\theta}^{(i)};\mathbf{y}) \;\geq\; \ell(\boldsymbol{\theta}^{(i-1)};\mathbf{y}).$$

Furthermore, if these iterations converge then they converge to a local stationary point of the log-likelihood function $\ell(\boldsymbol{\theta};\mathbf{y})$ (Wu, 1983).

The algorithm alternates between two stages, the E-step (Expectation) and the M-step (Maximisation), where the parameter values are updated repeatedly until a convergence criterion is met. More specifically, the E-step estimates the complete data log-likelihood by taking its conditional expectation given $\mathbf{y}$ and evaluating it using the current parameter values $\boldsymbol{\theta}^{(i-1)}$. Therefore the E-step evaluates the objective function

$$Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(i-1)}\right) = \mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[\ell\left(\boldsymbol{\theta};\mathbf{y},\mathbf{X}\right) \mid \mathbf{Y}=\mathbf{y}\right], \tag{4.2}$$

whilst the M-step maximises the objective function with respect to $\boldsymbol{\theta}$ to obtain an updated $\boldsymbol{\theta}^{(i)}$. The M-step is therefore finding

$$\boldsymbol{\theta}^{(i)} = \arg\max_{\boldsymbol{\theta}}\; Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{(i-1)}\right).$$

We have seen in Chapter 3 that the Gaussian copula model with specified marginals F_t is equivalent to

$$Y_t = F_t^{-1}(\Phi(X_t)), \qquad t = 1, \ldots, \textsc{n}, \tag{4.3}$$

where $\{X_t\}$ is an unobserved latent process, having the same Gaussian copula as $\{Y_t\}$. This latent variable perspective of the copula model facilitates maximum likelihood estimation via an EM algorithm. If the latent process $\{X_t\}$ is observed the complete data log-likelihood $\ell\,(\boldsymbol{\theta}; \mathbf{y}, \mathbf{x})$ is easier to compute. We consider the latent process as missing data, and augment the observed data $\mathbf{y}$ with the missing data $\mathbf{x}$, to apply the EM algorithm.

In the following sections we derive the E and M-steps of the EM algorithm for the Gaussian copula model and calculate the standard errors of the parameter estimates. The expectation of the latent process $\mathbf{X}$ given the observed count data, $\mathbf{Y} = \mathbf{y}$, is not easy to evaluate, even though we know the conditional distribution is truncated normal. A Monte Carlo E-step is therefore used resulting in a Monte Carlo EM algorithm (Wei and Tanner, 1990), where a Monte Carlo estimate of the Q function (4.2) is used in the M-step. We implement a Geweke-Hajivassilou-Keane (GHK) simulator (Geweke, 1989. Hajivassiliou and McFadden, 1998, and Keane, 1994) to sample directly from the conditional distribution of $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$ (Section 3.3.1).

4.2 The E-step

The latent process $\{X_t\}$ has a Gaussian structure and completely determines the observed data y_t. Therefore the log-likelihood of $\mathbf{X}$ is the complete data log-likelihood

$$\ell(\boldsymbol{\theta}; \mathbf{y}, \mathbf{x}) = -\frac{N}{2} \log{(2\pi)} - \frac{1}{2} \log{|\Sigma|} - \frac{1}{2} \mathbf{x}^\top \Sigma^{-1} \mathbf{x}. \tag{4.4}$$

This log-likelihood involves a large variance-covariance matrix that becomes difficult to invert for large dimensions. To overcome this, we use the time series approach (Brockwell and Davis, 1987, Ch. 5) to simplify the log-likelihood. Consider the prediction error decomposition of Σ and write the log-likelihood (4.4) in terms of

$$\begin{aligned} e_1 &= x_1, \\ e_2 &= x_2 - \phi_{1,1}x_1, \\ e_3 &= x_3 - (\phi_{2,1}x_2 + \phi_{2,2}x_1), \\ &\vdots \\ e_N &= x_N - (\phi_{N-1,1}x_{N-1} + \cdots + \phi_{N-1,N-1}x_1), \end{aligned}$$

where $\phi_{t1}, \ldots, \phi_{tt}$ are coefficients of the best linear predictor of x_{t+1} at time t using all previous values $x_t, \ldots, x_1$:

$$\widehat{x}_{\,t+1} = \phi_{t1}x_t + \phi_{t2}x_{t-2} + \cdots + \phi_{tt}x_1.$$

This one-to-one transformation between $\mathbf{e}$ and $\mathbf{x}$ in matrix form is

$$\mathbf{e} = \Psi\mathbf{x} \qquad \text{or} \qquad \Theta\mathbf{e} = \mathbf{x},$$

where the transformation matrix Ψ and inverse transformation matrix $\Theta = \Psi^{-1}$ are lower triangular with ones on its diagonal giving the determinant of the Jacobian=1.

Thus e_t is the error from predicting x_t (the prediction error), on the basis of the information available at time $t-1$. The prediction errors $\mathbf{e} = (e_1, \ldots, e_N)^\top$ are uncorrelated and it follows that

$$\Sigma = \mathrm{var}\,(\mathbf{x}) = \mathrm{var}\,(\Theta\mathbf{e}) = \Theta\mathrm{var}\,(\mathbf{e})\,\Theta^\top = \Theta\Gamma\Theta^\top,$$

where $\Gamma = \mathrm{diag}\,(\tau_1^2, \ldots, \tau_N^2)$ is the variance-covariance matrix of $\mathbf{e}$. The Levinson-Durbin algorithm given in (2.7) provides the elements of the matrices Γ and Ψ. We achieve this computational advantage via the Levinson-Durbin recursions or the innovations algorithm (Brockwell and Davis, 1987, p. 71) due to the ARMA form of the underlying model. Therefore the complete data log-likelihood is given by

$$\ell(\boldsymbol{\theta}; \mathbf{x}, \mathbf{y}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\,\log|\Gamma|\, - \frac{1}{2}\mathbf{x}^\top\Psi^\top\,\Gamma^{-1}\,\Psi\,\mathbf{x}.$$

The objective function in the E-step is

$$
\begin{aligned}
&Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(i-1)}) \\
&\quad = \ \mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[\ell(\boldsymbol{\theta};\mathbf{X},\mathbf{y}) \mid \mathbf{Y}=\mathbf{y}\right], \\
&\quad = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\,\log|\Gamma| \ - \frac{1}{2}\mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[\mathbf{X}^{\top}\Psi^{\top}\ \Gamma^{-1}\ \Psi\,\mathbf{X} \mid \mathbf{Y}=\mathbf{y}\right],
\end{aligned}
$$

where each term in the expression is a scalar and thus the matrix expression is equal to its trace and $\mathrm{tr}(\mathbf{X}^{\top}\Sigma^{-1}\mathbf{X}) = \mathrm{tr}(\Sigma^{-1}\mathbf{X}\mathbf{X}^{\top})$. Thus

$$
Q(\boldsymbol{\theta},\boldsymbol{\theta}^{(i-1)}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{N}\ \log(\tau_t^2)\ - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[X_i\ X_j \mid \mathbf{Y}=\mathbf{y}\right],
$$

where c_{ij} is the $(i,j)^{\text{th}}$ element of $\Sigma^{-1} = \Psi^{\top}\ \Gamma^{-1}\ \Psi$ used as the coefficient of the expectation of $X_i\ X_j$ given $\mathbf{Y}=\mathbf{y}$.

It was proved in Section 3.3.1 that for each $t = 1,\ldots,N$, the given information $Y_t = y_t$ translates into an interval for the unobserved/missing X_t:

$$
\Phi^{-1}\left(F_t(y_t-1)\right)\ <\ X_t\ \leq\ \Phi^{-1}\left(F_t(y_t)\right). \tag{4.5}
$$

Here X_t cannot be directly uncovered for a discrete distribution function F. Conditioning on the observed data $y_1,\ldots,y_N$, the N-dimensional multivariate normal distribution $N_N(\mathbf{0},\Sigma)$ of $X_1,\ldots,X_N$ becomes an N-dimesnional truncated Normal distribution, with truncation points determined by the marginal distributions F_t, for $t = 1,\ldots,N$. Denote the upper and lower truncation points respectively by $\mathbf{a} = (a_1,\ldots,a_N)$ and $\mathbf{b} = (b_1,\ldots,b_N)$, where

$$
a_t = \Phi^{-1}\left(F_t(y_t-1)\right) \ \text{ and } \ b_t = \Phi^{-1}\left(F_t(y_t)\right),
$$

for $t = 1,\ldots,N$. Note that this $\mathbf{a}$ and $\mathbf{b}$ are not the ARMA parameters but we use this notation to be consistent with the truncated multivariate normal and GHK (Section 4.3.1) literature. Hence $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$ follows an N-dimensional truncated normal distribution, $TN_N(\mathbf{0},\Sigma,\mathbf{a},\mathbf{b})$. Thus using the interval of X_t given in (4.5), the E-step is to evaluate

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \\ -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{N}\log(\tau_t^2) \; - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[X_iX_j \middle| \mathbf{a} < \mathbf{X} \leq \mathbf{b}\right]. \quad (4.6)$$

An important point to make here is that the τ_t^2 and the c_{ij} terms are functions of $\boldsymbol{\theta}$. The vector $\boldsymbol{\theta}^{(i-1)}$ is used only in the expectation term.

4.3 The Monte Carlo E-step

Direct evaluation of the $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ function is difficult because

$$\mathrm{E}_{\boldsymbol{\theta}^{(i-1)}}\left[X_iX_j \middle| \mathbf{a} < \mathbf{X} \leq \mathbf{b}\right] = \int_{a_N}^{b_N}\cdots\int_{a_1}^{b_1} x_ix_j\phi_\Sigma(x_1,\ldots,x_N)\mathrm{d}\,x_1\ldots\mathrm{d}\,x_N, \quad (4.7)$$

does not have a closed solution for N greater than 1 and numerical techniques must be employed to calculate the expectation for the truncated multivariate normal distribution. We approximate the N-dimensional integration (4.7) using Monte Carlo estimation.

Wei and Tanner (1990) first proposed a Monte Carlo method to estimate the E-step resulting in the MCEM algorithm. Chan and Ledolter (1995) showed that with suitable initial parameters, an MCEM sequence will converge with high probability to the set of parameter values at which the maximum of the likelihood occurs.

We utilise Monte Carlo estimation to obtain an approximation of the objective function by taking the following steps:

1. Generate m samples $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(m)}$, from the conditional distribution of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$ which is truncated normal $TN_N(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b})$ using the current parameters $\boldsymbol{\theta}^{(i-1)}$ to evaluate $\Sigma, \mathbf{a}$ and $\mathbf{b}$.

2. Set up

$$\widehat{Q}\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}\right) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{N}\log(\tau_t^2) \; - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\left(\frac{1}{m}\sum_{k=1}^{m} x_i^{(k)} x_j^{(k)}\right). \tag{4.8}$$

4.3.1 Sampling from $TN(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b})$

We implement a GHK (Geweke-Hajivassilou-Keane) simulator to sample directly from the conditional distribution of $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$. This conditional distribution is the multivariate truncated normal distribution $TN_N(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b})$.

The distribution arises from restricting the multivariate normal $N_N(\mathbf{0}, \Sigma)$ to the region $(\mathbf{a}, \mathbf{b}]$ as shown in (4.5). Hence

$$\mathbf{X} \sim TN_N(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b}) \qquad \Longleftrightarrow \qquad \mathbf{X} \sim N_N(\mathbf{0}, \Sigma) \;\text{ s.t }\; \mathbf{a} < \mathbf{X} \leq \mathbf{b}.$$

To obtain exact draws, rejection sampling is the simplest method however it is highly inefficient when the truncation region is small and high dimensional. Approximate methods such as Markov Chain Monte Carlo methods have been proposed to overcome inefficiency (Geweke, 1991), but they can suffer from poor mixing, convergence problems and correlation among samples (Wilhelm and Manjunath, 2010).

The GHK simulator allows direct sampling from the multivariate truncated normal distribution. The GHK method reduces the problem of sampling from an N-dimensional Gaussian with bounds in each dimension to recursively sampling N univariate truncated normals. It exploits the fact that a multivariate normal distribution function can be expressed as the product of sequentially conditioned univariate normal distribution functions, which can be easily and accurately evaluated. Therefore a sample from a truncated multivariate normal distribution can be derived from a recursive procedure using draws from a univariate truncated normal distribution which itself can be drawn smoothly from a uniform distribution.

To illustrate this idea we provide the details for simulating from $TN_N(\boldsymbol{\mu}, \Sigma, \mathbf{a}, \mathbf{b})$ with non-zero mean $\boldsymbol{\mu}$ (Greene, 2003). For any symmetric positive definite matrix Σ, we can write $\Sigma = LL^\top$, where L is a lower triangular matrix with real and positive diagonal elements,

$$L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{l2} & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ l_{N1} & l_{N2} & \cdots & l_{NN} \end{pmatrix}.$$

This decomposition is called the Cholesky decomposition and is commonly used in the Monte Carlo method for simulating systems with multiple correlated variables. Note, for our implementation in (4.5), $L^\top = \Gamma^{\frac{1}{2}}\Psi$ and $\mu = \mathbf{0}$. The key point is that through the simulation of independent standard normal variates, we can simulate multivariate normal variables by $\mathbf{X} = \boldsymbol{\mu} + L\mathbf{Z}$ where $\mathbf{Z} = (Z_1, \ldots, Z_N) \sim N_N(\mathbf{0}, I_N)$ denotes a vector of independent multivariate standard normal random variables.

Then $\mathbf{a} < \boldsymbol{\mu} + L\mathbf{Z} < \mathbf{b}$ so that $L^{-1}(\mathbf{a} - \boldsymbol{\mu}) < \mathbf{Z} \leq L^{-1}(\mathbf{b} - \boldsymbol{\mu})$ because L has positive diagonal elements and is triangular. Hence

$$\begin{pmatrix} \frac{a_1 - \mu_1}{l_{11}} \\ \frac{a_2 - \mu_2 - l_{21}x_1}{l_{22}} \\ \vdots \\ \frac{a_N - \mu_N - \sum_{i=1}^{N-1} l_{Ni}x_i}{l_{NN}} \end{pmatrix} < \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ \vdots \\ Z_N \end{pmatrix} \leq \begin{pmatrix} \frac{b_1 - \mu_1}{l_{11}} \\ \frac{b_2 - \mu_2 - l_{21}x_1}{l_{22}} \\ \vdots \\ \frac{b_N - \mu_N - \sum_{i=1}^{N-1} l_{Ni}x_i}{l_{NN}} \end{pmatrix}.$$

Simulating variates from a univariate normal distribution can be done by the Marsaglia polar method (G. Marsaglia, 1964) or the Box-Muller transform method (Box and Muller, 1958). We use standard method in R provided by the `qnorm` function which uses the C (Ritchie et al., 1975) translation of Wichura (1988). We simulate z_1 truncated to the region $\left(\frac{a_1-\mu_1}{l_{11}}, \frac{b_1-\mu_1}{l_{11}}\right)$, then z_2 truncated on the region $(\frac{a_2-\mu_2-l_{21}z_1}{l_{22}}, \frac{b_2-\mu_2-l_{21}z_1}{l_{22}})$ which is conditional on the simulated z_1. This recursion continues until $z_{\scriptscriptstyle N}$ is simulated resulting in a sample $\mathbf{z}$ from the distribution $TN_{\scriptscriptstyle N}(0, I_{\scriptscriptstyle N}, \mathbf{a}, \mathbf{b})$. The transformation $\mathbf{x} = \boldsymbol{\mu} + L\mathbf{z}$ gives a draw from the desired distribution $TN_{\scriptscriptstyle N}(\boldsymbol{\mu}, \Sigma, \mathbf{a}, \mathbf{b})$.

Algorithm 1 The GHK Simulator to generate samples from a truncated multivariate normal distribution.

Input: Truncated multivariate Normal distribution parameters, $\boldsymbol{\mu}, \Sigma, \mathbf{a}, \mathbf{b}$.
Output: m samples of length $\scriptstyle N$ from $TN_{\scriptscriptstyle N}(\boldsymbol{\mu}, \Sigma, \mathbf{a}, \mathbf{b})$.

```
$L = \text{chol}(\Sigma)$,                                        ▷ Cholesky decomposition
Generate $m$ random values $\mathbf{u} \leftarrow (u_1, \ldots, u_m)$ where $u_i \sim U[0,1]$,
Set $\mathbf{x}_1 \leftarrow \Phi^{-1}\left(\mathbf{u}\,(\Phi(\frac{b_1-\mu_1}{l_{11}}) - \Phi(\frac{a_1-\mu_1}{l_{11}})) + \Phi(\frac{a_1-\mu_1}{l_{11}})\right)$,
for $i = 2, \ldots, N$
    Compute lower bound $c \leftarrow \Phi\left((a_i - \sum_{k=1}^{i-1} l_{ik}z_k)/l_{ii}\right)$,
    Compute upper bound $d \leftarrow \Phi\left((b_i - \sum_{k=1}^{i-1} l_{ik}z_k)/l_{ii}\right)$,
    Generate $\mathbf{u} \leftarrow (u_1, \ldots, u_m)$ where $u_i \sim U[0,1]$.
    Set $\mathbf{x}_i = \Phi^{-1}(\mathbf{u}\,(d-c)+c)$,
end
Set $\mathbf{z} \leftarrow \boldsymbol{\mu} + L\mathbf{x}$,
Return $\mathbf{z} = (\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \ldots, \mathbf{z}^{(m)})$.
```

Using the GHK for large sample sizes has been viewed as difficult (Wang et al., 2012) and there are surprisingly not many examples of its use in the literature despite the simplicity of the idea. For this work we have vectorised the implementation across m columns rather than directly using m loops to increase efficiency resulting in a fast algorithm which is provided in pseudo-code in Algorithm 1. The R code is provided in Appendix C.1.

This method samples directly from the target distribution and therefore is exact resulting in a better estimator of the conditional expectation for the E-step than other Monte Carlo Markov Chain methods currently available (Wilhelm and Manjunath, 2014). Details of the properties of the simulator are given in Börsch-Supan and Hajivassilou (1990).

4.4 The M-step

Whilst an explicit solution to the M-step would be ideal and achieve convergence of the parameter values faster, there is no closed form solution available. Numerical optimisation routines should be used. For consistent results, a modified Nelder-Mead optimisation, the Nelder-Mead-Kelley approach of Kelley (1999) is used from the `dfoptim` package in `R` to complete the M-step (Varadhan et al., 2011). Care should be taken to consider the parameter constraints of the Gaussian copula model due to the ARMA(p, q) dependence structure, see Section 2.4. The full MCEM algorithm is given succinctly in pseudo-code in Algorithm 2 where we denote the log-likelihood $\ell(\boldsymbol{\theta}; \mathbf{y})$ as $\ell_{\mathbf{Y}}(\boldsymbol{\theta})$.

Algorithm 2 A MCEM algorithm for the Gaussian copula model for discrete data.

```
Input:
        Observed data $\mathbf{y}$,
        Estimated marginal distributions $\widehat{\mathbf{F}} = \widehat{\mathbf{F}}(\mathbf{y})$ and $\widehat{\mathbf{F}}^{-} = \widehat{\mathbf{F}}(\mathbf{y}-1)$,
        Initial parameter values $\boldsymbol{\theta}^{(0)}_{\text{copula}}$,
        Number of Monte Carlo samples $m$,
        A precision value for the stopping criteria $prec$.
Output:
        Maximum likelihood estimates $\widehat{\boldsymbol{\theta}}_{\text{copula}}$ of $\boldsymbol{\theta}_{\text{copula}}$.
Estimated marginals parametrically or non-parametrically,
Set lower truncation values $\mathbf{a} \leftarrow \Phi^{-1}(\widehat{\mathbf{F}}^{-})$,
Set upper truncation values $\mathbf{b} \leftarrow \Phi^{-1}(\widehat{\mathbf{F}})$,
$i \leftarrow 1$,
while $\widetilde{\Delta}\,\ell_{\mathbf{Y}} < prec$
    $\widetilde{Q}(\boldsymbol{\theta}) \leftarrow Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)})$ using current parameter values $\boldsymbol{\theta}^{(i-1)}$,    ▷ E-step
    $\boldsymbol{\theta}^{(i)} \leftarrow \arg\max_{\boldsymbol{\theta}} \widetilde{Q}(\boldsymbol{\theta})$,    ▷ M-step
    $\widetilde{\Delta}\,\ell_{\mathbf{Y}} \leftarrow |\ell(\boldsymbol{\theta}^{(i)}; \mathbf{y}) - \ell(\boldsymbol{\theta}^{(i-1)}; \mathbf{y})|$,
    $i \leftarrow i + 1$,
end
$\widehat{\boldsymbol{\theta}}_{\text{copula}} \leftarrow \boldsymbol{\theta}^{(i)}$,
Return $\widehat{\boldsymbol{\theta}}_{\text{copula}}$.
```

4.5 Practical considerations

As demonstrated by Wei and Tanner (1990), the number of Monte Carlo simulations m should begin with a small value and increase as the MCEM algorithm runs to allow the estimates to explore the parameter space; it should be large in the final few iterations to reduce the Monte Carlo error in the E-step. The approximate value of a local maximum is often easy to discern and further analysis with larger m can then be carried out, if necessary, to refine the approximation.

The choice of the initial parameter values for the EM algorithm and MCEM algorithm have the same concerns because every limit of an EM sequence $\boldsymbol{\theta}^{(i)}$ is a stationary point of $\ell(\boldsymbol{\theta};\mathbf{y})$. However it may be a local maximum rather than a global maximum, i.e. the maximum likelihood estimate. The goal is to find the parameters values which globally maximise the likelihood function. So in practice, we run the algorithm several times with different starting points. Increased confidence in an MCEM procedure can be achieved by running the procedure with different starting values representative of the parameter space to allow for easy searching for multiple modes and helps discover the landscape of the likelihood function. Also to note, the starting values $\boldsymbol{\theta}^{(0)}$ do not need to be close to the true $\boldsymbol{\theta}$; but the EM algorithm will converge very slowly if a poor choice of initial values $\boldsymbol{\theta}^{(0)}$ is used. Indeed, in some cases where the likelihood is unbounded on the edge of the parameter space, the sequence of estimates $\{\boldsymbol{\theta}^{(i)}\}$ generated by the EM algorithm may diverge if $\boldsymbol{\theta}^{(0)}$ is chosen too close to the boundary.

Due to the simulation variability introduced in the E-step, the parameter updates can still fluctuate after ‘convergence’ so it becomes difficult to certify convergence of $\{\boldsymbol{\theta}^{(i)}\}$ (Moller and Waagepetersen, 2003). The issue is expanded when the number of parameters is impractical to monitor. Chan and Ledolter (1995) suggest monitoring the change in likelihood along the sequence from $\boldsymbol{\theta}^{(i-1)}$ to $\boldsymbol{\theta}^{(i)}$. Define the log-likelihood of $\mathbf{Y}$ and $\mathbf{X}$, $\log\mathcal{L}_{\mathbf{Y}} = \ell(\boldsymbol{\theta};\mathbf{y})$ and $\log\mathcal{L}_{\mathbf{X}} = \ell(\boldsymbol{\theta};\mathbf{y})$ as $\ell_{\mathbf{Y}}(\boldsymbol{\theta})$ and $\ell_{\mathbf{X}}(\boldsymbol{\theta})$

respectively then

$$\Delta\ell_{\mathbf{Y}}(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}) = \ell_{\mathbf{Y}}(\boldsymbol{\theta}^{(i)}) - \ell_{\mathbf{Y}}(\boldsymbol{\theta}^{(i-1)}),$$

which can be estimated by

$$\widetilde{\Delta}\,\ell_{\mathbf{Y}}(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(i-1)}) = -\log\left(\frac{1}{m}\sum_{k=1}^{m} \mathcal{L}_{\mathbf{X}}(\mathbf{x}^{(k)}; \boldsymbol{\theta}^{(i-1)})/\mathcal{L}_{\mathbf{X}}(\mathbf{x}^{(k)}; \boldsymbol{\theta}^{(i)})\right), \tag{4.9}$$

where $\left\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right\}$ is the sample from the conditional distribution of $\mathbf{X} \mid \mathbf{Y} = \mathbf{y}$ using the current parameters $\boldsymbol{\theta}^{(i)}$. When the estimated change in log-likelihood value (4.9) is plotted against the number of iterations, convergence can be claimed when the plot appears to fluctuate randomly about the abscissa, for example when the absolute value of (4.9) is less than a precision value for the last 5 values.

4.6 Standard errors of estimates

The standard errors are required to assess the accuracy of the parameter estimates. The variance of the maximum likelihood estimates can be derived from the inverse of the negative Hessian matrix evaluated at the maximum likelihood estimates. The Hessian is a $\dim(\boldsymbol{\theta}) \times \dim(\boldsymbol{\theta})$ matrix of second derivatives of the log-likelihood function,

$$\mathbb{H}(\boldsymbol{\theta}) = \frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{Y}}(\boldsymbol{\theta}).$$

The negative of the Hessian is called the observed information matrix which we will denote by $I_Y(\boldsymbol{\theta})$. The standard errors are the square roots of the diagonal elements of the inverse observed information matrix.

The EM algorithm and its variant, the MCEM algorithm, do not provide standard errors automatically for the estimates. The second derivative of the observed log-likelihood $\ell_{\mathbf{Y}}(\boldsymbol{\theta})$ is too intensive to be computed directly since even the log-likelihood evaluation contains 2^N evaluations of the N-dimensional Gaussian distribution function and so we must seek another approach using the complete data log-likelihood $\ell_{\mathbf{X}}(\boldsymbol{\theta})$. Louis (1982) presented a formula for calculating standard errors using this approach. It was shown that the observed information matrix, $I_Y(\boldsymbol{\theta}) = -\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{Y}}(\boldsymbol{\theta})$, when using the EM algorithm can be calculated through $\ell_{\mathbf{X}}(\boldsymbol{\theta})$ using

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\theta}}\ell_{\mathbf{Y}}(\boldsymbol{\theta}) &= \mathrm{E}_{\boldsymbol{\theta}}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\ell_{\mathbf{X}}(\boldsymbol{\theta}) \mid \mathbf{y}\right], \\
-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{Y}}(\boldsymbol{\theta}) &= \mathrm{E}_{\boldsymbol{\theta}}\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{X}}(\boldsymbol{\theta}) \mid \mathbf{y}\right] - \mathrm{E}_{\boldsymbol{\theta}}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\ell_{\mathbf{X}}(\boldsymbol{\theta})\ \frac{\partial}{\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{X}}(\boldsymbol{\theta}) \mid \mathbf{y}\right] \\
&\quad + \frac{\partial}{\partial\boldsymbol{\theta}}\ell_{\mathbf{Y}}(\boldsymbol{\theta})\frac{\partial}{\partial\boldsymbol{\theta}^\top}\ell_{\mathbf{Y}}(\boldsymbol{\theta}). \qquad (4.10)
\end{aligned}$$

The last term in (4.10) is difficult to evaluate because it involves the derivative of the incomplete data log-likelihood. However the standard errors are required only for the MLE $\widehat{\boldsymbol{\theta}}$ and it can be taken as approximately zero when the MCEM algorithm stops.

To obtain the observed information matrix Chan and Ledolter (1995) approximate Louis' formula (4.10) by Monte Carlo. The derivative of the complete data log-likelihood is averaged over the Monte Carlo samples, to estimate the score and observed Fisher information of the incomplete data in a similar way to (4.9).

Hence

$$\begin{aligned}
\widetilde{I_Y}(\boldsymbol{\theta}) &= -\frac{1}{m}\sum_{j=1}^{m}\frac{\partial^2\ell_{\mathbf{X}}(\boldsymbol{\theta};\mathbf{x}^{(j)})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top} - \frac{1}{m}\sum_{j=1}^{m}\left(\frac{\partial\ell_{\mathbf{X}}(\boldsymbol{\theta};\mathbf{x}^{(j)})}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial\ell_{\mathbf{X}}(\boldsymbol{\theta};\mathbf{x}^{(j)})}{\partial\boldsymbol{\theta}^\top}\right) \\
&\quad + \left(\frac{1}{m}\sum_{j=1}^{m}\frac{\partial\ell_{\mathbf{X}}(\boldsymbol{\theta};\mathbf{x}^{(j)})}{\partial\boldsymbol{\theta}}\right)^2. \qquad (4.11)
\end{aligned}$$

so that the partial derivatives of the log-likelihood of an ARMA(p,q) model is required. As long as we can calculate $\frac{\partial}{\partial\boldsymbol{\theta}}\ell(\boldsymbol{\theta};\mathbf{x})$ and $\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\ell(\boldsymbol{\theta};\mathbf{x})$, the observed Fisher information can be calculated and the standard errors subsequently.

The expressions of (4.11) can be evaluated numerically as in this work. Analytic expressions for these terms can be found which would require the second derivatives of the ARMA log-likelihood. Burshtein (1993) gives exact analytical expressions for the likelihood and likelihood gradient of stationary ARMA(p,q) models, where the computational cost of the likelihood gradient requires $(2p{+}6q{+}2)N{+}o(N)$ operations. Burshtein's method reduces the inverse and determinant of the $N \times N$ covariance matrix to those of smaller matrices of dimension $M \times M$, where $M = \max(p, q)$.

Other methods of calculating the analytic gradient of the ARMA(p, q) likelihood involve the Kalman filter (Chevassu and Ortega, 2013) and similar to our approach, although in a multivariate time series setting with missing values, Jonasson and Ferrando (2008) uses the Cholesky decomposition.

However, these methods cannot be immediately applied as they assume that σ_ε^2 is a constant. For the Gaussian copula model, the Gaussian correlation matrix has unit diagonals and therefore σ_ε^2 is a function of $\boldsymbol{\theta}$. Hence one has to modify the current methods with the complexity of considering the derivative of $\sigma_\varepsilon^2(\boldsymbol{\theta})$, which we discuss later in Section 6.6.2 and further in Appendix A.

4.7 Simulation under model conditions

In this section we present the results of a simulation study to illustrate the performance of the theory in practice when applied to a Gaussian copula model with four parameters. For the continuous ARMA model, the parameter accuracy improves as the length of the time series N increases. To examine the accuracy as N increases for our discretised ARMA model, we present 3 simulation studies with time series of lengths N=250, N=500 and N=1000. We present the parameter estimates, bias, and relative mean squared error for the fully parametric Gaussian copula model. We are particularly interested in assessing the accuracy of the standard errors. The parametric model is used rather than the semi-parametric model to allow comparison with the subsequent model fitting methods described in Chapters 5 and 6.

The Gaussian copula model used is specified by an ARMA(1,1) dependence structure with parameters $\boldsymbol{\theta}_{\text{copula}} = (a_1, b_1) = (0.7, -0.5)$ and negative binomial marginals $NB(s, \pi)$ with parameters $\boldsymbol{\theta}_{\text{marg}} = (s, \pi) = (5, 0.5)$. We generate $S = 1000$ data sets from this model for each N.

The computations are all carried out in the R environment and the first of the two-stage estimation (Section 3.5) estimates the marginal parameters from the independence likelihood and is carried out using the `fitdistr` function in the *MASS* R package (Venables and Ripley, 2002). The upper and lower truncation points required for the E-step of the MCEM algorithm are computed for each $t = 1, \ldots, N$ as

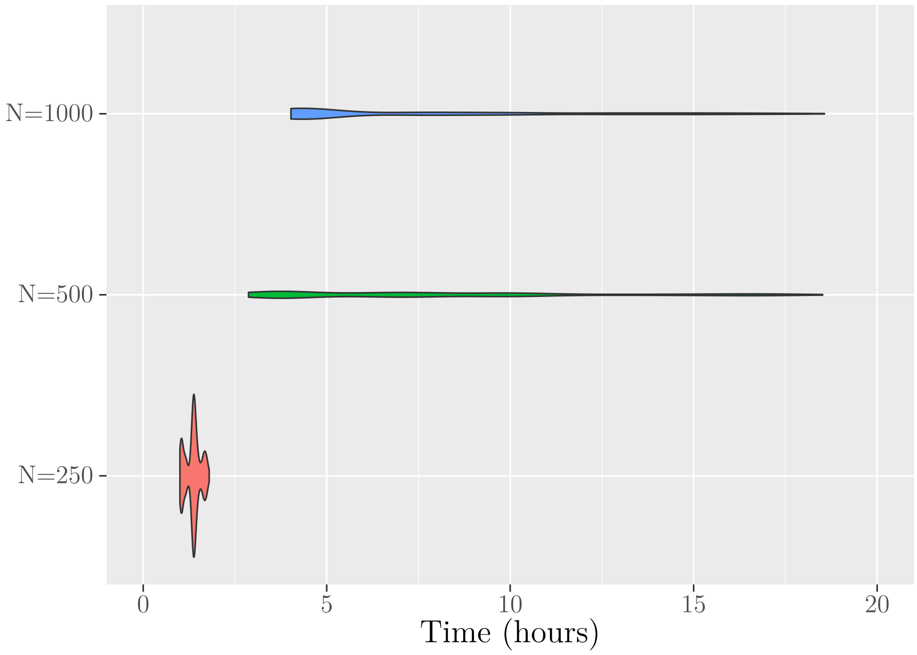
$$\widehat{k}_t = F_t(y_t - 1; \widehat{\pi}, \widehat{s}) \quad \text{and} \quad \widehat{l}_t = F_t(y_t; \widehat{\pi}, \widehat{s}).$$

The MCEM algorithm (Algorithm 2) is employed on the simulated data using initial values ($a_1 = 0, b_1 = 0$) for ten iterations. The maximisation in the M-step is carried out in the `dfoptim` package of the R software developed by Varadhan et al. (2011). The estimated change in likelihood (4.9) is computed at each step to monitor convergence. To aid convergence, the number of Monte Carlo samples begins with $m = 100$ for the first five iterations and is increased to $m = 500$ for the final five iterations. The mean and variance of the parameter estimates as well as the bias, mean squared error and root mean squared error are calculated (Walther and Moore, 2005). The MCEM estimation is run in parallel on a computer server on each dataset and we record the computational time taken to run ten iterations.

The distribution of the timing taken for the $S = 1000$ simulations is displayed using a violin plot in Fig. 4.1. The time scale is of the units hours. As expected we see the computational time increasing with the length of the integer-valued data, with average times of approximately 1.5, 3 and 5 hours respectively. The bottleneck of the algorithm occurs at the high-dimensional matrix multiplication required for (4.8). There is much variability in the timings for the time series of lengths N=250 and N=500. During sensitivity testing, the algorithm was very slow to run for longer time series of greater than length N=1000 and also for larger values of m, however, in this study we fixed m and selected moderate lengths of time series.

The distribution of the errors of the estimates ($\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_{\text{true}}$) are shown in Fig. 4.2 for each parameter in $\boldsymbol{\theta} = (a_1, b_1, s, \pi)$. By investigating the spread of the errors of MLEs in Fig. 4.2 we see that the estimates become closer to the true values as

Figure 4.1: The computational time taken for the $S = 1000$ MCEM chains for integer-valued time series of lengths N=250, N=500 and N=1000.

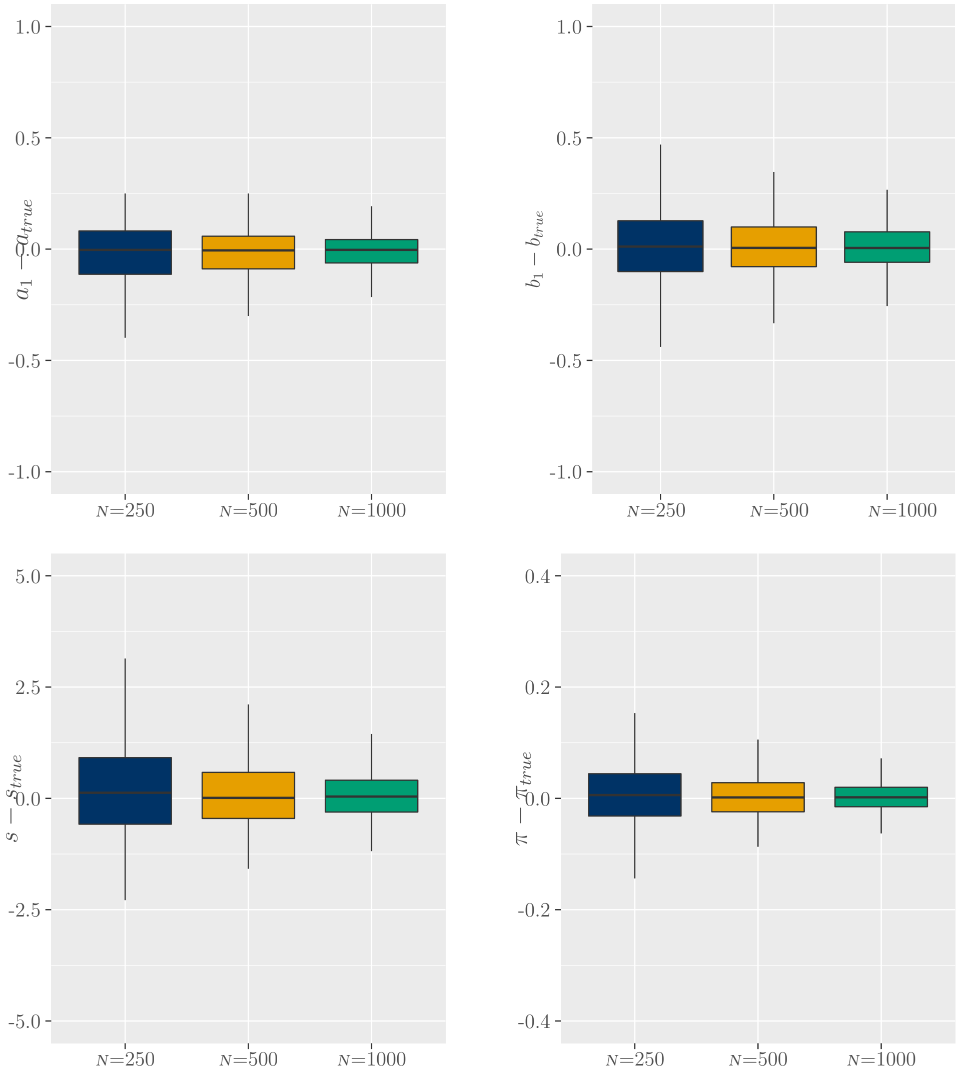


N increases. Fig. 4.2 shows that overall the parameters π and s are consistently accurately estimated for each of the sample sizes using the independence likelihood. This is important because when π and s are not estimated accurately, the estimation of the dependence parameters in the MCEM algorithm will be affected consequently (3.11). The whiskers of the distributions are slightly skewed for N=250 but become more symmetric as the length of the time series increases.

Table 4.1 gives the mean maximum likelihood estimates and their associated values of interest. As N increases, the mean values of the estimates become closer to their true values. The variance of the sample estimates decreases two-fold as N increases from N=250 to N=1000 and the bias is close to zero. Hence the MSE and RMSE also decrease as N increases. Table 4.1 shows that for moderate to large time series, the method works well and is reasonable for N=250.

The Gaussian copula model can be interpreted as borrowing the dependence of the underlying time series for the observed integer-valued time series, see Section

Figure 4.2: The distributions of error estimates for MCEM for $S = 1000$ simulated series each of length N=250, N=500 and N=1000.



True		N=250	N=500	N=1000
a_1=0.7	mean	0.676	0.677	0.685
	var	0.022	0.017	0.010
	bias	-0.024	-0.023	-0.015
	MSE	0.022	0.018	0.011
	RMSE	0.149	0.133	0.103
b_1=-0.5	mean	-0.481	-0.482	-0.489
	var	0.033	0.023	0.013
	bias	0.019	0.018	0.011
	MSE	0.033	0.023	0.013
	RMSE	0.183	0.152	0.115
s=5	mean	5.280	5.140	5.068
	var	1.405	0.676	0.269
	bias	0.280	0.140	0.068
	MSE	1.483	0.696	0.274
	RMSE	1.218	0.834	0.524
π=0.5	mean	0.508	0.504	0.503
	var	0.003	0.001	0.001
	bias	0.008	0.004	0.003
	MSE	0.003	0.001	0.001
	RMSE	0.054	0.038	0.025

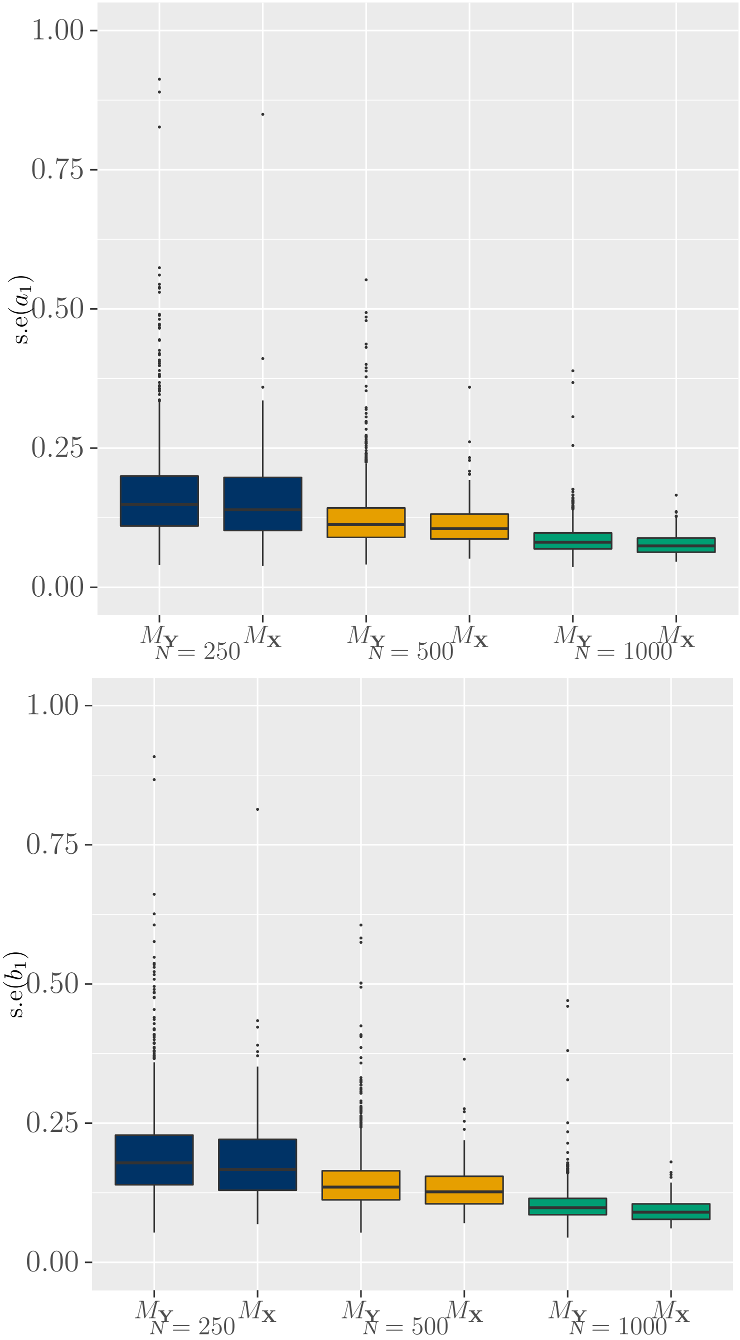
Table 4.1: The MCEM simulation results for ARMA(1,1) dependence parameters of the Gaussian copula model with negative binomial marginals (standard errors in parentheses) applied to $S = 1000$ data sets.

3.3. In a simulation study the underlying latent time series $\mathbf{X}$ is known. The exact likelihood can be computed via a state-space representation of the ARMA process, and the innovations and their variance found by a Kalman filter therefore standard maximum likelihood estimation can be used for the dependence parameters. These maximum likelihood estimates and their corresponding standard errors give an idea of how close to the well-established 'gold standard' ARMA methods we can get using the Gaussian copula model, allowing insight into the realistic accuracy of the parameter estimates that can be achieved.

Fig. 4.3 shows the distribution of the standard error estimates of a_1 and b_1 of the discretised ARMA (Gaussian copula model) applied to $S = 1000$ integer-valued data sets compared to the standard errors of the continuous underlying ARMA model. Let $M_{\mathbf{Y}}$ denote the estimates from the Gaussian copula model and $M_{\mathbf{X}}$ denote the estimates from the ARMA model. The navy, yellow and green represent the length of time series as $N = 250, 500, 1000$ respectively.

Fig. 4.3 demonstrates that the standard error estimates of the discretised model are favourably close to the estimates achieved if the underlying time series was known. The standard errors of the continuous ARMA are slightly smaller with fewer outliers but overall the difference in standard error distribution is negligible, demonstrating the strength and robustness of the MCEM algorithm derived in this chapter for this four parameter model of moderate length. Each MCEM algorithm was started at $a_1 = b_1 = 0$ and ran for ten iterations only, with five using $m = 100$ and five using $m = 500$ and each chain wasn't ran using different starting values or to ensure convergence was reached. This may explain the number of outliers in the distribution of estimates. However we see that when the ARMA model is fit using the `R` `arima` function that the outliers are not uncommon for time series of lengths N=250 and N=500. The number of outliers for $M_{\mathbf{Y}} : M_{\mathbf{X}}$ for $N = (250, 500, 1000)$ are $(3, 21, 13) : (5, 3, 3)$ for $\widehat{a}_1$ and $(8, 24, 10) : (5, 3, 3)$ for $\widehat{b}_1$.

Figure 4.3: The distributions of the estimated standard errors of a_1 and b_1 from the discrete data compared with the estimated continuous ARMA(1,1) standard errors when the underlying continuous time series is known.



4.8 Discussion

In this chapter, we have derived an MCEM algorithm to allow maximum likelihood estimation of the Gaussian copula model dependence parameters. As common in copula estimation, a two-stage approach has been implemented allowing practitioners to incorporate dependence into modelling discrete values over time.

We have also demonstrated the robustness of the MCEM method for estimation by considering scenarios of different length time series. In the four parameter model used here, the MCEM chains converged to global maxima using initial values $a_1 = b_1 = 0$ and ten iterations with $m = 100, 500$. However, a common criticism of the EM algorithm is that convergence can be relatively slow. We note that the computational time increases significantly for longer time series woth large m and we have shown that for large N=1000, the timings can come close to 18 hours. This may be due to memory allocation when parallelising the simulations. The minimum time for convergence of a length N=1000 time series was 4.2 hours and for shorter time series, the timings are much faster. Methods to accelerate the algorithm can be used but they may sacrifice the stability of the EM algorithm (Lange, 1995).

Limitations of this method include the failure to fully account for dependence when estimating the marginal parameters. However this is the ideal scenario and the two-stage estimation presented here is widely accepted in copula estimation (Xu, 1996; Joe, 2005). The independence likelihood allows inference to be made about the marginal parameters (Varin et al., 2011). We conducted a round of simulations which are not presented here for models using nonparametric marginals and we obtained maximum likelihood estimates closer to the true values for the dependence structure when a semi-parametric model was specified. An example with non-parametric marginals is given in Chapter 7.

Table 4.2 shows the results when the importance sampling approach of Masarotto et. al., (2012) is implemented using the `R` `gcmr` package (Masarotto and Varin,

Table 4.2: Parameter estimates from the importance sampling approach of Masarotto et al. for the S=1000 simulated data sets used for the MCEM algorithm study.

True		N=250	N=500	N=1000
a_1=0.7	mean	0.632	0.660	0.677
	var	0.053	0.013	0.004
	bias	-0.068	-0.040	-0.023
	MSE	0.058	0.014	0.005
	RMSE	0.241	0.119	0.071
b_1=-0.5	mean	-0.428	-0.458	-0.476
	var	0.056	0.017	0.007
	bias	0.072	0.042	0.024
	MSE	0.062	0.018	0.008
	RMSE	0.248	0.136	0.087
$s = 5$	mean	1.605	1.609	1.608
	var	0.004	0.002	0.001
	bias	-3.395	-3.391	-3.392
	MSE	11.531	11.498	11.505
	RMSE	3.396	3.391	3.392
$\pi = 0.5$	mean	0.196	0.198	0.202
	var	0.002	0.001	0.000
	bias	-0.304	-0.302	-0.298
	MSE	0.094	0.092	0.089
	RMSE	0.307	0.303	0.299

2012). The timing of their algorithm which simulates the likelihood is much faster than the MCEM algorithm we have presented, taking only a few minutes for a time series of length N=1000. Although we cannot compete with the computational time, the results from Table 4.1 are better in terms of mean, variance, bias, RMSE for the $S = 1000$ simulated data sets used for the MCEM algorithm study. The mean values of the estimates are closer to the true values while there is less bias in the estimators and the RMSE is smaller in all cases compared with Table 4.1. Overall, we achieve a higher level of accuracy at a computational cost.

Chapter 5

Estimation II: Approximate Bayesian Computation

The computational complexity when estimating the Gaussian copula model has been discussed in Chapter 3 and a MCEM algorithm derived in Chapter 4 to overcome the difficulties of maximum likelihood estimation. The algorithm works well and allows the model to be used in practice, however the running time becomes burdensome for longer length time series and a large number of Monte Carlo samples. A likelihood-free method for statistical inference of complex models is a natural step, which gives a simulation-based solution.

Given the parameters, the model (3.8) is not difficult to simulate from. It is given by

$$Y_t = F^{-1}(\Phi(X_t)), \quad t = 1, \ldots, \textsc{n}, \tag{5.1}$$

where F is the marginal distribution of Y_t, $t = 1, \ldots, \textsc{n}$ and $\{X_t\}$ is a Gaussian time series with variance/correlation matrix Σ (Section 3.3). By specifying an ARMA(p, q) model for $\{X_t\}$, the correlation matrix Σ is completely determined by the parameters $a_1, \ldots, a_p, b_1, \ldots, b_q$. Denote the marginal parameters of F as $\boldsymbol{\theta}_{\text{marginal}}$ then the model parameters are $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, b_q, \boldsymbol{\theta}_{\text{marginal}})$.

Approximate Bayesian Computation (ABC) is based on an acceptance-rejection algorithm. ABC methods simulate a large number of data sets using different parameter values. The parameter values whose simulated data sets are similar, in some sense, to the observed data set are saved and used as the basis of Monte Carlo inference. ABC has two main applications: parameter estimation and model choice. We use the former.

In this chapter we present an ABC algorithm for parameter estimation of the Gaussian copula model. In Section 5.1 we give the main ideas of ABC. In Section 5.2 and 5.3 we describe a novel method to compare the similarity of integer-valued time series in two stages. Section 5.4 presents the ABC algorithm and in Section 5.5, post-sampling techniques which have achieved great improvements elsewhere are considered. Simulation studies are presented in Section 5.6 and we finalise this chapter with a discussion in Section 5.7.

5.1 Introduction

The Bayesian approach to likelihood inference is based on the posterior distribution, that is the conditional distribution of the parameters given the observed data, $\pi(\boldsymbol{\theta}|\mathbf{y})$. Using a prior density $\pi(\boldsymbol{\theta})$ for the parameters $\boldsymbol{\theta}$ and the likelihood $\pi(\mathbf{y}|\boldsymbol{\theta})$, the posterior distribution of $\boldsymbol{\theta}$ can be written using the Bayes Theorem as

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \pi(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}),$$

up to a proportionality constant. The likelihood $\pi(\mathbf{y}|\boldsymbol{\theta})$ is intractable for the Gaussian copula model in high-dimensions and MCMC methods don't scale easily to high-dimensions typical of time series (Panagiotelis et al., 2012).

Approximate Bayesian Computation is a Bayesian approach to likelihood-free inference which generates samples from what is not necessarily the posterior distribution, but something close to it, i.e., sampling from an approximate posterior distribution and hence its name. The simplest ABC algorithm simulates data are rejects those not similar enough to the observed data. Such rejection sampling methods suffer from the 'curse of dimensionality', with the probability of rejection increasing exponentially with the dimension. Therefore to increase the efficiency of the algorithm, the data are compared using summary statistics and the similarity is quantified using a distance metric such as the Euclidean norm on the set of summary statistics.

The initial ABC algorithm was proposed by Tavaré et al. (1997) and has the following steps: Given observed data $\mathbf{y}$ from a multivariate distribution with prior density $\pi(\boldsymbol{\theta})$ for the parameters $\boldsymbol{\theta}$, calculate a summary statistic $S(\mathbf{y})$ of the data. Then at each iteration,

a) sample a parameter vector $\boldsymbol{\theta}^*$ from the prior $\pi(\boldsymbol{\theta})$;

b) simulate data $\mathbf{y}^*$ from the model with parameters $\boldsymbol{\theta}^*$;

c) calculate $S(\mathbf{y}^*)$ and a distance measure $d(S(\mathbf{y}), S(\mathbf{y}^*))$, *and*

d) accept $\boldsymbol{\theta}^*$ when the distance is less than a pre-determined value $\varepsilon > 0$.

The above is repeated until the desired number of $\boldsymbol{\theta}^*$ values have been collected. The collection of accepted parameter values $\boldsymbol{\theta}^*$ can be taken as a random sample of $\boldsymbol{\theta}$ from an approximation to the true posterior distribution, that is sampled from $\pi(\boldsymbol{\theta}|\mathbf{y})$.

The accepted parameter values $\boldsymbol{\theta}^*$ have a posterior density proportional to

$$\pi(\boldsymbol{\theta}) \int \pi(\mathbf{y}^*|\boldsymbol{\theta}) \mathbb{1}(d(S(\mathbf{y}), S(\mathbf{y}^*)) \leq \varepsilon) \,\mathrm{d}\,\mathbf{y}^*, \tag{5.2}$$

which (when a sufficient statistic $S(\mathbf{y})$ is used) converges point-wise to the true posterior $\pi(\boldsymbol{\theta}|\mathbf{y})$ as $\varepsilon \to 0$ (Prangle, 2011). We obtain the true posterior when $S(\mathbf{y})$ is a sufficient statistic because $S(\mathbf{y})$ contains all the information about $\boldsymbol{\theta}$ that $\mathbf{y}$ does. Thus the ABC rejection algorithm produces approximately i.i.d. samples from $\pi(\boldsymbol{\theta}|\mathbf{y})$ without the need to evaluate the likelihood $\pi(\mathbf{y}|\boldsymbol{\theta})$.

If the threshold ε, when $S(\mathbf{y})$ is a sufficient statistic, is zero then the ABC is exact and we draw from the true posterior which would result in a very high rejection rate. Alternatively, if ε is chosen to be large, such as infinity then the algorithm accepts every proposed set of parameters and we simply sample from the prior distribution. Decreasing ε would lead to a better concentration of the posterior density but the choice of the threshold is mostly a matter of computational power with the expenses being the size of the output or at a higher computing cost.

The choice of summary statistic $S(\mathbf{y})$, the distance metric d as well as the threshold ε affect the accuracy of the ABC estimates. There is a trade off between these to ensure a good selection of accepted values are accepted sufficiently often. There is little guidance on how to choose summary statistics and distance measures for complex models, despite a large amount of attention in the literature (Park et al., 2015; Stoehr et al., 2014; Marin et al., 2014; Fearnhead and Prangle, 2012; Aeschbacher et al., 2012; Prangle, 2011; Blum, 2010; Nunes and Balding, 2010).

The lack of guidance is due to the choice of summary statistics being specific to each application. Sufficient statistics for the model are ideal but typically a vector of informative statistics is used. For example the sample mean might be informative about a location parameter. We aim to balance capturing the information of the data against having low dimension to ensure a large enough acceptance rate. The choice is crucial to the efficiency and accuracy of the output (Prangle, 2011). Fearnhead and Prangle (2012) suggested a semi-automatic approach to choosing summary statistics and Park et al. (2015) proposed a non-parametric ABC paradigm using maximum mean discrepancy to quantify the similarity without using summary statistics.

For the Gaussian copula model applied to integer-valued time series data, we take an approach of summarising both marginal and dependence properties in a two part test; one test to compare the marginal distributions and one test to compare the dependence structures. The tests are independent and we accept $\boldsymbol{\theta}^*$ only if both tests are satisfied. In Section 5.2 we propose a method of comparing the marginal distributions taking advantage of the integer-valued nature of the data, whilst in Section 5.3 we describe the approach we take to comparing two dependence structures through their second order structure.

5.2 Comparing marginal distribuitons

For integer-valued data, we can detect a discrepancy between two time series marginal distributions of time series by counting the number of occurrences of each outcome. Thus the marginal distribution for $y_1, \ldots, y_N$ can be estimated by the sample proportions

$$\widehat{p}_j = \frac{1}{N}\sum_{j=1}^{N} \mathbb{1}(y_j = j), \qquad j = 0, 1, 2, \ldots. \tag{5.3}$$

These sample proportions are used as the summary statistics $S(\mathbf{y})$ in the first part of the test in the ABC algorithm.

To accept the set of parameter values $\boldsymbol{\theta}^*$, the distance between the set of summary statistics $S(\mathbf{y})$ and $S(\mathbf{y}^*)$ must be less than a pre-specified threshold. Any distance metric or function which quantifies the discrepancy numerically can be used and many applications use the Euclidean norm,

$$d_E(\widehat{p}, \widehat{p}^*) = \left(\sum_j (\widehat{p}_j - \widehat{p}_j^*)^2\right)^{1/2}. \tag{5.4}$$

However, the Kullback-Leibler statistic is a commonly used statistic to measure the dissimilarity between two probability distributions and this is the approach we take. The Kullback-Leibler statistic is given by

$$d_{KL}(\widehat{p}, \widehat{p}^*) = \sum_j \log\left(\frac{\widehat{p}_j}{\widehat{p}_j^*}\right)\widehat{p}_j. \tag{5.5}$$

The statistic is defined only if $\widehat{p}_j^* = 0$ implies $\widehat{p}_j = 0$, for all j (absolute continuity). Whenever $\widehat{p}_j$ is zero the contribution of the j^{th} term is interpreted as zero because $\lim_{\widehat{p}_j \to 0} \widehat{p}_j \log(\widehat{p}_j) = 0$.

This method of comparing marginal distributions of time series through sample proportions allows for the dependence between observations. Furthermore, the asymptotic distributions of the sample proportions $\widehat{p}_0$, $\widehat{p}_1$, $\ldots$, can be used to derive a formal hypothesis test to test the equality of sample proportions, however this is unnecessary for the ABC algorithm.

5.3 Spectral comparison of two stationary time series

For the second part of the rejection test to compare two integer-valued times series, we propose a comparison of the second order properties and we take a spectral domain approach. That is, we work with the Fourier transform of the data, rather than the time series data to compare their dependence structures. The observed integer valued time series can be converted to the frequency domain by the discrete Fourier transform (DFT) which can be calculated using the efficient fast Fourier transform (FFT).

The spectrum of a stationary time series is the distribution of variance of the series, as a function of frequency. It contains all the information about its second order properties including linear dependence. The classification of time series using spectral analysis approaches have been considered by Coates and Diggle (1986), Diggle and Fisher (1991), Dargahi-Noubary (1992), Chandler (1997), Diggle and al Wasel (1997), Kakizawa, Shumway and Taniguchi (1998), Maharaj (2002), Caiado, Crato and Peña (2006), among others.

The spectrum is a non-negative and integratable function $f(\omega)$ on $[-\pi, \pi)$ such that the autocovariance function $R(r)$ can be written as

$$R(r) = \int_{-\pi}^{\pi} e^{ir\omega}\, f(\omega)\, \mathrm{d}\,\omega, \qquad r \in \mathbb{Z}. \tag{5.6}$$

If $\sum_{r=-\infty}^{\infty} |R(r)| < \infty$, then $f(\omega)$ can be written in terms of its autocovariances as

$$f(\omega) = \frac{1}{2\pi} \sum_{r=-\infty}^{\infty} R(r)e^{-i\omega r}, \qquad \text{for } -\pi \leq \omega < \pi.$$

Therefore the linear dependence structures i.e., second-order properties of the time series can be compared through comparing spectra. Let $f(\omega)$ and $f^*(\omega)$ denote the spectrum of the observed and simulated time series respectively. We will now describe how we obtain estimates of the spectra at a fixed set of frequencies,

$$\omega_j = \frac{2\pi j}{N}, \qquad j = 0, 1, \ldots, N - 1,$$

before we derive a test to compare the set of estimated frequencies in Section 5.3.1. The vector of spectral estimates $f(\omega_j)$ will be considered as one set of summary statistics $S(\mathbf{y})$ in the ABC algorithm and the test derived in Section 5.3.1 will be considered as the distance metric for this set.

Various methods have been developed to estimate the spectrum from an observed time series (Chatfield, 2013, Ch. 7). The simplest estimator of the spectral density is the periodogram, given by

$$I_N(\omega) = \frac{1}{2\pi N} \left| \sum_{t=1}^{N} y_t e^{-i\omega t} \right|^2 .$$

The periodogram is an asymptotically unbiased estimator, however it is not a consistent estimator of the spectral density (Priestley, 1981). We can obtain consistent estimator in one of two ways:

1) by smoothing the periodogram using a spectral window $W(\theta)$

$$\widehat{f}(\omega) = \int_{-\pi}^{\pi} I_N(\omega) W(\omega - \theta) \, \mathrm{d}\,\theta,$$

or equivalently

2) by using weighted sample autocovariances $\widehat{R}(r)$ as follows:

$$\widehat{f}(\omega) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \lambda(r) \, \widehat{R}(r) e^{-ir\omega}, \tag{5.7}$$

where $\lambda(r) = k(\lambda/M)$ is a lag window with scale parameter M and $\widehat{R}(r)$ is the sample autocovariance at lag r. The scale parameter M controls the bandwidth of the spectral window given by

$$W(\theta) = \frac{1}{2\pi} \sum_{r=-(N-1)}^{N-1} \lambda(r) e^{-ir\theta}, \tag{5.8}$$

and thus M controls the degree of smoothing. There are many different forms of $\lambda(r)$ which lead to consistent estimates of $\widehat{f}(\omega)$.

The Bartlett-Priestley spectral window (Priestley, 1981, p. 444) has been shown to be optimal among a class of windows, in the sense of the smallest mean square error of the estimated spectral density function. Properties of the window include: it is non-negative valued; it integrates to one; it has a peak at $\theta = 0$, and has support on $(-\pi/M, \pi/M]$. The Bartlett-Priestley lag window generator is given by

$$k(r) = \frac{3}{(\pi r)^2}\left(\frac{\sin(\pi r)}{\pi r} - \cos(\pi r)\right), \quad r = 0, 1, \ldots, N-1,$$

with corresponding spectral window (5.8) given by

$$W(\theta) = \begin{cases} \frac{3M}{4\pi}\left(1 - \left(\frac{M\theta}{\pi}\right)^2\right) & \text{if } |\frac{\pi}{M}| \le \theta, \\ 0 & \text{if } |\theta| > \frac{\pi}{M}. \end{cases} \tag{5.9}$$

In theory we require $N/M \to \infty$ as $N \to \infty$ for $\widehat{f}(\omega)$ to be consistent.

The asymptotic variance of the spectral estimates using a Bartlett-Priestley window (5.9) (Priestley, 1981, p. 463) is

$$\operatorname{Var}\left(\widehat{f}(\omega)\right) \approx \frac{6M}{5N} f^2(\omega)(1 + \delta(\omega)),$$

where

$$\delta(\omega) = \begin{cases} 1 & \omega = 0, \pi, \\ 0 & \omega \ne 0, \pi, \end{cases} \tag{5.10}$$

With regard to deciding the degree of smoothing, the optimal choice of bandwidth M in the sense of the relative mean square error is

$$M = \left(\frac{8\pi^4 N}{15B^4} \right)^{\frac{1}{5}},$$

(Priestley, 1981, p. 515) where the bandwidth

$$B = 2\left(\inf_{\omega} |f(\omega)/f''(\omega)|^{1/2} \right),$$

is roughly the width of the narrowest peak. However, we do not have prior knowledge of the spectral bandwidth and therefore we take an empirical approach to selecting M; beginning with large M and decreasing until a suitable degree of smoothing is observed. Typically around $M = \sqrt{N}$ works well.

In the next section, we derive a test to reject simulated parameter values $\boldsymbol{\theta}^*$ in the ABC algorithm if the spectrum of the simulated time series $f^*(\omega)$ is sufficiently different to the spectrum $f(\omega)$ of the observed integer-valued time series. For the application of the ABC algorithm, this is the same as defining the summary statistics as the set of estimated spectral estimates at the fixed set of frequencies $S(\mathbf{y}) = \widehat{f}(\omega_j)$, $j = 0, \ldots, L-1$. Let ω_j be the set of fixed frequencies of the form $\omega_j = \frac{2\pi j}{L}$ where L is a fixed positive and even integer and suitably smaller than N. The test derived in Section 5.3.1 is based on an ANOVA test and makes the implicit assumption that the frequencies are independent.

That is we require the spectral estimates at fixed frequencies, say ω_1 and ω_2 for $\omega_1 \neq \pm\omega_2$, to be asymptotically uncorrelated (Priestley, 1981, p. 455),

$$\lim_{N \to \infty} \operatorname{cov}(\widehat{f}(\omega_1), \widehat{f}(\omega_2)) = 0.$$

Smoothing introduces correlation between neighbouring ordinates and by considering the covariance when N is large but finite,

$$\operatorname{cov}(\widehat{f}(\omega_1), \widehat{f}(\omega_2)) \approx \frac{2\pi}{N} \int_{-\pi}^{\pi} f^2(\theta) W_N(\omega_1 - \theta) W_N(\omega_2 - \theta) \, \mathrm{d}\theta, \tag{5.11}$$

we can see that when two frequencies are sufficiently close, $|\omega_1 - \omega_2|$ is the same order of magnitude as the width of spectral window. In this case the integral is not small and the spectral estimates will be correlated.

The spectral estimates corresponding to this window will be effectively uncorrelated if the separation between frequencies is greater than the bandwidth of the window $2\pi/M$. The distance between two fixed frequencies is $2\pi/L$ and therefore for uncorrelated spectral estimates we require $2\pi/M \leq 2\pi/L$, i.e.,

$$L \leq M.$$

5.3.1 A spectral comparison test

In this section, under the assumption of stationarity, we test the equality of two spectra at a fixed set of frequency points ω_j and the null hypothesis is that the two time series have identical spectra $f(\omega) = f^*(\omega)$ for all $\omega \in [-\pi, \pi)$. Priestley and Rao (1969) give a test for stationarity which uses estimated evolutionary spectra and tests the dependence on time and frequency using a two-way analysis of variance procedure when the variance is known. We simplify the test by omitting the time-dependence part to compare the spectral estimates of two time series at a fixed set of frequency points. Here we give the steps of the test before we describe the derivation in Section 5.3.2.

The procedure we take for comparing two spectra of integer-valued time series $\mathbf{y}$ and $\mathbf{y}^*$ is as follows:

1. Estimate the spectra $f(\omega)$ and $f^*(\omega)$ at a set of L fixed frequencies ω_j for $j = 0, 1, \ldots, L-1$.
2. Take a log transform of the estimates.
3. Calculate the asymptotic variance of the log spectral estimates.
4. Taking the frequency and the series ID as factors, compute a χ^2-statistic T and compare it with a critical value.

The test statistic for testing the null hypothesis that two time series have identical spectra is

$$T = \frac{1}{4\sigma^2}\sum_{j=0}^{L-1}\left(\log \widehat{f}\left(\frac{2\pi j}{L}\right) - \log \widehat{f}^{*}\left(\frac{2\pi j}{L}\right)\right)^2, \tag{5.12}$$

where $\widehat{f}$ is the smoothed periodogram (5.7) evaluated at ω_j and σ^2 is the variance of $\log \widehat{f}(\omega)$ (5.13) which is not dependent on ω (Priestley, 1981). The distribution of T under the null hypothesis is $\chi^2_{L/2+1}$ and L is the number of fixed frequencies ω_j.

5.3.2 Derivation of test statistic T

The test is based on the likelihood ratio principle. The probability distribution of the log-likelihood ratio test statistic, assuming that the null hypothesis is true, can be approximated using a χ^2 distribution.

Jenkins (1961) suggested that a logarithmic transform would bring the distribution of the spectral estimates $\widehat{f}(\omega_j)$ closer to normality and Priestley and Rao (1969) state that although this statement was not substantiated its validity is rendered highly plausible due to analogous result by Kendall and Stuart (1966). The logarithmic transform ensures the asymptotic variance (5.10) of the log spectral estimate is

$$\mathrm{Var}\left(\log \widehat{f}(\omega)\right) \approx \frac{6M}{5N}(1+\delta(\omega)), \tag{5.13}$$

which is constant except at $\omega = 0, \pi$. Following Priestley and Rao (1969), let

$$Y_{ij} = \log \widehat{f}(\omega_{ij}) + \varepsilon_{ij},$$

where i denotes the spectrum of time series with $i = 1$ for the observed $\mathbf{y}$ and $i = 2$ for the simulated time series $\mathbf{y}^*$ at $\omega_j = \frac{2\pi j}{L}$, $j = 0, \ldots, L/2$ with $\mathrm{E}(\varepsilon_{ij}) = 0$ and $\mathrm{var}(\varepsilon_{ij}) = \sigma^2(1+\delta(\omega))$. The model can be written as

$$Y_{ij} = \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ij},$$

for $i = 1, 2$ and $j = 0, 1, \ldots, L/2$, where α_i, β_j and γ_{ij} are the main effects for factor i and j and the interaction term respectively. We test the null hypothesis

$$H_0 : \alpha_i = \gamma_{ij} = 0, \qquad i = 1, 2, \quad j = 0, 1, \ldots, L/2,$$

against the alternative

$$H_1 : \alpha_i \neq 0 \text{ or } \gamma_{ij} \neq 0, \qquad \text{for some } i \text{ and } j.$$

Under the null hypothesis the model becomes

$$y_{ij} = \beta_j + \varepsilon_{ij}, \qquad i = 1, 2, \quad j = 0, 1, \ldots, L/2.$$

Denote the model under the null and alternate hypothesis as M_0 as M_1 respectively. Due to the periodicity $\widehat{f}(\omega_j) = \widehat{f}(\omega_{L-j})$ for $\omega_j = \frac{2\pi j}{L}$, $j = 0, \ldots, L/2$, the spectral estimates for j and $L - j$ have the same value and so we include all the frequency points $j = 0, \ldots, L - 1$. The maximised log-likelihood values are

$$\begin{aligned}
\ell_1 &= -\frac{L}{2} \log(2\pi\sigma^2) - \log 2, \\
\ell_0 &= -\frac{1}{2\sigma^2} \sum_{i=1}^{2} \sum_{j=0}^{L-1} \left(y_{ij} - \widehat{\beta}_j \right)^2 - \frac{L}{2} \log(2\pi\sigma^2) - \log 2,
\end{aligned}$$

where

$$\widehat{\beta}_j = \frac{y_{1j} + y_{2j}}{2}.$$

The likelihood ratio statistic is

$$\begin{aligned}
T &= -2\ (\ell_1 - \ell_0), \\
&= \frac{1}{2\sigma^2} \sum_{i=1}^{2} \sum_{j=0}^{L-1} \left(y_{ij} - \frac{y_{1j} + y_{2j}}{2} \right)^2, \\
&= \frac{1}{2\sigma^2} \sum_{j=0}^{L-1} \left\{ \left(\frac{y_{1j} - y_{2j}}{2} \right)^2 + \left(\frac{y_{2j} - y_{1j}}{2} \right)^2 \right\}, \\
&= \frac{1}{8\sigma^2} \sum_{j=0}^{L-1} \left\{ (y_{1j} - y_{2j})^2 + (y_{2j} - y_{1j})^2 \right\}, \\
&= \frac{1}{4\sigma^2} \sum_{j=0}^{L-1} \left(\log \widehat{f}_1(\omega_j) - \log \widehat{f}_2(\omega_j) \right)^2.
\end{aligned}$$

The asymptotic variance of $\log \widehat{f}_1(\omega_j) - \log \widehat{f}_2(\omega_j)$ is $2\sigma^2$ or $4\sigma^2$ as appropriate and therefore the test statistic T is a weighted sum of squares of the differences $\log \widehat{f}_1(\omega_j) - \log \widehat{f}_2(\omega_j)$, $j = 0, \ldots, L/2$ with weights equal to the reciprocals of their asymptotic variances.

Therefore the test statistic T has a χ^2 distribution with $L/2 + 1$ degrees of freedom under the null hypothesis of identical spectra,

$$T = \frac{1}{4\sigma^2} \sum_{j=0}^{L-1} \left(\log \widehat{f}_1 \left(\frac{2\pi j}{L} \right) - \log \widehat{f}_2 \left(\frac{2\pi j}{L} \right) \right)^2 \quad \dot{\sim} \quad \chi^2_{\frac{L}{2}+1} \qquad \text{if } f_1 \equiv f_2.$$

The distribution of the test statistic T (5.12) under the null hypothesis is known and therefore a formal threshold value can be determined by the critical value of $\chi^2_{L/2+1,\alpha}$ where α is the acceptable level of significance. However, it is not necessary for the ABC framework.

Standard practice is to pre-determine the number of simulated datasets and run the ABC algorithm, accepting each proposed parameter sets and computing the distance metric. Then post-sampling we select ε as a small percentile of the simulated distances $d(S(\mathbf{y}), S(\mathbf{y}^*)) < \varepsilon$. This can then be followed by some adjustments discussed in the Section 5.5 which have achieved major improvements. We modify this approach by setting a moderate threshold in the algorithm to ensure we don't save parameters far away so that a sufficient number of 'good ' estimates are in the sample. This is equivalent to assigning zero weight to these parameter values and will not change the weighted posterior estimates except near values of ε where parameter values would be in the tails of the posterior density but our interest is in the weighted means (5.14).

5.4 ABC for parameter estimation

In summary, the previous sections have proposed a two-stage test to compare two integer valued time series. The first test of Section 5.2 proposes using sample proportions as summary statistics with any distance metric d and the second test of

Section 5.3 proposes using spectral estimates as summary statistics and derived a specific metric T to compare the second order properties of the time series in a more formal manner. Denote the summary statistics of sample proportions as $S_1(\mathbf{y})$ and the summary statistics of spectral estimates as $S_2(\mathbf{y})$.

The algorithm to implement the approximate Bayesian computation for the Gaussian copula model (5.1) is given in Algorithm 3.

Algorithm 3 An approximate Bayesian computation (ABC) algorithm for the Gaussian copula model for discrete data.

Input: Observed integer-valued time series data $\mathbf{y}$ of length N,
Prior density distributions for parameters $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, b_q, \boldsymbol{\theta}_{\text{marginal}})$,
Number of desired iterations, $n \in \mathbb{Z}$,
Pre-determined moderate thresholds $\varepsilon_1, \varepsilon_2$.

Output: An i.i.d. sample of $\boldsymbol{\theta}^*$ with corresponding $d(S(\mathbf{y}), S(\mathbf{y}^*))$ and $S(\mathbf{y}^*)$ values.

```
1: while $i < n$,
2:     Sample parameter vector $\boldsymbol{\theta}^*$ from prior $\pi(\boldsymbol{\theta})$,
3:     Simulate data $\mathbf{y}^*$ from the model, conditional on $\boldsymbol{\theta}^*$,
4:     Set $D_1 \to d(S_1(\mathbf{y}), S_1(\mathbf{y}^*))$ where $S_1$ as (5.3) and $d$ as (5.5),
5:     Set $D_2 \to T(S_2(\mathbf{y}), S_2(\mathbf{y}^*))$ where $S_2$ as (5.7) and $T$ as (5.12),
6:     If both $D_1 < \varepsilon_1$ and $D_2 < \varepsilon_2$ then save $(\boldsymbol{\theta}^*, S(\mathbf{y}^*), D_1, D_2)$.
7: end
8: Return $(\boldsymbol{\theta}^*, S(\mathbf{y}^*), d(S_1(\mathbf{y}), S_1(\mathbf{y}^*)), T(S_2(\mathbf{y}), S_2(\mathbf{y}^*)))$ such that $D_1 < \varepsilon_1$ and
   $D_2 < \varepsilon_2$.
```

Choice of priors in step 2

The parameter values are sampled from probability distributions. We assume stationarity and invertibility for the Gaussian copula model with time dependence. Therefore for the ARMA dependence parameters, a simple prior distribution for

$\mathbf{a} = (a_1, \ldots, a_p)$ and $\mathbf{b} = (b_1, \ldots, b_q)$ would be a uniform prior over the area of the stationarity constraints (2.3). For example for AR(2) or MA(2), simple priors would be a uniform prior over the triangles,

$$\begin{aligned} & -2 < a_2 < 2, \qquad a_1 + a_2 < 1, \qquad a_2 - a_1 < 1. \\ \text{and} \qquad & -2 < b_1 < 2, \qquad b_1 + b_2 < 1, \qquad b_2 - b_1 > -1. \end{aligned}$$

The naive approach to sampling would be to sample from uniform priors and reject samples outside of the stationarity regions. However, sampling using a rejection approach is inefficient and more computationally efficient methods should be used. A standard method for simulation ARMA(p,q) parameters based on the beta distribution for partial autocorrelations can be used (Jones, 1987).

The Gaussian copula model lends itself naturally to ABC methods because for random variables Y_t in the Gaussian copula it can be shown (Section 3.3) that

$$Y_t = F^{-1}(\Phi(X_t)),$$

where $\{X_t\}$ follows an ARMA model. For simulation, we use the recursive formula of the ARMA model because of its easiness and simplicity but without the ARMA assumption, simulation is not as efficient. We could simulate from a Gaussian copula model by the following steps (Embrechts et al., 2005, p. 193):

1. Perform a Cholesky-decomposition $\Sigma = L^\top L$.
2. Simulate i.i.d. standard normal pseudo random variables $X'_1, X'_2, \ldots, X'_N$.
3. Compute $(X_1, X_2, \ldots, X_N) = \mathbf{X} = L\mathbf{X}'$ from $\mathbf{X}' = (X'_1, X'_2, \ldots, X'_N)$.
4. Set $U_t = \Phi(X_t), t = 1, \ldots N$ where Φ is the standard normal cumulative distribution function.
5. Return $Y_t = F^{-1}(U_t), t = 1, \ldots N$ where F is the marginal cumulative distribution function of Y_t.

Using the ARMA model, the steps 1-3 are avoided.

It should be noted that in general, copula models lend naturally to ABC methods and are not restricted to the Gaussian model. This natural idea is gaining traction for other copula models and a recent paper (Grazian and Liseo, 2015) uses the ABC theory with an empirical likelihood to estimate copulas non-parametrically. However, our interest is in parameter estimation for the fully parametric Gaussian copula model.

5.5 Post-sampling adjustment methods

Estimation of the posterior distribution can be improved by the use of regression techniques. Introduced by Beaumont et al. (2002) the idea is to give accepted values with smaller distances a larger weighting and reduce the approximation error due to choosing non-zero threshold value ε. The practice is as follows:

1. weight the parameters according to the value of $d(S(\mathbf{y}), S(\mathbf{y}^*))$; *and*
2. adjust the parameters by using

   A) local-linear regression,

   B) local nonlinear regression *or,*

   C) local-generalised linear regression,

as appropriate, to weaken the effect of the discrepancy between $S(\mathbf{y})$ and $S(\mathbf{y}^*)$ or to correct for heteroskedasticity.

Simulations that closely match $S(\mathbf{y})$ can be given more weight by assigning $K_\varepsilon(d(S(\mathbf{y}), S(\mathbf{y}^*)))$ to each simulation $(\boldsymbol{\theta}^*, S(\mathbf{y}^*))$. The ABC posterior density in (5.2) is the simplest case of an ABC algorithm in which equal weight is given for each accepted value i.e., a uniform kernel on $U(-\varepsilon, \varepsilon)$ is specified. For the general class of ABC algorithms the posterior density is

$$\pi_{ABC}(\boldsymbol{\theta}|\mathbf{y}) \propto \pi(\boldsymbol{\theta}) \int \pi(\mathbf{y}^*|\boldsymbol{\theta}) K_\varepsilon \left(d(S(\mathbf{y}), S(\mathbf{y}^*)) \right) \, \mathrm{d}\mathbf{y}^*,$$

where K_ε is a density kernel with bandwidth ε for any $\varepsilon > 0$. The kernel $K_\varepsilon(\mathbf{u})$ for d-dimensional vector $\mathbf{u}$ such that d is the dimension of $S(\mathbf{y})$, integrates to 1 and can be written as $K_\varepsilon(\mathbf{u}) = \varepsilon^{-d}K(\mathbf{u}/\varepsilon)$ where for simplicity, it is assumed $\max K(\mathbf{u}) = 1$. The uniform, Epanechnikov and Gaussian kernel are most common but any can be used. The weighted mean $\bar{\theta}^*_w$ for each parameter in $\boldsymbol{\theta}^*$ is

$$\bar{\theta}^*_w = \frac{\sum_{i=1}^n \theta^*_i K_\varepsilon\left(d(S(\mathbf{y})S(\mathbf{y}^*_i))\right)}{\sum_{i=1}^n K_\varepsilon\left(d(S(\mathbf{y}), S(\mathbf{y}^*_i))\right)}. \tag{5.14}$$

Let W be a diagonal matrix with i^{th} element given by $K_\varepsilon(d(S(\mathbf{y}), S(\mathbf{y}^*))$ giving more weight to the parameters with $d(S(\mathbf{y}), S(\mathbf{y}^*))$ small. There is no analytical equivalent to the standard error of the mean in weighted statistics and therefore numerical approximations must be used to compute an estimate of the standard error of the weighted dataset. A simple approximation using the weighted mean (5.14) can be found using

$$\mathrm{Var}_w(\boldsymbol{\theta}^*) = \sum_{i=1}^n \left(w_i(\theta^* - \bar{\theta}^*_w)^2 \times \left(\frac{\sum_{i=1}^n w_i}{(\sum_{i=1}^n w_i)^2 - \sum_{i=1}^n w_i^2)} \right) \right). \tag{5.15}$$

However, Cochran and Horne (1977) compares three methods from the literature with a standard method based on bootstrapping and gave results that were not statistically different from those of bootstrapping. The analytic formula that is closest to bootstrapping is

$$\begin{aligned} \mathrm{Var}_w(\boldsymbol{\theta}^*) = & \frac{n}{(n-1)(\sum_{i=1}^n w_i)^2} \times \\ & \left(\sum_{i=1}^n (w_i\theta^*_i - \bar{w}\bar{\theta}^*_w)^2 - 2\bar{\theta}^*_w \sum_{i=1}^n (w_i - \bar{w})(w_i\theta^*_i - \bar{w}\bar{\theta}^*_w) + \bar{\theta}^{*2}_w \sum_{i=1}^n (w_i - \bar{w})^2 \right), \end{aligned} \tag{5.16}$$

and is suggested to be the method of choice for routine computing of the standard error of the weighted mean (Gatz and Smith, 1995). This is the method we follow to compute the standard errors of a weighted dataset.

An alternative to a local mean adjustment is to do a local linear-regression of $\boldsymbol{\theta}^*_i$ on $S(\mathbf{y}^*_i) - S(\mathbf{y})$. The posterior density is a conditional density and can be described for some intercept $\boldsymbol{\alpha}$ and vector of regression coefficients $\boldsymbol{\beta}$,

$$\boldsymbol{\theta}^*_i = \boldsymbol{\alpha} + \boldsymbol{\beta}(S(\mathbf{y}^*_i) - S(\mathbf{y}))^\top + \epsilon_i, \qquad i = 1, \ldots, n,$$

where the ϵ_i are uncorrelated with mean zero and common variance. The least squares estimates of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are found by minimising

$$\sum_{i=i}^{n} \left(\theta_i^* - \boldsymbol{\alpha} - \boldsymbol{\beta}((S(\mathbf{y}_i^*) - S(\mathbf{y}))^\top)\right)^2 K_\varepsilon(d(S(\mathbf{y}_i^*), S(\mathbf{y}))).$$

Then the regression-adjusted sample can be taken as a draw from $\pi(\boldsymbol{\theta}|S(\mathbf{y}))$ (Nott et al., 2014) as

$$\boldsymbol{\theta}_{i,A}^* = \boldsymbol{\theta}_i^* + (S(\mathbf{y}) - S(\mathbf{y}_i^*))\,\widehat{\boldsymbol{\beta}} \approx \widehat{\boldsymbol{\alpha}} + \widehat{\epsilon}_i,$$

where the local-constant regression case (with $\boldsymbol{\beta} = 0$) corresponds to the local mean adjustment. In local linear approach, the choice of ε involves a bias-variance tradeoff since increasing ε reduces variance due to a larger sample size for fitting the regression, but also increases bias arising from departures from linearity and homoscedasticity (Beaumont et al., 2002).

Further variations of regression corrections have been proposed when the linearity and constant variance assumption are not appropriate. Blum (2010) proposes to model both the location and the scale of the θ_i using a nonlinear conditional heteroscedastic model. The details are given in Blum (2010) and the adjusted parameters are given by

$$\boldsymbol{\theta}_{i,B}^* = \widehat{m}(S(\mathbf{y})) + (\boldsymbol{\theta}_i^* - \widehat{m}(S(\mathbf{y}_i^*))\frac{\widehat{\sigma}(S(\mathbf{y}))}{\widehat{\sigma}(S(\mathbf{y}_i^*))}, \quad i = 1, \ldots, n,$$

where $\widehat{m}(S(\mathbf{y}_i^*))$ is the estimated conditional expectation $\mathrm{E}[\boldsymbol{\theta}^*|S(\mathbf{y}) = S(\mathbf{y}_i^*)]$ and $\widehat{\sigma}^2(S(\mathbf{y}_i^*))$ denotes the estimated conditional variance $\mathrm{Var}[\boldsymbol{\theta}^*|S(\mathbf{y}) = S(\mathbf{y}_i^*)]$. Blum (2010) investigates the asymptotic properties of regression correction and the method is provided in the R software package `nnet`. Extending this, Leuenberger and Wegmann (2010) use generalised linear models and weight all the accepted parameters similar to a Gaussian linear approximation in an inverse type regression. The details are given in Leuenberger and Wegmann (2010) with accompanying software named the `ABCtoolbox` written by Wegmann et al. (2010).

Example

To illustrate the theory we have presented and demonstrate the improvements that can be gained from post-sampling adjustments, we give an example using a simulated dataset. A Gaussian copula model with negative binomial marginals ($\pi = 0.3$, $s = 3$) and ARMA$(1,1)$ dependence ($a_1 = 0.7$, $b_1 = -0.5$) is used to simulate a dataset of length N=2000. We compute the sample proportions $S_1(\mathbf{y})$ and the spectral estimates $S_2(\mathbf{y})$ using the Bartlett-Priestley window with a bandwidth parameter of $M = 10$ and to ensure the covariance of the spectral estimates are close to zero we require $L \leq M$ (5.11) where L sets the spacing of the fixed frequencies $\omega_j = 2\pi j/L$. We let $L = 10$ resulting in $L/2 + 1 = 6$ dependence summary statistics in $S_2(\mathbf{y})$ (Section 5.3).

The ABC algorithm (Algorithm 3) is employed taking simple prior distributions,

$$\pi(a_1) = U[-1,1], \quad \pi(b_1) = U[-1,1], \quad \pi(s) = U(0,50], \quad \pi(\pi) = U[0,1].$$

We compute the metric T (5.12) on the spectral estimates and a Kullback-Leibler statistic (5.5) on the sample proportions. We run the algorithm until $n = 1000$ samples have been obtained and to save memory and ensure we do not save unnecessary simulated values far from the observed data, a moderate threshold of $\varepsilon_d = 0.2$ and $\varepsilon_T = 10$ is chosen (by preliminary testing) i.e., $d(S_1(\mathbf{y}), S_1(\mathbf{y}^*)) \leq 10$ and $T(S_2(\mathbf{y}), S_2(\mathbf{y}^*)) \leq 0.2$. Parallel computing is implemented for the algorithm which vastly decreases the computational time. For a time series of length N=2000, the time taken to obtain 10 samples is approximately two hours.

Fig. 5.1 shows the joint posterior densities for the accepted pair of copula and marginal parameter respectively. The range of axis represents the priors and we see they perform well, in particular the parameter s.

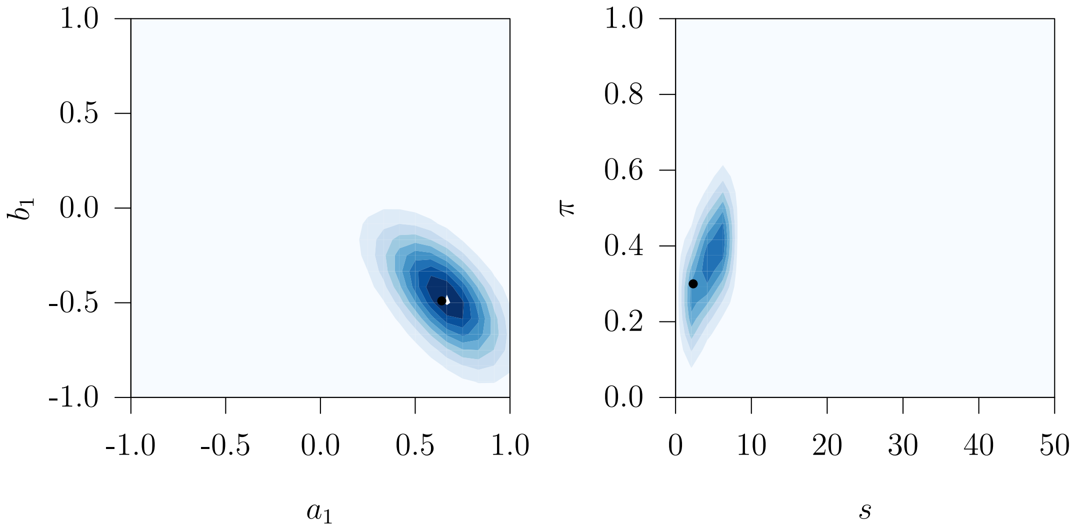


Figure 5.1: Plots of the joint posterior densities (a_1,b_1) and (s,π) for the ABC algorithm with $\varepsilon_d = 10$ and $\varepsilon_T = 0.2$. The true values are indicated by a circle.

Fig. 5.2 shows the accepted parameter values in grey plotted against their associated distance values (T for the copula parameters and d for the marginal parameters). The 0.1 quantile of the distance metrics are plotted in blue where the narrowing of the blue points around the troughs indicate the summary statistics are informative for the parameters. Fig. 5.2 suggests that perhaps a threshold of lower than 10 could have been used for T to tighten the range of a_1 and b_1 values accepted. However the trough is centred at the true value suggesting the regression adjustments will work well.

Figure 5.2: Output of $n = 1000$ parameter values of the ABC algorithm for the four parameter Gaussian copula model with true values ($a_1 = 0.7, b_1 = -0.5, s = 3, \pi = 0.3$). The 0.1 quantile values are shown in navy blue.

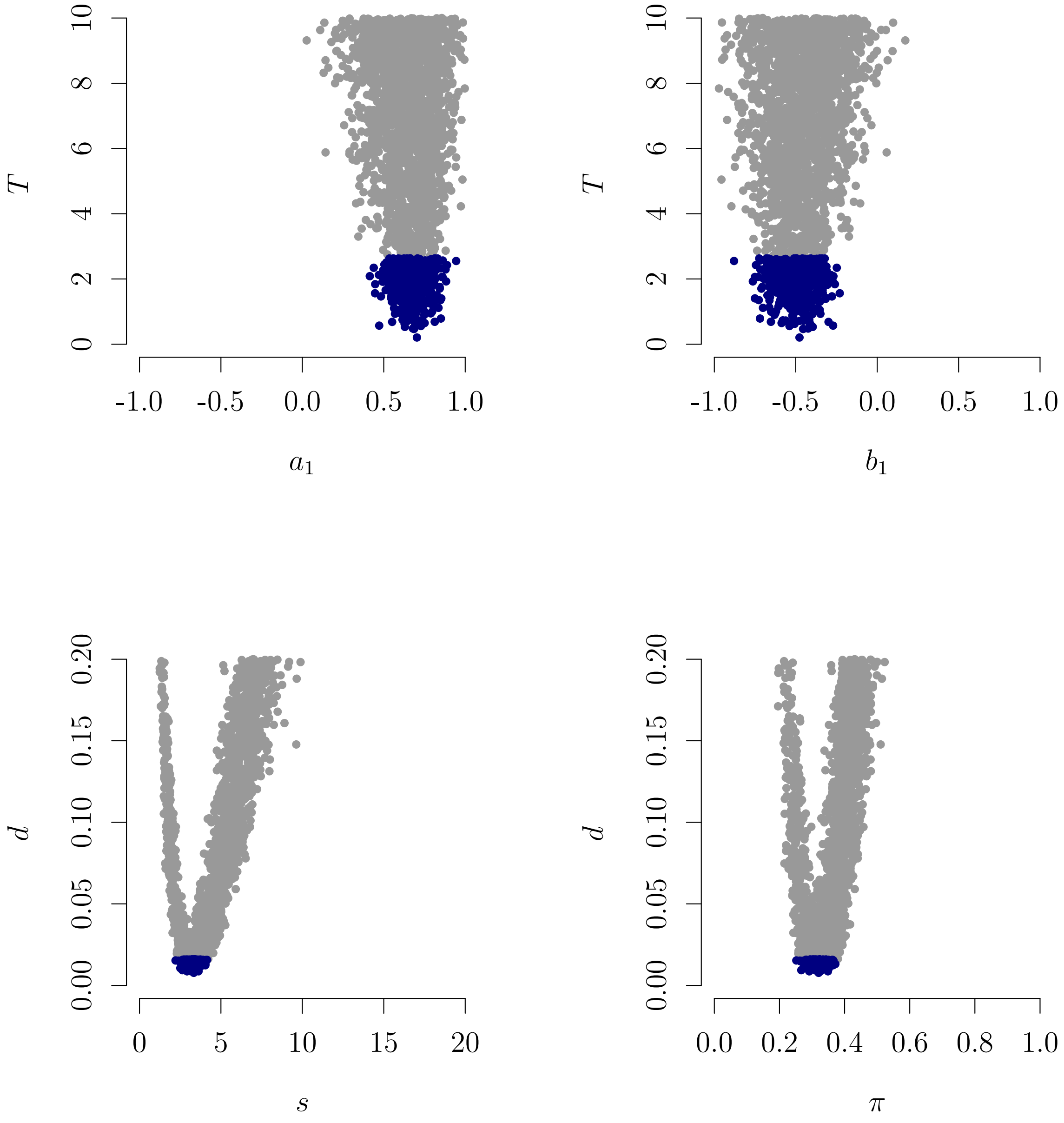
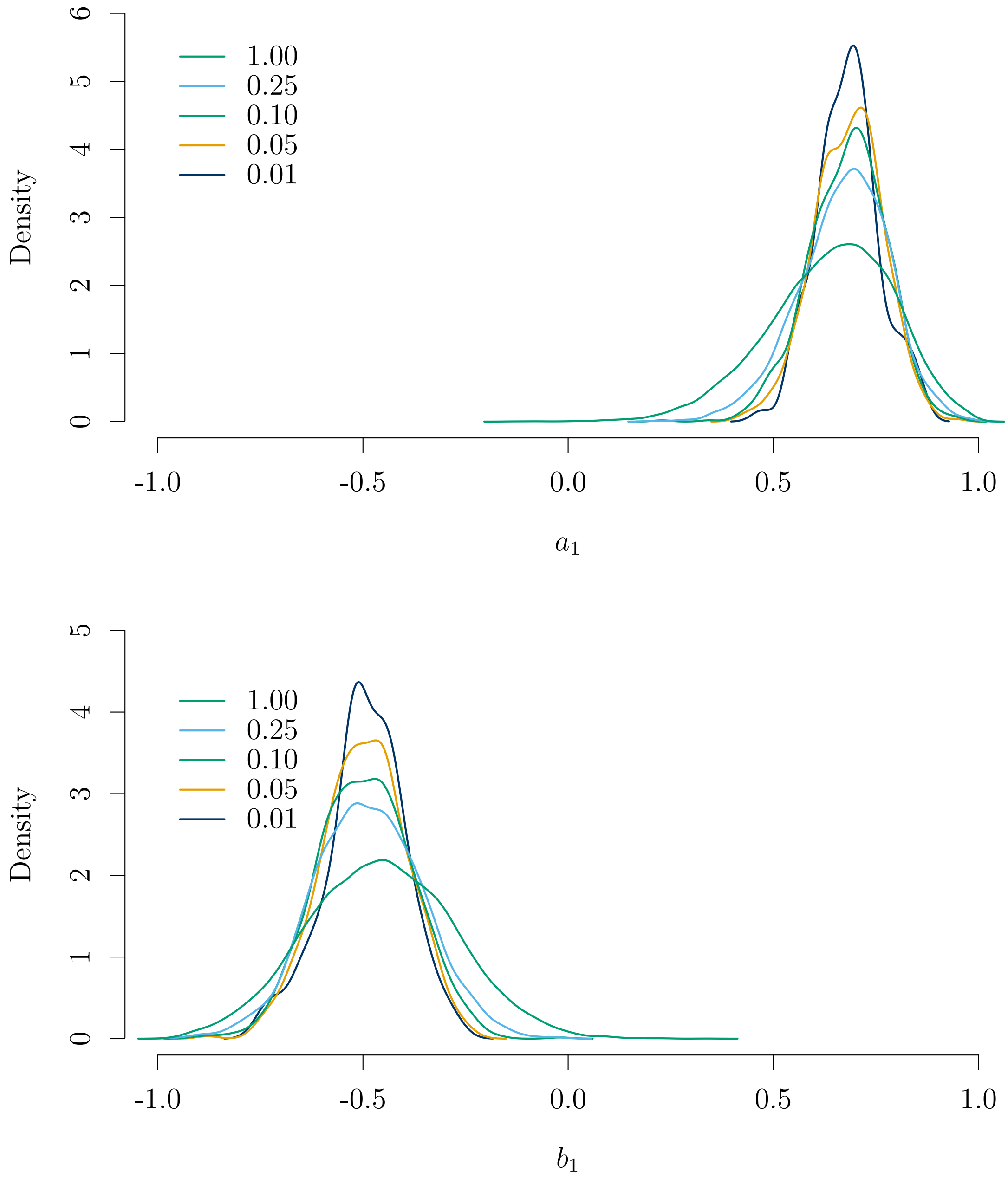


Figure 5.3: The posterior densities for the ABC output with the coloured lines corresponding to the 0.01, 0.05, 0.10 and 0.25 quantiles of the distances T for the parameters a_1 and b_1.



The posterior densities for the ABC output for a_1 and b_1 are given in Fig. 5.3 corresponding to 0.01, 0.05, 0.10 and 0.25 quantiles of the distance statistic T. There is a clear improvement as the distance metric decrease suggesting that weighting the smaller distances with larger weights will improve the posterior mean estimates further.

The regression adjustments of Section 5.5 are applied to the output of the $n = 1,000$ samples. Table 5.1 gives the mean of the raw ABC output and the weighted means of the regression adjusted values (5.14) and standard errors (5.16). All adjustments use the Epanechnikov kernel for weighting and the upper panel includes all $n = 1000$ samples and the lower panel includes the 0.1 quantile of values, i.e. $n = 100$. We see the algorithm performs well in all cases and there is a clear improvement with values getting closer to their true values after the post sampling adjustments. We see a reduction in the standard errors compared with the unadjusted estimates. We acknowledge that the reduction in standard errors could be sensitive to the choice of smoothing parameters in the regression adjustment and further work would assess the effect of this.

Table 5.1: The parameter estimates using the ABC algorithm and their standard errors for the Gaussian copula model with ARMA(1,1) dependence and negative binomial marginal distributions with true values are $a_1 = 0.7, b_1 = -0.5, \pi = 0.3$ and $s = 3$.

Method	$a_1 = 0.7$		$b_1 = -0.5$		$\pi = 0.3$		$s = 3$	
	mean	sd	mean	sd	mean	sd	mean	sd
$n = 1000$								
Unadjusted	0.674	(0.14)	-0.484	(0.17)	0.319	(0.06)	3.868	(1.46)
Linear adjust	0.706	(0.08)	-0.519	(0.10)	0.297	(0.01)	3.033	(0.01)
Nonlinear adjust	0.723	(0.07)	-0.524	(0.08)	0.288	(0.01)	2.797	(0.01)
0.1 quantile								
Unadjusted	0.691	(0.11)	-0.493	(0.13)	0.288	(0.02)	2.782	(0.36)
Linear adjust	0.715	(0.05)	-0.516	(0.08)	0.283	(0.01)	2.774	(0.01)
Nonlinear adjust	0.695	(0.03)	-0.481	(0.05)	0.290	(0.01)	2.831	(0.01)

5.6 Simulation under model conditions

To investigate the robustness of the model and to examine the effect on N on the application, we carry out a simulation study using $S = 100$ time series of lengths $N = 100, 250, 500, 1000, 2000$ from the Gaussian copula model of the previous example with negative binomial marginals ($\pi = 0.3$, $s = 3$) and ARMA$(1, 1)$ dependence ($a_1 = 0.7$, $b_1 = -0.5$). We also compare our ABC method for parameter estimation against Masarotto et al. (2012)'s importance sampling method from the `gcmr` R package (Masarotto and Varin, 2012). The mean, standard error, bias and RMSE are presented in Table 5.2.

Table 5.2 shows as N increase the estimates improve for each method. The regression adjustments improve the mean values and reduce the variance values significantly. Although the ABC algorithm does not compare to the `gcmr` for computational time, by comparing the ABC output with the importance sampling output we see that the ABC algorithm with the local nonlinear regression adjustment of (Blum, 2010) outperforms the `gcmr` in each scenario for $N \geq 250$ in terms of the mean estimates are closer to the true values and smaller regression adjusted sample variances.

Table 5.2: ABC Simulation Study of copula parameters for different length N.

		$a_1 = 0.7$						$b_1 = -0.5$					
Length	Method	mean	var	bias	MSE	RMSE	$\overline{\text{var}}(\widehat{a}_1)$	mean	var	bias	MSE	RMSE	$\overline{\text{var}}(\widehat{b}_1)$
N=100	ABC	0.121	0.004	-0.579	0.340	0.583	0.23609	0.069	0.004	0.569	0.328	0.573	0.23084
	+ linear	0.245	0.012	-0.455	0.219	0.468	0.00018	-0.019	0.015	0.481	0.246	0.496	0.00018
	+ nonlinear	0.421	0.029	-0.279	0.107	0.327	0.00015	-0.172	0.026	0.328	0.134	0.366	0.00015
	gcmr	0.481	0.142	-0.219	0.190	0.436	0.32062	-0.270	0.139	0.230	0.192	0.438	0.34158
N=250	ABC	0.265	0.011	-0.435	0.200	0.447	0.20342	-0.077	0.008	0.423	0.187	0.432	0.18953
	+ linear	0.428	0.011	-0.272	0.085	0.292	0.00013	-0.221	0.012	0.279	0.089	0.299	0.00013
	+ nonlinear	0.618	0.013	-0.082	0.020	0.142	0.00011	-0.387	0.019	0.113	0.032	0.178	0.00011
	gcmr	0.618	0.066	-0.082	0.073	0.270	0.16218	-0.415	0.067	0.085	0.074	0.272	0.18628
N=500	ABC	0.427	0.011	-0.273	0.086	0.292	0.12308	-0.223	0.013	0.277	0.089	0.299	0.12439
	+ linear	0.428	0.011	-0.272	0.085	0.292	0.00013	-0.221	0.012	0.279	0.089	0.299	0.00013
	+ nonlinear	0.669	0.008	-0.031	0.009	0.094	0.00006	-0.460	0.014	0.040	0.016	0.125	0.00006
	gcmr	0.646	0.014	-0.054	0.017	0.129	0.11779	-0.438	0.018	0.062	0.022	0.150	0.13695
N=1000	ABC	0.511	0.016	-0.189	0.051	0.227	0.07224	-0.320	0.014	0.180	0.047	0.217	0.07993
	+ linear	0.593	0.013	-0.107	0.025	0.157	0.00003	-0.401	0.016	0.099	0.026	0.160	0.00004
	+ nonlinear	0.645	0.010	-0.055	0.014	0.116	0.00003	-0.448	0.014	0.052	0.017	0.130	0.00003
	gcmr	0.676	0.005	-0.024	0.005	0.071	0.08031	-0.475	0.007	0.025	0.008	0.089	0.09558
N=2000	ABC	0.647	0.003	-0.053	0.006	0.077	0.02156	-0.452	0.004	0.048	0.006	0.080	0.02954
	+ linear	0.681	0.002	-0.019	0.003	0.053	0.00001	-0.489	0.004	0.011	0.004	0.061	0.00002
	+ nonlinear	0.697	0.002	-0.003	0.002	0.045	0.00001	-0.500	0.004	0.00*	0.004	0.060	0.00001
	gcmr	0.681	0.002	-0.019	0.003	0.051	0.05639	-0.483	0.003	0.017	0.004	0.059	0.06747

Table 5.3: ABC Simulation Study of marginal parameters for different length N (* denotes $< 1 \times 10^{-5}$).

		$\pi = 0.3$						$s = 3$					
Length	Method	mean	var	bias	MSE	RMSE	$\overline{\text{var}}(\widehat{\pi})$	mean	var	bias	MSE	RMSE	$\overline{\text{var}}(\widehat{s})$
N=100	ABC	0.436	0.005	0.136	0.024	0.154	0.01969	8.266	8.913	5.266	36.639	6.053	48.39249
	+ linear	0.406	0.005	0.106	0.016	0.126	0.00001	6.488	8.018	3.488	20.183	4.493	0.01925
	+ nonlinear	0.345	0.003	0.045	0.005	0.073	0.0000*	3.783	1.371	0.783	1.985	1.409	0.00972
	gcmr	0.325	0.006	0.025	0.006	0.079	0.07935	1.947	0.010	-1.053	1.119	1.058	0.10444
N=250	ABC	0.421	0.002	0.121	0.017	0.129	0.01228	7.539	2.901	4.539	23.503	4.848	22.54218
	+ linear	0.352	0.002	0.052	0.005	0.069	0.0000*	4.429	2.999	1.429	5.041	2.245	0.00597
	+ nonlinear	0.320	0.001	0.020	0.001	0.038	0.0000*	3.263	0.323	0.263	0.392	0.626	0.00237
	gcmr	0.329	0.002	0.029	0.003	0.057	0.04986	1.941	0.005	-1.059	1.126	1.061	0.06729
N=500	ABC	0.407	0.001	0.107	0.012	0.111	0.00901	7.080	0.810	4.080	17.459	4.178	13.31925
	+ linear	0.352	0.002	0.052	0.005	0.069	0.0000*	4.429	2.999	1.429	5.041	2.245	0.00597
	+ nonlinear	0.309	0.001	0.009	0.001	0.025	0.0000*	3.100	0.117	0.100	0.127	0.357	0.00088
	gcmr	0.336	0.001	0.036	0.002	0.050	0.03547	1.941	0.002	-1.059	1.124	1.060	0.04748
N=1000	ABC	0.407	0.001	0.107	0.012	0.110	0.00765	7.045	0.492	4.045	16.851	4.105	11.31999
	+ linear	0.309	0.001	0.009	0.001	0.026	0.0000*	3.068	0.453	0.068	0.458	0.677	0.00109
	+ nonlinear	0.307	0.00*	0.007	0.00*	0.020	0.0000*	3.098	0.069	0.098	0.078	0.280	0.00040
	gcmr	0.334	0.001	0.034	0.002	0.043	0.02486	1.945	0.001	-1.055	1.115	1.056	0.03372
N=2000	ABC	0.340	0.000*	0.040	0.002	0.042	0.00341	4.278	0.078	1.278	1.710	1.308	2.60583
	+ linear	0.304	0.000*	0.004	0.00*	0.015	0.0000*	3.050	0.073	0.050	0.075	0.274	0.00019
	+ nonlinear	0.307	0.000*	0.007	0.00*	0.015	0.0000*	3.086	0.037	0.086	0.044	0.210	0.00005
	gcmr	0.331	0.000*	0.031	0.001	0.036	0.01742	1.945	0.001	-1.055	1.113	1.055	0.02377

5.7 Discussion

We have demonstrate that by taking advantage of the fact we can generate data under the model conditional on the parameters, good estimates of the both the copula and the marginal model parameters can be found. We have improved the simple ABC estimation further by using local linear and nonlinear regression techniques allowing less samples to be required.

ABC can be limited by the availability of informative statistics but the simulation study has shown that the parameters with the lowest distances are closest to the true values confirming that the choice of summary statistics for integer-valued time series are informative about the dependence structure as well as the marginal structure.

A major advantage of the ABC rejection algorithm is the output are independent samples allowing parallel computation to be used. Besides decreasing computation time, an advantage of this is that the error from the approximation ε, decreases when the as computational effort increases and so we can computationally reduce this error by increasing the number of samples.

The inefficiency of the computation can also be improved by ensuring simulated parameters have a high probability of being accepted. Prangle (2014) proposes a method to stop the checking process early if data set is clearly different by using an initial checking system. This approach is termed 'lazy ABC' and uses a random stopping rule and appropriate re-weighting step so not to change the target distribution of standard ABC. For our model this approach is not necessary however it is worth noting for more complex models.

Other methods such as Monte Carlo Markov Chain (MCMC) and Sequential Monte Carlo (SMC) methods can be incorporated into ABC to help sample values in areas of high ABC-posterior probability. Incorporating MCMC methods into the ABC algorithm will make the algorithm more efficient by drawing parameter values from a proposal distribution which only depends on the previous iteration. ABC-MCMC was introduced by Marjoram et al. (2003) to explore the parameter space iteratively using the distance between the observed and simulated summary statistics to update the current parameter values. This ensures that parameter values which produce similar simulations close to the observed data are visited regularly. Simulated parameter values that fall within the threshold are not immediately included in the posterior sample and can still be rejected with a certain probability through the acceptance probability α. Csilléry et al. (2010) showed in the ABC-MCMC algorithm will converge to the target approximate distribution with absolute certainty.

Chapter 6

Estimation III: A Discrete Vine Pair-Copula Construction

The Gaussian copula model (3.8) at the core of this thesis has the ability to allow any desired discrete marginal distributions as well as the acclaimed ARMA structure to account for temporal dependence between observations.

The likelihood of the Gaussian copula model poses computational problems for maximum likelihood estimation (MLE) as discussed in Section 3.7 because it is the sum of 2^N terms. We have tried to overcome these using an MCEM algorithm in Chapter 4 and approximate Bayesian computation in Chapter 5. Each of these methods have their own merits and limitations: the MCEM approach works well for time series of length over 100 but becomes slow for lengths greater than 500, whilst our ABC approach works well for lengths greater than 1000 and above but fails to achieve accurate estimates for shorter time series as we will illustrate in Chapter 7. The details have been discussed in their chapter conclusions and will be elucidated further in the thesis discussion in Chapter 8. In this chapter we aim to speed up the direct computation of the log-likelihood using a vine pair-copula construction (PCC). The algorithm is so efficient that the marginal parameters as well as the dependence parameters can be estimated at the same time. Before proceeding to describe the decomposition of the log-likelihood, it is necessary to introduce the key ideas of vine pair-copula constructions also known simply as vine copulas.

6.1 Vine Pair-Copula Constructions (PCCs)

A vine pair-copula construction is a structure that represents an N-dimensional joint probability mass/density function in terms of bivariate copulas, called pair-copulas. Vines give a simple way of specifying conditionally dependent random variables (Bedford and Cooke, 2002) allowing the joint density to be decomposed into bivariate copula densities. The graphical representation resembles grape vines, lending its name to the class of models. It has been a significant advance in modelling high-dimensional data (Kurowicka and Cooke, 2002, 2006b; Brechmann et al., 2012; Stöber and Czado, 2012; Brechmann and Czado, 2013).

6.1.1 Literature on vines

The evolving literature on vine copulas contains seminal contributions from Aas et al. (2009), Cooke (1997), Kurowicka and Cooke (2006a), Bedford and Cooke (2001), Bedford and Cooke (2002) and Joe (1996) to name a few. Vine constructions are highly flexible due to the number of admissible R-vine structures in N-dimensions and the freedom to select arbitrary bivariate copulas for which there is a large library to choose from, many of which are listed in Joe (1997). The basic ideas have been refined and developed into strong mathematical objects that are highly practical in nature. Joint modelling using vines can be viewed in two ways:

- the first based on building high dimensional distributions from a collection of bivariate 'blocks' to give the overall joint distribution; *and*

- the second based on decomposing a multivariate density into a cascade of bivariate copulas.

The majority of the literature and applications of vines has focused on the former whilst our work focuses on the latter. The necessary information on vine copulas is included in this chapter. More details are provided in Czado (2010), Kurowicka and Cooke (2006b), Kurowicka and Joe (2010), Mai and Scherer (2012), and Joe (2015). Inference and estimation techniques for PCCs include sequential estimation (Haff et al., 2013; Dißmann et al., 2013), maximum likelihood (Czado, 2010) and Bayesian techniques (Min and Czado, 2010). Vine copulas have been used in a variety of applications from finance to medical disciplines requiring the need for efficient estimation methods.

Vine copulas allows us to factorise a joint density by successive conditioning and there are many options available for choosing the order in which you consider the variables. In the Gaussian time series context it makes sense to start at both ends of the series X_1 and X_N and work inwards because the required conditional distributions can then be calculated via the partial correlations. Before we present the general case, we will illustrate the decomposition of a joint density into pair-copula densities in two, three and then four dimensions. The method is recursive in nature and this is instructive because it allows very high-dimensional densities to be decomposed. We generalise to N dimensions and further to the discrete case (Section 6.3).

6.1.2 Motivating examples

The case in two dimensions

For the case of two dimensions, by Sklar's Theorem, the joint cdf F of X_1 and X_2 with marginals F_1 and F_2 can be written in terms of its copula C as

$$F(x_1, x_2) = C_{12}(F_1(x_1), F_2(x_2)),$$

where C is written with indices as C_{12} to identify the random variables in the distribution that it corresponds to. The joint density $f(x_1, x_2)$ can be found by taking partial derivatives wrt x_1 and x_2,

$$f(x_1, x_2) = c_{12}(F_1(x_1), F_2(x_2))f_1(x_1)f_2(x_2), \tag{6.1}$$

where c_{12} is the copula density C_{12} (Definition 9). Note that for *symmetric* copulas, such as the Gaussian copula, the arguments of C can be permuted so that $C_{12} = C_{21}$.

The case in three dimensions

For the case in three dimensions, the joint density can be factorised first into a product of univariate conditional densities, say

$$f(x_1, x_2, x_3) = f_{3|21}(x_3|x_2, x_1)\, f_{2|1}(x_2|x_1)\, f_1(x_1). \tag{6.2}$$

From (6.1) we can write the conditional density of X_2 given $X_1 = x_1$ as

$$\begin{aligned} f(x_2|x_1) &= \frac{f_{12}(x_1, x_2)}{f_1(x_1)}, \\ &= c_{12}(F_1(x_1), F_2(x_2))f_2(x_2). \end{aligned} \tag{6.3}$$

The conditional density $f(x_3|x_2, x_1)$ can be written in terms of a bivariate density by bringing x_1 to the front and then using a conditioned version of (6.1) in terms of its associated copula density,

$$\begin{aligned} f(x_3|x_2, x_1) &= \frac{f(x_3, x_1|x_2)}{f(x_1|x_2)}, \\ &= \frac{c_{13|2}(F_{3|2}(x_3|x_2), F_{1|2}(x_1|x_2))f_{1|2}(x_1|x_2)f_{3|2}(x_3|x_2)}{f_{1|2}(x_1|x_2)}, \\ &= c_{13|2}(F_{3|2}(x_3|x_2), F_{1|2}(x_1|x_2))f_{1|2}(x_1|x_2). \end{aligned}$$

The copula density $c_{13|2}(\cdot, \cdot)$ corresponds to the bivariate distribution $F_{13|2}$ of $(X_1, X_3)|X_2$. This is instructive because by putting x_1 together with x_3 in $c_{13|2}$ the conditioning set is reduced by one variable. The use of the univariate conditional densities is key to extending recursively to higher dimensions as demonstrated in (6.2).

The general form is

$$f_{i|jk}(x_i|x_j, x_k) = c_{ij|k}(F_{i|k}(x_i|x_k), F_{j|k}(x_j|x_k))f_{i|k}(x_i|x_k), \tag{6.4}$$

where k can be a single index or multiple indices. When k is empty (6.4) reduces to (6.3). Hence the joint density (6.2) can be written as

$$\begin{aligned} f(x_1, x_2, x_3) &= f_{3|21}(x_3|x_2, x_1)\, f_{2|1}(x_2|x_1)\, f_1(x_1), \\ &= c_{13|2}(F_{3|2}(x_3|x_2), F_{1|2}(x_1|x_2))\, f_{3|2}(x_3|x_2) \\ &\qquad \times c_{12}(F_1(x_1), F_2(x_2))\, f_1(x_1)\, f_2(x_2), \\ \\ &= c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))\, c_{23}(F_2(x_2), F_3(x_3))\, f_3(x_3) \\ &\qquad \times c_{12}(F_1(x_1), F_2(x_2))\, f_1(x_1)\, f_2(x_2). \end{aligned} \tag{6.5}$$

Thus the 3-variate density can be written in terms of 3 pair-copula densities and 3 marginal densities.

The case in four dimensions

For notational simplicity we omit the arguments $F_{1|2}$ and $F_{4|2}$ of $c_{14|2}(F_{1|2}(x_1|x_2), F_{4|2}(x_4|x_2))$ and simply write $c_{14|2}$.

Consider the case of a 4-variate density decomposed using (6.4) as

$$\begin{aligned} f(x_1, x_2, x_3, x_4) &= f_{4|321}(x_4|x_3, x_2, x_1) f_{3|21}(x_3|x_2, x_1) f_{2|1}(x_2|x_1) f_1(x_1), \\ &= c_{14|23} f_{4|23}(x_4|x_2, x_3) \cdot c_{13|2} f_{3|2}(x_3|x_2) \cdot c_{12} f_2(x_2) \cdot f_1(x_1), \\ &= c_{14|23}\, c_{24|3}\, f_{4|2}(x_4|x_2)\, c_{13|2}\, c_{23}\, f_3(x_3)\, c_{12}\, f_2(x_2)\, f_1(x_1), \\ &= c_{14|23}\, c_{24|3}\, c_{24}\, f_4(x_4)\, c_{13|2}\, c_{23}\, f_3(x_3) c_{12} f_2(x_2) f_1(x_1), \\ &= c_{14|23}\, c_{13|2}\, c_{24|3}\, c_{34}\, c_{23}\, c_{12}\, f_1(x_1)\, f_2(x_2)\, f_3(x_3)\, f_4(x_4), \end{aligned} \tag{6.6}$$

Thus the 4-variate joint density can be written in terms of six pair-copula densities:

$$\begin{aligned} &c_{12}(F_1(x_1), F_2(x_2)), && c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2)), \\ &c_{23}(F_2(x_2), F_3(x_3)), && c_{14|2}(F_{1|2}(x_1|x_2), F_{4|2}(x_4|x_2)), \\ &c_{34}(F_3(x_3), F_4(x_4)), && c_{14|23}(F_{1|23}(x_1|x_2, x_3), F_{4|23}(x_4|x_2, x_3)), \end{aligned}$$

and 4 marginal densities:

$$f_1(x_1),\ f_2(x_2),\ f_3(x_3),\ f_4(x_4).$$

This is a vine copula construction: the resulting copula is built from bivariate and conditional bivariate copulas. Essentially, by recursively applying (6.4) we can write high-dimensional densities as a product of bivariate copula and marginal densities. For continuous random variables, the joint density expressed using a vine copula construction with N continuous margins $F_1, \ldots, F_N$ is a product of $N(N-1)/2$ bivariate copula densities and N marginal densities. Bivariate copula are much lower-dimensional than N and thus computational demand in maximum likelihood estimation can be substantially reduced.

6.1.3 Regular Vines and graphical models

To keep the decompositions manageable and to ensure the decomposition chosen corresponds to a true joint density, each vine is graphically represented as a nested set of trees (Bedford and Cooke, 2001). The term *vine* arises because the hierarchical graphical structure commonly resembles a grape vine. Bedford and Cooke (2001) introduced the representation where the indices of the edges correspond to the bivariate conditional specifications. The nodes in tree one represent random variables. Tree one reflects the conditioning of the variables. The edges of tree one become the nodes of tree two and each edge of subsequent trees are the nodes of the current tree. The edges of the trees are associated with pair-copula densities because they identify the variable decomposition i.e., they describe which conditional specifications are being made on the joint distribution. By the formation of subsequent trees, the vine is unique to each factorisation up to relabelling.

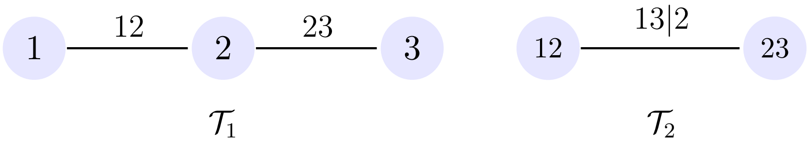


Figure 6.1: A vine structure on three elements.

The vine copula in Fig. 6.1 represents the three variable decomposition of (6.5),

$$
\begin{aligned}
f(x_1, x_2, x_3) = c_{12}(F_1(x_1), F_2(x_2))\, c_{23}(F_2(x_2), F_3(x_3)) \\
\times\, c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))\, f_1(x_1)\, f_2(x_2)\, f_3(x_3),
\end{aligned}
$$

where the edges of vine correspond to the pair-copula indices. Each pair of variables occurs once on the left hand side of the copula density index. Another example with four variables is given in Fig. 6.2. This vine structure represents the decomposition

$$
\begin{aligned}
f(x_1, x_2, x_3, x_4) = c_{12}(F_1(x_1), F_2(x_2))\, c_{23}(F_2(x_2), F_3(x_3))\, c_{24}(F_4(x_4), F_4(x_4)) \\
\times\, c_{13|2}(F_{1|2}(x_1|x_2), F_{3|2}(x_3|x_2))\, c_{14|2}(F_{1|2}(x_1|x_2), F_{4|2}(x_4|x_2)) \\
\times\, c_{34|12}(F_{3|12}(x_3|x_1, x_2), F_{4|12}(x_4|x_1, x_2)) f_1(x_1) f_2(x_2) f_3(x_3) f_4(x_4),
\end{aligned}
$$

where the edges of the trees are the copula density indices. Each subsequent tree corresponds to a higher order conditioning.

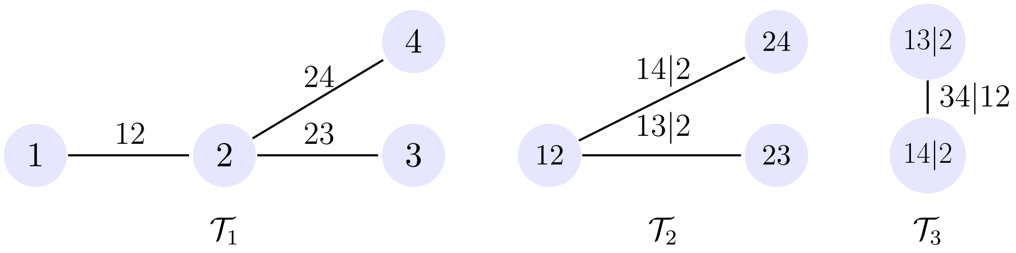


Figure 6.2: A vine structure on four elements.

6.1.4 R-vines

In graph theory, a tree is defined as a simple undirected connected graph. A vine structure on N elements is a nested set of trees $\{\mathcal{T}_j; j = 1, \ldots, N-1\}$, where the edges of tree j are the nodes of the following tree $j+1$, as shown in Fig.'s 6.1 and 6.2. Each tree has the maximum number of edges and letting $\mathcal{V}$ denote the vine, we write $\mathcal{V} = (\mathcal{T}_1, \ldots, \mathcal{T}_{N-1})$. A regular vine (R-vine) on N elements is a vine which satisfies the proximity condition.

Definition 10 (Regular Vine, R-vine)**.** *Let N_j denote the set of nodes of tree j and $\mathcal{E}_j$ denote the set of edges. A* Regular vine *(R-vine) $\mathcal{V}$ on N elements is a nested set of trees, $\mathcal{T}_j = (N_j, \mathcal{E}_j)$, $j = 1, \ldots, N-1$ satisfying the following conditions:*

1. $\mathcal{V} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{N-1}\}$;
2. $\mathcal{T}_1$ *is a connected tree with nodes* $\mathcal{N}_1 = \{1, \ldots, N\}$ *and edges* $\mathcal{E}_1$,
   $\mathcal{T}_j$ *has nodes* $\mathcal{N}_j = \mathcal{E}_{j-1}$, $j = 2, \ldots, N-1$,
3. (Proximity Condition) *Two nodes are joined by an edge in $\mathcal{T}_j$ if their corresponding edges in $\mathcal{T}_{j-1}$ share a common node, for $j = 2, \ldots, N-1$.*

Each edge of $\mathcal{T}_j$ corresponds to a bivariate copula density. Combining these bivariate copula densities with marginal densities gives an N-dimesnional joint density, as shown in (6.5) and (6.6). The combination for discrete variables is considered in the Section 6.3. The proximity condition of a vine ensures that the N-dimesnional vine copula construction is a true distribution function in the sense that the decomposition into bivariate copulas is well-defined (Czado et al., 2013). Graphically, if tree one of a vine is a connected tree, then the complete vine structure is classified as an R-vine. In summary, the graphical representation of a vine is simply a way of displaying the unique vine copula.

6.1.5 Special cases: the D-vine and the C-vine

Each factorisation of the joint density into bivariate copulas and marginal densities is unique, up to relabelling. Thus each vine represents a unique decomposition. For the case of three random variables there are three decompositions of $f(x_1, x_2, x_3)$ corresponding to three possible vines,

$$c_{12|3}\ c_{13}\ c_{23}\ f_1\ f_2\ f_3, \qquad c_{13|2}\ c_{12}\ c_{23}\ f_1\ f_2\ f_3, \qquad c_{23|1}\ c_{12}\ c_{13}\ f_1\ f_2\ f_3.$$

The decomposition in (6.5) puts x_3 with x_1, rather than x_2, resulting in the second decomposition here. Each of these vines correspond to choosing a different decomposition or a different variable to put together in the factorisation.

Two common factorisations are the C-vine and the D-vine (Aas et al., 2009). These are special cases of R-vines.

A D-vine factorisation chooses the variable from the conditioning set such that the pairs of the random variables have the largest difference, i.e. for N variables $f(x_1, x_N|x_2, \ldots, x_{N-1})$, then $f(x_2, x_{N-1}|x_3, \ldots, x_{N-2})$ etc. These structures are most effective in situations when all the variables are of equal importance. A D-vine corresponds to each node having no more than 2 edges attached with each node present. Hence the trees of a D-vine are simply strings of nodes.

Definition 11 (Drawable Vine, D-vine)**.** *A* D-vine *is an R-vine such that no node in any tree $\mathcal{T}_j$ is connected to more than two edges.*

An example of a D-vine on 4 elements is given in Fig. 6.3.

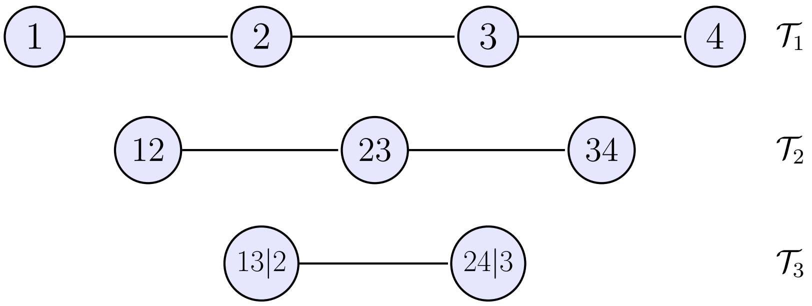


Figure 6.3: An example of a 4-dimensional D-vine.

The decomposition of $f(x_1, \dots, x_N)$ according to a D-vine can be written as

$$\begin{aligned} f(x_1, \dots, x_N) &= \prod_{k=1}^{N} f_k(x_k) \\ &\times \prod_{i=1}^{N-1} \prod_{j=1}^{N-i} c_{j,j+i|(j+1):(j+i-1)} \left(F(x_j | x_{j+1:j+i-1}), F(x_{j+i} | x_{(j+1):(j+i-1)}) \right), \end{aligned} \tag{6.7}$$

where $c_{j,j+i|(j+1):(j+i-1)}$ is the density of Y_j and Y_{j+i} conditioned on $Y_{j+1},\ dots, Y_{j+i-1}$. The index i represents the tree while j runs over the edges of the trees. For example,

$$f = f_1\ f_2\ f_3\ f_4\ c_{12}\ c_{23}\ c_{34}\ c_{13|2}\ c_{24|3}\ c_{14|23}, \qquad \text{for } N = 4.$$

A C-vine factorisation chooses the variable which is considered to have most dependence and the variable that has the smallest difference e.g. $f(x_1, x_2|x_3, \dots, x_N)$. A C-vine corresponds to setting a node in the first tree as a pivot node to which every other node is attached. The dependence with respect to that pivot node is modelled for each pair and the resulting trees of a C-vine have a star-like tree structure.

Definition 12 (Canonical Vine, C-vine)**.** *A* C-vine *is an R-vine such that each tree $\mathcal{T}_j$ has a unique node that is connected to $N - j$ edges.*

An example of a C-vine on 4 elements is given in Fig. 6.4 where X_1 is assumed to be of most importance in the dependence structure.

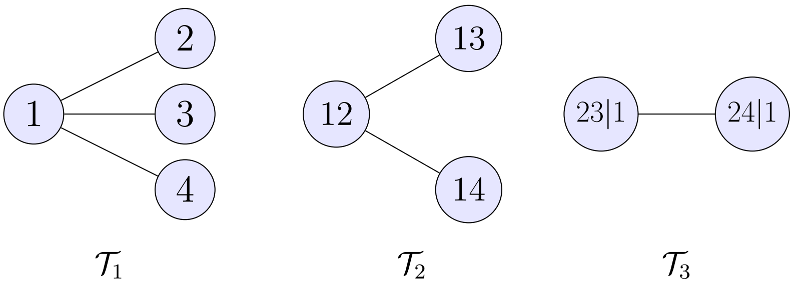


Figure 6.4: An example of a 4-dimensional C-vine.

The decomposition of $f(x_1, \ldots, x_N)$ according to a C-vine can be written as

$$f(x_1, \ldots, x_N) = \prod_{k=1}^{N} f_k(x_k) \times \prod_{i=1}^{N-1} \prod_{j=1}^{N-i} c_{i,i+j|1,\ldots,i-1} \left(F(x_i|x_{1:i-1}), F(x_{i+j}|x_{1:i-1}) \right). \quad (6.8)$$

For example,

$$f = f_1 \; f_2 \; f_3 \; f_4 \; c_{12} \; c_{13} \; c_{14} \; c_{23|1} \; c_{24|1} \; c_{34|12}, \qquad \text{for } N = 4.$$

We take the approach of the D-vine because for the Gaussian time series context, the required conditional densities can be calculated via the partial correlations.

6.2 Evaluation of an R-vine copula density

The D-vine and C-vine are special cases of the R-vine. In this section we outline an algorithm for evaluating an continuous R-vine density (6.12) proposed by Dißmann et al. (2013). In Section 6.5 we modify the algorithm of Dißmann et al. (2013, Algorithm 2.1, p. 58) to evaluate a D-vine with discrete margins for maximum likelihood estimation. Before we introduce the algorithm for the continuous R-vine in Section 6.2.3, we address the issue of evaluating the arguments of the copula densities. We then describe the discrete analogue for our model in Section 6.3.

6.2.1 Evaluation of the copula arguments

The joint density (6.12) involves the evaluation of copula densities. The copula density (Section 3.1)

$$c_{ij|k} \left(F_{i|k} \left(x_i | x_k \right), F_{j|k} \left(x_j | x_k \right) \right)$$

is applied to the arguments $F_{i|k}(x_i|x_k)$ and $F_{j|k}(x_j|x_k)$. The evaluation of the joint density is sequential in that it builds up the conditional distributions in a hierarchical manner from those with a lower-dimensional conditioning set. For example, in the three dimensional case of (6.5)

1. first C_{12} can be applied to the marginal cdfs F_1 and F_2 allowing $F_{1|2}$ and $F_{2|1}$ to be found via (6.9), similarly for C_{13}.
2. then C_{13} can be applied to the marginal cdfs F_1 and F_3 allowing $F_{1|3}$ and $F_{3|1}$ to be found
3. Then $C_{23|1}$ can be applied to $F_{2|1}$ and $F_{3|1}$ (allowing $F_{2|31}$ and $F_{3|21}$ to be obtained if desired).

Formally, this idea to evaluate the arguments $F_{i|k}\left(x_i|x_k\right)$ and $F_{j|k}\left(x_j|x_k\right)$, a key result of pair-copula constructions given by Joe (1996). Consider for conditional distributions $F_{ij|k}$ of $(X_i, X_j) \mid X_k = x_k$,

$$F_{ij|k}\left(x_i, x_j|x_k\right) = C_{ij|k}\left(F_{i|k}\left(x_i|x_k\right), F_{j|k}\left(x_j|x_k\right)\right),$$

where $C_{i,j|k}$ is the corresponding copula (by Sklar's Theorem), assuming its functional form does not depend on x_k. For continuous random variables the conditional distribution function of X_i given $X_j = x_j$ and $X_k = x_k$ is equal to the partial derivative of the copula $C_{ij|k}$ with respect to its second argument $F_{j|k}(x_j|x_k)$. It is written as

$$F_{i|jk}\left(x_i|x_j, x_k\right) = \frac{\partial\, C_{i,j|k}\left(F_{i|k}\left(x_i|x_k\right), F_{j|k}\left(x_j|x_k\right)\right)}{\partial F_{j|k}\left(x_j|x_k\right)}. \tag{6.9}$$

To see how 6.9 is useful and provide introductory details for Section 6.4, consider the distribution function $F(x_1, x_2)$ for 2 continuous random variables (X_1, X_2) then by Sklar's Theorem we have

$$F(x_1, x_2) = C(u_1, u_2) \quad \text{where } u_i = F(x_i), \quad i = 1, 2.$$

For illustrative purposes consider differentiating with respect to $F_2(x_2)$. We have the partial derivative of the copula C with respect to one of its arguments, say u_2, equal to the conditional distribution of X_1 at x_1 given $X_2 = x_2$ i.e.,

$$\frac{\partial}{\partial F_2(x_2)} C(F_1(x_1), F_2(x_2)) = F_{1|2}(x_1|x_2).$$

This is because

$$\begin{aligned}
\frac{\partial}{\partial F(x_2)} C(F(x_1), F(x_2)) &= \frac{\partial}{\partial F(x_2)} F(x_1, x_2), \\
&= \frac{\partial}{\partial x_2} \left(F(x_1, x_2)\right) \frac{\mathrm{d}x_2}{\mathrm{d}F(x_2)}, \\
&= \frac{\partial}{\partial x_2} \left(F(x_1, x_2)\right) \frac{1}{f(x_2)}, \\
&= \frac{1}{f(x_2)} \frac{\partial}{\partial x_2} \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} f(x_1', x_2')\, \mathrm{d}x_1'\, \mathrm{d}x_2', \\
&= \frac{1}{f(x_2)} \int_{-\infty}^{x_1} f(x_1', x_2)\, \mathrm{d}x_1', \\
&= \int_{-\infty}^{x_1} f(x_1'|x_2)\, \mathrm{d}x_1', \\
&= F(x_1|x_2).
\end{aligned}$$

The N-dimensional Gaussian copula model (3.8) can be decomposed into $N(N-1)/2$ bivariate Gaussian copulas as we go on to show. For the Gaussian pair-copula of $F_{ij|k}(x_i, x_j|x_k)$,

$$C(u_1, u_2; \rho) = \Phi_\rho(\Phi^{-1}(u_1), \Phi^{-1}(u_2)), \tag{6.10}$$

with $u_1 = F_{i|k}(x_i|x_k)$ and $u_2 = F_{j|k}(x_j|x_k)$, (6.9) becomes

$$\begin{aligned}
F_{i|jk}\left(x_i|x_j, x_k\right) &= \frac{\partial}{\partial u_2} C(u_1, u_2)\Big|_{u_1=F(x_1), u_2=F(x_2)}, \\
&= \Phi\left(\frac{\Phi^{-1}(F_{j|k}(x_j|x_k)) - \rho\Phi^{-1}(F_{i|k}(x_i|x_k))}{\sqrt{1-\rho^2}}\right).
\end{aligned} \tag{6.11}$$

Therefore (6.11) is a key part of the computation.

6.2.2 R-vine density and R-vine matrices

The R-vine distribution density is

$$\begin{aligned}
&f(x_1, \ldots, x_N) \\
&= \prod_{j=1}^{N} f_j(x_j) \times \prod_{k=N-1}^{1} \prod_{i=N}^{k+i} c_{m_{k,k}, m_{i,k}|m_{i+1,k},\ldots,m_{N,k}} \left(F_{m_{k,k}|m_{i+1,k},\ldots,m_{N,k}}, F_{m_{i,k}|m_{i+1,k},\ldots,m_{N,k}}\right),
\end{aligned} \tag{6.12}$$

where $M = (m_{i,j})_{i,j=1,\ldots,N}$ is an R-vine matrix. The densities according to the D- (6.7) and C-vine (6.8) are special cases of (6.12).

Each vine structure can be stored in a lower triangular N by N matrix called an R-vine matrix which is used to select copula indices (vine edges) to aid computational inference of the models. Dißmann et al. (2013) introduced the R-vine matrix for efficient storage of the R-vine tree structure in an algorithm to compute the log-likelihood of any R-vine. A description and example of constructing an R-vine matrix and reconstructing the corresponding tree sequence from an R-vine matrix is given in Stöber (2013, p. 13). There is a one-to-one correspondence for each R-vine matrix and there is a general form for the special cases of the D- and C-vine structures. The general form of a D-vine matrix is the lower triangular elements of a Toeplitz matrix with elements 0 to $N-1$ and then with a diagonal of N to 1 added:

$$M_D = \begin{pmatrix} N & 0 & \ldots & \ldots & 0 \\ 1 & N-1 & \ddots & \vdots & 0 \\ 2 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & 1 & 2 & 0 \\ N-1 & N-2 & \ldots & 1 & 1 \end{pmatrix}, \quad \text{e.g. for } N=4 \quad M_D = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 2 & 1 & 2 & 0 \\ 3 & 2 & 1 & 1 \end{pmatrix}. \tag{6.13}$$

The general form of a C-vine matrix is a lower triangular matrix with rows elements equal to diagonal elements such that the diagonal is N to 1:

$$M_C = \begin{pmatrix} N & 0 & \ldots & \ldots & 0 \\ N-1 & N-1 & \ddots & \vdots & 0 \\ N-2 & N-2 & \ddots & 0 & 0 \\ \vdots & \ldots & 2 & 2 & 0 \\ 1 & 1 & \ldots & 1 & 1 \end{pmatrix}, \quad \text{e.g. for } N=4 \quad M_C = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 2 & 2 & 2 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

6.2.3 Evaluation of an R-vine copula density

In addition to storing the R-vine structure, the algorithm of Dißmann et al. (2013, Algorithm 2.1, p 58) stores the associated copula parameters in a strictly lower triangular $N \times N$ matrix, denoted P^*. For the Gaussian copula model with all Gaussian bivariate copulas all elements of P^* are the conditional correlations. The matrix P^* is defined as follows and for illustrative purposes the special case for the D-vine matrix in (6.13) is given, (the vine is shown in Fig. 6.3):

$$P^* = \begin{pmatrix} 0 & 0 & 0 & 0 \\ p_{4m_{2,1}|m_{3,1}m_{4,1}} & 0 & 0 & 0 \\ p_{4m_{3,1}|m_{4,1}} & p_{3m_{3,2}|m_{4,2}} & 0 & 0 \\ p_{4|m_{4,1}} & p_{3|m_{4,2}} & p_{4|m_{4,3}} & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ p_{14|23} & 0 & 0 & 0 \\ p_{24|3} & p_{13|2} & 0 & 0 \\ p_{34} & p_{23} & p_{12} & 0 \end{pmatrix},$$

The algorithm stores the evaluations of the conditional distribution functions in two lower triangular $N \times N$ matrices denoted v^{direct} and v^{indirect} where the subscripts are used to represent the different index set. The general form and corresponding matrices for our example (6.13) are

$$\begin{aligned} v^{\text{direct}} &= \begin{pmatrix} F_{4|m_{2,1},m_{3,1},m_{4,1}} & 0 & 0 & 0 \\ F_{4|m_{3,1},m_{4,1}} & F_{3|m_{3,2},m_{4,2}} & 0 & 0 \\ F_{4|m_{4,1}} & F_{3|m_{4,2}} & F_{2|m_{4,3}} & 0 \\ F_4 & F_3 & F_2 & F_1 \end{pmatrix} = \begin{pmatrix} F_{4|321} & 0 & 0 & 0 \\ F_{4|32} & F_{3|21} & 0 & 0 \\ F_{4|3} & F_{3|2} & F_{2|1} & 0 \\ F_4 & F_3 & F_2 & F_1 \end{pmatrix}, \\ v^{\text{indirect}} &= \begin{pmatrix} F_{m_{2,1}|m_{3,1}m_{4,1}4} & 0 & 0 & 0 \\ F_{m_{3,1}|m_{4,1},4} & F_{m_{3,2}|m_{4,2}3} & 0 & 0 \\ F_{m_{4,1}|4} & F_{m_{4,2}|3} & F_{m_{4,3}|2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} F_{1|234} & 0 & 0 & 0 \\ F_{2|34} & F_{1|23} & 0 & 0 \\ F_{3|4} & F_{2|3} & F_{1|2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{6.14}$$

We now describe the algorithm to compute an R-vine density (6.12), and hence the log-likelihood, in Algorithm 4. The algorithm consists of two loops; one loop iterates over the column entries and one iterates over the rows. Essentially this equates to filling in the elements of (6.14) column by column; beginning in the final column, from bottom to the top, until the top left corner is reached. This allows higher order conditional distributions to be calculated from the elements of the row below as in (6.9). In each loop, there are three steps:

1. The copula arguments are selected and saved as z_1 and z_2 (lines 7 through to 9) then,
2. the copula density function is applied to the arguments and R-vine density updated (line 10) and finally,
3. the higher-order conditional distributions are computed via (6.9) (line 11).

The $\textsc{n} \times \textsc{n}$ lower triangular matrix $\widetilde{M}$ is introduced to work with the R-vine matrix M to select the pair-copula indices. The matrix $\widetilde{M}$ has elements $\widetilde{m}_{i,k} = \max\{m_{i,k}, \ldots, m_{\textsc{n},k}\}$, $k = 1, \ldots, \textsc{n}$ and $i = k, \ldots, \textsc{n}$. That is $\widetilde{M}$ is the maximum array of M. For example continuing with our D-vine example (6.13),

$$\widetilde{M} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 \\ 3 & 2 & 2 & 0 \\ 3 & 2 & 1 & 1 \end{pmatrix}.$$

Algorithm 4 Evaluation of an R-vine density for continuous variables (Dißmann et al., 2013).

Input: R-vine matrix M with associated matrix P^* and a dataset $\mathbf{x}$.
Output: Density of an R-vine PCC at P^*.

```
Set $\mathrm{f} = 1$
Allocate $N \times N$ matrices, $v^{\text{direct}}, v^{\text{indirect}}$ as in (6.14).
Set $\widetilde{M}$ such that $\widetilde{m}_{i,k} = \max\{m_{i,k}, \dots, m_{N,k}\}, k = 1, \dots, N$ and $i = k, \dots, N$
    $\in \mathbb{R}^{N \times N}$.
Set $(v^{\text{direct}}_{N,1}, \dots, v^{\text{direct}}_{N,N}) = (F_N(x_N), \dots, F_1(x_1))$.

for $k = N - 1, \dots 1$
    for $i = N, \dots k + 1$
        Set $z_1 = v^{\text{direct}}_{i,k}$.                                      ▷ Select copula arguments
        if $\widetilde{m}_{i,k} = m_{i,k}$ then   set $z_2 = v^{\text{direct}}_{k,(N+1-\widetilde{m}_{i,k})}$,
        else                       set $z_2 = v^{\text{indirect}}_{k,(N+1-\widetilde{m}_{i,k})}$.

        $f = f \cdot c(z_1, z_2; p_{i,k})$.                                  ▷ Evaluate copula density

        $v^{\text{direct}}_{i-1,k} = \frac{\partial}{\partial z_1} C(z_1, z_2; p_{i,k})$ and        ▷ Conditional distributions
          $v^{\text{indirect}}_{i-1,k} = \frac{\partial}{\partial z_2} C(z_2, z_1; p_{i,k})$.                ▷ as in (6.9)
    end
end
Return $f$                          ▷ For the log-likelihood replace with $\ell = \log f$.
```

6.3 Pair-copula constructions for discrete data

Earlier research has largely focused on the case of purely continuous random variables (Aas et al., 2009). Panagiotelis et al. (2012) extended the principles of PCC to discrete variables. Their discrete analogue to vine PCCs, which we provide the details of in this section, requires only $2N(N-1)$ pair-copula evaluations. This is significantly less demanding than the 2^N evaluations of an N-dimensional copula required to evaluate the pmf of a copula model. Hence it is a substantial step towards simplifying the evaluation of multivariate discrete probabilities and consequently in fitting the models.

Stöber (2013) adapted the systematic approach of evaluating continuous R-vine densities of Dißmann et al. (2013) (Section 6.2.3), to the cases of both continuous, discrete and mixed margins by building on Panagiotelis et al. (2012)'s discussion of discrete PCCs. Stöber et al. (2015) illustrates this by implementing a mixed continuous and discrete R-vine PCC for modelling the co-morbidity on a longitudinal study of ageing (LSOAII) dataset with 6 variables.

At the time of writing, no R code is publicly available and software is not currently available for statisticians to apply the methods. For the case of integer-valued time series, typically the sample size or dimension N is very large. The increase in computational complexity is tackled by making use of existing methods in time series analysis. We apply the ideas of Panagiotelis et al. (2012) and Stöber (2013) to extend to the integer-valued time series model with discrete margins and an ARMA(p, q) dependence structure of the Gaussian copula. Therefore our contribution allows joint maximum likelihood estimation of the Gaussian copula model (3.8) for length N integer-valued time series with $p + q + \dim(\boldsymbol{\theta}_{\text{marginal}})$ parameters.

The joint probability mass function can be recursively factorised similar to the continuous case described in Section 6.1.2. Before we describe the method to decompose a multivariate probability mass function into a function of bivariate copulas, we introduce the notation that will be used. The standard notation for a cumulative distribution function is

$$F_{Y_i,Y_j|Y_k}(y_i, y_j|y_k) = \Pr(Y_i \leq y_i, Y_j \leq y_j|Y_k = y_k).$$

However we will drop the variables and retain the indices only, e.g., $F_{i,j|k}(y_i, y_j|y_k)$. For the unconditional and conditional distribution functions evaluated at the two integer-values y_i and $y_i^- = y_i - 1$, we use

$$\begin{aligned} F_i &= F_i\left(y_i\right), & F_i^- &= F_i(y_i - 1), \\ F_{i|k} &= F_{i|k}(y_i \mid Y_k = y_k), & F_{i|k}^- &= F_{i|k}(y_i - 1 \mid Y_k = y_k). \end{aligned}$$

By Sklar's Theorem, the bivariate distribution function of $(Y_i, Y_j) \mid Y_k = y_k$ can be expressed in terms of a copula. For example,

$$F_{ij|k}(y_i, y_j|y_k) = C_{ij|k}(F_{i|k}(y_i|y_k), F_{j|k}(y_j|y_k)).$$

For the copula evaluated at four different pairs of values, let

$$\begin{aligned} C_{ij|k}^{00} &= C_{ij|k}(F_{i|k}, F_{j|k}), & C_{ij|k}^{01} &= C_{ij|k}(F_{i|k}, F_{j|k}^-), \\ C_{ij|k}^{10} &= C_{ij|k}(F_{i|k}^-, F_{j|k}), & C_{ij|k}^{11} &= C_{ij|k}(F_{i|k}^-, F_{j|k}^-). \end{aligned}$$

Therefore the conditional probability

$$\Pr(Y_i = y_i, Y_j = y_j|Y_k = y_k) = C_{ij|k}^{00} - C_{ij|k}^{01} - C_{ij|k}^{10} + C_{ij|k}^{11}.$$

For simplification of notation to make equations shorter we write

$$\nabla C_{ij|k} = C_{ij|k}^{00} - C_{ij|k}^{01} - C_{ij|k}^{10} + C_{ij|k}^{11}.$$

It is clear from this definition that

$$\begin{aligned} \Pr(Y_i = y_i|Y_j = y_j, Y_k = y_k) &= \frac{C_{ij|k}^{00} - C_{ij|k}^{01} - C_{ij|k}^{10} + C_{ij|k}^{11}}{F_{j|k}(y_j|y_k) - F_{j|k}(y_j - 1|y_k)}, \\ &= \frac{\nabla C_{ij|k}}{\nabla F_{j|k}(y_j|y_k)}. \end{aligned} \tag{6.15}$$

We use the backward difference notation ∇ to denote the finite difference of step length one. Thus the expression consists of two terms for a univariate function e.g. a distribution function $F_{j|k}$ and four terms for a bivariate function e.g. a copula $C_{ij|k}$.

In a hierarchical manner we can build up the conditional distributions from those with a lower-dimensional conditioning set by writing the conditional distribution function in terms of a bivariate copula,

$$\begin{aligned} F_{i|j,k}\left(y_i|y_j, y_k\right) &= \frac{C_{i,j|k}\left(F\left(y_i|y_k\right), F\left(y_j|y_k\right)\right) - C_{i,j|k}\left(F\left(y_i|y_k\right), F(y_j^-|y_k)\right)}{\Pr\left(Y_j = y_j|Y_k = y_k\right)}, \\ &= \frac{C^{00}_{ij|k} - C^{01}_{ij|k}}{\nabla F_{j|k}}. \end{aligned} \tag{6.16}$$

This can be considered the discrete analogue of (6.9) for computing the arguments of the pair-copulas.

For the bivariate Gaussian copula, (6.16) is

$$F_{i|jk}(y_i|y_j, y_k) = \frac{\Phi_\rho(\Phi^{-1}(F_{i|k}(y_i|y_k)), \Phi^{-1}(F_{j|k}(y_j|y_k))) - \Phi_\rho(\Phi^{-1}(F_{i|k}(y_i|y_k)), \Phi^{-1}(F_{j|k}(y_j^-|y_k)))}{F_{j|k}(y_j|y_k) - F_{j|k}(y_j^-|y_k)}.$$

6.3.1 Motivating examples in the discrete case

Let $(y_1, y_2, \ldots, y_N)$ be an integer-valued time series of length N. The joint probability expressed in terms of its copula C is

$$\Pr(Y_1 = y_1, \ldots, Y_N = y_N) = \sum_{\ell_1=0}^{1} \cdots \sum_{\ell_N=0}^{1} (-1)^{\ell_1+\cdots+\ell_N}\, C(F_1(y_1 - \ell_1), \ldots, F_N(y_N - \ell_N)), \tag{6.17}$$

where F_i is the discrete marginal distribution function of Y_i, $i = 1, \ldots, \textsc{n}$, (Section 3.6). The computation of the joint pmf requires $2^{\textsc{n}}$ $\textsc{n}$-dimensional copula evaluations. The $\textsc{n}$-dimensional Gaussian copula does not have a closed form and is defined by an $\textsc{n}$-dimensional integral requiring numerical methods for evaluation. We decompose the joint pmf (6.17) into many bivariate copulas to facilitate maximum likelihood estimation.

The case in two dimensions

Sklar's Theorem on two variables Y_1, Y_2 gives us

$$F(y_1, y_2) = C_{12}(F_1(y_1), F_2(y_2)).$$

Thus the joint probability is

$$\begin{aligned}
\Pr(Y_1 = y_1, Y_2 = y_2) &= F(y_1, y_2) - F(y_1^-, y_2) - F(y_1, y_2^-) + F(y_1^-, y_2^-), \\
&= C_{12}\left(F(y_1), F(y_2)\right) - C_{12}\left(F(y_1), F(y_2^-)\right) \\
&\qquad - C_{12}\left(F(y_1^-), F(y_2)\right) + C_{12}\left(F(y_1^-), F(y_2^-)\right), \\
&= C_{12}^{00} - C_{12}^{01} - C_{12}^{10} + C_{12}^{11}, \\
&= \frac{\nabla C_{12}}{\nabla F_1 \nabla F_2} \nabla F_1 \nabla F_2,
\end{aligned}$$

which is the analogue of (6.1).

The analogue to (6.3) is

$$\begin{aligned}
\Pr(Y_1 = y_1 \mid Y_2 = y_2) &= \frac{\Pr(Y_1 = y_1, Y_2 = y_2)}{\Pr(Y_2 = y_2)}, \\
&= \frac{\Big(C_{12}\left(F(y_1), F(y_2)\right) - C_{12}\left(F(y_1), F(y_2^-)\right) - C_{12}\left(F(y_1^-), F(y_2)\right) + C_{12}\left(F(y_1^-), F(y_2^-)\right)\Big)}{F(y_2) - F(y_2^-)}, \\
&= \frac{C_{12}^{00} - C_{12}^{01} - C_{12}^{10} + C_{12}^{11}}{F_2 - F_2^-}, \\
&= \frac{\nabla C_{12}}{\nabla F_1 \nabla F_2} \nabla F_1.
\end{aligned}$$

The case in three dimensions

Similarly, the joint probability of (Y_1, Y_2, Y_3) can be decomposed as

$$\begin{aligned}
&\Pr(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)\\
&= \Pr(Y_3 = y_3|Y_2 = y_2, Y_1 = y_1)\ \Pr(Y_2 = y_2|Y_1 = y_1)\ \Pr(Y_1 = y_1),\\
&= \frac{\Pr(Y_3 = y_3, Y_1 = y_1|Y_2 = y_2)}{\Pr(Y_1 = y_1|Y_2 = y_2)}\frac{\Pr(Y_2 = y_2, Y_1 = y_1)}{\Pr(Y_1 = y_1)}\Pr(Y_1 = y_1),\\
&= \frac{\Pr(Y_3 = y_3, Y_1 = y_1|Y_2 = y_2)}{\Pr(Y_1 = y_1, Y_2 = y_2)}\Pr(Y_2 = y_2)\Pr(Y_1 = y_1, Y_2 = y_2),\\
&= \Pr(Y_1 = y_1, Y_3 = y_3|Y_2 = y_2)\Pr(Y_2 = y_2),\\
&= (C^{00}_{13|2} - C^{01}_{13|2} - C^{10}_{13|2} + C^{11}_{13|2})\Pr(Y_2 = y_2),\\
&= \nabla C_{13|2}\nabla F_2, \qquad (6.18)
\end{aligned}$$

where $\nabla C_{13|2}$ has arguments $F_{1|2}$ and $F_{3|2}$ which are separately computed using (6.16) and the copulas C_{12} and C_{23}. This can be considered the discrete analogue to the continuous C-vine presented in the three dimensional case (6.5). Although (6.18) is not directly comparable, through the cancellation of terms, all of the pair-copulas of (6.5) are involved indirectly through the copula arguments and thus must be computed.

6.3.2 D-vine PCC for discrete data

As in the continuous case there are a number of ways to decompose the joint probability. In this section, we factorise the joint pmf (6.17) so that the pair-copula correlations are the partial autocorrelations in time series analysis (Section 6.4). The resulting vine PCC turns out to be a discrete D-vine PCC.

For $(y_1, \ldots, y_{N})$, a time series of length N, we order the indices by pairing the first and last time points and conditioning on the intermediate time points, subsequently moving inwards from both sides. This leads to the following factorisation into bivariate probabilities of the joint pmf,

$$\begin{aligned}&\Pr\left(Y_1 = y_1, \ldots, Y_{N} = y_{N}\right)\\&= \prod_{i=1}^{n} \Pr\left(Y_i = y_i, Y_{N-i+1} = y_{N-i+1} \mid Y_{i+1} = y_{i+1}, \ldots, Y_{N-i} = y_{N-i}\right),\end{aligned} \tag{6.19}$$

where $n = \frac{N}{2}$ if N is even, or $n = \frac{N+1}{2}$ if N is odd.

The random variables $Y_{i+1}, \ldots, Y_{N-i}$ are sandwiched between Y_i and Y_{N-i+1}. For example for $N = 3$,

$$\begin{aligned}&\Pr\left(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3\right)\\&= \Pr\left(Y_1 = y_1, Y_3 = y_3 \mid Y_2 = y_2\right)\ \Pr\left(Y_2 = y_2, Y_2 = y_2\right),\end{aligned}$$

and for $N = 4$,

$$\begin{aligned}&\Pr\left(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3, Y_4 = y_4\right)\\&= \Pr\left(Y_1 = y_1, Y_4 = y_4 \mid Y_2 = y_2, Y_3 = y_3\right)\ \Pr\left(Y_2 = y_2, Y_3 = y_3\right).\end{aligned}$$

The pair-copula decomposition is

$$\begin{aligned}&\Pr\left(Y_1 = y_1, Y_4 = y_4 \mid Y_2 = y_2, Y_3 = y_3\right)\ \Pr\left(Y_2 = y_2, Y_3 = y_3\right)\\&= \nabla C_{14|23}\left(F_{1|23}(y_1 \mid y_2, y_3), F_{4|23}(y_4 \mid y_2, y_3)\right)\ \nabla C_{23}\left(F_2(y_2), F_3(y_3)\right).\end{aligned}$$

This pairwise decomposition is recognised as a D-vine, in particular as the D-vine shown in Fig. 6.5. We can see the copula $C_{14|23}$ and C_{23} are explicit in the decomposition. All of the pair-copulas of the continuous D-vine are involved and hence must be computed. However only a subset of size n are expressed directly in the likelihood (6.21). Working hierarchically from tree $\mathcal{T}_1$, the terms C_{12}, C_{23} and C_{34} are required to compute the marginal terms $F_{1|2}, F_{3|2}, F_{2|3}$ and $F_{3|4}$. These terms are needed for the arguments for the pair-copulas in $\mathcal{T}_2$ and the copulas $C_{13|2}$ gives $F_{1|23}$

and $C_{24|3}$ gives $F_{4|23}$ which are the marginal cdfs of the highest order copula $C_{14|23}$. Equation (6.19) encompasses the example of three-dimensional given in Panagiotelis et al. (2012).

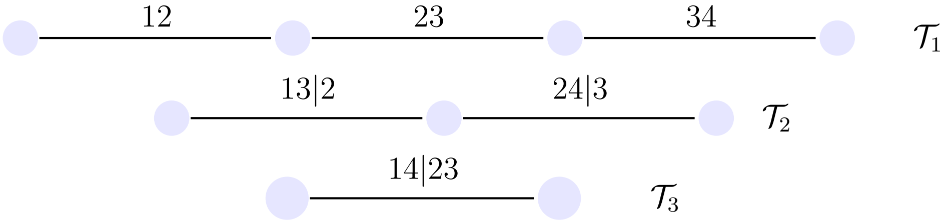


Figure 6.5: A D-vine with four variables.

6.3.3 The discrete D-vine log-likelihood

Let $\mathcal{V}$ be an N-dimensional D-vine on the variables $\mathbf{Y} = (Y_1, \dots Y_N)$. In this section we contribute to the literature by providing a general form for the pmf and the log-likelihood of a discrete D-vine To the best of our knowledge, a general formula has not been given in the literature. Let us define $n = \frac{N}{2}$ if N is even, or $n = \frac{N+1}{2}$ if N is odd, then the joint pmf corresponding to the factorisation (6.19), can be written as

$$\begin{aligned}
&\Pr(Y_1 = y_1, \dots, Y_N = y_N) \\
&= \prod_{i=1}^{n} \Pr\left(Y_i = y_i, Y_{N-i+1} = y_{N-i+1} \mid Y_{(i+1):(N-i)} = y_{(i+1):(N-i)}\right), \\
&= \prod_{i=1}^{n} \nabla C_{i,\,(N-i+1)|(i+1):(N-i-1)}, \\
&= \begin{cases} \nabla C_{1,N|2:(N-1)}\, \nabla C_{2,N-1|3:(N-2)}\, \nabla C_{3,N-2|4:(N-3)} \cdots, \nabla C_{n,n+1}, & \text{if } N \text{ is even,} \\ \nabla C_{1,N|2:(N-1)}\, \nabla C_{2,N-1|3:(N-2)}\, \nabla C_{3,N-2|4:(N-3)} \cdots, \nabla C_{n}, & \text{if } N \text{ is odd,} \end{cases}
\end{aligned} \tag{6.20}$$

where we write C_n for the univariate probability $\Pr(Y_n \leq y_n)$. For example with N=15,

$$\Pr(Y_1 = y_1, \dots, Y_N = y_N) = \nabla C_{1,\,15|2:14}\, \nabla C_{2,\,14|3:13} \cdots \nabla C_{7,\,9|8}\, \nabla C_8.$$

Therefore the general form of the log-likelihood for the discrete D-vine PCC with N margins is given by

$$\ell\left(\boldsymbol{\theta};\mathbf{y}\right)=\sum_{i=1}^{n}\log\nabla C_{i,\,N-i+1|i+1:N-i}. \tag{6.21}$$

For example with N=15 we have

$$\ell\left(\boldsymbol{\theta};\mathbf{y}\right)=\log\nabla C_{1,\,15|2:14}+\log\nabla C_{2,\,14|3:13}+\cdots+\log\nabla C_{7,\,9|8}+\log\nabla C_{8}.$$

The general form (6.21) is simple and has repetition for generalise to large N easily. The importance of this form for the time series context is that the correlations of the Gaussian conditional correlations are the partial autocorrelations of time series. Therefore the Levinson Durbin algorithm (Section 2.1.1) can be used to compute these efficiently, removing the need to compute the conditional distribution correlations. Therefore by choosing successive conditioning in this way we facilitate computationally efficient joint maximum likelihood. We modify the algorithm of (Dißmann et al., 2013) to compute (6.21) and also use (6.21) to obtain analytic standard errors of the parameter estimates.

6.4 The Gaussian copula model as a discrete D-vine PCC

Now we consider the details of the decomposition of the Gaussian copula pmf (3.8) into a combination of pair-copulas. In this section we justify the D-vine decomposition of the Gausian copula by showing that

1. the Gaussian copula model for discrete data can be represented as an R-vine construction where all the pair-copula are bivariate Gaussian copulas;
2. Gaussian pair-copulas parameters are identical to the partial correlations; *and*
3. these partial autocorrelations are the same as those from time series analysis.

Gaussian pair-copulas are the natural choice of family for the vine structure for the Gaussian copula model (3.8) because it is well known that conditional distributions of the multivariate Gaussian are themselves Gaussian. Write $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)^\top$ where $\mathbf{Y}_1$ is $N_1 \times 1$ and $\mathbf{Y}_2$ is $N_2 \times 1$ with $N = N_1 + N_2$. If $\mathbf{Y} \sim N_N(\boldsymbol{\mu}, \Sigma)$ with $\boldsymbol{\mu} = (\boldsymbol{\mu}_1^\top, \boldsymbol{\mu}_2^\top)^\top$ and $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$, then

$$\mathbf{Y}_2 | \mathbf{Y}_1 = \mathbf{y}_1 \sim N_{N_2}(\boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1), \Sigma_{22} - \Sigma_{21}\Sigma_{22}^{-1}\Sigma_{12}), \qquad (6.22)$$

where the conditional mean vector $\boldsymbol{\mu}_{2|1} = \boldsymbol{\mu}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1)$ is linear in $\mathbf{y}_1$. The conditioned variable $\mathbf{y}_1$ affects the mean of the conditional distribution but not its variance. The correlation matrix Σ is given by

$$\Sigma = \begin{pmatrix} 1 & r(1) & r(2) & \cdots & r(N-1) \\ r(1) & 1 & r(1) & \ddots & \vdots \\ r(2) & r(1) & 1 & \ddots & r(2) \\ \vdots & \ddots & \ddots & 1 & r(1) \\ r(N-1) & \cdots & r(2) & r(1) & 1 \end{pmatrix} \in \mathbb{R}^{N \times N},$$

where $r(i)$, $i = 1, \ldots, N$ is the autocorrelation at lag i, (Definition 4). For the Gaussian copula model (3.8), we allow all $N(N-1)/2$ pair-copulas of an R-vine to be bivariate Gaussian copulas (Joe, 1996). A pair-copula construction consists of $N(N-1)/2$ pair-copulas each with its own parameter set. A Gaussian pair-copula has only one correlation parameter resulting in a total of $N(N-1)/2$ pair-copula parameters in an all Gaussian R-vine structure.

Kurowicka and Cooke (2003) discuss a general correspondence between the parameters on a vine with all Gaussian copulas and the correlation matrix Σ of a multivariate Gaussian via the partial correlations. For the multivariate Gaussian distribution, the bivariate conditional distribution correlations coincide with the partial correlations (Baba et al., 2004). Partial correlations can be computed from the correlations $r(i)$, $i = 1, \ldots, N - 1$, using the formula (Yule and Kendall, 1965)

$$\rho_{i,j|k,l} = \frac{\rho_{ij|k} - \rho_{il|k}\rho_{jl|k}}{\sqrt{1-\rho_{il|k}^2}\sqrt{1-\rho_{jl|k}^2}}. \tag{6.23}$$

For time series the partial correlations are the partial autocorrelations which have been discussed in Chapter 2. The decomposition of the pmf (6.20), with the conditioning indices $i+1, \ldots, N-i$ sandwiched between i and $N - i + 1$, for $i = 1, \ldots, \lfloor (N+1)/2 \rfloor$, gives the partial correlations to be the partial autocorrelations, allowing efficient methods of time series to be used. The computationally efficient Levinson-Durbin algorithm given in (2.7) can then be used instead of using (6.23). As Daniels and Pourahmadi (2009) point out, the reparameterisation of the correlation matrices involving partial autocorrelations is not a recent result. They state 'the notion of PACF is known to be indispensable and' ...... 'be can be traced to a notable and somewhat neglected paper of Yule (1907)'. The one-to-one correspondence between Gaussian pair-copula correlations and the Gaussian correlation matrix is key to the D-vine representation of the Gaussian copula model.

6.4.1 Conditional distributions and conditional copulas

In a hierarchical manner we can build up the conditional distributions from those with a lower-dimensional conditioning set, as shown and described in Section 6.2.1, by

$$\begin{aligned} F_{i|j,k}\left(y_i|y_j, y_k\right) &= \frac{C_{i,j|k}\left(F\left(y_i|y_k\right), F\left(y_j|y_k\right)\right) - C_{i,j|k}\left(F\left(y_i|y_k\right), F(y_j^-|y_k)\right)}{\Pr\left(Y_j = y_j|Y_k = y_k\right)}, \\ &= \frac{C_{ij|k}^{00} - C_{ij|k}^{01}}{\nabla F_{j|k}} \end{aligned} \tag{6.24}$$

where k can be a scalar or multiple indices. By assuming the latent variables $\mathbf{X}$ (Section 3.3) follow a multivariate normal distribution, the dependence structure between the original variables is created. We borrow the dependence from the Gaussian copula for the discrete dependent random variables $\mathbf{Y}$. The justification of

(6.24) and the use of the Gaussian copula model for discrete random variables stems from using the conditional copula of the Gaussian ARMA(p, q) for the conditional copula of the discrete random variables $\mathbf{Y}$.

Consider the case N=3, we have Y_1, Y_2, Y_3 with distribution function given by $F_{123}(y_1, y_2, y_3)$ and the conditional distribution function of $Y_1, Y_3|Y_2 = y_2$ given by $F_{13|2}(y_1, y_3|y_2)$. By Sklar's Theorem, the appropriate bivariate copula of the distribution of $Y_1, Y_3|Y_2 = y_2$ is $C_{13|2}$. Let $C_{13|2}$ be the copula corresponding to $F_{13|2}$ according to Sklar's Theorem. Then

$$\begin{aligned}F_{13|2}(y_1, y_3|y_2) &= \Pr(Y_1 \le y_1, Y_3 \le y_3|Y_2 = y_2),\\ &= \frac{\Pr(Y_1 \le y_1, Y_3 \le y_3, Y_2 = y_2)}{\Pr(Y_2 = y_2)},\\ &= \frac{\Pr(Y_1 \le y_1, Y_2 \le y_2, Y_3 \le y_3) - \Pr(Y_1 \le y_1, Y_2 \le y_2 - 1, Y_3 \le y_3)}{\Pr(Y_2 \le y_2) - \Pr(Y_2 \le y_2 - 1)},\\ &= \frac{F_{123}(y_1, y_2, y_3) - F_{123}(y_1, y_2 - 1, y_3)}{F_2(y_2) - F_2(y_2 - 1)},\\ &= \frac{C(F_1(y_1), F_2(y_2), F_3(y_3)) - C(F_1(y_1), F_2(y_2 - 1), F_3(y_3))}{F_2(y_2) - F_2(y_2 - 1)}.\end{aligned}$$

Let $u_i = F_i(y_i)$ and $u_i^- = F_i(y_i - 1)$ for $i = 1, 2, 3$. Then

$$\begin{aligned}F_{13|2}(y_1, y_3|y_2) &= \frac{C(u_1, u_2, u_3) - C(u_1, u_2^-, u_3)}{u_2 - u_2^-},\\ &= \frac{1}{u_2 - u_2^-}\int_{u_2^-}^{u_2} \frac{\partial}{\partial u_2'} C(u_1, u_2', u_3)\, \mathrm{d}u_2'.\end{aligned}$$

The partial derivative of a copula with respect to one of its arguments is equal to the conditional distribution function given the random variable that the argument corresponds to (Section 6.2.1). Patton (2006) extends Sklar's Theorem to conditional distributions.

$$\frac{\partial}{\partial u_2} C(u_1, u_2, u_3) = C_{13|2}(u_1, u_3|u_2) = C_{13|2}(C_{1|2}(u_1|u_2), C_{3|2}(u_3|u_2)).$$

Therefore

$$F_{13|2}(y_1, y_3|y_2) = \frac{1}{u_2 - u_2^-} \int_{u_2^-}^{u_2} C_{13|2}\left(C_{1|2}(u_1|u_2'), C_{3|2}(u_3|u_2')\right) \, \mathrm{d}u_2'.$$

The conditional distribution function $C_{13|2}(C_{1|2}, C_{3|2})$ can be written as a pair-copula with arguments $C_{1|2}$ and $C_{3|2}$ (Patton, 2006) such that $C_{1|2} = \frac{\partial}{\partial u_2} C(u_1, u_2, 1)$ and $C_{3|2} = \frac{\partial}{\partial u_2} C(1, u_2, u_3)$ since copulas are distribution functions by definition. Thus the copula $C(u_1, u_2, u_3)$ is a cdf on $[0,1]^3$ and $C(u_1, u_2, 1) = C_{12}(u_1, u_2)$, $C(1, u_2, u_3) = C_{32}(u_3, u_2)$ and $\frac{\partial}{\partial u_2} C(u_1, u_2, 1) = C_{1|2}(u_1, u_2)$, $\frac{\partial}{\partial u_2} C(1, u_2, u_3) = C_{3|2}(u_3, u_2)$.

$$F_{13|2}(y_1, y_3|y_2) = \frac{1}{u_2 - u_2^-} \int_{u_2^-}^{u_2} C_{13|2}\Big(\frac{\partial}{\partial u_2'} C(u_1, u_2', 1), \frac{\partial}{\partial u_2'} C(1, u_2', u_3)\Big) \, \mathrm{d}u_2'.$$

The copula C is uniquely defined at $u_i = F_i(0), F_i(1), \ldots,$. We can linearly interpolate the values of C (Genest and Nešlehová, 2007) so that it is piecewise linear over $[0,1]^3$. Then C is a constant and

$$\begin{aligned}
&F_{13|2}(y_1, y_3|y_2) \\
=&C_{13|2}\Big(\frac{C(u_1, u_2, 1) - C(u_1, u_2^-, 1)}{u_2 - u_2^-}, \frac{C(1, u_2, u_3) - C(1, u_2^-, u_3)}{u_2 - u_2^-}\Big), \\
=&C_{13|2}\Big(\frac{F_{12}(y_1, y_2) - F_{12}(y_1, y_2 - 1)}{F_2(y_2) - F_2(y_2 - 1)}, \frac{F_{23}(y_2, y_3) - F_{23}(y_2 - 1, y_3)}{F_2(y_2) - F_2(y_2 - 1)}\Big), \\
=&C_{13|2}\Big(\frac{\Pr(Y_1 \le y_1, Y_2 = y_2)}{\Pr(Y_2 = y_2)}, \frac{\Pr(Y_2 = y_2, Y_3 \le y_3)}{\Pr(Y_2 = y_2)}\Big), \\
=&C_{13|2}\Big(F_{1|2}(y_1|y_2), F_{3|2}(y_3|y_2)\Big).
\end{aligned}$$

Therefore the conditional copula is the conditional cdf for the discrete distribution.

6.4.2 Equivalence of the vine copula and Gaussian copula model

In this section we compare the evaluation of the joint probabilities computed via the discrete D-vine PCC and the Gaussian copula model directly.

We numerically verify the equivalence of the Gaussian copula model and the discrete D-vine copula model in dimensions N=4. For N=4,

$$\Pr(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3, Y_4 = y_4) =$$
$$\sum_{\ell_1=0}^{1}\sum_{\ell_1=2=0}^{1}\sum_{\ell_3=0}^{1}\sum_{\ell_4=0}^{1}(-1)^{\ell_1+\ell_2+\ell_3+\ell_4}\Phi_\Sigma(\Phi^{-1}(F_1(y_1)), \Phi^{-1}(F_2(y_2)), \Phi^{-1}(F_3(y_3)), \Phi^{-1}(F_4(y_4))),$$

and

$$\Pr(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3, Y_4 = y_4) =$$
$$\nabla C_{14|23}(F_{1|23}(y_1|y_2, y_3), F_{4|23}(y_4|y_2, y_3))\, \nabla C_{23}(F_2(y_2), F_3(y_3)),$$

where $\nabla C = C^{00} - C^{01} - C^{10} + C^{11}$.

Direct evaluation of the Gaussian copula model probability mass function is computationally expensive and one evaluation for a time series of length N=20 takes longer than six days. However it can be evaluated directly for low-dimensions and hence it can be used to compare the accuracy of the D-vine PCC. The N-dimensional Gaussian copula is a cumulative distribution function which does not have a closed form. The evaluation of multivariate Gaussian probability require numerical methods due to the multivariate integrals. In low dimensions numerical integration techniques can be used, but as the dimension increases other numerical approximation methods must be used. Tong (1990, pg 186) gives an overview of some numerical methods that have been proposed such as Monte Carlo simulations (Abbe 1964; Deak 1978; Moran 1984), dimensional reductions in numerical integration (Steck 1962; Bacon 1963), change-of-variables, infinite series expansions and quadrature methods to name a few. For the direct evaluation, the multivariate Gaussian cdf is required. We use a built-in function of `R` for evaluating the multivariate Gaussian cdf in high-dimensions, namely `pmnorm()` in *mnormt* (Azzalini and Genz, 2015) which works via a non Monte Carlo method using the `sadvm` function (Genz, 1992).

The evaluation of the discrete D-vine PCC using Algorithm 5 involves bivariate Gaussian copulas which require the bivariate Gaussian distribution function. These can be computed using the `R` function `pbivnorm()` in the *pbivnorm* package (Genz, 2015). This is Fortran code written by Alan Genz, translated into `R` code by Brenton Kenkel, based on the *mnormt* package. Alternatively the `pCopula()` function in the *copula* package (Hofert et al., 2015) can be used.

Numercal Verification

The Gaussian copula model with negative binomial margins ($\pi = 0.3, s = 3$) and ARMA$(1, 1)$ dependence ($a_1 = 0.6, b_1 = 0.8$) is fit to 160 000 datasets of length N=4, in both ways; directly using the `pmnorm()` and via the D-vine PCC. The datasets of length N=4 consist of all the combinations of the integers from one to twenty, e.g. (1, 7, 3, 8), (2, 4, 8, 1), (12, 3, 1, 19) e.t.c. There are $20^4 = 160\,000$ datasets in total.

Fig. 6.6 shows the pmf of both methods plotted against each other. Perfect agreement corresponds to all points lying on the line $y = x$ and we can there is very good agreement between the methods for the case of N=4.

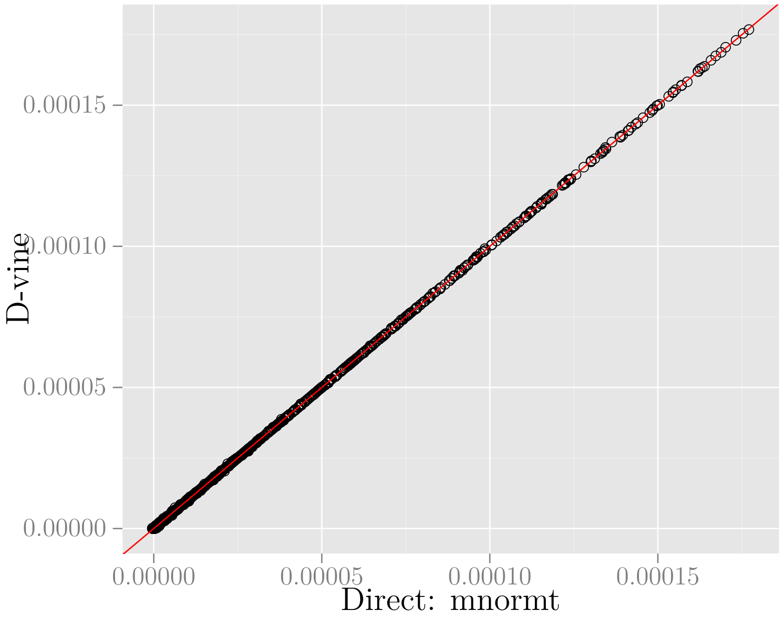


Figure 6.6: Comparing D-vine PCC and direct Gaussian copula pmf evaluations on datasets of length N=4.

6.4.3 Discussion

We make use of the vine methodology and computational techniques and adapt them for an efficient algorithm for maximum likelihood estimation of the Gaussian copula model (3.8).

It happens that the vine methodology corresponding to the Gaussian copula model which is built with Gaussian pair-copulas is not a complex vine and has many simplifications over other vine models. Firstly, the bivariate Gaussian copula has only one parameter and has established efficient computation procedures. Secondly, inference for many vine models require a simplifying assumption (Haff et al., 2013) being that the the pair-copula corresponding to the conditional cumulative distribution function $F_{ij|k}$ of $Y_i, Y_j|Y_k$ is unaffected by the value of the conditioning variable Y_k. That is

$$C_{ij|k}(F_{i|k}(y_i|y_k), F_{i|k}(y_i|y_k); y_k) = C_{ij|k}(F_{i|k}(y_i|y_k), F_{i|k}(y_i|y_k)).$$

From (6.22) we can see that the conditional Gaussian distribution depends on the conditioned variable only through its mean value. Hence the Gaussian copula is a 'copula of the simplified form' (Stöber, 2013) and therefore the condition is satisfied. Finally, vines with all Gaussian pair-copulas have been viewed as simplified vines and they have been compared with more complex vine structures to investigate whether the results of the simplified vine varies significantly with more complex models (Valdesgo, 2009; Brechmann, 2010).

The evaluation of the discrete D-vine PCC representation (6.21) of the Gaussian copula pmf with N discrete margins involves at most $2N(N-1)$ bivariate Gaussian distribution functions which is much simpler to evaluate than the 2^N finite differences of N-dimesional Gaussian copula required for the Gaussian copula pmf (3.8). In the next section we provide an algorithm to evaluate the discrete D-vine PCC log-likelihood discussed in Chapter 6 and present analytic forms of the score and the Hessian of the log-likelihood for maximum likelihood estimation.

6.5 Evaluation of the discrete D-vine log-likelihood

Panagiotelis et al. (2012) give an algorithm for computing a discrete D-vine density. The algorithm is specific to the graphical tree structure of the D-vine where the conditional copula distributions corresponding to each edge are evaluated by beginning at tree one, working downwards to finish at tree $N-1$. Dißmann et al. (2013) provides a more general approach for evaluating any R-vine continuous density which was described in Section 6.2.3 and has led to analytic forms of the score and Hessian (Stöber and Schepsmeier, 2013) in the continuous case.

We combine the ideas of the discrete D-vine of Panagiotelis et al. with the general and efficient computation of Dißmann et al. (2013, Algorithm 2.1, p. 58) as does Stöber (2013) to evaluate a discrete D-vine log-likelihood in Algorithm 4. We also provide the novel details of an analytic form of the score and Hessian in the discrete case which has not been done before in the literature. [1] Panagiotelis et al. suggest the bootstrap method to compute the standard errors in the discrete case.

To evaluate the discrete D-vine, let the matrices in (6.14) be reformed to $v^{\text{direct}} = (v^{\text{direct}+}, v^{\text{direct}-})$ and $v^{\text{indirect}} = (v^{\text{indirect}+}, v^{\text{indirect}-})$ where v_i^+ is evaluated at y_i and v_i^- is evaluated at $y_i - 1$. Introduce lower triangular $N \times N$ matrices, f^{direct} and f^{indirect} where

$$f_k = \Pr(Y_k = y_k) = F_k(y_k) - F_k(y_k - 1) = F_k - F_k^-, \tag{6.25}$$

where k can be multiple indices or a single index. Continuing with the example of a D-vine on 4 elements (6.13), the form of $v^{\text{direct}+}$, $v^{\text{direct}-}$, f^{direct} and f^{indirect} are

$$v^{\text{direct}+} = \begin{pmatrix} F_{4|321} & 0 & 0 & 0 \\ F_{4|23} & F_{3|21} & 0 & 0 \\ F_{4|3} & F_{3|2} & F_{2|1} & 0 \\ F_4 & F_3 & F_2 & F_1 \end{pmatrix}, \quad v^{\text{direct}-} = \begin{pmatrix} F_{4|321}^- & 0 & 0 & 0 \\ F_{4|23}^- & F_{3|21}^- & 0 & 0 \\ F_{4|3}^- & F_{3|2}^- & F_{2|1}^- & 0 \\ F_4^- & F_3^- & F_2^- & F_1^- \end{pmatrix},$$

$$f^{\text{direct}} = \begin{pmatrix} f_{4|321} & 0 & 0 & 0 \\ f_{4|23} & f_{3|21} & 0 & 0 \\ f_{4|3} & f_{3|2} & f_{2|1} & 0 \\ f_4 & f_3 & f_2 & f_1 \end{pmatrix}, \quad f^{\text{indirect}} = \begin{pmatrix} f_{1|234} & 0 & 0 & 0 \\ f_{2|34} & f_{1|23} & 0 & 0 \\ f_{3|4} & f_{2|3} & f_{1|2} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{6.26}$$

Therefore the log-likelihood (6.21) can be alternatively computed using the diagonals of the matrix f^{direct} as

$$\ell(\boldsymbol{\theta}; \mathbf{y}) = \sum_{j=1}^{N} \log f_{j,j}^{\text{direct}}.$$

[1] At the time of writing, Harry Joe released an excellent resource (Joe, 2015). The book surveys a significant proportion of the copula literature which will no doubt become core reading for the advancement of copula modelling. Section 6.12 includes a discussion on the derivatives of the the discrete R-vine density of which encompasses much of this work.

The matrix P^* stores the pair-copula parameters which coincide with the partial correlations for the Gaussian distribution. For a D-vine the partial correlations are the partial autocorrelations (Section 6.4). For a stationary time series, the partial autocorrelations ρ_{kk} depends on the time points only through its lag k, therefore all the pair-copulas have the same correlations when assigned in the same tree as illustrated in Fig. 6.7.

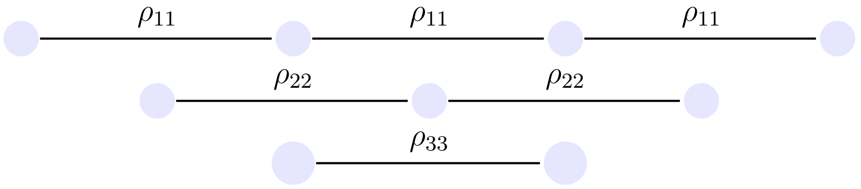


Figure 6.7: The conditional distribution correlation parameters of a D-vine for a stationary time series in four dimensions.

There are $N-1$ trees and hence there are at most $N-1$ distinct values. The matrix P^* for a D-vine for time series data is

$$P^* = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \rho_{N-1\,N-1} & 0 & 0 & 0 \\ \vdots & \ddots & 0 & 0 \\ \rho_{11} & \rho_{11} & \rho_{11} & 0 \end{pmatrix}, \tag{6.27}$$

where ρ_{kk} are the partial autocorrelations at lag k, $k = 1, \ldots, N-1$. The Levinson-Durbin algorithm provides a computationally efficient method of computing the partial autocorrelations by solving the Yule-Walker equations recursively (Chapter 2).

The algorithm to compute the log-likelihood function of a discrete D-vine (6.21) is given in Algorithm 5. It takes the same form as the continuous R-vine given in Algorithm 4 in that it consists of two loops which iterate over the columns of (6.26) with index k and row index i. Each loop has the same three main steps;

1. Select the copula arguments	$F^+_{i\|k}, F^-_{i\|k}, F^+_{j\|k}, F^-_{j\|k}$ as z_1^+, z_1^- and z_2^+, z_2^- (lines 8-10)
2. Evaluate the pair-copula at each pair of arguments	$C^{++}_{ij\|k}, C^{+-}_{ij\|k}, C^{-+}_{ij\|k}, C^{--}_{ij\|k}$ (lines 11)
3. Compute the conditional distribution and densities	$F^+_{i\|jk}, F^-_{i\|jk}, F^+_{j\|ik}, F^-_{j\|ik}$ $f^+_{i\|jk}, f^-_{i\|jk}, f^+_{j\|ik}, f^-_{j\|ik}$ (lines 12-13)

These are looped over for $i = \text{N}, \ldots, k+1$ and $k = \text{N} - 1, \ldots, 1$ and then the likelihood returned (line 17). The order of these steps are shown in Table 6.1. The difference between Dißmann et al. (2013) and Panagiotelis et al. (2012)'s evaluation of the algorithm is simply the order in which the pair-copulas are applied to the arguments. The algorithm provided in Panagiotelis et al. (2012) corresponds to the order $C_{12}, C_{23}, C_{34}, C_{13|2}, C_{24|3}$ then $C_{14|23}$ rather than as shown in Table 6.1 and therefore would correspond to iterating over rows then columns, rather than columns then rows and the indices are fixed as a special case with no R-vine matrices, M and $\widetilde{M}$.

Table 6.1: The order of copula evaluations in for a D-vine in N=4 dimension for Algorithm 5.

Loop		Copula	Entries of	
$(i,$	$k)$	$C_{jk\|l}$	v^{direct}	v^{indirect}
4	3	C_{12}	$F_{2\|1}$	$F_{1\|2}$
4	2	C_{23}	$F_{3\|2}$	$F_{2\|3}$
3	2	$C_{13\|2}$	$F_{3\|21}$	$F_{1\|23}$
4	1	C_{34}	$F_{4\|3}$	$F_{3\|4}$
3	1	$C_{24\|3}$	$F_{4\|32}$	$F_{2\|34}$
2	1	$C_{14\|23}$	$F_{4\|321}$	$F_{1\|234}$

The differences between the continuous R-vine (Section 6.2.3)and the discrete given above are in the details of these steps

a) Each step requires an extra evaluation at $y_i - 1$;

b) The copula distribution function rather than the copula density is applied (line 11);

c) The conditional distributions are updated using (6.16) rather than (6.9) (line 12);

d) the corresponding conditional densities are computed $f_k = F_k - F_k^-$ (line 13);

e) the log-likelihood has a different expression in the final step: If the copula is directly expressed in the likelihood then we save the value as $c_j = \nabla C_j$. The algorithm returns then returns a combination of c_j as in (6.21) or alternatively $\ell(\boldsymbol{\theta}; \mathbf{y}) = \sum\limits_{j=1}^{N} \log f_{j,j}^{\text{direct}}$.

For our case of all Gaussian pair-copula, line 11 becomes

$$14: \qquad (C^{00}, C^{01}, C^{10}, C^{11}) = \Phi_{p_{i,k}} \left(\Phi^{-1} \left(z_1^a \right), \Phi^{-1}(z_2^b) \right),$$

where $\Phi_{p_{i,k}}$ is a bivariate Gaussian distribution function with correlation $p_{i,k}$ and Φ^{-1} is the inverse of the univariate standard normal cdf.

Maximum likelihood estimation is an optimisation procedure and good optimisation methods should be adhered to.

Algorithm 5 Log-likelihood for the D-vine density for discrete variables.

Input: Model parameters $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, b_q, \boldsymbol{\theta}_{\text{marginal}})$ and an integer-valued time series $(y_1, \ldots, y_N)$ of length N.

Output: Log-likelihood of a discrete R-vine Gaussian PCC at $\boldsymbol{\theta}$.

Required: Roots of $\alpha(z) = 1 - a_1 z - \cdots - a_p z^p$ and $\beta(z) = 1 + b_1 z + \cdots + b_q z^q$ to lie outside the unit circle.

```
Create $N \times N$ matrices $v^{\text{direct}+}, v^{\text{direct}-}, v^{\text{indirect}+}, v^{\text{indirect}-}, f^{\text{direct}}, f^{\text{indirect}}$ as in (6.26).
Set $M$ and $\widetilde{M}$ $N \times N$ matrices as in (6.13).
$\rho_{kk}$ = Levinson-Durbin$(\mathbf{a}, \mathbf{b})$ into $P^*$ matrix as in (6.27).   ▷ Copula parameters

Set $(u_1^+, \ldots, u_N^+) = (F(y_1), \ldots, F(y_N))$,   ▷ Compute marginals
        and $(u_1^-, \ldots, u_N^-) = (F(y_1 - 1), \ldots, F(y_N - 1))$.
Store $(v_{N,1}^{\text{direct}+}, \ldots, v_{N,N}^{\text{direct}+}) = (u_N^+, \ldots, u_1^+)$,
        and $(v_{N,1}^{\text{direct}-}, \ldots, v_{N,N}^{\text{direct}-}) = (u_N^-, \ldots, u_1^-)$,
        and $(f_{N,1}^{\text{direct}-}, \ldots, f_{N,N}^{\text{direct}-}) = (u_N^+, \ldots, u_1^+) - (u_N^-, \ldots, u_1^-)$ as (6.25).

for $k = N - 1, \ldots 1$
    for $i = N, \ldots k + 1$

        $(z_1^+, z_1^-) = (v_{i,k}^{\text{direct}+}, v_{i,k}^{\text{direct}-})$.   ▷ Select copula arguments
        if $\widetilde{m}_{i,k} = m_{i,k}$ then $(z_2^+, z_2^-) = (v_{k,(N+1-\widetilde{m}_{i,k})}^{\text{direct}+}, v_{k,(N+1-\widetilde{m}_{i,k})}^{\text{direct}-})$,
        else $(z_2^+, z_2^-) = (v_{k,(N+1-\widetilde{m}_{i,k})}^{\text{indirect}+}, v_{k,(N+1-\widetilde{m}_{i,k})}^{\text{indirect}-})$.

        $(C^{00}, C^{01}, C^{10}, C^{11}) =$   ▷ Evaluate copulas
                $C\left(z_1^a, z_2^b; t_{i,k}, p_{i,k}\right)$ where $ab = \{+, -\}$.

                                                ▷ Conditional distributions
        $v_{i-1,k}^{\text{direct}+-} = \dfrac{C(z_1, z_2; p_{i,k}) - C(z_1, z_2^-; p_{i,k})}{f(z_2)}$ and
        $v_{i-1,k}^{\text{indirect}+-} = \dfrac{C(z_1, z_2; p_{i,k}) - C(z_1^-, z_2; p_{i,k})}{f(z_1)}$ as in (6.16).
        Set $f_{i-1,k}^{\text{direct}} = v_{i-1,k}^{\text{direct}+} - v_{i-1,k}^{\text{direct}-}$,
        and $f_{i-1,k}^{\text{indirect}} = v_{i-1,k}^{\text{indirect}+} - v_{i-1,k}^{\text{indirect}-}$.

        If $C$ expressed in log-likelihood
                then save $c_j = \nabla C = C^{00} - C^{01} - C^{10} + C^{11}$.
    end
end

Return $\ell = \sum_{j=1}^{\lfloor (N+1)/2 \rfloor} c_j$ as (6.21)   or $\ell = \sum_{j=1}^{N} \log f_{j,j}^{\text{direct}}$ as (6.25).
```

6.5.1 Model selection using AIC and BIC

In classical time series the order of a suitable ARMA(p, q) model is identified by looking at the acf and pacf plots or by some criteria, such as the Akaike's and Bayesian Information Criteria. We follow the latter approach to select an appropriate ARMA(p, q) dependence structure in the Gaussian copula model.

Let $\widehat{\ell} = \ell(\widehat{\boldsymbol{\theta}}; \mathbf{y})$ be the maximum likelihood value evaluated at the MLE $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ and let $k = \dim(\boldsymbol{\theta})$ which in this case is $p + q$. The AIC and BIC are given in terms of $\widehat{\ell}$ and k .

Definition 13. ***Akaike Information Criterion*** *(AIC, Akaike 1974)*
The Akaike Information criterion *is*

$$AIC\ (p, q) = -2\,\widehat{\ell} + 2k.$$

Definition 14. ***Bayesian Information Criterion*** *(BIC)*
The Bayesian Information criterion *is*

$$BIC\ (p, q) = -2\,\widehat{\ell} + k \log \textsc{n}.$$

Both of these take the form of a penalised maximised likelihood but with different penalties. The BIC typically selects a more parsimonious model.

6.5.2 Assumption of the underlying ARMA process

Masarotto et al. (2012) consider the Gaussian copula model as a marginal regression model with serially correlated errors. They check the model adequacy by residual analysis. We use this idea to check the assumption of the underlying ARMA process.

If the marginals of the Gaussian copula model are continuous, the model adequacy can be checked by confirming that

$$r_i = \Phi^{-1}\left(F_i(Y_i \mid Y_{i-1}, Y_{i-2}, \ldots, Y_1; \widehat{\boldsymbol{\theta}})\right) \quad i = 1, \ldots, \textsc{n},$$

behave like realisations of N uncorrelated standard normal variables. This is because in the continuous case Rosenblatt (1952) showed that

$$M_i = F_i(Y_i \mid Y_{i-1}, Y_{i-2}, \ldots, Y_1; \boldsymbol{\theta}),$$

are uniformly and independently distributed in the unit interval.

In the discrete case, Dunn and Smyth (1996) introduced *randomised quantile residuals*, for discrete dependent y_i where $m_i = F(y_i \mid y_{i-1}, \ldots, y_1 \;\; \widehat{\boldsymbol{\theta}})$ and $m_i^- = F(y_i^- \mid y_{i-1}, \ldots, y_1 \;\; \widehat{\boldsymbol{\theta}})$ where $\widehat{\boldsymbol{\theta}}$ is the MLE of $\boldsymbol{\theta}$. These are given by

$$r_i^* = \Phi^{-1}\left(m_i^- + u_i(m_i - m_i^-)\right),$$

where the u_i is a draw from a $(0, 1)$ uniform variate. If the model assumptions are satisfied, then the values r_i^* for $i = 1, \ldots,$ N are N independent standard normal variables and normality of the r_i's can be checked using the usual residual checks for example acf and qq-plots.

6.6 The analytic score and Hessian

The Hessian of the log-likelihood is required to study the covariance structure of maximum likelihood estimators, specifically in computation of the standard errors. An analytic form allows estimation of parameter uncertainty and may avoid numerical issues which arise from numerical differentiation, whilst the score function can be used to speed up the optimisation of ML estimation.

Panagiotelis et al. (2012) suggested to use a bootstrap method to obtain standard errors for the discrete D-vine PCC. Stöber and Schepsmeier (2013) provided algorithms for computing the first and second derivatives of the continuous R-vine log-likelihood. The possible direct and indirect dependences through the arguments (the conditional cdfs) of the copula terms on the pair-copula parameters are discussed in their appendix and they are with respect to the pair-copula parameters only. At the time of writing, there is no publicly available software for computing the score or Hessian for a discrete D-vine or more general R-vine in the literature[2].

In this section we present general form of the score and Hessian of a discrete D-vine PCC and we present the algorithms to compute the score function and the Hessian of a discrete D-vine PCC. We extend the methodology to include the marginal derivatives as well as the cross derivatives. This is a novel extension of Schepsmeier and Stöber (2014) to include the marginal model parameters, yet also a special case with all Gaussian pair-copulas and an ARMA(p, q) dependence structure. We provide pseudo-code in Algorithms 6 and 7 given in Appendix B.

The simplification of the vine having only $p+q+\dim(\boldsymbol{\theta}_{\text{marginal}})$ parameters leads to all of the conditional copulas and conditional arguments in the likelihood function to be directly or indirectly functions of these $p+q+\dim(\boldsymbol{\theta}_{\text{marginal}})$ parameters. This

[2] At the time of writing, Joe (2015)[Section 6.12] discusses the derivatives for factor copula models

is in contrast to the $N(N-1)/2$ partial derivatives of an R-vine Schepsmeier and Stöber (2014).

The log-likelihood (6.21) is

$$\ell(\boldsymbol{\theta};\mathbf{y}) = \sum_{k=1}^{n} \log \nabla C_{k,\ N-k+1|k+1:N-k},$$

where $\nabla C = C^{00} - C^{01} - C^{10} + C^{11}$ (Section 6.3). We will use partial notation for the partial derivatives but will use Lagrangian notation when necessary for simplification. The score of the log-likelihood of the discrete D-vine is given by

$$\begin{aligned}
\ell'_{\theta_i}(\boldsymbol{\theta};\mathbf{y}) &= \sum_{i=1}^{\lfloor (N+1)/2 \rfloor} \frac{\nabla C'_{k,\ N-k+1|k+1:N-k}}{\nabla C_{k,\ N-k+1|k+1:N-k}}, \\
&= \begin{cases} \dfrac{\nabla C'_{1N|2:N-1,\ \theta_i}}{\nabla C_{1N|2:N-1}} + \cdots + \dfrac{\nabla C'_{n,n+1,\ \theta_i}}{\nabla C_{n,n+1}} & \text{if } N = 2n \quad \text{i.e. } N \text{ is even,} \\ \dfrac{\nabla C'_{1N|2:N-1,\ \theta_i}}{\nabla C_{1N|2:N-1}} + \cdots + \dfrac{\nabla C'_{n,\ \theta_i}}{\nabla C_n} & \text{if } N = 2n-1 \quad \text{i.e. } N \text{ is odd.} \end{cases} \\
&= \begin{cases} \sum_{k=1}^{n} \dfrac{\nabla C'_{k\ N-k+1|k+1:N-k-1,\ \theta_i}}{\nabla C_{k\ N-k+1|k+1:N-k-1}} & \text{if } N = 2n \quad \text{i.e. } N \text{ is even,} \\ \sum_{k=1}^{n} \dfrac{\nabla C'_{k\ N-k+1|k+1:N-k-1,\ \theta_i}}{\nabla C_{k\ N-k+1|k+1:N-k-1}} + \dfrac{f'_{\theta_i}(y_n)}{f(y_n)} & \text{if } N = 2n-1 \quad \text{i.e. } N \text{ is odd.} \end{cases}
\end{aligned} \tag{6.28}$$

where $f = F^{+} - F^{-} = F(y) - F(y-1)$. For example for N=5,

$$\ell'_{\theta_i}(\boldsymbol{\theta};\mathbf{y}) = \frac{\nabla C'_{15|234;\theta_i}}{\nabla C_{15|234}} + \frac{\nabla C'_{24|3;\theta_i}}{\nabla C_{24|3}} + \frac{f'_{3,\theta_i}(y_3)}{f_3(y_3)}.$$

For notational simplicity, denote the copula indices by $l = \{k,\ N-k+1 \mid k+1 : N-k\}$ for $k = 1, \ldots, n$.

The Hessian of the log-likelihood is

$$\ell''_{\theta_i\theta_j}(\boldsymbol{\theta};\mathbf{y}) = \begin{cases} \sum\limits_{l=1}^{n} \dfrac{\nabla C''_{l,\,\theta_i\theta_j}}{\nabla C_l} - \dfrac{\nabla C'_{l,\,\theta_i}\nabla C'_{l,\,\theta_j}}{\nabla C_l^2} & \text{if } N \text{ is even,} \\ \\ \sum\limits_{l=1}^{n} \dfrac{\nabla C''_{l,\,\theta_i\theta_j}}{\nabla C_l} - \dfrac{\nabla C'_{l,\,\theta_i}\nabla C'_{l,\,\theta_j}}{\nabla C_l^2} & \\ \quad + \dfrac{f''_{n,\theta_i\theta_j}(y_n)}{f_n(y_n)} - \dfrac{f'_{n,\theta_i}(y_n)\,f'_{n,\theta_j}(y_n)}{f_n(y_n)^2} & \text{if } N \text{ is odd.} \end{cases} \tag{6.29}$$

For example with N=3

$$\ell''_{\theta_i\theta_j} = \frac{\nabla C''_{13|2,\,\theta_i\theta_j}}{\nabla C_{13|2}} - \frac{\nabla C'_{13|2,\,\theta_i}\nabla C'_{13|2,\,\theta_j}}{\nabla C^2_{13|2}} + \frac{f''_{2,\theta_i\theta_j}(y_2)}{f_2(y_2)} - \frac{f'_{2,\theta_i}(y_2)\,f'_{2,\theta_j}(y_2)}{f_2(y_2)^2}.$$

The term ∇C involves the copula cdf evaluated at four pairs of points. The partial derivatives of the copula cdfs C^{ℓ_1,ℓ_2} are required for the calculations of the gradient (6.28) and the Hessian (6.29) of (6.21) with respect to the marginal and ARMA(p,q) dependence parameters,

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_{\text{copula}}, \boldsymbol{\theta}_{\text{marginal}}) = (a_1,\ldots,a_p,b_1,\ldots,b_q,\boldsymbol{\theta}_{\text{marginal}}).$$

These can be analytically found using the chain rule. The partial derivatives of the copula distribution functions with respect to $\boldsymbol{\theta}$ are

$$\frac{\partial C}{\partial \theta_i} = \frac{\partial C}{\partial \rho}\cdot\frac{\partial \rho}{\partial \theta_i} + \frac{\partial C}{\partial u_1}\cdot\frac{\partial u_1}{\partial \theta_i} + \frac{\partial C}{\partial u_2}\cdot\frac{\partial u_2}{\partial \theta_i},$$

and

$$\begin{aligned}\frac{\partial^2 C}{\partial\theta_i\partial\theta_j} = \frac{\partial}{\partial\theta_j}\left(\frac{\partial C}{\partial\rho}\right)\cdot\frac{\partial\rho}{\partial\theta_i} + \frac{\partial C}{\partial\rho}\cdot\frac{\partial^2\rho}{\partial\theta_i\partial\theta_j} + \frac{\partial}{\partial\theta_j}\left(\frac{\partial C}{\partial u_1}\right)\cdot\frac{\partial u_1}{\partial\theta_i} \\ + \frac{\partial C}{\partial u_1}\cdot\frac{\partial^2 u_1}{\partial\theta_i\partial\theta_j} + \frac{\partial}{\partial\theta_j}\left(\frac{\partial C}{\partial u_2}\right)\cdot\frac{\partial u_2}{\partial\theta_i} + \frac{\partial C}{\partial u_2}\cdot\frac{\partial^2 u_2}{\partial\theta_i\partial\theta_j}.\end{aligned} \tag{6.30}$$

We compute the derivatives of the conditional distribution functions, $u = F_{j|kl}$, by differentiation of (6.16), that is $F_{j|kl} = (C^{00}_{jk|l} - C^{01}_{jk|l})/f_{k|l}$, using the quotient rule resulting in

$$\frac{\partial u}{\partial \theta_i} = \frac{\partial F_{j|kl}}{\partial \theta_i} = F'_{j|kl,\theta_i} = \frac{\left(C^{00\,\prime}_{jk|l,\theta_i} - C^{01\,\prime}_{jk|l,\theta_i}\right) \cdot f_{k|l} - \left(C^{00}_{jk|l} - C^{01}_{jk|l}\right) \cdot f'_{k|l,\theta_i}}{f^2_{k|l}}, \tag{6.31}$$

where C'_{θ_i} and f'_{θ_i} is Langrangian notation for the derivatives wrt θ_i.

The second derivative is given by applying the quotient rule a second time as

$$\begin{aligned} \frac{\partial^2 u}{\partial \theta_i \partial \theta_j} &= F''_{j|kl,\theta_i\theta_j} \\ &= \frac{\begin{gathered}\left(C^{00\,\prime\prime}_{jk|l,\theta_i\theta_j} - C^{01\,\prime\prime}_{jk|l,\theta_i\theta_j}\right) \cdot f_{k|l} + \left(C^{00\,\prime}_{jk|l,\theta_i} - C^{01\,\prime}_{jk|l,\theta_i}\right) \cdot f'_{k,\theta_j} \\ - \left(C^{00\,\prime}_{jk|l\theta_j} - C^{01\,\prime}_{jk|l,\theta_j}\right) \cdot f'_{k|l,\theta_i} + \left(C^{00}_{jk|l} - C^{01}_{jk|l}\right) \cdot f''_{k|l,\theta_i\theta_j}\end{gathered}}{f^4_{k|l}}. \end{aligned} \tag{6.32}$$

The derivatives f'_{θ_i} and $f''_{\theta_i\theta_j}$ are simple to compute due to the simple form of $f_{jk|l} = F_{jk|l} - F^-_{jk|l}$.

When the marginals are assumed to be negative binomial NB (π, s) and the pair-copulas are Gaussian copulas $C(u_1, u_2; \rho)$ with correlations that are the ARMA(p,q) partial autocorrelations, the following are required,

$$\frac{\partial C}{\partial u_1}, \frac{\partial C}{\partial u_2}, \quad \frac{\partial C}{\partial \rho}, \frac{\partial u}{\partial a_i}, \frac{\partial u}{\partial b_j}, \quad \frac{\partial u}{\partial \pi}, \frac{\partial u}{\partial s}, \frac{\partial \rho}{\partial a_i}, \frac{\partial \rho}{\partial b_j},$$

where a_i and b_j are the i^{th} autoregressive and j^{th} moving average parameters respectively, $0 \le i \le p$ or $0 \le j \le q$. Note that $\dfrac{\partial \rho}{\partial \pi} = \dfrac{\partial \rho}{\partial s} = 0$.

For the second and cross partial derivatives, the terms required are

$$\begin{gathered} \frac{\partial^2 C}{\partial {u_1}^2}, \frac{\partial^2 C}{\partial u_1 \partial u_2}, \frac{\partial^2 C}{\partial {u_2}^2}, \quad \frac{\partial c}{\partial u_1}, \frac{\partial c}{\partial u_2}, \frac{\partial c}{\partial \rho}, \\ \frac{\partial^2 u}{\partial {a_i}^2}, \frac{\partial^2 u}{\partial a_i \partial b_j}, \frac{\partial^2 u}{\partial a_i \partial \pi}, \frac{\partial^2 u}{\partial a_i \partial s}, \frac{\partial^2 u}{\partial {b_j}^2}, \frac{\partial^2 u}{\partial b_j \partial \pi}, \frac{\partial^2 u}{\partial b_j \partial s}, \frac{\partial^2 u}{\partial \pi^2}, \frac{\partial^2 u}{\partial \pi \partial s}, \frac{\partial^2 u}{\partial s^2}, \\ \frac{\partial^2 \rho}{\partial {a_i}^2}, \frac{\partial^2 \rho}{\partial a_i \partial b_j} \text{ and } \frac{\partial^2 \rho}{\partial {b_j}^2}, \end{gathered}$$

where C and c are the Gaussian pair-copula distribution (6.10) and density (3.6) functions respectively. We provide the details in Sections 6.6.1 through 6.6.3. The algorithms to compute the score (6.28) and Hessian (6.29) using these terms are given in Section 6.6.4.

6.6.1 The derivatives of the bivariate Gaussian copula cdf

We compute the first and second derivatives of the bivariate Gaussian copula distribution function wrt $\boldsymbol{\theta}$. These include the bivariate Gaussian cdf $\Phi(x, y; \rho)$ wrt its arguments, x and y and correlation parameter ρ.

The partial derivative of the Gaussian copula C with respect to its first and second arguments respectively (Aas et al., 2009),

$$\begin{aligned} \frac{\partial \Phi}{\partial u_1}\left(u_1, u_2; \rho\right) &= \Phi\left(\frac{\Phi^{-1}\left(u_2\right) - \rho\Phi^{-1}\left(u_1\right)}{\sqrt{1-\rho^2}}\right), \\ \frac{\partial \Phi}{\partial u_2}\left(u_1, u_2; \rho\right) &= \Phi\left(\frac{\Phi^{-1}\left(u_1\right) - \rho\Phi^{-1}\left(u_2\right)}{\sqrt{1-\rho^2}}\right). \end{aligned}$$

Plackett (1954) states that the derivative of the bivariate Gaussian cdf wrt its correlation parameter is equivalent to the mixed partial derivative of its arguments,

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\rho}\left(x, y; \rho\right) = \frac{\partial^2 \Phi}{\partial x \partial y}\left(x, y; \rho\right).$$

Therefore the derivative of the bivariate Gaussian copula cdf wrt its correlation parameter is the Gaussian copula density e.g.,

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\rho}\left(x, y; \rho\right) = \frac{\partial^2 \Phi}{\partial x \partial y}\left(x, y; \rho\right) = \phi\left(x, y; \rho\right),$$

and let $x = \Phi^{-1}\left(u_1\right)$ and $y = \Phi^{-1}\left(u_2\right)$.

When C is the bivariate Gaussian copula with correlation parameter ρ, the terms in (6.30) are

$$\begin{aligned} \frac{\partial}{\partial \theta_j}\left(\frac{\partial C}{\partial \rho}\right) &= \frac{\partial}{\partial \theta_j}\phi\left(\Phi^{-1}\left(u_1\right), \Phi^{-1}\left(u_2\right); \rho\right), \\ \frac{\partial}{\partial \theta_j}\left(\frac{\partial C}{\partial u_1}\right) &= \frac{\partial}{\partial \theta_j}\Phi\left(\frac{\Phi^{-1}\left(u_2\right) - \rho\Phi^{-1}\left(u_1\right)}{\sqrt{1-\rho^2}}\right), \\ \frac{\partial}{\partial \theta_j}\left(\frac{\partial C}{\partial u_2}\right) &= \frac{\partial}{\partial \theta_j}\Phi\left(\frac{\Phi^{-1}\left(u_1\right) - \rho\Phi^{-1}\left(u_2\right)}{\sqrt{1-\rho^2}}\right). \end{aligned}$$

A key result for the differentiation is

$$\frac{\partial \Phi^{-1}\left(u\right)}{\partial u} = \frac{1}{\phi\left(\Phi^{-1}\left(u\right)\right)}.$$

We present the second derivatives of the bivariate Gaussian distribution:

$$\begin{aligned}
\Phi''_{\rho^2}\left(x,y;\rho\right) &= \phi'_{\rho}\left(x,y;\rho\right), \\
&= -\frac{\left(\rho x^2-\left(\rho^2+1\right)xy+\rho\left(\rho^2+y^2-1\right)\right)}{2\pi\left(1-\rho^2\right)^{5/2}}\exp\left\{\frac{x^2-2\rho xy+y^2}{-2\left(1-\rho^2\right)}\right\}, \\
\Phi''_{\rho x}\left(x,y;\rho\right) &= \phi'_{x}\left(x,y;\rho\right)=\phi\left(\frac{y-\rho\, x}{\sqrt{1-\rho^2}}\right)\cdot\frac{y\rho-x}{1-\rho^2}, \\
\Phi''_{\rho y}\left(x,y;\rho\right) &= \phi'_{y}\left(x,y;\rho\right)=\phi\left(\frac{x-\rho\, y}{\sqrt{1-\rho^2}}\right)\cdot\frac{x\rho-y}{1-\rho^2}, \\
\Phi''_{x^2}\left(x,y;\rho\right) &= \phi\left(\frac{y-\rho\, x}{\sqrt{1-\rho^2}}\right)\cdot\frac{-\rho}{\sqrt{1-\rho^2}}, \\
\Phi''_{y^2}\left(x,y;\rho\right) &= \phi\left(\frac{x-\rho y}{\sqrt{1-\rho^2}}\right)\cdot\frac{-\rho}{\sqrt{1-\rho^2}}, \\
\Phi''_{xy}\left(x,y;\rho\right) &= \phi\left(x,y;\rho\right). \qquad (6.33)
\end{aligned}$$

6.6.2 The derivatives of the partial autocorrelations

The partial autocorrelations, ρ_{kk}, satisfy the Yule-Walker equations given in (2.6). In this section, we directly differentiate the partial autocorrelations from the Yule-Walker equations and compute them using an intuitive way of computing the inverse of the leading sub-matrices of the autocorrelation matrix.

However, it has since been acknowledged that the derivatives of the partial autocorrelations (and autocorrelations) with respect to ARMA parameters were derived in the signal processing literature in the 1980's and which also implement these steps more efficiently (Porat (1983, p. 346);Friedlander and Porat (1984);Kohn and Ansley (1985);Porat and Friedlander (1986);Stoica and Moses (1997)).

The derivatives of ρ_{kk} wrt $\boldsymbol{\theta}_{\text{copula}}$ can be found by solving the differentiated Yule-Walker equations for $\dfrac{\partial \rho_{kk}}{\partial \theta_i}$ given by

$$\underbrace{\begin{bmatrix} 1 & r(1) & r(2) & \cdots & r(k-1) \\ r(1) & 1 & r(1) & & r(k-2) \\ r(2) & r(1) & 1 & \ddots & \vdots \\ \vdots & & \vdots & \vdots & r(1) \\ r(k-1) & r(k-2) & \cdots & r(1) & 1 \end{bmatrix}}_{:=R_{k\times k}} \begin{bmatrix} \dfrac{\partial \rho_{k1}}{\partial \theta_i} \\ \dfrac{\partial \rho_{k2}}{\partial \theta_i} \\ \vdots \\ \vdots \\ \dfrac{\partial \rho_{kk}}{\partial \theta_i} \end{bmatrix}$$

$$= \begin{bmatrix} 1-\rho_{k2} & -\rho_{k3} & \cdots & \cdots & -\rho_{kk} & 0 \\ -\rho_{k1}-\rho_{k3} & 1-\rho_{k4} & \cdots & -\rho_{kk} & 0 & 0 \\ -\rho_{k2}-\rho_{k4} & -\rho_{k1}-\rho_{k5} & \iddots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & 0 & 0 & 0 \\ -\rho_{k,k-1} & -\rho_{k,k-2} & -\rho_{k,k-3} & \cdots & -\rho_{k1} & 1 \end{bmatrix} \begin{bmatrix} \dfrac{\partial r(1)}{\partial \theta_i} \\ \dfrac{\partial r(2)}{\partial \theta_i} \\ \vdots \\ \vdots \\ \dfrac{\partial r(k)}{\partial \theta_i} \end{bmatrix}. \tag{6.34}$$

where $r(k)$, $k = 1, 2, \ldots, N-1$ are the autocorrelations of the ARMA(p, q) model which are functions of $\boldsymbol{\theta}_{\text{copula}}$. To solve (6.34) for $\dfrac{\partial \rho_{kk}}{\partial \theta_i}$ we need to compute $R^{-1}_{k\times k}$ for $k = 1, 2, \ldots, N-1$. The Levinson Durbin algorithm (see 2.7) has been used in an earlier step of the algorithm to compute the partial autocorrelations for the pair-copula parameters. The Levinson Durbin algorithm computes the terms $\rho_{i,k}$, $i = 1, \ldots, k$ and $k = 1, \ldots, N$ in the triangular Ψ matrix and the τ_k^2 terms in a diagonal Γ matrix and so Σ^{-1} can be considered as a by-product,

$$R^{-1}_{N\times N} = \Sigma^{-1} = \Phi^{\top}\Gamma^{-1}\Phi.$$

However, this only gives $R^{-1}_{k\times k}$ for $k = N$. We can sequentially compute the inverse of the leading $(k-1)\times(k-1)$ sub-matrices (6.34) $k = N-1, \ldots, 2$ from $R^{-1}_{k\times k}$ by the following result.

Consider the inverse of the $N \times N$ matrix to be written in the form

$$\begin{pmatrix} R_{k\times k} & b \\ c^\top & d \end{pmatrix}^{-1} = \begin{pmatrix} E & f \\ g^\top & h \end{pmatrix},$$

where b and c are vectors of length $k-1$ and d is a scalar. Then the inverse of the leading sub-matrix $R_{k\times k}$ with dimension reduced by one is

$$R^{-1}_{k\times k} = E - \frac{fg^\top}{h}.$$

We apply this result sequentially for $k = N-1, \ldots, 1$.

The derivatives with respect to the marginal parameters $\boldsymbol{\theta}_{\text{marginal}}$ are zero since the partial autocorrelations are functions of the ARMA(p,q) parameters only. For the remainder of this chapter we write $\boldsymbol{\theta}$ instead of $\boldsymbol{\theta}_{\text{copula}}$ or $\boldsymbol{\theta}_{\text{marginal}}$.

Now we obtain the derivatives of the autocorrelations, $\dfrac{\partial r(k)}{\partial \theta_i}$ for $k = 1, \ldots, N-1$, wrt the ARMA(p,q) parameters which can be done by solving the differentiated system of equations of (2.5) for $\dfrac{\partial r(k)}{\partial \theta_i}$. The system of equations (2.5) are simplified in our model because the Gaussian copula restricts the covariance matrix to have unit diagonals so $r(0) = 1$.

For simplicity of notation, an example with ARMA($p = 5$,q) is shown below but can be easily extended for general p,

$$\underbrace{\begin{bmatrix} 1-a_2 & -a_3 & -a_4 & -a_5 & 0 \\ -a_1-a_3 & 1-a_4 & -a_5 & 0 & 0 \\ -a_2-a_4 & -a_1-a_5 & 1 & 0 & 0 \\ -a_3-a_5 & -a_2 & -a_1 & 1 & 0 \\ -a_4 & -a_3 & -a_2 & -a_1 & 1 \end{bmatrix}}_{:=\mathbf{C}} \begin{bmatrix} \frac{\partial r(1)}{\partial \theta_i} \\ \frac{\partial r(2)}{\partial \theta_i} \\ \frac{\partial r(3)}{\partial \theta_i} \\ \frac{\partial r(4)}{\partial \theta_i} \\ \frac{\partial r(5)}{\partial \theta_i} \end{bmatrix} = \frac{\partial \mathbf{C}}{\partial \theta_i} +$$

$$+ \frac{\partial \sigma_\varepsilon^2}{\partial \theta_i} \cdot \begin{bmatrix} b_1 + b_2\psi_1 + b_3\psi_2 + \cdots + b_q\psi_{q-1} \\ b_2 + b_3\psi_1 + b_4\psi_2 + \cdots + b_q\psi_{q-2} \\ \vdots \\ b_p + b_{p+1}\psi_1 + b_{p+2}\psi_2 + \ldots\ldots \end{bmatrix} + \sigma_\varepsilon^2 \cdot \frac{\partial}{\partial \theta_i} \left(\begin{bmatrix} b_1 + b_2\psi_1 + b_3\psi_2 + \cdots + b_q\psi_{q-1} \\ b_2 + b_3\psi_1 + b_4\psi_2 + \cdots + b_q\psi_{q-2} \\ \vdots \\ b_p + b_{p+1}\psi_1 + b_{p+2}\psi_2 + \ldots\ldots \end{bmatrix} \right), \tag{6.35}$$

where $\frac{\partial \mathbf{C}}{\partial \theta_i}$ is the derivative of the matrix $\mathbf{C}$ element-wise wrt θ_i and is a vector of length p. The final two matrices in (6.35) become zero for the AR(p) model, simplifying the computation greatly. Computationally, we exploit the form of C in (6.35) and (similarly in 6.34) by considering matrix C as the sum of two triangular matrices and and identity matrix.

The Gaussian copula restricts the covariance matrix Σ to be a correlation matrix, i.e. have unit diagonals, that is $r(0)$ is one and $r(i) \in [-1, 1]$. Hence the ARMA(p, q) dependence structure is such that the covariance $R(0)$ is restricted to be one. Therefore the term σ_ε^2 is not a constant and the term σ_ε^2 is a function of the ARMA parameters $\mathbf{a}$ and $\mathbf{b}$ and so the derivative with respect to $\boldsymbol{\theta}$ must be accounted for.

For example, if the dependence structure is specified as an AR(2) process then

$$R(0) = \frac{1-a_2}{1+a_2} \frac{\sigma_\varepsilon^2}{(1-a_2)^2 - a_1^2}.$$

and setting $R(0) = 1$ and rearranging for σ_ε^2 we have

$$\sigma_\varepsilon^2(\boldsymbol{\theta}) = \frac{1+a_2}{1-a_2}((1-a_2)^2 - a_1^2).$$

For a MA(q) dependence structure

$$\sigma_\varepsilon^2(\boldsymbol{\theta}) = \frac{1}{1+b_1^2+\cdots+b_q^2},$$

and the derivatives are given by

$$\begin{cases} \dfrac{\partial}{\partial b_i}\sigma_\varepsilon^2(\boldsymbol{\theta}) = \dfrac{-2b_i}{\left(1+b_1^2+\cdots+b_q^2\right)^2}, \\ \dfrac{\partial^2}{\partial {b_i}^2}\sigma_\varepsilon^2(\boldsymbol{\theta}) = \dfrac{8b_i^2 - 2\left(1+b_1^2+\cdots+b_q^2\right)}{\left(1+b_1^2+\cdots+b_q^2\right)^3}, \\ \dfrac{\partial^2}{\partial b_i \partial b_j}\sigma_\varepsilon^2(\boldsymbol{\theta}) = \dfrac{8b_ib_j}{\left(1+b_1^2+\cdots+b_q^2\right)^3}. \end{cases}$$

Similarly for an ARMA(1,1) dependence structure,

$$\sigma_\varepsilon^2(\boldsymbol{\theta}) = \frac{1-a_1^2}{1+b_1^2+2a_1b_1}$$

and the derivatives are given by

$$\begin{cases} \dfrac{\partial}{\partial a_1}\sigma_\varepsilon^2(\boldsymbol{\theta}) = -\dfrac{2\left(a_1+b_1\right)\left(a_1b_1+1\right)}{\left(2a_1b_1+b_1^2+1\right)^2}, \\ \dfrac{\partial}{\partial b_1}\sigma_\varepsilon^2(\boldsymbol{\theta}) = \dfrac{2(a_1^2-1)(a_1+b_1)}{\left(2a_1b_1+b_1^2+1\right)^2}, \\ \dfrac{\partial^2}{\partial {a_1}^2}\sigma_\varepsilon^2(\boldsymbol{\theta}) = -\dfrac{2\left(b_1^2-1\right)^2}{\left(2a_1b_1+b_1^2+1\right)^3}, \\ \dfrac{\partial^2}{\partial {b_1}^2}\sigma_\varepsilon^2(\boldsymbol{\theta}) = -\dfrac{2\left(a_1^2-1\right)\left(6a_1b_1+4a_1^2+3b_1^2-1\right)}{\left(2a_1b_1+b_1^2+1\right)^3}, \\ \dfrac{\partial^2}{\partial a_1 \partial b_1}\sigma_\varepsilon^2(\boldsymbol{\theta}) = \dfrac{4a_1^3b_1+6a_1^2\left(b_1^2+1\right)+4a_1b_1\left(b_1^2+2\right)+6b_1^2-2}{\left(2a_1b_1+b_1^2+1\right)^3}. \end{cases}$$

Note that the derivatives for AR(p) are not required in the score and Hessian of the discrete D-vine PCC. The form of the derivative of $\sigma_\varepsilon^2(\boldsymbol{\theta})$ quickly becomes complicated for ARMA(p,q) when $p \neq 0, q > 2$. See Appendix A for the example of the partial derivatives for ARMA(2,2). The second derivatives are found by solving the second derivatives of the system of equations in (6.35).

6.6.3 The derivatives of the negative binomial distribution functions

Any arbitrary discrete marginal can be assumed (Section 3.3.3) such as Poisson, zero-inflated Poisson, zero-inflated negative binomial and can include covariates. The negative binomial is one of the most frequently and widely applied discrete distributions along with the Poisson distribution (Johnson and Kotz, 1969). For illustration we consider the case of the negative binomial margins with parameters π, s for analytic differentiation. The partial derivatives of the univariate negative binomial distribution function $F(y; \pi, s)$ wrt π and s are sought. For the unconditional univariate distributions, the first and second and any mixed partial derivatives wrt the ARMA(p, q) parameters are zero. The negative binomial distribution function can be expressed in terms of an incomplete beta function ratio which is the commonly recognised form given by (Rider, 1962),

$$F_Y(k; \pi, s) = \Pr(Y \leq k) = \sum_{y=-\infty}^{k} \Pr(Y = k) = \frac{B(\pi; s, 1+k)}{B(s, 1+k)}, \tag{6.36}$$

where $B(\alpha, \beta)$, $B(z; \alpha, \beta)$ are the complete and incomplete beta functions respectively.

We differentiate the negative binomial distribution function $F(k; \pi, s)$ with parameters π and s to give the first partial derivatives as

$$\begin{aligned}
\frac{\partial}{\partial \pi} F(k) &= \frac{\pi^{s-1}(1-\pi)^k}{B(s, 1+k)}, \\
\frac{\partial}{\partial s} F(k) &= \frac{B(\pi; s, 1+k)}{B(s, 1+k)} \left(\log \pi - \frac{\Gamma'(s)}{\Gamma(s)} + \frac{\Gamma'(1+s+k)}{\Gamma(1+s+k)} \right) \\
&\quad - \pi^s \frac{\Gamma(s)^2}{B(s, 1+k)} \frac{1}{\Gamma(s+1)^2} \, {}_3F_2 \left[\begin{matrix} s, s, -k \\ s+1, s+1 \end{matrix} \right],
\end{aligned} \tag{6.37}$$

where ${}_3F_2\left[\begin{smallmatrix} a_1, a_2, a_3 \\ b_1, b_2 \end{smallmatrix}\right]$ is a generalised hypergeometric function (Abramowitz and Stegun, 1964) and $\Gamma(x)$ is the gamma function.

The second and mixed partial derivatives are

$$\begin{aligned}
\frac{\partial^2}{\partial\pi^2}F(k) &= \frac{\pi^{s-2}(1-\pi)^{k-1}}{B(s,k+1)}((1-\pi)(s-1)-\pi\ k),\\
\frac{\partial^2}{\partial\pi\partial s}F(k) &= \frac{\pi^{s-1}(1-\pi)^{k}}{B(s,k+1)}(\psi(s+k+1)+\log(\pi)-\psi(s)),\\
\frac{\partial^2}{\partial s^2}F(k) &= \frac{B_p(s,k+1)}{B(s,k+1)} \quad\times\quad \left(-\psi^1(s)+\psi^1(1+s+k)\right)\\
&\quad + \left(\log\pi-\psi^0(s)+\psi^0(1+s+k)\right) \quad\times\quad F_s'(k)\\
&\quad - \pi^s\frac{\Gamma(s)^2}{B(s,1+k)}\frac{1}{\Gamma(s+1)^2} \quad\times\quad \partial s\left({}_3F_2\left[\begin{matrix}s,s,-k\\ s+1,s+1\end{matrix}\right]\right)\\
&\quad + {}_3F_2\left[\begin{matrix}s,s,-k\\ s+1,s+1\end{matrix}\right] \quad\times\quad \frac{-\pi^s\Gamma(s)^2}{B(s,k+1)}\left(\log\pi-\psi^0(s)+\psi^0(1+s+k)\right),
\end{aligned}\tag{6.38}$$

where the digamma function $\psi(x)$ is defined by $\psi(x) = \Gamma'(x)/\Gamma(x)$ and $\psi^n(x)$ is the n^{th} derivative of the digamma function.

The second derivative wrt s can also be expressed as

$$\begin{aligned}
\frac{\partial^2 s}{\partial F^2}(k) = \frac{B_p(s,k+1)}{B(s,k+1)}&\left((\psi(s)-\psi(s+k+1))\cdot(\gamma-H_{s+k}-2\log(\pi)+\psi(s))\right.\\
&\left.+\psi^1(s+k+1)+\log^2(\pi)-\psi^1(s)\right)\\
&-\pi^s\Gamma(s)^2\left(2(\psi^0(s+k+1)+\log(\pi))\,{}_3\widetilde{F}_2(s,s,-k;s+1,s+1;\pi)\right.\\
&+\partial_{(\{0,0,0\},\{0,1\},0)}\ {}_3F_2\left(\{s,s,-k\},\{s+1,s+1\},\pi\right)+\partial_{(\{0,0,0\},\{1,0\},0)}\ {}_3F_2\left(\{s,s,-k\},\{s+1,s+1\},\pi\right)\\
&\left.+\partial_{(\{0,1,0\},\{0,0\},0)}\ {}_3F_2\left(\{s,s,-k\},\{s+1,s+1\},\pi\right)+\partial_{(\{1,0,0\},\{0,0\},0)}\ {}_3F_2\left(\{s,s,-k\},\{s+1,s+1\},\pi\right)\right),
\end{aligned}$$

where H_n is the n^{th} harmonic number such that $H_n := \sum_{k=1}^{n}\frac{1}{k}$, and γ is the Euler-Mascheroni constant, $\gamma \approx 0.57721566\ldots$. The $\partial_{(\{i,0,0\},\{0,0\},0)}$ notation is used to give the argument of the function the derivative is with respect to.

6.6.4 Evaluation of the score function and Hessian

The algorithm to compute the score function of the log-likelihood (6.21) is given in Algorithm 6, and the algorithm to the compute the Hessian function is given in Algorithm 7 both in Appendix B. Both these algorithms extend the algorithm to compute the log-likelihood function of a discrete D-vine (6.21). In this section we provide commentary for the two algorithms to explain how they differ.

The main steps of the likelihood evaluation are:

- select the arguments;
- evaluate the copula functions, *and*
- compute the conditional distributions.

At each of these steps, the computation of the derivatives are added into the algorithm. The details of these extra steps have been discussed in Sections 6.6.1, 6.6.2 and 6.6.3. The extra steps are summarised in Table 6.2.

Let T be the number of parameters in the log-likelihood so $T = p + q + \dim(\boldsymbol{\theta}_{\text{marginal}})$. Define an array of dimension $(2, T)$ of $N \times N$ matrices for $S1^{\text{direct}}$ and $S1^{\text{indirect}}$ to store the derivatives relating to the matrices $v^{\text{direct}\,+}$,$v^{\text{direct}\,-}$, $v^{\text{indirect}\,+}$ and $^{\text{indirect}\,-}$.

First, we compute the copula arguments and marginal cdfs. We now also compute the derivative of the marginal cdfs. The algorithm consists of two loops which iterate over the columns of (6.26) with index k and row index i. Note: The loops are indexed as (i, k) to select the matrix positions. The notation θ_i has been used to denote the i^{th} element of $\boldsymbol{\theta}$; and these are not the same i.

Benchmark Study

The availability of an analytic gradient evaluation allows numerical maximisation of the log-likelihood to be based on the analytic gradient, rather than finite-difference

Table 6.2: A comparison of the steps for the three computational algorithms.

Task	Line of algorithm		
	Log-likelihood	Score	Hessian
Compute copula arguments	3	3-4	3-5
Compute marginals	4-5	5-10	6-11
Begin loops			
Select the copula arguments $F^+_{i\|k}, F^-_{i\|k}, F^+_{j\|k}, F^-_{j\|k}$ as z_1^+, z_1^- and z_2^+, z_2^-	8-10	13-16	13-19
Evaluate the pair-copulas $C^{++}_{ij\|k}, C^{+-}_{ij\|k}, C^{-+}_{ij\|k}, C^{--}_{ij\|k}$	11	17-24	22-32
Compute the conditional distributions $F^+_{i\|jk}, F^-_{i\|jk}, F^+_{j\|ik}, F^-_{j\|ik}$	12	26-29	34-37
and densities $f^+_{i\|jk}, f^-_{i\|jk}, f^+_{j\|ik}, f^-_{j\|ik}$	13	30-31	38-42
End loops			
Calculate log-likelihood	17	34	43

approximations of the gradient. This decreases the computation time and becomes increasingly important when the N is large. These algorithms allow the Gaussian copula model to be applied to large discrete-valued datasets, which is considered infeasible due to the vast computation of the evaluation and hence the optimisation of the likelihood of the Gaussian copula model (4.1).

A benchmark study is carried out in order to compare the decrease in computation time using the analytic score function from numerical approximation by finite-differencing (Newtons quotient). Both the numerical and analytical score functions are computed for 100 simulated datasets of length N=100. Fig. 6.8 shows violin plots of the resulting times to see the distribution of computational time of each method. As expected, we can see that there is a clear improvement when gradient is computed analytically. There is no overlap in the results and the speed up is approximately by a factor of 4. There is a smaller range of computation time for the analytic evaluation and overall a clear advantage to the analytic method. We also carried out a sensitivity analysis and repeated the numerical differentiation using the R package `numDeriv` using the `grad` functions (Gilbert and Varadhan, 2015) which uses the Richardson extrapolation approach (Fornberg and Sloan, 1994) and the results are similar.

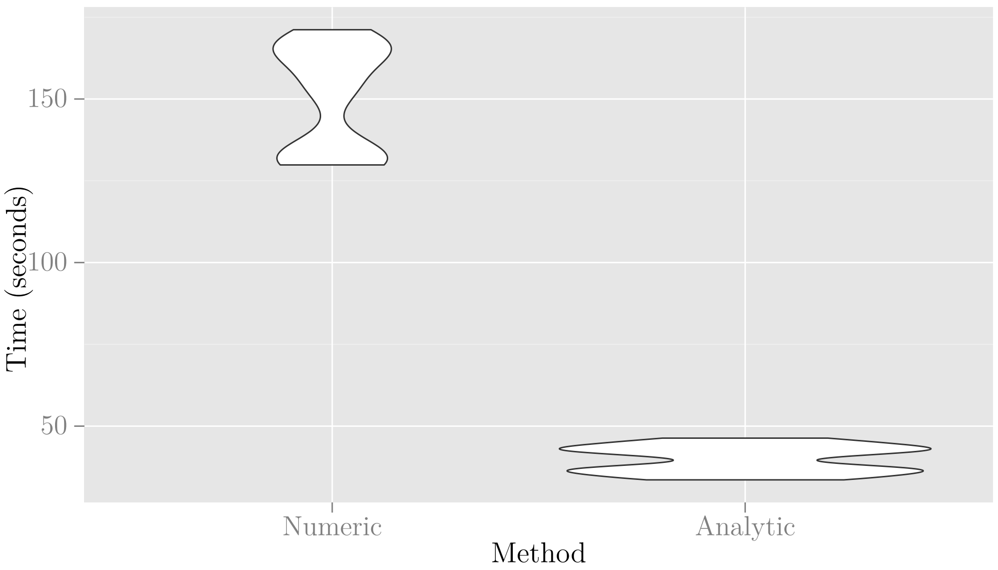


Figure 6.8: A comparison study of the time to compute the score function of the discrete D-vine log-likelihood numerically and analytically for 100 simulated integer-valued time series datasets of length $N = 100$.

A second study is carried out to investigate the time requirement of the algorithms as the length of the time series increases. This was carried out by repeating the first simulation study, i.e. recording the time taken for 100 evaluations, on data sets of differing lengths $N = 10, 20, \ldots, 160$. We record the computational time for the evaluations of both the score function and Hessian. The numerical Hessian calculations are carried out using the `hessian` function in the `numDeriv` package which uses the Richardson extrapolation approach (Fornberg and Sloan, 1994). We could potentially speed up the numerical approximations by taking advantage of the known zeros in the Hessian matrix. However, the `hessian` function is commonly used in R and this is beyond the scope of this exercise. Fig. 6.9 shows a significant decrease in time required for the analytic evaluation of the Hessian over the numerical evaluation. We can see the difference in time between both numerical and analytical methods increases non linearly as N increases.

Figure 6.9: Average time in seconds to compute the Hessian of the discrete D-vine log-likelihood numerically and analytically for 100 simulated integer-valued time series of length $N = 10, 20, \ldots, 160$.

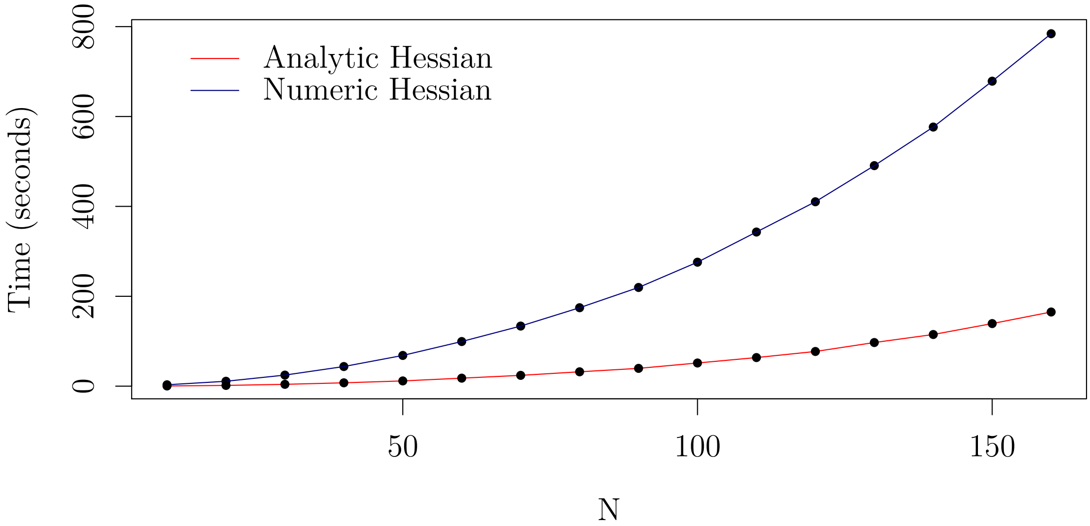


Fig. 6.10 plots the log transform of the length N against log time and we obtain a straight line. This suggests we have a power law of the form $T = \mathcal{O}(N^b)$, specified by $T = aN^b +$ lower order terms

$$T = -3.596\, N^{2.007} \qquad \text{(analytic Hessian)},$$

$$T = -5.896\, N^{2.146} \qquad \text{(numeric Hessian)},$$

which are approximately N^2 laws. It would be of interest to obtain an analytic result.

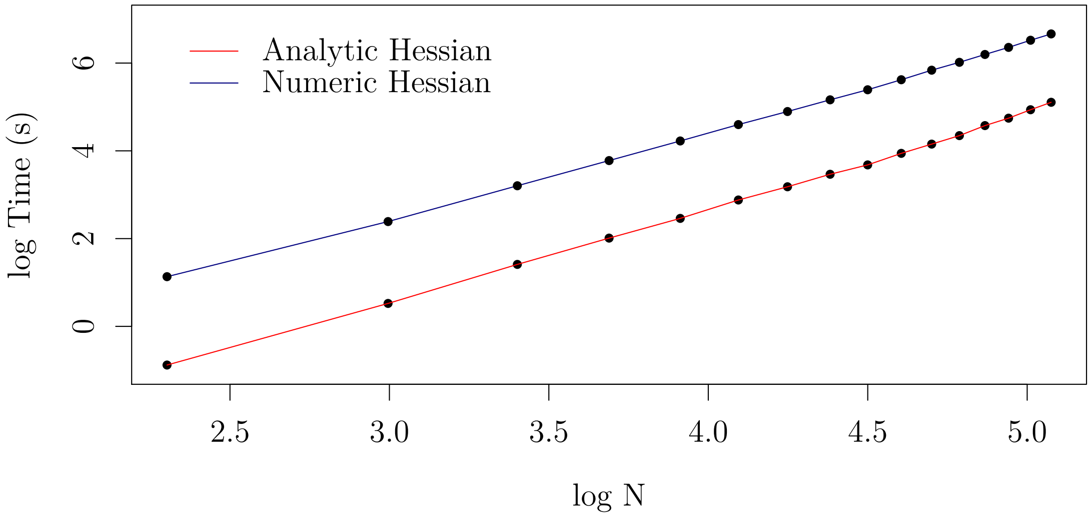


Figure 6.10: Numerical analysis for the time complexity of the analytic and numerical Hessian evaluations.

6.7 Simulations under model conditions

Two simulation studies are presented to demonstrate the methodology and estimation methods in situations where the process is accurately known before undertaking a real world data set in Chapter 7. The first study aims to illustrate that the theory works and this is demonstrated on 6 time series with underlying AR(1) and ARMA(1,1) structures for lengths N=100, N=250 and N=500. We investigate the sensitivity of the estimates and compare the method to the best possible estimates if the underlying time series is known. The final simulation study looks at different numerical optimisation methods for maximum likelihood estimation of the discrete D-vine and the effect of providing the analytic score function when optimising.

Simulation Study I

We consider time series of different lengths N=100, N=250 and N=500. We generate $S = 100$ realisations from two Gaussian copula models: with AR(1) and ARMA(1,1) dependence structure with parameters $a_1 = 0.7$ and ($a_1 = 0.8$, $b_1 = -0.4$) respectively and negative binomial parameters ($\pi = 0.5, s = 3$). The Gaussian copula model is fitted to the data using joint maximum likelihood using the numerical optimisation routine `constrOptim.nl` in R (Varadhan, 2014) and the standard errors are computed both analytically (Section 6.6.4) and numerically using `numDeriv` for comparison.

The computational times elapsed to optimise the likelihood (perform maximum likelihood estimation) for the $S = 100$ simulations are recorded and shown using a violin plot in Fig. 6.11. The average time is 0.5, 5 and 8 hours respectively for the lengths N when performing parallel computing.

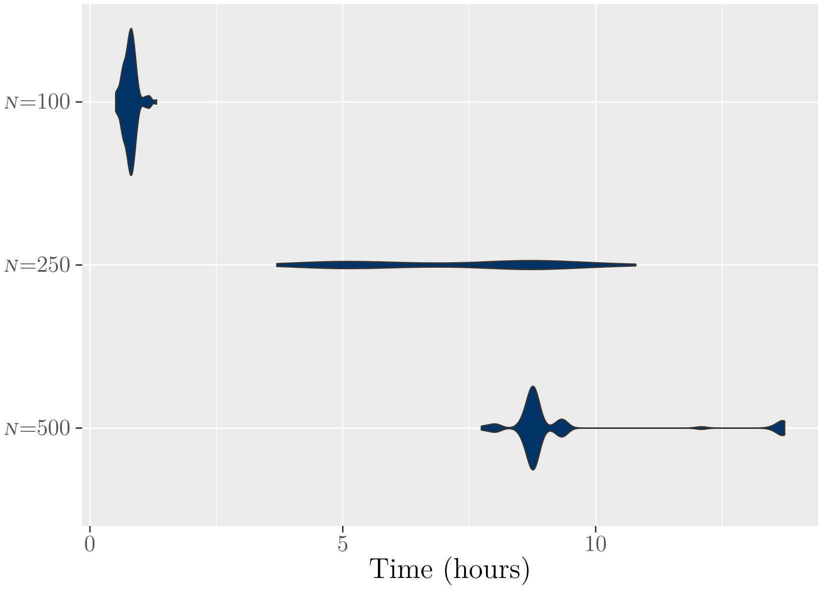


Figure 6.11: A violin plot of the computational time to perform maximum likelihood estimation to fit a Gaussian copula model with AR(1) dependence to integer-valued time series of lengths $N = 100, N = 250$ and $N = 500$.

The joint maximum likelihood estimates of $S = 100$ simulations for the AR(1) model are given in Fig. 6.12. The plots suggest that the method performs well for the AR(1) model and the sample variance decreases as N increases. The mean values, sample variance, bias, RMSE of the estimates and mean of the analytical standard errors are calculated and given in Table 6.3. The bias gets closer to zero as N increases, while the variance decreases also.

Figure 6.12: Boxplots of maximum likelihood estimates for the Gaussian copula model with negative binomial margins from $S = 100$ simulated data of length $N = 100, N = 250$ and $N = 500$, with true values $a_1 = 0.7$, $s = 3$, $\pi = 0.5$ represented by a red dashed line.

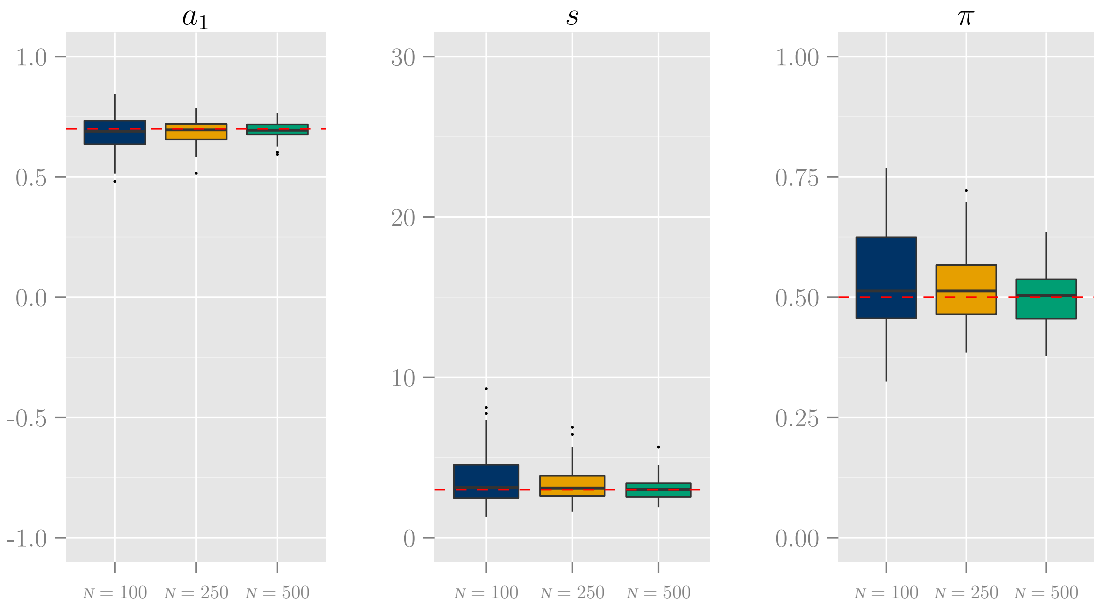


Table 6.3: The mean and standards deviation of maximum likelihood estimates and the mean of the analytic standard errors for simulation results using $S = 100$ realisations of the model with AR(1) dependence structure and negative binomial margins for lengths $N = 100$, $N = 250$ and $N = 500$. True parameters are $a_1 = 0.7$, $\pi = 0.5$ and $s = 3$.

Parameter		N=100	N=250	N=500
a_1	mean	0.679	0.689	0.692
	var	0.005	0.002	0.002
	bias	-0.021	-0.011	-0.090
	RMSE	0.075	0.003	0.003
π	mean	0.541	0.519	0.510
	var	0.015	0.005	0.040
	bias	0.041	0.019	0.011
	RMSE	0.128	0.006	0.004
s	mean	4.320	3.311	3.055
	var	24.942	1.036	0.981
	bias	1.320	0.311	0.221
	RMSE	5.166	1.132	0.890

Table 6.5 displays the results of our method on the ARMA(1,1) model. All the numerical standard errors are within 3 decimal places of the analytical presented. To investigate further, we compare our results with two other methods. Let us denote our joint MLE method as $M_{\mathbf{Y}}$ since we apply the model directly to the integer-valued time series $\mathbf{Y}$. The two other scenarios which we consider as a 'best ' case and a 'computationally fast but an approximation' case;

i. through a simulation study, we can directly estimate the dependence parameters and corresponding standard errors from the underlying continuous time series $\mathbf{X}$ and estimate the marginal parameters under the independence likelihood, denoting this method $M_{\mathbf{X}}$.

ii. we compare with the importance sampling approach of Masarotto et al. (2012) using the `gcmr` R function and denote this simply as $M_{\mathbf{Z}}$.

Boxplots of the distribution of the errors of the estimates of our method and $M_{\mathbf{X}}$

and $M_{\mathbf{Z}}$ on the ARMA(1,1) model are given in Fig. 6.13. In all cases, the discrete D-vine fit outperforms the other two methods in terms of narrowest range of errors and consistently centred symmetrically at zero. Joint maximisation under the discrete D-vine estimates the dependence parameters comparably well to the case when the underlying time series is known, in particular for the marginal parameter estimates. The estimates under the marginal assumption gives better estimates. We can get an idea of how close to the well-established gold standard ARMA methods we can achieve by comparing the estimates through $M_{\mathbf{Y}}$ to $M_{\mathbf{X}}$. Fig. 6.13 show that the discrete D-vine compares favourably to knowing the underlying time series, which is impossible in practice. The negative binomial parameters π, s are estimated well under the assumption there is no dependence structure ($M_{\mathbf{X}}$) and the estimates of the marginal parameters under the independence likelihood are given in Table 6.5. It should be noted that the standard errors of the marginal parameters estimated using $M_{\mathbf{X}}$ presented here are not accurate because they have not been adjusted for dependence (Chandler and Bate, 2007). The importance sampling approach of $M_{\mathbf{Z}}$ performs overall the worst for this model.

Figure 6.13: The distribution of errors of estimates for $S = 100$ simulated data of length $N = 100$ and $N = 250$ with true values $a_1 = 0.8$, $b_1 = -0.4$, $s = 3$, $\pi = 0.5$. The navy colour corresponds to the joint maximum likelihood estimates from the discrete D-vine $M_{\mathbf{Y}}$, the yellow corresponds to the continuous ARMA and independent marginal likelihood, $M_{\mathbf{X}}$, and the green corresponds to the estimates using the gcmr R package, $M_{\mathbf{Z}}$.

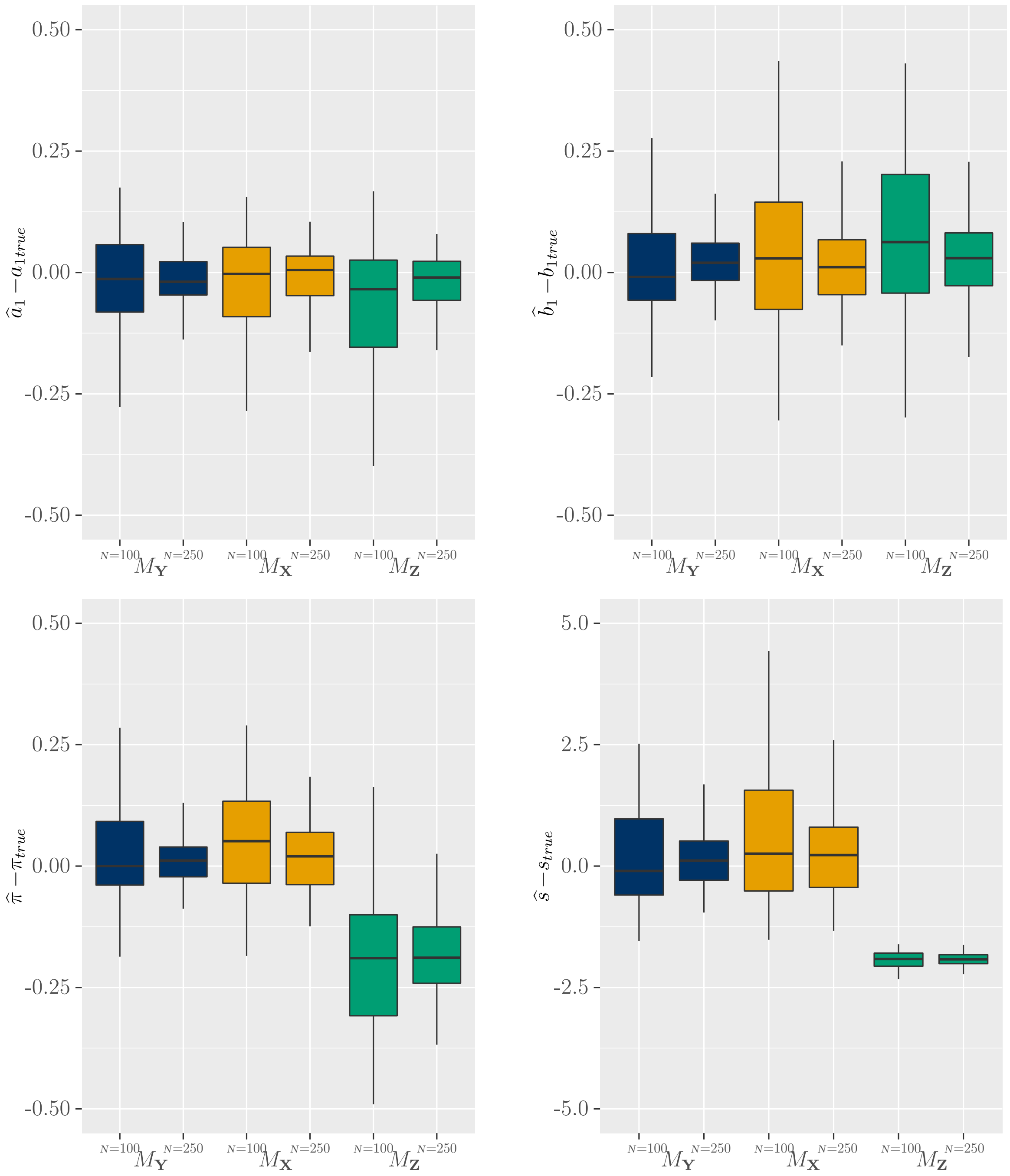


Table 6.4: ARMA(1,1) Gaussian copula model estimates

		Method														
		D-vine					True ARMA					gcmr				
Length	$\boldsymbol{\theta}$	mean	var	bias	RMSE	$\overline{\text{var}(a)}$	mean	var	bias	RMSE	$\overline{\text{var}(a)}$	mean	var	bias	RMSE	$\overline{\text{var}(a)}$
	a_1=0.8	0.78	0.011	-0.02	0.10	0.116	0.76	0.016	-0.04	0.13	0.015	0.72	0.021	-0.08	0.16	0.132
N=100	b_1=-0.4	-0.39	0.011	0.01	0.10	0.163	-0.36	0.026	0.04	0.16	0.029	-0.32	0.029	0.08	0.19	0.180
	s =3	3.91	11.92	0.91	3.57	1.66	4.36	11.85	1.36	3.70	35.66	1.07	0.035	-1.93	1.94	0.183
	π=0.5	0.53	0.012	0.03	0.11	0.138	0.55	0.014	0.05	0.13	0.056	0.31	0.021	-0.19	0.24	0.154
	a_1=0.8	0.78	0.003	-0.02	0.06	0.072	0.79	0.005	-0.01	0.07	0.005	0.77	0.005	-0.03	0.07	0.073
N=250	b_1=-0.4	-0.38	0.003	0.02	0.06	0.084	-0.39	0.008	0.01	0.09	0.011	-0.37	0.008	0.03	0.09	0.108
	s=3	3.16	0.429	0.16	0.68	0.592	3.38	1.177	0.38	1.15	0.689	1.08	0.016	-1.92	1.92	0.119
	π=0.5	0.51	0.003	0.01	0.05	0.027	0.52	0.006	0.02	0.08	0.023	0.32	0.009	-0.18	0.20	0.096

Table 6.5: The mean and standards deviation of maximum likelihood estimates and the mean of the analytic standard errors for simulation results using $S = 100$ realisations of the model with ARMA(1,1) dependence structure and negative binomial margins for lengths $N = 100$ and $N = 250$ (* denotes $< 1 \times 10^{-4}$).

Simulation Study II

A number of numerical optimisation methods can be used to optimise the likelihood function. Available algorithms for optimising smooth nonlinear objective functions with constraint in R include augmented Lagrangian without and with adaptive barriers (`auglag`, `constOpt.nl` respectively), Nelder-mead derivative-free (`nmkb`) and box-constrained minisisers (`nlminb`). Table 6.6 compares the optimal value of the likelihood values when an AR(3) Gaussian copula model is fitted to simulated data. The estimates and the time taken is recorded. The preferred optimisation in R for our Gaussian D-vine PCC is the `constOpt.nl` function in the R package `alabama` (Varadhan, 2014).

There is a significant decrease in time taken for all functions when the analytic gradient is supplied. However the `nmkb` method, which is a derivative free method, performs consistently well for shorter lengths of $N \leq 200$. Other methods such as `bobyqa`, `Rcgmin`, `Rvmmin`, `spg`, `hjkb` were also tested but did not compare as well as the methods shown in Table 6.6 and encountered issues such as of non-convergence. It should be noted that the objective function (the log-likelihood function) should be defined outside of constrained region so that when the optimiser requires an evaluation outside the constrained region close to the boundary, the function is directed back inside the constrained region.

Table 6.6: Comparing different numerical optimisation methods on an AR(3) Gaussian copula model with negative binomial margins. True parameters are $a_1 = -0.8$, $a_2 = 0.6$, $a_3 = 0.7$, $s = 2$, $\pi = 0.5$. The methods denoted with a + included the analytic gradient of Algorithm 6.

Length	Optimisation		Parameters					Time	Time
N	method		$\widehat{a}_1$	$\widehat{a}_2$	$\widehat{a}_3$	$\widehat{s}$	$\widehat{\pi}$	(min)	Rank
20	auglag	+	-0.7637	0.5492	0.6096	4.3241	0.6864	1.4851	-
30	auglag	+	-0.7726	0.5690	0.7209	3.0833	0.6113	3.2002	-
40	auglag	+	-0.7734	0.6231	0.6607	2.4434	0.5338	5.2868	-
50	auglag	+	-0.8060	0.5834	0.6359	2.6956	0.5615	7.3724	-
100	nmkb		-0.7787	0.5716	0.6022	2.6105	0.5692	26.8857	2
	constOpt		-0.7795	0.5709	0.6021	2.5964	0.5677	53.2930	5
	constOpt	+	-0.7787	0.5716	0.6022	2.6108	0.5693	25.4747	1
	auglag		-0.7794	0.5712	0.6024	2.5910	0.5672	60.6307	6
	auglag	+	-0.7787	0.5716	0.6023	2.6102	0.5692	40.1581	4
	nlminb		-0.7787	0.5716	0.6023	2.6102	0.5692	39.1909	3
200	nmkb		-0.7571	0.5936	0.5856	2.6466	0.5765	109.0634	2
	constOpt		-0.7572	0.5933	0.5852	2.6457	0.5764	190.9778	5
	constOpt	+	-0.7571	0.5937	0.5856	2.6464	0.5764	115.6024	3
	auglag		-0.7578	0.5934	0.5859	2.6264	0.5744	223.5922	6
	auglag	+	-0.7571	0.5937	0.5857	2.6460	0.5764	90.9063	1
	nlminb		-0.7571	0.5937	0.5857	2.6459	0.5764	164.6186	4
300	nmkb		-0.7456	0.5851	0.5889	3.2141	0.6310	320.1936	3
	constOpt		-0.7461	0.5849	0.5891	3.1923	0.6292	472.9344	4
	constOpt	+	-0.7456	0.5850	0.5889	3.2136	0.6309	243.1077	1
	auglag		-0.7463	0.5848	0.5893	3.1921	0.6292	673.7797	5
	auglag	+	-0.7456	0.5850	0.5889	3.2160	0.6311	263.9167	2
	nlminb		0.1821	0.4125	0.4699	4.8365	0.6686	41.4495	-

6.8 Discussion

Dependence modelling using vines can done in two ways: the first is to build from a collection of bivariate 'blocks' to give the overall joint distribution, and the second is to decompose a multivariate density into a cascade of bivariate copulas. Our work follows the latter.

We have shown that the Gaussian copula model (3.9) is equivalent to a discrete D-vine PCC and we have verified this with numerical results. The likelihood has a computationally tractable form compared with the likelihood of a Gaussian copula model with discrete margins. The vine methodology allows the likelihood to be written in a computationally tractable form and facilitates maximum likelihood estimation. This has also led to analytic solutions for the standard errors of the estimates. Simulation studies have confirmed the accuracy that can be achieved by using vine copula methods and the method compares favourably to estimating the copula parameters in the best case scenario of knowing the underlying time series.

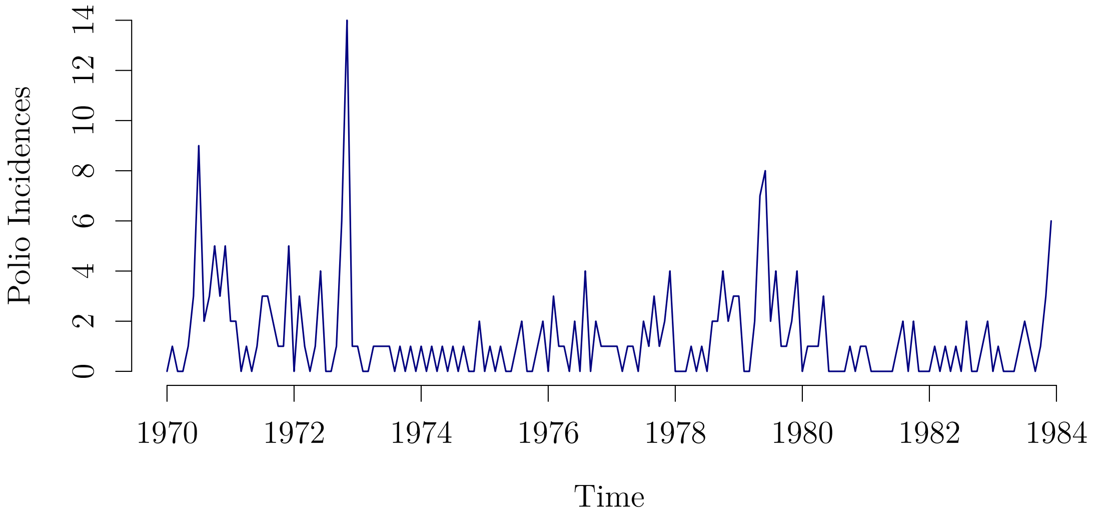
Chapter 7

Empirical Results and Comparisons

In this chapter we analyse Zeger's (1988) US polio data, a well known integer-valued time series from the literature, to compare the Gaussian copula model estimation methods developed in Chapters 4, 5 and 6 in real life scenarios. This data set has been extensively discussed by many authors allowing us to draw comparisons to the integer-valued time series literature reviewed in Chapter 2 (Zeger, 1988; Zeger and Qaqish, 1988; Li, 1994; Jørgensen et al., 1999; Davis et al., 2000; Heinen, 2003; Neal and Subba Rao, 2007; Davis and Wu, 2009; Masarotto et al., 2012; Neal and Kypraios, 2015).

Time series of the number of polio incidences in the US between 1978-1983 are reported monthly (N= 168) by the Center for Disease Control. The original interest was to detect a decreasing trend in the number of cases of the infectious disease since the introduction of the polio injection vaccine in 1955 and the oral vaccine in 1962. The data reveal some seasonality and the possibility of a slight decreasing trend.

Figure 7.1: Monthly number of US cases of poliomyelitis from 1970 to 1983.



Zeger (1988) proposed a Poisson regression model with explanatory variables to model the trend and seasonality. These include an intercept term, a linear trend and harmonics at periods of 6 and 12 months. Zeger (1988) models the trend and seasonality explicitly conditional on a latent weakly stationary AR(1) process (a parameter-driven model). Zeger adopts an estimating equation approach for this model and concludes that the decrease in trend becomes less significant once the correlation structure of the time series is taken into account. Chan and Ledolter (1995) apply a MCEM algorithm for Zeger's model, modified to have a Gaussian distribution assumption on a latent stationary AR(1) process. Benjamin et al. (2003) fit a Generalised ARMA model which can be implemented in the `gamlss.util`. The data are taken from the `R` package `gamlss.data` (Stasinopoulos and Rigby, 2015) because the GARMA is a GAM with an exponential family. Masarotto et al. (2012) fit the Gaussian copula model with discrete marginal regression using a simulated likelihood approach as discussed in Section 3.7 to the data.

We fit the model of Masorotto & Varin using the estimation methods described in Chapters 4, 5 and 7 and later we fit similar models to compare. The MCEM algorithm is applied in Section 7.1, the ABC algorithm in Section 7.2 and joint maximum likelihood estimation using the D-vine representation in Section 7.3. We investigate other models for the data in Section 7.4. At the end of this chapter in Table 7.9 we give a full comparison of all of the estimates obtained using the different methods, as well as the those reported by Masarotto are displayed.

Serial correlation is accommodated by assuming an ARMA(2,1) model for the underlying time series. The negative binomial distribution is considered for the marginal distribution of Y_i with parameterisation

$$\Pr(Y_i = y) = \frac{\Gamma(y + \frac{1}{\kappa})}{y!\Gamma(\frac{1}{\kappa})} \left(\frac{1}{1+\kappa\mu_i}\right)^{\frac{1}{\kappa}} \left(\frac{\kappa\mu_i}{1+\kappa\mu_i}\right)^{y},$$

where $\mu_i = \mathrm{E}(Y_i \mid \mathbf{x}_i)$ and variance $\mathrm{var}(Y_i \mid \mathbf{x}_i) = \mu_i + \kappa\mu_i^2$ where κ is the overdispersion parameter. The log-linear model with covariates (Masarotto et al., 2012) is

$$\begin{aligned} \log\,(\mu_i) = \beta_0 + \beta_1 t_i + \beta_2 \cos\left(\frac{2\pi t_i}{12}\right) + \beta_3 \sin\left(\frac{2\pi t_i}{12}\right) \\ + \beta_4 \cos\left(\frac{2\pi t_i}{6}\right) + \beta_5 \sin\left(\frac{2\pi t_i}{6}\right). \end{aligned} \tag{7.1}$$

The linear trend is rescaled and centred at January 1976, by $t_i = (t'_i - 73)/1000$. The parameter vector for this model is $\boldsymbol{\theta} = (a_1, a_2, b_1, \beta_0, \ldots, \beta_5, \kappa)$. This parameterisation corresponds to the negative binomial distribution with parameters (s, π) previously used in the simulation studies with

$$\mu = \frac{s(1-\pi)}{\pi} \text{ and } \kappa = \frac{1}{s}.$$

7.1 MCEM algorithm

In this section, the MCEM algorithm described in Chapter 4 is applied to fit the model (7.1). Estimation of the parameters $\boldsymbol{\theta}$ is carried out in two-stages (Section 3.5); first we estimate the marginal structure and then the dependence structure. Section 7.1.1 considers parametric estimates of the marginals and Section 7.1.2 considers a nonparametric estimation.

7.1.1 Parametric marginals

At the first stage, the marginal parameters are estimated under the assumption Y_i are i.i.d. via the method of iteratively re-weighted least squares. The estimates of the the regression coefficients and the size parameter κ of the negative binomial are

$$(\widehat{\beta_0}, \widehat{\beta_1}, \widehat{\beta_2}, \widehat{\beta_3}, \widehat{\beta_4}, \widehat{\beta_5}, \widehat{\kappa}) = (0.209, -4.332, -0.143, -0.503, 0.168, -0.421, 0.567).$$

At the second stage the MCEM algorithm (Algorithm 2) is implemented using the estimated marginal parameters. The initial values for the copula parameters are $a_1 = a_2 = b_1 = 0$ and the number of Monte Carlo samples begins small with $m = 10$ and increases at every tenth iteration as $m =$ 50, 100, 500, 1000, 5000, 10000, 50000, 100000.

The MCEM parameter chains are plotted in Fig. 7.2. Fig. 7.3 shows the estimated change in log-likelihood (4.9). The parameter iterations suggest convergence after around iteration 50 and the change in likelihood confirms this; we see the chain begins to stabilise after 50 iterations for the Polio dataset of length N=168. This which corresponds to the number of Monte Carlo samples of $m > 5000$. After we establish convergence, the parameters and the estimated change in likelihood continues to fluctuate randomly around zero, even for large m due to the Monte Carlo E-step.

Figure 7.2: MCEM iterations for each ARMA(2,1) parameter for parametric model.

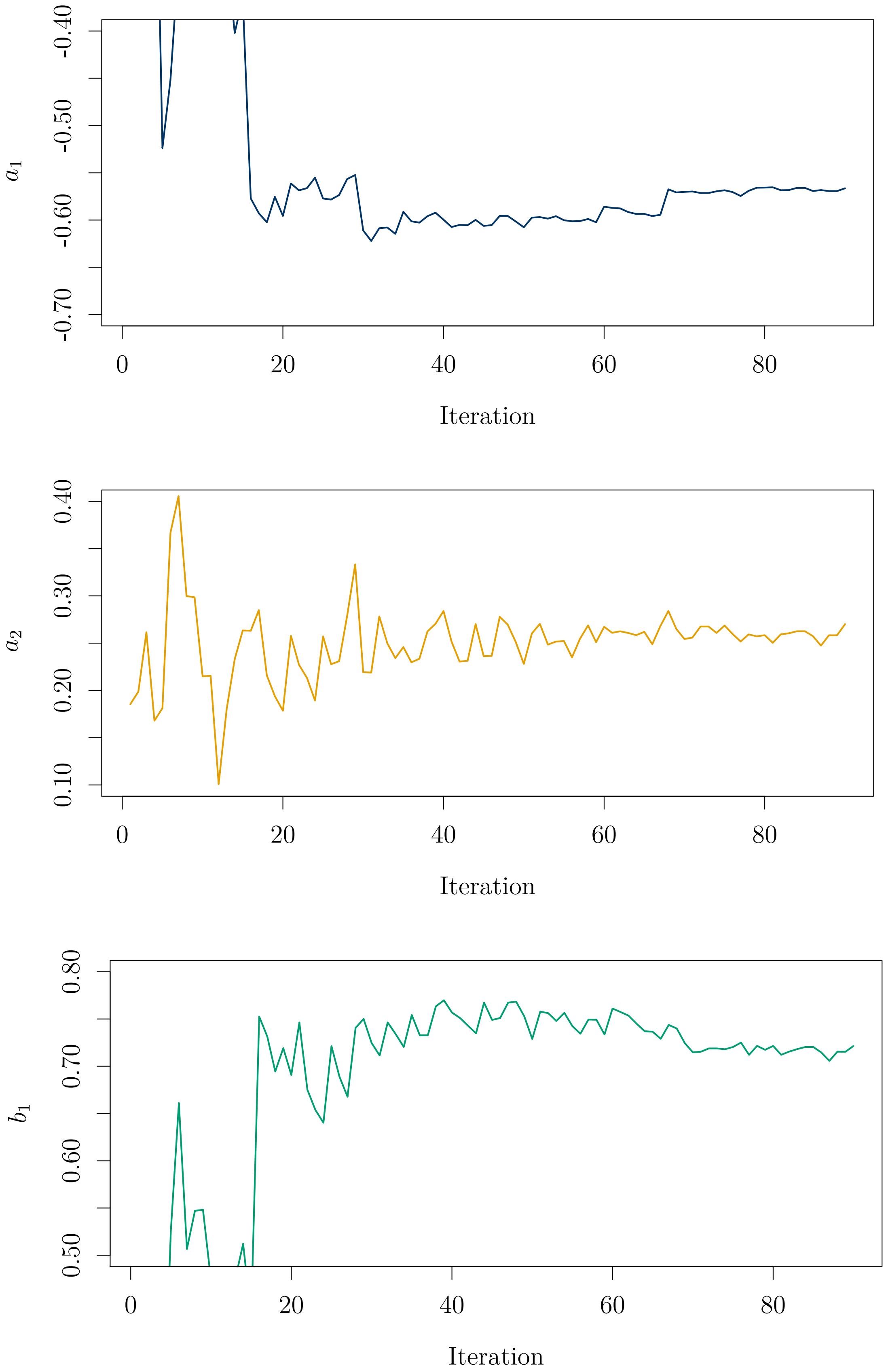
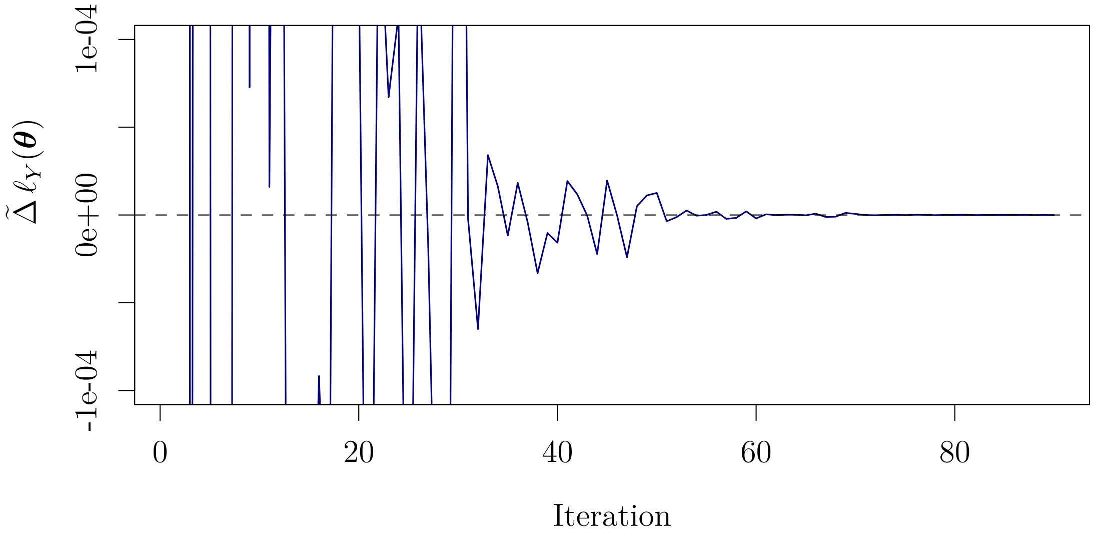


Figure 7.3: Estimated change in log-likelihood over the iterations 1 to 90 for the polio dataset, where m increases every 10 iterations.



The maximum likelihood estimates of the copula parameters $\boldsymbol{\theta}_{\text{copula}}$ are to 3 decimal places

$$\widehat{a}_1 = -0.566, \quad \widehat{a}_2 = 0.270 \quad \text{and} \quad \widehat{b}_1 = 0.721.$$

The observed information matrix (4.10) is calculated at $\widehat{\boldsymbol{\theta}}$ as

$$I_{\mathbf{Y}}(\widehat{\boldsymbol{\theta}}) = \begin{pmatrix} 310.9 & -230.9 & 246.9 \\ -230.9 & 295.6 & -169.3 \\ 246.9 & -169.3 & 206.6 \end{pmatrix},$$

giving the standard errors (Section 4.6) as

$$\text{s.d}(a_1) = 0.260, \, \text{s.d}(a_2) = 0.099, \, \text{s.d}(b_1) = 0.272.$$

We acknowledge that the length of the time series $N = 168$ may be too small for asymptotic results of the inference function for margins (IFM) to hold (Joe, 1996).

7.1.2 Non-parametric marginals

For comparison, we repeat the MCEM algorithm with non-parametric estimates of the marginal distribution in the first stage of the estimation, and run the MCEM algorithm as before. The MCEM parameter values are plotted in Fig. 7.4 with the MCEM chains of the parametric model displayed by dashed lines for comparison.

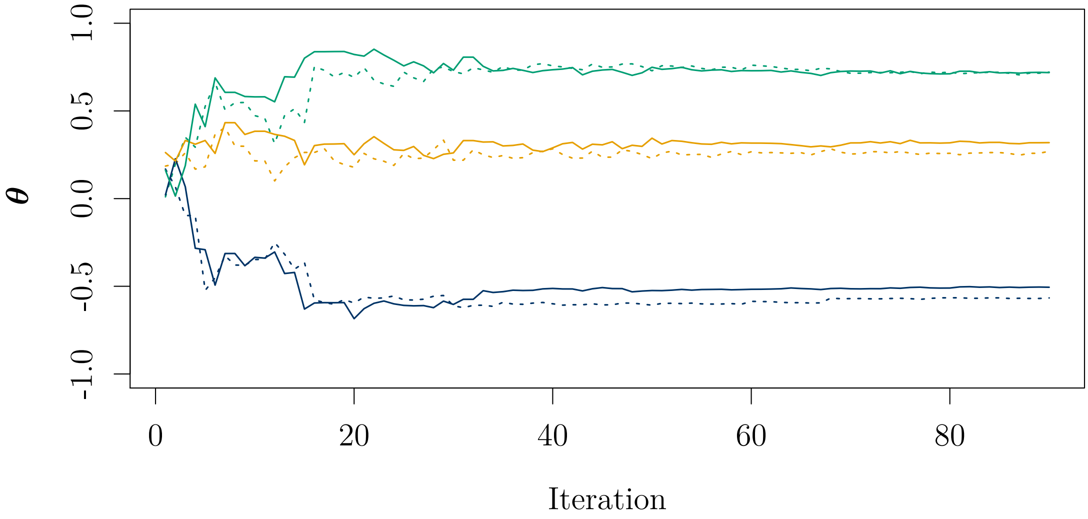


Figure 7.4: MCEM iterations for a semi-parametric Gaussian copula model with ARMA(2,1) parameters where m increases every 10 iterations. The fully parametric MCEM chains of Fig. 7.2 are shown by dashed lines.

The parameter chains are similar to the fully parametric model and the estimated change in log-likelihood, not presented here, was similar also, suggesting convergence after 50 iterations ($m > 5000$). The updated parameter values for the semi-parametric model are slightly higher in the semi-parametric model and agree closer with the results presented by Masarotto et al. (2012). The parameter estimates and standard errors of both the paramertic and semiparametric models are given in Table 7.1.

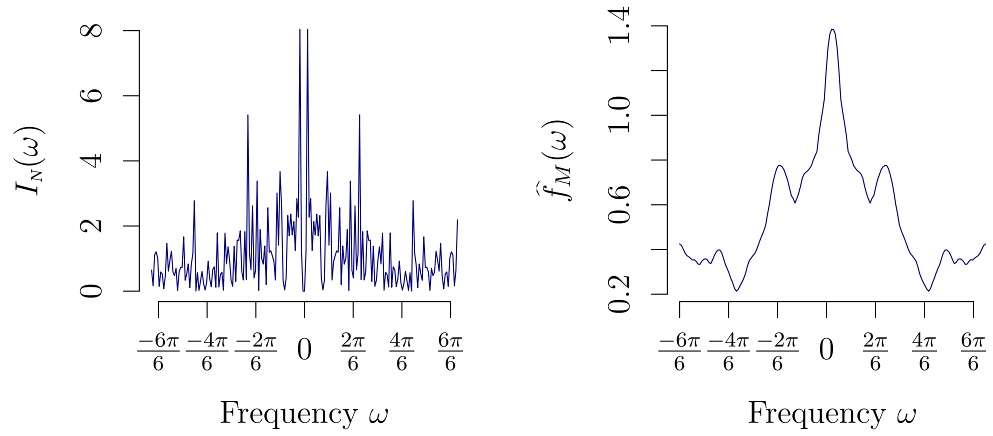
Table 7.1: The parameter estimates of the Gaussian copula model with negative binomial regression margins (7.1) for the Polio data set.

	MCEM				Masarotto	
	Semi-parametric		Parametric			
	$\widehat{\boldsymbol{\theta}}$	s.e	$\widehat{\boldsymbol{\theta}}$	s.e	$\widehat{\boldsymbol{\theta}}$	s.e
a_1	-0.5054	(0.209)	-0.5664	(0.260)	-0.5229	(0.220)
a_2	0.3192	(0.089)	0.2701	(0.099)	0.3046	(0.090)
b_1	0.7186	(0.217)	0.7214	(0.272)	0.6959	(0.229)
β_0	-	-	0.2093	(0.096)	0.2095	(0.121)
β_1	-	-	-4.3318	(1.895)	-4.3151	(2.284)
β_2	-	-	-0.1430	(0.129)	0.1215	(0.147)
β_3	-	-	-0.5025	(0.138)	-0.4967	(0.157)
β_4	-	-	0.1682	(0.131)	0.1903	(0.129)
β_5	-	-	-0.4214	(0.132)	-0.4030	(0.128)
κ	-	-	0.5671	(0.484)	0.5700	(0.170)

7.2 Parameter estimation via ABC

To apply the ABC algorithm we compute the observed summary statistics for the Polio data set $\mathbf{y}$ following Chapter 5. These are the spectral estimates at a fixed set of frequencies and sample proportions of Chapter 5. Fig. 7.5 shows the periodogram using the discrete Fourier transforms in the left panel and the spectral estimate in the right panel smoothed using the Bartlett-Priestley window (5.9) where a bandwidth of $M = 10$ is found suitable giving $\widehat{\sigma}^2 = 0.0714$ (5.13).

Figure 7.5: Periodogram and estimated spectrum for the Polio data set.



We let $L = 10$ resulting in $L/2 + 1 = 6$ summary statistics (Section 5.3). The observed summary statistics are given in Table 7.2.

Table 7.2: Observed summary statistics of the Polio data.

Frequency	ω_0	ω_1	ω_2	ω_3	ω_4	ω_5
Spectral estimate	1.3394	0.7024	0.6546	0.2459	0.3635	0.3959

(a) Summary statistics for the dependence structure.

Integer-value	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Sample	55	22	12	6	3	2	1	1	1	0	0	0	0	1

(b) Summary statistics for the dependence structure.

Data are simulated from the model, given parameters $(\boldsymbol{\theta}_{\text{copula}}, \boldsymbol{\theta}_{\text{marginal}}) = (\mathbf{a}, \mathbf{b}, \boldsymbol{\beta}, \kappa)$ by

1. Simulate $\mathbf{Z} \sim$ARMA(2,1) such that $\text{var}(Z) = 1$ using $\mathbf{a}, \mathbf{b}$.
2. Transform to $\mathbf{y} = F^{-1}(\Phi(\mathbf{Z}))$ where Φ is the standard normal cdf and F^{-1} is the inverse cdf of F with mean $\mu_i = \exp(\mathbf{x}_i^\top \boldsymbol{\beta})$ and overdispersion parameter κ.

The ABC algorithm (Algorithm 3) is implemented using the statistic T (5.12) and the Euclidean distance metric for the dependence and marginal distance metrics respectively. For the priors of the marginal parameters, we used normal distributions centred around the estimates under found from the independence likelihood with variance 1 and for the copula parameters we simulated uniformly over the stationarity constraints,

$$\begin{aligned} -1 < b_1 < 1, &\qquad -2 < a_2 < 2, \\ a_1 + a_2 < 1, &\qquad a_2 - a_1 < 1. \end{aligned}$$

The ABC output consists of $n = 42,000$ parameter sets with their corresponding distances and summary statistics using $\varepsilon_T = \varepsilon_d = 20$. We use local linear and local nonlinear regression giving most weight to parameter values with smaller distances. We compare simulated datasets in two parts and thus we use two distance measures. Therefore there are two sets of summary statistics to regress onto; $\boldsymbol{\theta}_{\text{copula}} = (a_1, a_2, b_2)$ are regressed on S_1^* using T and $\boldsymbol{\theta}_{\text{marginal}}$=($\beta_0, \beta_1$, β_2, β_3, β_4, β_5, κ) are regressed on S_2^* using d. The parameter estimates are given in Table 7.3.

	ABC			MCEM	Masarotto
	$\varepsilon_T = 5$	Adjusted	nonlinear		(2012)
a_1	0.160 (0.49)	0.075 (0.50)	0.057 (0.51)	-0.566 (0.26)	-0.523 (0.22)
a_2	0.160 (0.32)	0.119 (0.34)	0.107 (0.32)	0.270 (0.10)	0.305 (0.09)
b_1	-0.067 (0.48)	-0.142 (0.47)	-0.128 (0.47)	0.721 (0.27)	0.696 (0.23)
β_0	0.048 (0.30)	0.018 (0.26)	0.028 (0.25)	0.209 (0.10)	0.210 (0.12)
β_1	-4.522 (1.02)	-4.767 (1.01)	-4.719 (0.97)	-4.330 (1.89)	-4.320 (2.28)
β_2	-0.023 (0.49)	-0.011 (0.52)	-0.021 (0.51)	-0.143 (0.13)	0.120 (0.15)
β_3	-0.119 (0.50)	-0.041 (0.53)	0.010 (0.51)	-0.503 (0.14)	-0.497 (0.16)
β_4	0.061 (0.56)	0.185 (0.55)	0.183 (0.54)	0.168 (0.13)	0.190 (0.13)
β_5	-0.131 (0.56)	-0.165 (0.56)	-0.193 (0.54)	-0.421 (0.13)	-0.403 (0.13)
κ	0.642 (0.89)	0.577 (0.82)	0.608 (0.81)	0.567 (0.48)	0.570 (0.17)

Table 7.3: The ABC estimates and standard errors given in parentheses for the polio data compared with our MCEM and importance sampling estimates of Masarotto et al. (2012).

The marginal parameter $\beta_1, \beta_4, \beta_5$ and κ compare reasonably well with the results from the previous analysis and Masarotto et al. (2012). However, the dependence parameters are far from the MLEs from the MCEM algorithm. Fig. 7.6 plots the joint densities for each of the pairs of parameters. The regression adjusted output is shown in blue with the unadjusted ABC output in grey. The peaks of the contours are far from the MLEs found using the MCEM algorithm. The adjustment brings them slightly closer to the values, suggesting the estimates may improve with more samples and lower thresholds.

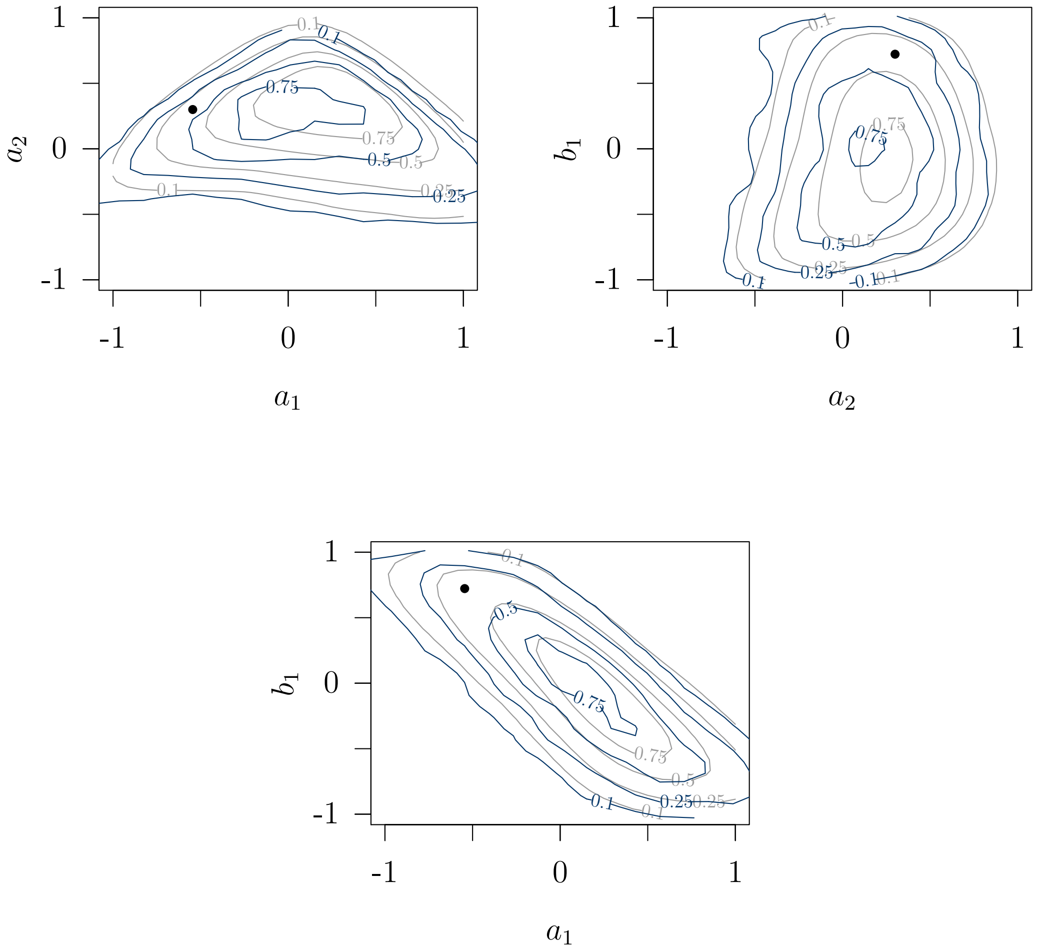


Figure 7.6: The joint posteriors for the three weighted regression adjusted dependence parameters are shown in blue, and the unadjusted joint posterior are shown in grey for comparison for quantiles The MLEs from the MCEM algorithm are indicated by the black dot.

The copula parameters are not close to what we expected and the standard errors are relatively large. Initially we think this maybe because using the threshold value $\varepsilon_T = 20$ is too high and more computational effort (collecting more samples) would improve the results. But the regression adjustments (Section 5.5) can deal well with improving the results so we have a closer inspection of the accepted values. Fig. 7.7 shows the distance values for each of the parameters and we can see for the copula parameters are not well captured because it is not similar to the example of Fig. 5.2, i.e., the trough is not as sharp and peaked or close to the red line.

We suspect the difference in parameter estimates between the ABC, MCEM and Masorotto is because the ABC algorithm may perform poorly for shorter time series since maybe there is not enough dependence information in shorter time series. The polio data is quite short $N = 168$ and the simulation study in Chapter 5 performed better for longer time series. Table 5.2 shows that the ARMA(1,1) parameters performed poorly for $N < 250$. The test statistic T uses only the second order structure of the series to estimate the dependence parameters and we suspect these are not captured well for $N < 250$. The MCEM algorithm may be picking up information from elsewhere in the data that the ABC algorithm doesn't pick up.

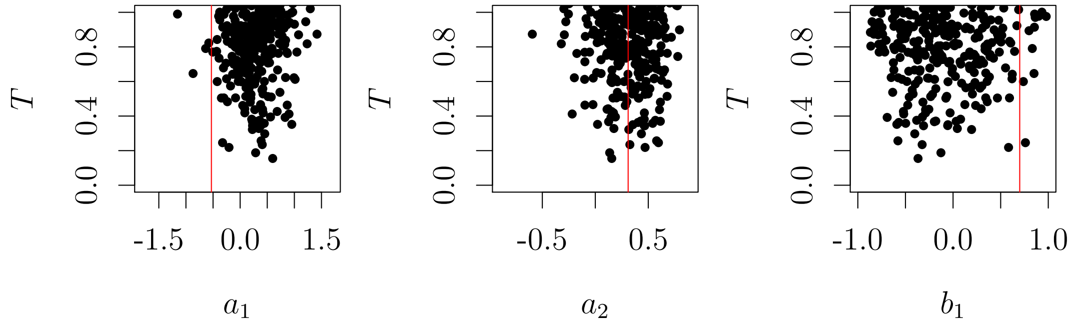


Figure 7.7: ABC distance measures for ARMA(2,1) parameters. The MLEs from the MCEM analysis are shown by the red line.

7.3 MLE estimation via the discrete D-vine PCC

Full joint maximum likelihood estimation is carried out using numerical optimisation of `nmkb()` from the R package `dfoptim()`. The maximum likelihood estimates are given in Table 7.4

	D-vine PCC		Masarotto	
Parameters	$\widehat{\boldsymbol{\theta}}$	s.e	$\widehat{\boldsymbol{\theta}}$	s.e
a_1	-0.5769	(0.20)	-0.5229	(0.22)
a_2	0.2932	(0.09)	0.3046	(0.09)
b_1	0.7585	(0.20)	0.6959	(0.23)
β_0	0.2100	(0.12)	0.2095	(0.12)
β_1	-4.3866	(2.25)	-4.3151	(2.28)
β_2	-0.1207	(0.15)	-0.1215	(0.15)
β_3	-0.4938	(0.16)	-0.4967	(0.16)
β_4	0.1917	(0.13)	0.1903	(0.13)
β_5	-0.4084	(0.13)	-0.4030	(0.12)
$1/\kappa$	1.7701	(0.51)	1.7543	-

Table 7.4: Maximum likelihood estimates for the discrete D-vine PCC compared with the MCEM mles and Masarotto's results.

Randomised quantile residuals (Section 6.5.2) are computed several times for the results of Table 7.4. and the distributional assumptions are checked. The results are shown in Fig. 7.8. Fig. 7.8 shows normal probability and autocorrelation plots for a set of residuals. The left plot alludes that the model is a good fit and the right plot suggests that no residual serial correlation is present in the data.

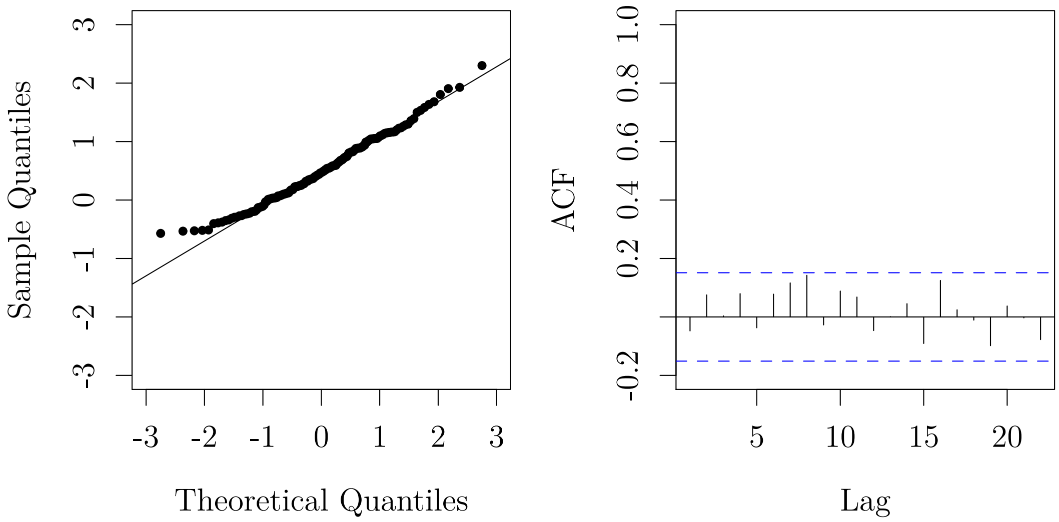


Figure 7.8: Polio data. Normal probability (left panel) and autocorrelation plots (right panel) for a set of randomised residuals.

For a further comparison with the semiparametric model in Section 7.1.2, we estimate the marginals nonparametrically. We take a grid of points for the parameters a_1, a_2 and b_1 and evaluate the likelihood at each triplet of values. The negative log-likelihood surface is uni-modal and is plotted in Fig. 7.9 for each pair of parameters around the maximum values. The red dots represent the lower parts of the surfaces where the maximum likelihood estimates are achieved. The log-likelihood surface is maximised at $\widehat{\ell} = 252.363$. The maximum likelihood estimates for the Gaussian discrete D-vine with non-parametric marginals are

$$\widehat{a}_1 = -0.485, \quad \widehat{a}_2 = 0.350, \quad \text{and } \widehat{b}_1 = 0.720.$$

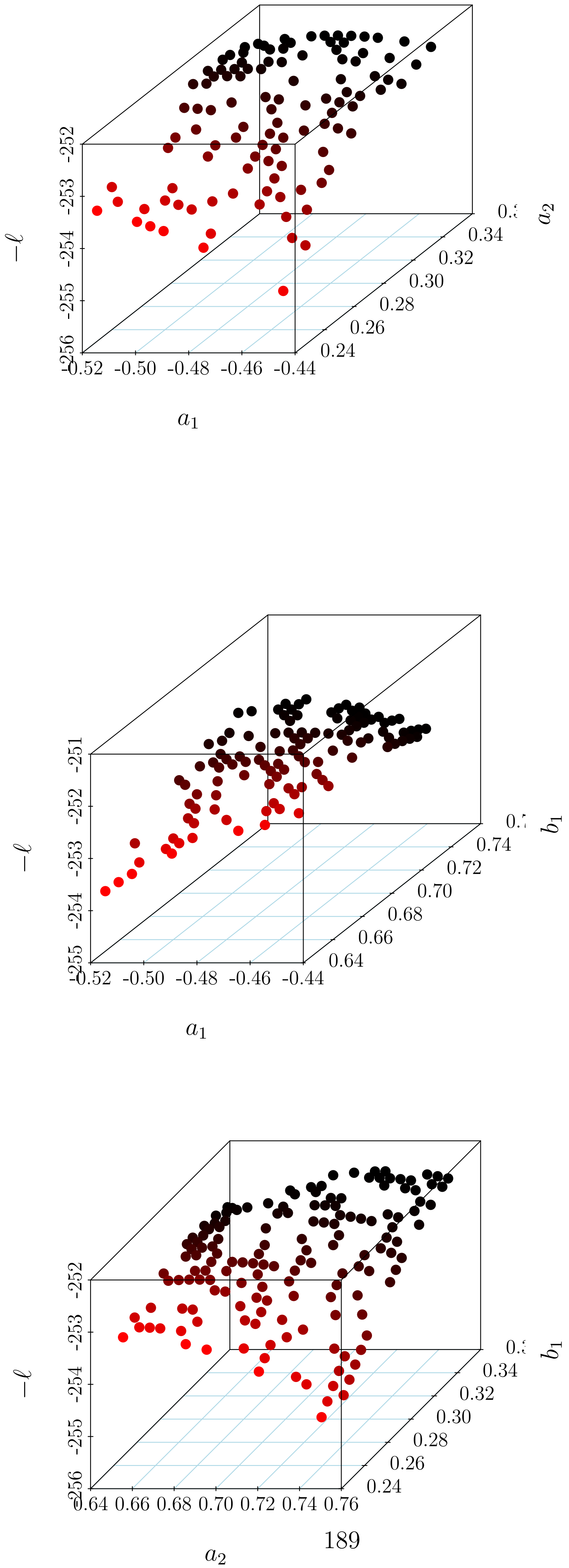


Figure 7.9: The likelihood surface.

7.4 Model Selection

Following Masarotto et al. (2012) we have fitted the ARMA(2,1) dependence structure but due to the ease of fitting the discrete D-vine to an integer-valued time series of length N=168 we carry out a model fitting exercise with differing dependence structure to the polio data using full maximum likelihood (in contrast to the simulated likelihood approach of Masarotto).

The results and corresponding AIC and BIC values are given in Table 7.5. The model selected according to the AIC is the ARMA(2,1) , however the BIC selects the AR(1) model. Of these models, the model with lowest AIC is the model fitted above suggested by Masarotto et al. (2012). As expected the time taken for the numerical optimisation increases as the number of parameters increase. These timings are much faster than the MCEM algorithm which is of the order hours.

Table 7.5: The maximised log-likelihood values, AIC, BIC and computational times for the discrete D-vine copula model with covariates (7.1) for the polio data.

		ARMA(p,q)									
p	q	a_1	a_2	a_3	b_1	b_2	b_3	-$\widehat{\ell}$	AIC	BIC	Time (m)
1	0	0.162						252.3	520.6	545.6	29
2	0	0.123	0.213					249.8	517.5	545.6	42
3	0	0.146	0.222	-0.100				249.2	518.4	549.7	46
0	1				0.112			252.8	521.6	546.5	29
0	2				0.163	0.208		249.8	517.7	545.8	28
0	3				0.169	0.196	-0.067	249.6	519.1	550.4	43
1	1	0.896			-0.779			249.9	517.9	546.0	57
1	2	-0.826			0.979	0.261		248.7	517.3	548.6	74
2	1	-0.577	0.293		0.759			248.0	516.0	547.2	48
3	1	-0.660	0.321	0.064	0.830			247.9	517.7	552.1	71
3	2	0.135	0.843	-0.134	0.003	-0.665		246.7	517.5	555.0	110

We further fitted a number of models to the polio data with different marginal structures. These include the geometric, Poisson and negative binomial with no covariates and the results are given in Tables 7.6, 7.7 and 7.8. Of the range of models compared, the model with the lowest AIC is the copula model with Poisson marginals and ARMA(2,1) dependence ($\widehat{\ell} = 259.2$, AIC=526.4). Comparing the AIC values with Table 7.5, it is clear that the model with covariates included are much more suitable.

Table 7.6: Fitted AR(p) dependence structures.

	AR(p)												Marginals		-Log-lik	AIC
p	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}			-$\widehat{\ell}$	
Poisson													λ			
1	0.269												0.4		263.7	531.3
2	0.206	0.221											0.4		261.1	528.1
3	0.235	0.243	-0.130										0.4		260.2	528.4
4	0.245	0.213	-0.157	0.150									0.4		259.1	528.2
Geometric													π			
1	0.220												1.4		291.3	586.7
2	0.189	0.110											1.4		289.3	584.6
3	0.195	0.130	-0.066										1.4		288.7	585.3
4	0.207	0.116	-0.092	0.088									1.4		287.5	585.0
Negative Binomial													s	π		
1	0.259												1.28	0.49	263.1	532.2
2	0.201	0.201											1.24	0.48	260.7	529.4
3	0.228	0.224	-0.123										1.24	0.48	259.8	529.6
4	0.241	0.196	-0.152	0.139									1.23	0.48	258.7	529.5
5	0.236	0.203	-0.161	0.129	0.039								1.24	0.48	258.6	531.3
6	0.231	0.194	-0.144	0.107	0.0163	0.092							1.23	0.48	258.2	532.3
7	0.230	0.194	-0.146	0.109	0.014	0.090	0.010						1.23	0.48	258.1	534.3
8	0.229	0.186	-0.145	0.103	0.022	0.077	-0.004	0.064					1.23	0.48	257.9	535.9
9	0.235	0.185	-0.134	0.102	0.028	0.069	0.012	0.081	-0.082				1.22	0.48	257.6	537.1
10	0.245	0.175	-0.140	0.087	0.022	0.060	0.033	0.056	-0.103	0.123			1.21	0.47	256.8	537.6
11	0.230	0.185	-0.151	0.078	0.008	0.058	0.023	0.076	-0.131	0.093	0.138		1.21	0.47	255.8	537.6
12	0.232	0.186	-0.153	0.080	0.008	0.060	0.023	0.076	-0.132	0.096	0.139	-0.011	1.21	0.47	255.8	539.6

Table 7.7: Fitted MA(q) dependence structures.

	MA(q)												Marginals		-Log-lik	AIC
q	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}			$-\widehat{\ell}$	
Poisson													λ			
1	0.178												0.4		265.0	534.1
2	0.262	0.270											0.4		260.6	527.2
3	0.261	0.255	-0.065										0.4		260.4	528.7
4	0.246	0.276	-0.049	0.087									0.4		259.8	529.6
Geometric													π			
1	0.174												1.4		293.2	590.4
2	0.204	0.169											1.4		288.6	583.3
3	0.204	0.171	-0.044										1.4		288.3	584.6
4	0.202	0.167	-0.043	0.030									1.4		288.1	586.2
Negative Binomial													s	π		
1	0.177												1.27	0.49	264.5	535.0
2	0.252	0.255											1.26	0.48	260.1	528.2
3	0.251	0.242	-0.065										1.27	0.49	259.9	529.7
4	0.239	0.258	-0.053	0.076									1.25	0.48	259.4	530.8
5	0.237	0.251	-0.056	0.092	0.035								1.26	0.48	259.3	532.7
6	0.233	0.239	-0.041	0.102	0.037	0.073							1.27	0.49	259.1	534.2
7	0.229	0.247	-0.041	0.095	0.038	0.073	0.027						1.26	0.49	259.0	536.1
8	0.292	0.277	-0.115	0.046	-0.019	0.018	0.113	0.230					1.15	0.46	257.3	534.5
9	0.264	0.306	-0.074	0.101	0.028	0.051	0.067	0.155	-0.147				1.18	0.47	256.1	534.2
10	0.268	0.307	-0.073	0.104	0.029	0.051	0.067	0.153	-0.152	-0.013			1.18	0.47	256.1	536.2
11	0.246	0.265	-0.184	0.096	-0.011	0.024	0.115	0.272	0.077	0.091	0.285		1.08	0.44	253.9	533.7
12	0.257	0.258	-0.186	0.117	0.017	0.034	0.142	0.284	-0.091	0.120	0.329	0.100	1.04	0.43	253.3	534.6

Table 7.8: Fitted ARMA (p, q) dependence structures.

	ARMA(p,q)								Marginals		-Log-lik	AIC
p,q	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4			-$\widehat{\ell}$	
Poisson									λ			
1, 1	0.810				-0.610				0.4		261.9	529.9
1, 2	-0.397				0.645	0.349			0.4		260.1	528.1
2, 1	-0.463	0.376			0.726				0.4		259.2	526.4
2, 2	-0.393	0.481			0.645	-0.123			0.4		259.1	528.2
Geometric									π			
1, 1	0.626				-0.443				1.4		289.8	585.6
1, 2	-0.285				0.490	0.230			1.4		288.2	584.4
2, 1	-0.478	0.251			0.686				1.4		288.1	584.1
2, 2	0.150	0.083			0.038	0.037			1.4		289.2	588.4
Negative Binomial									s	π		
1, 1	0.757				-0.553				1.26	0.48	261.5	531.1
1, 2	-0.384				0.626	0.332			1.26	0.49	259.6	529.2
1, 3	-0.768				1.000	0.423	0.117		1.25	0.48	259.2	530.3
1, 4	0.893				-0.668	0.031	-0.272	0.151	1.26	0.48	258.5	531.0
2, 1	-0.474	0.358			0.730				1.22	0.48	258.9	527.7
2, 2	-0.415	0.449			0.662	-0.105			1.22	0.48	258.8	529.6
3, 1	-0.552	0.379	0.048		0.799				1.22	0.48	258.8	529.6
3, 3	0.076	-0.076	0.675		0.177	0.302	-0.752		1.21	0.47	256.3	528.5
4, 1	0.952	0.042	-0.315	0.203	-0.733				1.24	0.48	258.2	530.4
4, 2	0.549	0.494	-0.291	0.120	-0.319	-0.389			1.22	0.48	258.0	532.1
4, 3	0.557	-0.742	0.218	-0.660	-0.660	-0.564	0.409		1.19	0.47	257.8	533.6

Table 7.9 summarises the estimates from this chapter. Both the D-vine and MCEM show similar estimates and standard errors to that of Masarotto et al. (2012) using the sequential importance sampling. The ABC algorithm performs poorly for the copula parameters and we elude that this is due to the length of the time series. The simulation study in Chapter 5 suggests this is reasonable. Masarotto et al. (2012)'s method is faster than any of our three discussed here but it uses only an approximation to the likelihood. Our contribution allows exact maximum likelihood estimation for the Gaussian copula and discrete marginals.

The standard errors in Table 7.9 for the MCEM marginal estimates are the smallest standard errors but they are underestimated because they have not been adjusted for dependence. By assuming i.i.d. data in the first of the two stage estimation approach, IFM (Section 3.5) and therefore have not been adjusted for dependence and are underestimated. Masarotto et al. (2012) compute the standard errors of the independence likelihood estimates using the heteroskedasticity and autocorrelation consistent (HAC) sandwich estimator for time series of Andrews (1991). In further work, we could take this approach to take into account of the two stage estimation of the MCEM algorithm.

Table 7.9: Parameter estimates of the Gaussian copula model with negative binomial regression marginals (7.1) and ARMA(2,1) dependence for the Polio data set.

Panel I: Parametric Gaussian copula model

$\boldsymbol{\theta}$	$\widehat{\boldsymbol{\theta}}_{\text{MCEM}}$		$\widehat{\boldsymbol{\theta}}_{\text{ABC}}$		$\widehat{\boldsymbol{\theta}}_{\text{PCC}}$		$\widehat{\boldsymbol{\theta}}_{\text{Masarotto}}$	
a_1	-0.566	(0.260)	0.057	(0.508)	-0.577	(0.199)	-0.5229	(0.220)
a_2	0.270	(0.099)	0.107	(0.321)	0.293	(0.091)	0.3046	(0.090)
b_1	0.721	(0.272)	-0.128	(0.470)	0.759	(0.203)	0.696	(0.229)
β_0	0.209	(0.096)	0.028	(0.250)	0.210	(0.119)	0.210	(0.121)
β_1	-4.332	(1.895)	-4.719	(0.969)	-4.387	(2.252)	-4.315	(2.284)
β_2	-0.143	(0.129)	-0.021	(0.506)	-0.121	(0.147)	0.122	(0.147)
β_3	-0.503	(0.138)	0.506	(0.505)	-0.494	(0.157)	-0.497	(0.157)
β_4	0.168	(0.131)	0.010	(0.539)	0.192	(0.132)	0.190	(0.129)
β_5	-0.421	(0.132)	-0.193	(0.542)	-0.408	(0.131)	-0.403	(0.128)
κ	0.567	(0.484)	0.608	(0.810)	(κ^{-1}=) 1.770	(0.513)	-0.570	(0.170)
$\widehat{\ell}$	-		-		247.988		247.906	

Panel II: Semi-parametric Gaussian copula model

$\boldsymbol{\theta}$	$\widehat{\boldsymbol{\theta}}_{\text{MCEM}}$		$\widehat{\boldsymbol{\theta}}_{\text{PCC}}$
a_1	-0.505	(0.209)	-0.45
a_2	0.319	(0.089)	0.350
b_1	0.719	(0.217)	0.720
$\widehat{\ell}$	252.642		252.363

Chapter 8

Summary

Copulas allow complex relationships between variables to be specified in a simple manner. Gaussian copula models are a natural choice for integer-valued time series with interpretable parameters. We have tackled a number of challenges in the application of the Gaussian copula model. Under the assumption of an underlying ARMA process we have dealt with maximum likelihood estimation problems, by incorporating classic ideas from time series analysis into modern methods of model fitting, namely EM, ABC and vine copula construction.

A review of current models for integer-valued time series is given in Chapter 2. These models (Markov Chain, MTD, DARMA, INARMA) do not generalise easily for a desired marginal distribution or adapt to inclusion of explanatory variables. The Gaussian copula model with discrete margins is extremely flexible allowing the marginal distributions to be specified, unspecified and with or without explanatory variables.

Chapter 3

In Chapter 3, the Gaussian copula model and the interpretation of the underlying ARMA process is discussed. The conditional distribution of the underlying time series given the observed data was shown to be multivariate truncated normal and the proof is given. The computational cost of likelihood evaluation was discussed and a literature review of current methods for estimation given.

The main focus of this thesis has been on the three methods for parameter inference in Chapters 4, 5 and 6 developed from a range of different areas.

Chapter 4

In Chapter 4, an EM algorithm was considered where the ARMA time series is treated as missing data. A GHK simulator was used to sample directly from the multivariate truncated normal. The R code for a vectorised implementation is given in Appendix C.1. Numerical studies showed this works well for small to moderate N and becomes slow for $N>$ 1000. Standard errors are found, and compared with those for ARMA estimation and found to be as nearly good as.

Chapter 5

In Chapter 5 we gave a likelihood-free simulation approach based on a rejection-acceptance algorithm. A spectral comparison of two integer-valued time series was used for the rejection criterion. The test was based on Priestley and Rao (1969)'s test for non-stationarity and consisted of comparing two spectral density estimates at a fixed set of frequencies with an ANOVA test. The ABC algorithm samples from an approximation to the true posterior and Monte Carlo inference can be used. Simulation studies show the algorithm performs well for large N and post-sampling adjustments improved the estimates further whilst reducing the standard errors.

Chapter 6

In Chapter 6, we write the Gaussian copula model in an R-vine copula representation and use a sequential algorithm for the computation of the log-likelihood. We evaluate the score and Hessian and give analytic solutions for the standard errors. The proposed methodology is illustrated using simulation studies and highlight the numerous advantages of the vine copula approach. The derivatives of the autocorrelations and partial autocorrelations are given and implemented using a modified version of the Levinson Durbin algorithm. Additionally, we report the first and second partial derivatives for the negative binomial cdf. Analytic and numerical evaluations of the score and Hessian are compared and a simulation study is carried out to suggest optimisation routines for joint maximum likelihood.

Chapter 7

In Chapter 7, an empirical study was conducted using Zeger's (1988) polio data set. A selection of models were compared using criteria such as AIC and BIC and the underlying assumption of Gaussian ARMA model checked. The methods were compared with the INAR(p) models. The R-vine representation was shown to outperform in terms of computational time, simplicity for likelihood based model section and the availability of an analytic score function.

The full set of R code related to the topics used in Chapters 4, 5, 6 and 7 can be developed into an R package. A selection of code is given in Appendix C.

8.1 Limitations and Future work

The Gaussian copula model we have studied allows covariates to be specified in the marginal distributions allowing us to study the effects of covariates on the marginal structure. However, the model does not allow us to study the effect of covariates on the dependence structure. This is not an issue for integer-valued time series but for other applications such as spatial statistics or longitudinal data it may be a limitation because other copulas are suitable, such as vine copulas of Archimedean copulas, where covariate functions for the dependence structure can be used.

We have improved on the quality of the parameter estimation of Masarotto et al. (2012) by using the exact likelihood. The methods presented here have performed well in simulation studies and each have their merits and together provide options for parameter estimation. Investigation of the robustness of these methods to real world data is perhaps the most interesting direction for future research.

Further developments of this work could include

- an analytic form of the M-step in the EM algorithm to increase efficiency. It is an open problem as to whether an analytic solution exists. We envisage a state space approach to this.
- We assume stationarity for the latent ARMA model, but a test for stationarity of integer-valued data would be interesting.
- We model the dependence in the observed time series Y_t indirectly through the unobserved Gaussian ARMA series X_t. The autocorrelation of Y_t is not available and this could be part of future work.

Bibliography

K. Aas, C. Czado, A. Frigessi, and H. Bakken. Pair-Copula Constructions of Multiple Dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.

M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables.* Number 55 in Applied Mathematics Series. Courier Corporation, 1964.

S. Aeschbacher, M. A. Beaumont, and A. Futschik. A novel approach for choosing summary statistics in approximate bayesian computation. *Genetics*, 192(3): 1027–1047, 2012.

M. A. Al-Osh and A. A. Alzaid. First-order integer-valued autoregressive (inar (1)) process. *Journal of Time Series Analysis*, 8(3):261–275, 1987.

A. Alzaid and M. Al-Osh. An integer-valued pth-order autoregressive structure (inar (p)) process. *Journal of Applied Probability*, page 314–324, 1990.

D. W. Andrews. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica: Journal of the Econometric Society*, page 817–858, 1991.

J. Ashford and R. Sowden. Multi-variate probit analysis. *Biometrics*, page 535–546, 1970.

A. Azzalini. Logistic regression for autocorrelated data with application to repeated measures. *Biometrika*, 81(4):767–775, 1994.

A. Azzalini and A. Genz. *The R package `mnormt`: The multivariate normal and t distributions (version 1.5-2)*, 2015. URL `http://azzalini.stat.unipd.it/SW/Pkg-mnormt`.

K. Baba, R. Shibata, and M. Sibuya. Partial Correlation and Conditional Correlation as measures of Conditional Independence. *Australian & New Zealand Journal of Statistics*, 46(4):657–664, 2004.

M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

T. Bedford and R. M. Cooke. Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines. *Annals of Mathematics and Artificial Intelligence*, 32(1–4):245–268, 2001. ISSN 1012–2443.

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

M. A. Benjamin, R. A. Rigby, and D. M. Stasinopoulos. Generalized autoregressive moving average models. *Journal of the American Statistical association*, 98(461): 214–223, 2003.

L. A. Bernotas, P. E. Crago, and H. J. Chizeck. A discrete-time model of electrcally stimulated muscle. *Biomedical Engineering, IEEE Transactions on*, 9:829–838, 1986.

M. Blum. Choosing the summary statistics and the acceptance rate in approximate bayesian computation. In Y. Lechevallier and G. Saporta, editors, *Proceedings of COMPSTAT2́010*, page 47–56. Physica-Verlag HD, 2010.

G. E. Box and M. E. Muller. A note on the generation of random normal deviates. *The annals of mathematical statistics*, 29(2):610–611, 1958.

K. Brännäs and P. Johansson. Time series count data regression. *Communications in Statistics - Theory and Methods*, 23(10):2907–2925, 1994.

E. C. Brechmann. Truncated and simplified regular vines and their applications, 2010.

E. C. Brechmann and C. Czado. Risk management with high-dimensional vine copulas: An analysis of the euro stoxx 50. *Statistics & Risk Modeling*, 30(4): 307–342, 2013.

E. C. Brechmann, C. Czado, and K. Aas. Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85, 2012.

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods.* Springer, 2 edition, 1987.

D. Burshtein. An Efficient Algorithm for Calculating the Likelihood and Likelihood Gradient of ARMA Models. *Automatic Control, IEEE Transactions on*, 38(2): 336–340, 1993.

A. C. Cameron and P. K. Trivedi. *Regression analysis of count data*, volume 53. Cambridge University Press, 2013.

H. Carley. Maximum and minimum extensions of finite subcopulas. *Communications in Statistics-Theory and Methods*, 31(12):2151–2166, 2002.

K. S. Chan and J. Ledolter. Monte Carlo EM Estimation for Time Series Models Involving Counts. *Journal of the American Statistical Association*, 90(429): 242–252, 1995.

R. E. Chandler. A spectral method for estimating parameters in rainfall models. *Bernoulli*, page 301–322, 1997.

R. E. Chandler and S. Bate. Inference for clustered data using the independence loglikelihood. *Biometrika*, 94(1):167–183, 2007.

T. J. Chang, M. Kavvas, and J. Delleur. Daily precipitation modeling by discrete autoregressive moving average processes. *Water Resources Research*, 20(5):565–580, 1984.

C. Chatfield. *The analysis of time series: an introduction.* CRC press, 2013.

S. X. Chen and T.-M. Huang. Nonparametric estimation of copula functions for dependence modelling. *Canadian Journal of Statistics*, 35(2):265–282, 2007.

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

F. Chevassu and J.-P. Ortega. Maximum likelihood estimation of state-space models. *Interstat, paper*, 1, 2013.

S. Chib and R. Winkelmann. Markov chain monte carlo analysis of correlated count data. *Journal of Business & Economic Statistics*, 19(4):428–435, 2001.

R. N. Cochran and F. H. Horne. Statistically weighted principal component analysis of rapid scanning wavelength kinetics experiments. *Analytical Chemistry*, 49(6): 846–853, 1977.

R. M. Cooke. Markov and entropy properties of tree-and vine-dependent variables. In *Proceedings of the ASA Section of Bayesian Statistical Science*, volume 27, 1997.

K. Csilléry, M. G. Blum, O. E. Gaggiotti, and O. François. Approximate bayesian computation (abc) in practice. *Trends in ecology & evolution*, 25(7):410–418, 2010.

C. Czado. Pair-copula constructions of multivariate copulas. In *Copula theory and its applications*, page 93–109. Springer, 2010.

C. Czado, E. C. Brechmann, and L. Gruber. Selection of vine copulas. In *Copulae in Mathematical and Quantitative Finance*, page 17–37. Springer, 2013.

P. Danaher and M. Smith. Modeling multivariate distributions using copulas: Applications in marketing. *Marketing Science*, 30:4–21, 2011.

M. J. Daniels and M. Pourahmadi. Modeling covariance matrices via partial autocorrelations. *Journal of Multivariate Analysis*, 100(10):2352–2363, 2009.

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

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

P. Deheuvels. ila fonction de dépendance empirique et ses propriétés. *Un Test non Paramétrique Dhindependance, Bullettin de l'académie Royal de Belgique, Classe des Sciences*, page 274–292, 1979.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

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

J. Dißmann, E. C. Brechmann, C. Czado, and D. Kurowicka. Selecting and estimating regular vine copulae and application to financial returns . *Computational Statistics & Data Analysis*, 59(0):52–69, 2013.

F. C. Drost, R. v. d. Akker, and B. J. Werker. Efficient estimation of auto-regression parameters and innovation distributions for semiparametric integer-valued ar (p) models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):467–485, 2009.

P. K. Dunn and G. K. Smyth. Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5(3):236–244, 1996.

P. Embrechts. Copulas: a personal view. *Journal of Risk and Insurance*, 76(3): 639–650, 2009.

P. Embrechts, R. Frey, and A. McNeil. Quantitative risk management. *Princeton Series in Finance, Princeton*, 10, 2005.

V. Erhardt. *Modeling different dependence structures involving count data with applications to insurance, economics and genetics.* Dissertation, Technische Universität München, München, 2010.

H.-B. Fang, K.-T. Fang, and S. Kotz. The meta-elliptical distributions with given marginals. *Journal of Multivariate Analysis*, 82(1):1–16, 2002.

R. W. Farebrother. Simplified Samuelson Conditions for Cubic and Quartic Equations*. *The Manchester School*, 41(4):396–400, 1973.

P. Fearnhead and D. Prangle. Constructing summary statistics for approximate bayesian computation: semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3): 419–474, 2012.

K. Fokianos. Integer-valued time series. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):361–364, 2009. ISSN 1939-0068.

B. Fornberg and D. M. Sloan. A review of pseudospectral methods for solving partial differential equations. *Acta Numerica*, 3(1):203–267, 1994.

C. Francq and J.-M. Zakoïan. Estimating the marginal law of a time series with applications to heavy-tailed distributions. *Journal of Business & Economic Statistics*, 31(4):412–425, 2013.

R. Frey, A. J. McNeil, and M. Nyfeler. Copulas and credit models. *Risk*, 10 (111114.10), 2001.

B. Friedlander and B. Porat. The modified yule-walker method of arma spectral estimation. *Aerospace and Electronic Systems, IEEE Transactions on*, (2):158–173, 1984.

T. A. B. G. Marsaglia. A convenient method for generating normal variables. *SIAM Review*, 6(3):260–264, 1964. ISSN 00361445.

D. F. Gatz and L. Smith. The standard error of a weighted mean concentration—i. bootstrapping vs other methods. *Atmospheric Environment*, 29(11):1185–1193, 1995.

C. Genest and J. Nešlehová. A primer on copulas for count data. *Astin Bulletin*, 37 (02):475–515, 2007.

C. Genest and L. Rivest. Statistical Inference Procedures for Bivariate Archimedean Copulas. *Journal of the American Statistical Association*, 88(423):1034–1043, September 1993.

C. Genest, K. Ghoudi, and L.-P. Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82 (3):543–552, 1995.

A. Genz. Numerical computation of multivariate normal probabilities. *Journal of computational and graphical statistics*, 1(2):141–149, 1992.

A. Genz. *pbivnorm: Vectorized Bivariate Normal CDF (version 0.6.0)*, 2015. URL `http://CRAN.R-project.org/package=pbivnorm`. Fortran code by Alan Genz. R code by Brenton Kenkel and based on Adelchi Azzalini's 'mnormt' package.

J. Geweke. Efficient Simulation from the Multivariate Normal and Student-t Distributions Subject to Linear Constraints and the Evaluation of Constraint Probabilities. In *Computing science and statistics: Proceedings of the 23rd symposium on the interface*, page 571–578. Citeseer, 1991.

P. Gilbert and R. Varadhan. *numDeriv: Accurate Numerical Derivatives*, 2015. URL `https://CRAN.R-project.org/package=numDeriv`. R package version 2014.2-1.

C. Grazian and B. Liseo. Approximate Bayesian Computation for Copula Estimation. *arXiv:1503.02912 [stat]*, March 2015. URL `http://arxiv.org/abs/1503.02912`. arXiv: 1503.02912.

W. H. Greene. Accounting for excess zeros and sample selection in poisson and negative binomial regression models. 1994.

W. H. Greene. Econometric analysis, 5th. *Ed.. Upper Saddle River, NJ*, 2003.

I. H. Haff et al. Parameter estimation for pair-copula constructions. *Bernoulli*, 19 (2):462–491, 2013.

A. Heinen. Modelling time series count data: an autoregressive conditional poisson model. CORE Discussion Papers 2003062, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2003.

H. Herwartz, N. Klein, and C. Strumann. Modelling hospital admission and length of stay by means of generalised count data models. *Journal of Applied Econometrics*, pages n/a–n/a, 2015. ISSN 1099-1255.

D. P. Heyman and T. Lakshman. Source models for vbr broadcast-video traffic. *IEEE/ACM Transactions on Networking (TON)*, 4(1):40–48, 1996.

D. P. Heyman, A. Tabatabai, and T. Lakshman. Statistical analysis and simulation study of video teleconference traffic in atm networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 2(1):49–59, 1992.

M. Hofert, I. Kojadinovic, M. Maechler, and J. Yan. *copula: Multivariate Dependence with Copulas*, 2015. URL `http://CRAN.R-project.org/package=copula`. R package version 0.999-13.

P. D. Hoff. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics*, page 265–283, 2007.

P. A. Jacobs and P. A. Lewis. Discrete time series generated by mixtures. i: Correlational and runs properties. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 94–105, 1978a.

P. A. Jacobs and P. A. Lewis. Discrete time series generated by mixtures ii: asymptotic properties. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 222–228, 1978b.

P. A. Jacobs and P. A. Lewis. Discrete time series generated by mixtures iii: Autoregressive processes (dar (p)). Technical report, Monterey, California. Naval Postgraduate School, 1978c.

P. A. Jacobs and P. A. Lewis. Stationary discrete autoregressive-moving average time series generated by mixtures. *Journal of Time Series Analysis*, 4(1):19–36, 1983.

G. M. Jenkins. General considerations in the analysis of spectra. *Technometrics*, 3 (2):133–166, 1961.

D. Jin-Guan and L. Yuan. The integer-valued autoregressice (inar(p)) model. *Journal of time series analysis*, 12(2):129–142, 1991.

H. Joe. Families of m-Variate Distributions with Given Margins and m(m-1)/2 Bivariate Dependence Parameters. *Lecture Notes-Monograph Series*, 28:120–141, 1996.

H. Joe. *Multivariate models and multivariate dependence concepts.* CRC Press, 1997.

H. Joe. Asymptotic efficiency of the two-stage estimation method for copula-based models. *Journal of Multivariate Analysis*, 94(2):401–419, 2005.

H. Joe. *Dependence Modeling with Copulas. No. 134 in monographs on statistics and applied probability.* CRC Press, 2015.

N. L. Johnson and S. Kotz. *Discrete distributions: Distributions in statistics.* Houghton Mifflin, 1969.

N. L. Johnson, A. W. Kemp, and S. Kotz. *Univariate Discrete Distributions*, volume 444. John Wiley & Sons, 2005.

K. Jonasson and S. E. Ferrando. Evaluating exact VARMA likelihood and its gradient when data are incomplete. *ACM Transactions on Mathematical Software (TOMS)*, 35(1):5, 2008.

M. C. Jones. Randomly choosing parameters from the stationarity and invertibility region of autoregressive-moving average models. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 36(2):134–138, 1987.

R. H. Jones. Maximum likelihood fitting of arma models to time series with missing observations. *Technometrics*, 22(3):389–395, 1980.

B. Jørgensen, S. Lundbye-Christensen, P.-K. Song, and L. Sun. A state space model for multivariate longitudinal count data. *Biometrika*, 86(1):169–181, 1999.

D. Karlis and I. Ntzoufras. Analysis of sports data by using bivariate poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(3): 381–393, 2003. ISSN 1467-9884.

D. Karlis and E. Xekalaki. Maximum likelihood estimation for integer valued time series models. In *Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications*, page 778–783, 2001.

B. Kedem and K. Fokianos. *Regression models for time series analysis*, volume 488. John Wiley & Sons, 2005.

C. T. Kelley. *Iterative methods for optimization*, volume 18. Siam, 1999.

M. Kendall and A. Stuart. Vol. 3: Design and analysis, and time-series. 1966.

G. Kimeldorf and A. Sampson. Uniform Representations of Bivariate Distributions. *Communications in Statistics-simulation and Computation*, 4:617–627, 1975.

R. Kohn and C. F. Ansley. Computing the likelihood and its dierivatives for a gaussian arma model. *Journal of Statistical Computation and Simulation*, 22 (3–4):229–263, 1985.

D. Kurowicka and R. Cooke. Completion problem with partial correlation vines. *Linear algebra and its applications*, 418(1):188–220, 2006a.

D. Kurowicka and R. M. Cooke. Advanced input modeling: the vine copula method for representing high dimensional dependent distributions: application to continuous belief nets. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, page 270–278. Winter Simulation Conference, 2002.

D. Kurowicka and R. M. Cooke. A parameterization of positive definite matrices in terms of partial correlation vines . *Linear Algebra and its Applications*, 372(0): 225–251, 2003.

D. Kurowicka and R. M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling.* John Wiley & Sons, 2006b.

D. Kurowicka and H. Joe, editors. *DEPENDENCE MODELING:Vine Copula Handbook.* World Scientific Publishing Co. Pte. Ltd., 2010.

K. Lange. A quasi-newton acceleration of the em algorithm. *Statistica sinica*, 5(1): 1–18, 1995.

A. Latour. The multivariate ginar (p) process. *Advances in Applied Probability*, page 228–248, 1997.

A. Latour. Existence and stochastic structure of a non-negative integer-valued autoregressive process. *Journal of Time Series Analysis*, 19(4):439–455, 1998. ISSN 1467-9892.

C. Leuenberger and D. Wegmann. Bayesian computation and model selection without likelihoods. *Genetics*, 184(1):243–252, 2010.

P. A. Lewis. Simple models for positive-valued and discrete-valued time series with arma correlation structure. *In Multivariate analysis: North Holland, Amsterdam*, page 151–166, 1980.

W. K. Li. Time Series Models Based on Generalized Linear Models: Some Further Results. *Biometrics*, 50(2):506–511, 1994.

T. A. Louis. Finding the Observed Iinformation matrix when using the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 226–233, 1982.

I. L. MacDonald and W. Zucchini. *Hidden Markov and other models for discrete-valued time series*, volume 110. CRC Press, 1997.

J.-F. Mai and M. Scherer. *Simulating copulas: stochastic models, sampling algorithms, and applications*, volume 4 of *Series in Quantitative Finance.* World Scientific, Singapore, June 2012.

J.-M. Marin, N. S. Pillai, C. P. Robert, and J. Rousseau. Relevant statistics for bayesian model choice. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(5):833–859, 2014.

P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.

G. Masarotto and C. Varin. Gaussian copula marginal regression. *Electronic Journal of Statistics*, 6:1517–549, 2012. URL `http://projecteuclid.org/euclid.ejs/1346421603`.

G. Masarotto, C. Varin, et al. Gaussian Copula Marginal Regression. *Electronic Journal of Statistics*, 6:15171549, 2012.

B. P. M. McCabe, G. M. Martin, and D. Harris. Efficient probabilistic forecasts for counts. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):253–272, 2011. ISSN 1467–9868.

P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

E. McKenzie. Some simple models for discrete variate time series1. *JAWRA Journal of the American Water Resources Association*, 21(4):645–650, 1985. ISSN 1752–1688.

E. McKenzie. Autoregressive moving-average processes with negative-binomial and geometric marginal distributions. *Advances in Applied Probability*, page 679–705, 1986.

E. McKenzie. Some arma models for dependent sequences of poisson counts. *Advances in Applied Probability*, page 822–835, 1988.

E. McKenzie. Ch 16 discrete variate time series. *Handbook of statistics*, 21:573–606, 2003.

G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*, volume 382. John Wiley & Sons, 2007.

T. Mikosch. Copulas: Tales and facts. *Extremes*, 9(1):3–20, 2006.

A. Min and C. Czado. Bayesian inference for multivariate copulas using pair-copula constructions. *Journal of Financial Econometrics*, 8(4):511–546, 2010.

J. Moller and R. P. Waagepetersen. *Statistical inference and simulation for spatial point processes*, volume 100. Chapman & Hall/CRC, 2003.

J. Mullahy. Specification and testing of some modified count data models. *Journal of Econometrics*, 33(3):341–365, 1986.

P. Neal and T. Kypraios. Exact bayesian inference via data augmentation. *Statistics and Computing*, 25(2):333–347, 2015. ISSN 0960-3174.

P. Neal and T. Subba Rao. Mcmc for integer-valued arma processes. *Journal of Time Series Analysis*, 28(1):92–110, 2007.

J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

R. B. Nelsen. *An introduction to copulas*, volume 139. Springer Science & Business Media, 1999.

A. K. Nikoloulopoulos. Copula-based models for multivariate discrete response data. In *Copulae in Mathematical and Quantitative Finance*, page 231–249. Lecture Notes in Statistics, Springer, 2013.

A. K. Nikoloulopoulos and D. Karlis. Modeling multivariate count data using copulas. *Communications in Statistics-Simulation and Computation*, 39(1):172–187, 2009.

D. J. Nott, Y. Fan, L. Marshall, and S. Sisson. Approximate bayesian computation and bayes' linear analysis: Toward high-dimensional abc. *Journal of Computational and Graphical Statistics*, 23(1):65–86, 2014.

M. A. Nunes and D. J. Balding. On optimal selection of summary statistics for approximate bayesian computation. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

A. Panagiotelis, C. Czado, and H. Joe. Pair Copula Constructions for Multivariate Discrete Data. *Journal of the American Statistical Association*, 107(499): 1063–1072, 2012.

M. Park, W. Jitkrittum, and D. Sejdinovic. K2-ABC: Approximate Bayesian Computation with Infinite Dimensional Summary Statistics via Kernel Embeddings. *ArXiv e-prints*, Feb. 2015.

A. J. Patton. Modelling asymmetric exchange rate dependence*. *International economic review*, 47(2):527–556, 2006.

M. Pitt, D. Chan, and R. Kohn. Efficient bayesian inference for gaussian copula regression models. *Biometrika*, 93(3):537–554, 2006.

R. L. Plackett. A Reduction Formula for Normal Multivariate Integrals. *Biometrika*, 41(3/4):351–360, 1954.

V. Pless. *Introduction to the theory of error-correcting codes*, volume 48. John Wiley & Sons, 2011.

B. Porat. Arma spectral estimation based on partial autocorrelations. *Circuits, Systems and Signal Processing*, 2(3):341–360, 1983.

B. Porat and B. Friedlander. Computation of the exact information matrix of gaussian time series with stationary random components. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(1):118–130, 1986.

D. Prangle. *Summary statistics and sequential methods for approximate Bayesian computation.* PhD thesis, Lancaster University, 2011.

D. Prangle. Lazy abc. *Statistics and Computing*, page 1–15, 2014.

M. Priestley and T. S. Rao. A test for non-stationarity of time-series. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 140–149, 1969.

M. B. Priestley. *Spectral Analysis and Time Series.* Academic Press, 1981.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`.

A. E. Raftery. A model for high-order Markov chains. *Journal of the Royal Statistical Society. Series B (Methodological)*, page 528–539, 1985.

B. Renard and M. Lang. Use of a gaussian copula for multivariate extreme value analysis: some case studies in hydrology. *Advances in Water Resources*, 30(4): 897–912, 2007.

P. R. Rider. The negative binomial distribution and the incomplete beta function. *The American Mathematical Monthly*, 69(4):302–304, 1962.

D. M. Ritchie, B. W. Kernighan, and M. E. Lesk. *The C programming language.* Bell Laboratories, 1975.

M. Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, page 470–472, 1952.

U. Schepsmeier and J. Stöber. Derivatives and Fisher information of bivariate copulas. *Statistical Papers*, 55(2):525–542, 2014.

P. Shi and E. A. Valdez. Longitudinal modeling of insurance claim counts using jitters. *Scandinavian Actuarial Journal*, 2014(2):159–179, 2014.

M. S. Smith. Bayesian approaches to copula modelling. *Available at SSRN 1974297*, 2011.

M. S. Smith and M. A. Khaled. Estimation of copula models with discrete margins via bayesian data augmentation. *Journal of the American Statistical Association*, 107(497):290–303, 2012.

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

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

M. Stasinopoulos and B. Rigby. *gamlss.data: GAMLSS Data*, 2015. URL `http://CRAN.R-project.org/package=gamlss.data`. R package version 4.3-0.

F. Steutel and K. Van Harn. Discrete analogues of self-decomposability and stability. *The Annals of Probability*, page 893–899, 1979.

J. Stöber. *Regular Vine Copulas with the simplifying assumption, time-variation, and mixed discrete and continuous margins.* Phd thesis, Technische Universität München, München, 2013.

J. Stöber and C. Czado. Detecting regime switches in the dependence structure of high dimensional financial data. *arXiv preprint arXiv:1202.2009*, 2012.

J. Stöber and U. Schepsmeier. Estimating standard errors in regular vine copula models. *Computational Statistics*, 28(6):2679–2707, 2013.

J. Stöber, H. G. Hong, C. Czado, and P. Ghosh. Comorbidity of chronic diseases in the elderly: Patterns identified by a copula design for mixed responses. *Computational Statistics & Data Analysis*, 88:28–39, 2015.

J. Stoehr, P. Pudlo, and L. Cucala. Adaptive abc model choice and geometric summary statistics for hidden gibbs random fields. *Statistics and Computing*, 25 (1):129–141, 2014.

P. Stoica and R. L. Moses. *Introduction to spectral analysis*, volume 1. Prentice hall Upper Saddle River, 1997.

G. M. Tallis. The moment generating function of the truncated multi-normal distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23 (1):223–229, 1961. ISSN 00359246.

S. Tavaré, D. J. Balding, R. C. Griffiths, and P. Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

Y. L. Tong. *Multivariate Normal Distribution.* Springer-Verlag, 1990.

P. K. Trivedi and D. M. Zimmer. *Copula modeling: an introduction for practitioners.* Now Publishers Inc, 2007.

H. Tsukahara. Semiparametric estimation in copula models. *Canadian Journal of Statistics*, 33(3):357–375, 2005.

A. Valdesgo. *Multivariate volatility models using copulas.* PhD thesis, Center for Operations Research and Econometrics, Université Catholique de Louvain, Belgium, 2009.

R. Varadhan. *alabama: Constrained Nonlinear Optimization*, 2014. URL `http://CRAN.R-project.org/package=alabama`. R package version 2014.12-1.

R. Varadhan, J. H. University, H. W. Borchers, and A. C. Research. *dfoptim: Derivative-free Optimization*, 2011. URL `http://CRAN.R-project.org/package=dfoptim`. R package version 2011.8-1.

C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, page 5–42, 2011.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S.* Springer, New York, fourth edition, 2002.

G. Venter. Tails of copulas. In *In Proceedings ASTIN Washington, USA*, volume 27, page 68–113, 2001.

B. A. Walther and J. L. Moore. The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature review of estimator performance. *Ecography*, 28(6):815–829, 2005.

H. Wang, E. M. Iglesias, and J. M. Wooldridge. Partial Maximum Likelihood Estimation of Spatial Probit Models. *Journal of Econometrics*, 2012.

D. Wegmann, C. Leuenberger, S. Neuenschwander, and L. Excoffier. Abctoolbox: a versatile toolkit for approximate bayesian computations. *BMC Bioinformatics*, 11(1):1–7, 2010.

G. C. Wei and M. A. Tanner. A monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

C. H. Weiß. Generalized choice models for categorical time series. *Journal of Statistical Planning and Inference*, 141(8):2849–2862, 2011.

M. J. Wichura. Algorithm as 241: The percentage points of the normal distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 37(3): 477–484, 1988.

S. Wilhelm and B. Manjunath. tmvtnorm: A package for the truncated multivariate normal distribution. *sigma*, 2:2, 2010.

S. Wilhelm and G. Manjunath, B. *tmvtnorm: Truncated Multivariate Normal and Student t Distribution*, 2014. URL `http://CRAN.R-project.org/package=tmvtnorm`. R package version 1.4-9.

B. Wu. *Contributions to Copula Modeling of Mixed Discrete-Continuous Outcomes.* PhD thesis, University of Calgary, Alberta, 2013.

C. F. J. Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

J. J. Xu. *Statistical modelling and inference for multivariate and longitudinal discrete response data.* PhD thesis, University of British Columbia, 1996.

Q. Yao and P. J. Brockwell. Gaussian Maximum Likelihood Estimation For ARMA Models. I: Time Series. *Journal of Time Series Analysis*, 27(6):857–875, 2006.

G. U. Yule. On the theory of correlation for any number of variables, treated by a new system of notation. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 79(529):182–193, 1907.

G. U. Yule and M. Kendall. An introduction to the theory of statistics charles griffin & co. *Belmont, CA*, 1965.

S. L. Zeger. A regression model for time series of counts. *Biometrika*, 75(4):621–629, 1988.

S. L. Zeger and B. Qaqish. Markov regression models for time series: a quasi-likelihood approach. *Biometrics*, page 1019–1031, 1988.

Y. Zhao and H. Joe. Composite likelihood estimation in multivariate data analysis. *Canadian Journal of Statistics*, 33(3):335–356, 2005.

D. M. Zimmer and P. K. Trivedi. Using trivariate copulas to model sample selection and treatment effects: application to family health care demand. *Journal of Business & Economic Statistics*, 24(1):63–76, 2006.

Appendix A

Derivatives of σ_ε^2 in the ARMA(2,2) case

$$
\begin{aligned}
\frac{\partial \sigma_\varepsilon^2(\mathbf{a},\mathbf{b})}{\partial a_1} &= \frac{2a_1\,(a_2+1)}{b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1} \\
&\quad + \frac{((a_2-1)^{\,2}-a_1^2)\,(a_2+1)\,(-4b_2a_1-2b_1\,(b_2+1))}{(b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1)^{\,2}},
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \sigma_\varepsilon^2(\mathbf{a},\mathbf{b})}{\partial a_2} &= -\frac{(a_2-1)^{\,2}-a_1^2}{b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1} \\
&\quad - \frac{2\,(a_2-1)\,(a_2+1)}{b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1} \\
&\quad + \frac{(a_2+1)\,((a_2-1)^{\,2}-a_1^2)\,(b_2\,(b_2+4a_2-2)+b_1^2+1)}{(b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1)^{\,2}},
\end{aligned}
$$

$$
\frac{\partial \sigma_\varepsilon^2(\mathbf{a},\mathbf{b})}{\partial b_1} = \frac{((a_2-1)^{\,2}-a_1^2)\,(a_2+1)\,(2b_1\,(a_2-1)-2\,(b_2+1)\,a_1)}{(b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1)^{\,2}},
$$

$$
\frac{\partial \sigma_\varepsilon^2(\mathbf{a},\mathbf{b})}{\partial b_2} = \frac{((a_2-1)^{\,2}-a_1^2)\,(a_2+1)\,(-2b_1a_1+2b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))}{(b_1^2\,(a_2-1)-2\,(b_2+1)\,b_1a_1+b_2\,(b_2\,(a_2-1)+2a_2^2-2\,(a_1^2+a_2))+a_2-1)^{\,2}}.
$$

Appendix B

Score and Hessian of Discrete D-vine Log-likelihood

Algorithm 6 Score function for the D-vine log-likelihood for discrete variables.

Input: Model Parameters $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, b_q, \boldsymbol{\theta}_{\text{marginal}})$ and integer-valued time series of length N

Output: The first derivative of the discrete D-vine Gaussian PCC log-likelihood with respect to $\boldsymbol{\theta}$.

Require: Roots of $\alpha(z) = 1 - a_1 z - \cdots - a_p z^p$ and $\beta(z) = 1 + b_1 z + \cdots + b_q z^q$ to lie outside the unit circle.

```
Create $N \times N$ matrices $v^{\text{direct}+}, v^{\text{direct}-}, v^{\text{indirect}+}, v^{\text{indirect}-}, f^{\text{direct}}, f^{\text{indirect}}, S1^{\text{direct}+},$
        $S1^{\text{direct}-}, S1^{\text{indirect}+}$ and $S1^{\text{indirect}-}$ as in (6.26).
Set $M$ and $\widetilde{M}$ $N \times N$ matrices as in (6.13).
$\rho_{kk}$ = Levinson-Durbin($\mathbf{a}, \mathbf{b}$) into $P^*$ matrix as in (6.27)    ▷ Copula parameters
$\partial_{\theta_i} \rho_{kk}$ into $P'^*$ vector of length $T$ and $k = 1, \ldots, N - 1$.

Set $(u_1^+, \ldots, u_N^+) = (F(y_1), \ldots, F(y_N))$,
        and $(u_1^-, \ldots, u_N^-) = (F(y_1 - 1), \ldots, F(y_N - 1))$.
Store $(v_{N,1}^{\text{direct+}}, \ldots, v_{N,N}^{\text{direct+}}) = (u_N^+, \ldots, u_1^+)$,
        and $(v_{N,1}^{\text{direct−}}, \ldots, v_{N,N}^{\text{direct−}}) = (u_N^-, \ldots, u_1^-)$,
        and $(f_{N,1}^{\text{direct−}}, \ldots, f_{N,N}^{\text{direct−}}) = (u_N^+, \ldots, u_1^+) - (u_N^-, \ldots, u_1^-)$ as (6.25)
```

```
Using (6.37), $(s1^{\text{direct}}_{N,1,1,\theta_i},\dots,s1^{\text{direct}}_{N,N,1,\theta_i}) = (F'_{\theta_i}(y_N),\dots,F'_{\theta_i}(y_1))$,
        and $(s1^{\text{direct}}_{N,1,2,\theta_i},\dots,s1^{\text{direct}}_{N,N,2,\theta_i}) = (F'_{\theta_i}(y_N-1),\dots,F'_{\theta_i}(y_1-1))$,
$(f'^{\text{direct}}_{N,1,\theta_i},\dots,f'^{\text{direct}}_{N,N,\theta_i}) = (F'_{\theta_i}(y_N),\dots,F'_{\theta_i}(y_1)) - (F'_{\theta_i}(y_N-1),\dots,F'_{\theta_i}(y_1-1))$.

for $k = N-1,\dots 1$
    for $i = N,\dots k+1$

        $(z_1^+, z_1^-) = (v^{\text{direct}+}_{i,k}, v^{\text{direct}-}_{i,k})$.          ▷ Select copula arguments
        $(z^{+'}_{1,\theta_i}, z^{-'}_{1,\theta_i}) = (s1^{\text{direct}}_{i,k,\theta_i}{}^{+}, s1^{\text{direct}}_{i,k,\theta_i}{}^{-})$ for $i = 1,\dots,T$.
        if $\widetilde{m}_{i,k} = m_{i,k}$ then $(z_2^+, z_2^-) = (v^{\text{direct}+}_{k,(N+1-\widetilde{m}_{i,k})}, v^{\text{direct}-}_{k,(N+1-\widetilde{m}_{i,k})})$.
                    $(z^{+'}_{2,\theta_i}, z^{-'}_{2,\theta_i}) = (s1^{\text{indirect}}_{k,(N+1-\widetilde{m}_{i,k})}{}^{+}, s1^{\text{indirect}}_{k,(N+1-\widetilde{m}_{i,k})}{}^{-})$
        else        $(z_2^+, z_2^-) = (v^{\text{indirect}+}_{k,(N+1-\widetilde{m}_{i,k})}, v^{\text{indirect}-}_{k,(N+1-\widetilde{m}_{i,k})})$.
                    $(z^{+'}_{2,\theta_i}, z^{-'}_{2,\theta_i}) = (s1^{\text{indirect}}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i}{}^{+}, s1^{\text{indirect}}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i}{}^{-})$

        for $ab = \{+,-\}$          ▷ Evaluate copulas
            Set $(C^{00}, C^{01}, C^{10}, C^{11}) = \Phi\left(\Phi^{-1}(z_1^a), \Phi^{-1}(z_2^b); p_{i,k}\right)$.
            Set $\nabla C_{i,k} = C^{00} - C^{01} - C^{10} + C^{11}$.
            Set $d_1^{00,01,10,11} = \phi\left(\Phi^{-1}(z_1^a), \Phi^{-1}(z_2^b); p_{i,k}\right)$.
            Set $d_2^{00,01,10,11} = \Phi\left(\frac{\Phi^{-1}(z_2^a) - \theta_{i,k}\Phi^{-1}(z_1^a)}{\sqrt{1-p_{i,k}^2}}\right)$.
            Set $d_3^{00,01,10,11} = \Phi\left(\frac{\Phi^{-1}(z_1^a) - \theta_{i,k}\Phi^{-1}(z_2^a)}{\sqrt{1-p_{i,k}^2}}\right)$.
            Set $\mathbf{C}' = d_1 \cdot \rho'_{N-i+1,N-i+1,\theta_i} + d_2 \cdot z'^{a}_{1,\theta_i} + d_3 \cdot z'^{b}_{2,\theta_i}$.
            Set $\nabla C'_{i,k,\theta_i} = C^{00'}_{\theta_i} - C^{01'}_{\theta_i} - C^{10'}_{\theta_i} + C^{11'}_{\theta_i}$.
        end
                                                      ▷ Conditional distributions
        $v^{\text{direct}+-}_{i-1,k} = \frac{C(z_1, z_2; t_{i,k}, p_{i,k}) - C(z_1, z_2^-; t_{i,k}, p_{i,k})}{f(z_2)}$ and
        $v^{\text{indirect}+-}_{i-1,k} = \frac{C(z_1, z_2; t_{i,k}, p_{i,k}) - C(z_1^-, z_2; t_{i,k}, p_{i,k})}{f(z_1)}$ as in (6.16).
        Set $f^{\text{direct}}_{i-1,k} = v^{\text{direct}+}_{i-1,k} - v^{\text{direct}-}_{i-1,k}$ and $f^{\text{indirect}}_{i-1,k} = v^{\text{indirect}+}_{i-1,k} - v^{\text{indirect}-}_{i-1,k}$.
        $v^{\text{direct}+-}_{i-1,k} = H(z_1, z_2 \mid \mathbf{C})$ and $v^{\text{indirect}+-}_{i-1,k} = H(z_1, z_2 \mid \mathbf{C})$
        $s1^{\text{direct}+-}_{i-1,k,1:T} = H'_{\theta_i}(z'_1, z'_2 \mid \mathbf{C}, \mathbf{C}')$ and $s1^{\text{indirect}+-}_{i-1,k,1:2,1:T} = H'_{\theta_i}(z'_2, z'_1 \mid \mathbf{C}, \mathbf{C}')$
        $f^{\text{direct}}_{i-1,k} = v^{\text{direct}+}_{i-1,k,1:2,1:T} - v^{\text{direct}-}_{i-1,k}$ and $f^{\text{indirect}}_{i-1,k} = v^{\text{indirect}+}_{i-1,k} - v^{\text{indirect}-}_{i-1,k}$
        $f^{\text{direct}}_{i-1,k,\theta_i}{}' = s1^{\text{direct}+}_{i-1,k} - s1^{\text{direct}-}_{i-1,k}$ and $f^{\text{indirect}}_{i-1,k,\theta_i}{}' = s1^{\text{indirect}+}_{i-1,k} - s1^{\text{indirect}-}_{i-1,k}$

        if $\nabla C$ expressed in log-likelihood then save $c_j = \nabla C$, and $c'_j = \nabla C'$.
end end
Set $\ell'_{\theta_i} = \begin{cases} \sum_{k=1}^{n} \frac{\nabla C'_{k\ N-k+1|k+1:N-k-1,\ \theta_i}}{\nabla C_{k\ N-k+1|k+1:N-k-1}} & \text{if } N = 2n \text{ i.e. } N \text{ is even,} \\ \sum_{k=1}^{n} \frac{\nabla C'_{k\ N-k+1|k+1:N-k-1,\ \theta_i}}{\nabla C_{k\ N-k+1|k+1:N-k-1}} + \frac{f'_{n,\theta_i}(y_n)}{f_n(y_n)} & \text{if } N = 2n-1 \text{ i.e. } N \text{ is odd.} \end{cases}$.
Return $\ell'_{\theta_i}$
```

Algorithm 7 Hessian for the D-vine log-likelihood for discrete variables.

Input: Model Parameters $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, b_q, \boldsymbol{\theta}_{\text{marginal}})$ and integer-valued time series of length N

Output: The Hessian of the discrete D-vine Gaussian PCC log-likelihood with respect to the parameters $\boldsymbol{\theta} = (a_1, \ldots, a_p, b_1, \ldots, \boldsymbol{\theta}_{\text{marginal}})$.

There are $p + q + \dim(\boldsymbol{\theta}_{\text{marginal}}) := T$ parameters in the log-likelihood and therefore there are $T(T+1)/2 := T^*$ second mixed derivatives. We define an array of dimension $(2, T)$ of $N \times N$ matrices each for $S1^{\text{direct}}$ and $S1^{\text{indirect}}$, and an array of $(2, T^*)$ of $N \times N$ matrices each for $S2^{\text{direct}}$ and $S2^{\text{indirect}}$, to store the derivatives relating to the matrices $v^{\text{direct}\,+}$,$v^{\text{direct}\,-}$, $v^{\text{indirect}\,+}$ and $v^{\text{indirect}\,-}$.

```
Create $N \times N$ matrices $v^{\text{direct}+}, v^{\text{direct}-}, v^{\text{indirect}+}, v^{\text{indirect}-}, f^{\text{direct}}, f^{\text{indirect}}, S1^{\text{direct}+}$,
    $S1^{\text{direct}-}$, $S1^{\text{indirect}+}$ and $S1^{\text{indirect}-}$ as in (6.26).
Set $M$ and $\widetilde{M}$ $N \times N$ matrices as in (6.13).
$\rho_{kk}$ = Levinson-Durbin($\mathbf{a}, \mathbf{b}$) into $P^*$ matrix as in (6.27)   ▷ Copula parameters
$\partial_{\theta_i} \rho_{kk}$ into $P'^*$ vector of length $T$ and $k = 1, \ldots, N-1$.
$\partial^2_{\theta_i \theta_j} \rho_{kk}$ into $P^*\ast$ $i, j = 1, \ldots, T$, $i < j$, and $k = 1, \ldots, N-1$.

Set $(u_1^+, \ldots, u_N^+) = (F(y_1), \ldots, F(y_N))$,
    and $(u_1^-, \ldots, u_N^-) = (F(y_1 - 1), \ldots, F(y_N - 1))$.
Store $(v_{N,1}^{\text{direct}+}, \ldots, v_{N,N}^{\text{direct}+}) = (u_N^+, \ldots, u_1^+)$,
    and $(v_{N,1}^{\text{direct}-}, \ldots, v_{N,N}^{\text{direct}-}) = (u_N^-, \ldots, u_1^-)$,
    and $(f_{N,1}^{\text{direct}-}, \ldots, f_{N,N}^{\text{direct}-}) = (u_N^+, \ldots, u_1^+) - (u_N^-, \ldots, u_1^-)$ as (6.25)

Using (6.37), $(s1^{\text{direct}}_{N,1,1,\theta_i}, \ldots, s1^{\text{direct}}_{N,N,1,\theta_i}) = (F'_{\theta_i}(y_N), \ldots, F'_{\theta_i}(y_1))$,
    and $(s1^{\text{direct}}_{N,1,2,\theta_i}, \ldots, s1^{\text{direct}}_{N,N,2,\theta_i}) = (F'_{\theta_i}(y_N - 1), \ldots, F'_{\theta_i}(y_1 - 1))$,
Using (6.38), Set $(s2^{\text{direct}}_{N,1,1,\theta_i\theta_j}, \ldots, s2^{\text{direct}}_{N,N,1,\theta_i\theta_j}) = (\partial^2_{\theta_i\theta_j} u_N^+, \ldots, \partial^2_{\theta_i\theta_j} u_1^+)$,
    and $(s2^{\text{direct}}_{N,1,2,\theta_i\theta_j}, \ldots, s2^{\text{direct}}_{N,N,2,\theta_i\theta_j}) = (\partial^2_{\theta_i\theta_j} u_N^-, \ldots, \partial^2_{\theta_i\theta_j} u_1^-)$.
$(f'^{\text{direct}}_{N,1,\theta_i}, \ldots, f'^{\text{direct}}_{N,N,\theta_i}) = (F'_{\theta_i}(y_N), \ldots, F'_{\theta_i}(y_1)) - (F'_{\theta_i}(y_N - 1), \ldots, F'_{\theta_i}(y_1 - 1))$.
$(f''^{\text{direct}}_{N,1,\theta_i\theta_j}, \ldots, f''^{\text{direct}}_{N,N,\theta_i\theta_j}) = (F''_{\theta_i\theta_j}(y_N), \ldots, F''_{\theta_i\theta_j}(y_1)), -(F''_{\theta_i\theta_j}(y_N^-), \ldots, F''_{\theta_i\theta_j}(y_1^-))$.

for $k = N - 1, \ldots 1$
    for $i = N, \ldots k + 1$

        $(z_1^+, z_1^-) = (v_{i,k}^{\text{direct}+}, v_{i,k}^{\text{direct}-})$.   ▷ Select copula arguments
        $(z^{+'}_{1,\theta_i}, z^{-'}_{1,\theta_i}) = (s1^{\text{direct}\,+}_{i,k,\theta_i}, s1^{\text{direct}\,-}_{i,k,\theta_i})$ for $i = 1, \ldots, T$.
        $(z^{+''}_{1,\theta_i\theta_j}, z^{-''}_{1,\theta_i\theta_j}) = (s2^{\text{direct}\,+}_{i,k,\theta_i\theta_j}, s2^{\text{direct}\,-}_{i,k,\theta_i\theta_j})$ for $i, j = 1, \ldots, T$.
        if $\widetilde{m}_{i,k} = m_{i,k}$ then $(z_2^+, z_2^-) = (v^{\text{direct}+}_{k,(N+1-\widetilde{m}_{i,k})}, v^{\text{direct}-}_{k,(N+1-\widetilde{m}_{i,k})})$.
                $(z^{+'}_{2,\theta_i}, z^{-'}_{2,\theta_i}) = (s1^{\text{indirect}\,+}_{k,(N+1-\widetilde{m}_{i,k})}, s1^{\text{indirect}\,-}_{k,(N+1-\widetilde{m}_{i,k})})$
        else
                $(z_2^+, z_2^-) = (v^{\text{indirect}+}_{k,(N+1-\widetilde{m}_{i,k})}, v^{\text{indirect}-}_{k,(N+1-\widetilde{m}_{i,k})})$.
                $(z^{+'}_{2,\theta_i}, z^{-'}_{2,\theta_i}) = (s1^{\text{indirect}\,+}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i}, s1^{\text{indirect}\,-}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i})$
                $(z^{+''}_{2,\theta_i\theta_j}, z^{-''}_{2,\theta_i\theta_j}) = (s2^{\text{indirect}+}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i\theta_j}, s2^{\text{indirect}-}_{k,(N+1-\widetilde{m}_{i,k}),\theta_i\theta_j})$.
        for $ab = \{+, -\}$ do                                                    ▷ Evaluate copulas
            Set $(C^{00}, C^{01}, C^{10}, C^{11}) = \Phi\left(\Phi^{-1}\left(z_1^a\right), \Phi^{-1}(z_2^b); p_{i,k}\right)$.
            Set $\nabla C_{i,k} = C^{00} - C^{01} - C^{10} + C^{11}$.
            Set $d_1^{00,01,10,11} = \phi\left(\Phi^{-1}\left(z_1^a\right), \Phi^{-1}(z_2^b); p_{i,k}\right)$.
            Set $d_2^{00,01,10,11} = \Phi\left(\frac{\Phi^{-1}\left(z_2^a\right) - \theta_{i,k}\Phi^{-1}\left(z_1^a\right)}{\sqrt{1-p_{i,k}^2}}\right)$.
            Set $d_3^{00,01,10,11} = \Phi\left(\frac{\Phi^{-1}\left(z_1^a\right) - \theta_{i,k}\Phi^{-1}\left(z_2^a\right)}{\sqrt{1-p_{i,k}^2}}\right)$.
            Set $\mathbf{C}' = d_1 \cdot \rho'_{N-i+1,N-i+1,\theta_i} + d_2 \cdot z'^{a}_{1,\theta_i} + d_3 \cdot z'^{b}_{2,\theta_i}$.
            Set $\nabla C'_{i,k,\theta_i} = C^{00'}_{\theta_i} - C^{01'}_{\theta_i} - C^{10'}_{\theta_i} + C^{11'}_{\theta_i}$.
            Set $\mathbf{C}''$ as in (6.30) using $d_{4:10}$ terms.

            Set $\nabla C'_{i,k} = C^{00'} - C^{01'} - C^{10'} + C^{11'}$.
            Set $\nabla C''_{i,k} = C^{00''} - C^{01''} - C^{10''} + C^{11''}$.
        end
                                                                      ▷ Conditional distributions
        $v_{i-1,k}^{\text{direct}+-} = \frac{C(z_1, z_2; t_{i,k}, p_{i,k}) - C(z_1, z_2^-; t_{i,k}, p_{i,k})}{f(z_2)}$ and
        $v_{i-1,k}^{\text{indirect}+-} = \frac{C(z_1, z_2; t_{i,k}, p_{i,k}) - C(z_1^-, z_2; t_{i,k}, p_{i,k})}{f(z_1)}$ as in (6.16).
        Set $v_{i-1,k}^{\text{direct}+-} = H\left(z_1, z_2 \mid \mathbf{C}\right)$
            and $v_{i-1,k}^{\text{indirect}+-} = H_{\theta_i}\left(z_1, z_2 \mid \mathbf{C}\right)$
        Set $s1_{i-1,k,1:T}^{\text{direct}+-} = H'_{\theta_i}\left(z_1, z_2, z'_1, z'_2 \mid \mathbf{C}, \mathbf{C}'\right)$
            and $s1_{i-1,k,1:2,1:T}^{\text{indirect}+-} = H'_{\theta_i}\left(z_2, z_1, z'_2, z'_1 \mid \mathbf{C}, \mathbf{C}'\right)$, using (6.31).
        Set $s2_{i-1,k,1:T*}^{\text{direct}+-} = H''_{\theta_i\theta_j}\left(z_1, z_2, z'_1, z'_2, z''_1, z''_2 \mid \mathbf{C}, \mathbf{C}', \mathbf{C}''\right)$
      and $s2_{i-1,k,1:2,1:T*}^{\text{indirect}+-} = H''_{\theta_i\theta_j}\left(z_2, z_1, z'_2, z'_1, z''_2, z''_1 \mid \mathbf{C}, \mathbf{C}', \mathbf{C}''\right)$, using (6.32).

        Set $f_{i-1,k}^{\text{direct}} = v_{i-1,k,1:2,1:T}^{\text{direct}+} - v_{i-1,k}^{\text{direct}-}$,
       and $f_{i-1,k}^{\text{indirect}} = v_{i-1,k}^{\text{indirect}+} - v_{i-1,k}^{\text{indirect}-}$.
        Set $f_{i-1,k,\theta_i}^{\text{direct}}{}' = s1_{i-1,k}^{\text{direct}+} - s1_{i-1,k}^{\text{direct}-}$,
       and $f_{i-1,k,\theta_i}^{\text{indirect}}{}' = s1_{i-1,k}^{\text{indirect}+} - s1_{i-1,k}^{\text{indirect}-}$.
        Set $f_{i-1,k,\theta_i}^{\text{direct}}{}'' = s2_{i-1,k}^{\text{direct}+} - s2_{i-1,k}^{\text{direct}-}$,
       and $f_{i-1,k,\theta_i}^{\text{indirect}}{}'' = s2_{i-1,k}^{\text{indirect}+} - s2_{i-1,k}^{\text{indirect}-}$.

        if $\nabla C$ expressed in log-likelihood then save $c_j = \nabla C$, and $c'_j = \nabla C'$.
end end
Set
    $\ell''_{\theta_i\theta_j} = \begin{cases} \sum_{j=1}^{\lfloor (N+1)/2 \rfloor} \frac{\nabla C''_{k,\,\theta_i\theta_j}}{\nabla C_k} - \frac{\nabla C'_{k,\,\theta_i\theta_j} \nabla C'_{k,\,\theta_j}}{\nabla C_k^2}, & \text{if } N \text{ is even,} \\ \sum_{j=1}^{\lfloor (N+1)/2 \rfloor} \frac{\nabla C''_{k,\,\theta_i\theta_j}}{\nabla C_k} - \frac{\nabla C'_{k,\,\theta_i\theta_j} \nabla C'_{k,\,\theta_j}}{\nabla C_k^2} & \\ \quad + \frac{f''_{\theta_i\theta_j}(y_n)}{f(y_n)} - \frac{f'_{\theta_i}(y_n) f'_{\theta_j}(y_n)}{f^2(y_n)} & \text{if } N \text{ is odd.} \end{cases}$
Return $\ell'_{\theta_i}$
```

Appendix C

All computations in thesis are written and implemented in the R statistical software environment (R Core Team, 2013). A selection of R code is given in this section.

C.1 GHK Simulator

The Geweke-Hajivassilou-Keane (GHK) importance sampler discussed in section 4.3.1 is implemented in R using the vectorised code. The GHK function returns m samples from the truncated multivariate normal distribution $TN_N(\mathbf{0}, \Sigma, \mathbf{a}, \mathbf{b})$.

```
GHK <- function (m, Sigma, a, b){
    N       <- length(a)  # dimension of distribution
    xt      <- x <- matrix(0, N, m)  # allocate space
    pc      <- pnorm(a[1])  # compute initial values
    pd      <- pnorm(b[1])
    xt[1, ] <- runif(m) * (pd - pc) + pc
    x[1, ]  <- qnorm(ut[1, ])
    L <- t(chol(Sigma))  # compute Cholesky decomposition of Sigma
    for (i in 2:N) {
        dummy   <- L[i, ] %*% x
        pc      <- pnorm((a[i] - dummy)/L[i, i])
        pd      <- pnorm((b[i] - dummy)/L[i, i])
        xt[i, ] <- runif(m) * (pd - pc) + pc
        x[i, ]  <- qnorm(xt[i, ])
    }
    return(L %*% x) }  # transform values to TN(0,Sigma,a,b)
```

C.2 MCEM Algorithm

The MCEM algorithm of Chapter 4 is implemented in R code with the following code. The MCEM function returns the maximum likelihood estimates of the dependences parameters $(a_1, \ldots, a_p, b_1, \ldots, b_q)$ of the Gaussian copula model (3.8) with discrete margins. The input arguments are:

`obs.data`: Observed integer-valued time series with no missing values.

`Monte_Carlo_samples` : The number of Monte Carlo samples to be used in E-Step.

`p, q`: The order of the underlying ARMA model.

`initial`: The initial values to begin the MCEM algorithm.

`num_iterations`: The length of the output of the MCEM sequence.

`prec`: The numeric value to determine the stopping criteria, default to 0.01.

`marginal_dist`: The specification of the marginal distribution, default=”negbin”.

`optim_method`: The numerical optimisation procedure to be used, default=”nmkb” in Varadhan et al. (2011) package.

`compute_stderrors`: Option for the output to include the standard errors of the MLEs, default=TRUE.

```
MCEM <- function(obs.data, Monte_Carlo_samples=mm, p, q, initial,  num_iterations,
                prec=0.01,  marginal_dist="non-parametric", optim_method="nmkb"){

# Allocate space/Initial setup
n         <- length(obs.data)
t         <- 1
para.star <- initial

# Specify estimation methods for distribution functions, e.g.
if(marginal_dist=="non-parametric"){
Fn        <- ecdf(obs.data)
up        <- qnorm(Fn(obs.data))
low       <- qnorm(Fn(obs.data-1))
}

for(k in 1:ml){  # loop to increase the number of Monte Carlo samples
m  <- mm[k]  # Set number of samples for Monte Carlo
# Begin MCEM algorithm iterations where each loop is one iteration
for(j in 1:num_iterations){
```

```
                    current <- para.star
                    # The E and the M-step
                    para <- optim(current, Q_function, para.star=para.star, m=m, low=low,
                    up=up,p=p, q=q, method="L-BFGS-B", control=list(maxit=3000))$par
                    # Different numerical optimisers can be chosen

                    # Save the samples from the Monte Carlo Step
                    xdata <- xxx

                    # Check: Ensure the stationarity of the updated parameters
                    para[1:p]        <-  ensure_causality_invertibility(para[1:p])
                    para[(p+1):(p+q)]<- -ensure_causality_invertibility(-para[(p+1):(p+q)])

                    # Estimate the change in log-likelihood
                    ll[t]            <- Delta_l_x(xdata, para.star, para, c(p,q))
                    diff[t]          <- ll[t]
                    # Check: Has the stopping criteria been satisfied?
                    finish      <- all(diff[t:(t-5)]<prec)
                    # Save and print updated parameter values
                    t                <- t + 1
                    save[t, ]        <- para.star  <- para
}}

se <- compute_std_errors(para, obs.data,  m, p, q, low, up)  # Computing the Std errs

return(list(MCEM_Iterations=save[1:t,], diff=diff[1:t], dall=dall[1:t],
Monte_Carlo_samples=Monte_Carlo_samples, t=t, MLEs=MLEs))
}
```