



Technische Universität München

DEPARTMENT OF MATHEMATICS

General Vine Copula Models for Stationary Multivariate Time Series

Master Thesis

by

Daniel Krüger

Supervisor: PD Dr Aleksey Min

Advisors: PD Dr Aleksey Min

M.Sc. Thomas Nagler

Submission Date: 27th of November 2018

I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Munich, 27th of November 2018

Abstract

Describing the serial, cross-serial and cross-sectional (conditional) dependence is an important task in the analysis of multivariate time series. While the classical vector autoregressive (VAR) model only captures linear dependence, copula functions, introduced by Sklar (1959) enable us to describe the dependence more flexibly. For high dimensional data, one often uses pair-copula constructions, where the joint copula density is decomposed using only bivariate copulas, for a more flexible description. In the literature there are mainly three copula time series models using regular vines in combination with pair-copula constructions. Regular vines are collections of trees, enabling us to build a plan for the pair-copula constructions. The two most common examples are D-vines (each tree is a path) and C-vines (each tree is a star). While the models of Smith (2015) and Beare and Seo (2015) use D-vines to capture the cross-sectional dependence and only deviate in the cross-serial connection, the model developed by Brechmann and Czado (2015) uses a cross-sectional C-vine. A convenient way to store all informations of the copula models (vine structure, copula families and parameters) is given by the regular vine matrices introduced by Dissman et al. (2012).

In this thesis, we generalise those approaches to the so called R(egular)-T(emporal)-vine model. On basis of the M-vine model (Beare and Seo (2015)) we will not only allow for arbitrary regular vines for the cross-sectional dependence structure, but also for different edges connecting the cross-sectional structures at each time step. We will analyse the influence on the overall vine structure and come up with an algorithm yielding the R-vine matrix representation of our R-T-Vine model. Further, we will show that stationarity as well as the Markov property can easily be imposed to the new model by constraining the pair-copula models at some edges in our vine structure.

In the empirical part of the master thesis, we will illustrate our model at work. We consider a data set consisting of monthly log-returns of five selected stock indices. We will see a clear out-performance of the R-T-vine model versus the classical M-vine model. Finally, we use this estimated R-T-Vine model for prediction of monthly log-returns and compare the outcome with the classical M-vine.

Abstract

Eine wichtige Aufgabe in der Analyse multivariater Zeitreihen ist die Beschreibung der seriellen, seriell übergreifenden und der sektionalen (bedingten) Abhängigkeiten. Während das klassische autoregressive Vektormodell (VAR) nur die lineare Abhängigkeit erfasst, ermöglichen die von Sklar (1959) eingeführten Copulafunktionen eine bessere Beschreibung der Abhängigkeiten. Für eine flexiblere Darstellung hochdimensionaler Daten werden häufig Paar-Copula Konstruktionen verwendet, bei der die Dichte der gemeinsamen Copula nur mit Hilfen von bivariaten Copulaen dargestellt wird. In der Literatur gibt es im Wesentlichen drei Copula-Zeitreihenmodelle, die reguläre Vine-Strukturen in Kombination mit Paar-Copula Konstruktionen verwenden. Reguläre Vines sind Sammlungen von Bäumen und können zur Erstellung eines Plans für die Paar-Copula Konstruktionen verwendet werden. Die bekanntesten Vine-Strukturen sind D-Vines (jeder Baum ist ein Pfad) und C-Vines (jeder Baum ist ein Stern). Während die Modelle von Smith (2015) und Beare und Seo (2015) D-Vines verwenden, um die sektionale Abhängigkeit zu erfassen und nur in den seriell übergreifenden Verbindung voneinander abweichen, verwendet das von Brechmann und Czado (2015) entwickelte Model C-Vines. Eine Möglichkeit alle Informationen von Copula-Modellen (Vine-Strukturen, Copula-Familien und -Parameter) zu speichern, bietet die von Dissman et al. (2012) eingeführte reguläre Vine-Matrix.

In dieser Arbeit verallgemeinern wir die Ansätze auf das sogenannte R(egular)-T(emporal)-Vine-Modell. Auf der Grundlage des M-Vine-Modells (Beare und Seo (2015)) berücksichtigen wir nicht nur jeden beliebigen regulären Vine für die sektionale Abhängigkeitsstruktur, sondern auch unterschiedliche Kanten, die diese Strukturen in den Zeitschritten verbinden. Wir werden den Einfluss auf die Gesamtstruktur des Vines auswerten und einen Algorithmus entwickeln, der die R-Vine-Matrixdarstellung unseres R-T-Vine-Modells liefert. Außerdem zeigen wir, dass Stationarität sowie die Markov-Eigenschaft im neuen Modell einfach umzusetzen sind, indem wir die Paar-Copulae Modelle für einige Kanten in unserer Vine-Struktur beschränken.

Im empirischen Teil dieser Arbeit wenden wir unser Model auf einen Datensatz an, der aus monatlichen Log>Returns von fünf ausgewählten Aktienindizes besteht. Gegenüber dem klassischen M-Vine-Modell werden wir eine deutliche Outperformance sehen. Schließlich verwenden wir das geschätzte R-T-Vine-Modell zur Vorhersage und vergleichen das Ergebnis mit dem des klassischen M-Vines.

Contents

1	Introduction	2
2	Mathematical Preliminaries	4
2.1	Time Series Analysis	4
2.1.1	Stationarity	5
2.1.2	Classical Time Series Models	8
2.2	Copulas	10
2.2.1	Introduction to Copulas	11
2.2.2	Copula Families	15
2.3	Dependence Measures	16
2.3.1	Linear Correlation	16
2.3.2	Kendall's τ	17
3	Vine Copula Specification	19
3.1	Regular Vines	19
3.2	Pair Copula Construction	23
3.2.1	General Scheme	24
3.2.2	R-Vine Pair Copula Construction	25
3.3	Matrix Notation of Regular Vines	27
4	R-Vine Matrices for Time Series	30
4.1	Introduction into Regular Vine Models for Multivariate Time Series	30
4.2	Cross-Sectional D-Vine Structure	33

4.2.1	Classical M-Vine Approach	33
4.2.2	Comparison with D-Vine Approach	37
4.3	Cross-Sectional C-Vine Structure	39
4.3.1	COPAR Approach	39
4.3.2	Generalised M-Vine Approach for Cross-Sectional C-Vine	42
4.3.3	Deep-Dive Differences COPAR and (Generalised) M-Vine	45
4.4	More General Cross-Sectional R-Vine Structure	50
4.5	Summary and Algorithm for Generalised M-Vine	54
5	Generalised M-Vine	58
5.1	Formal Definition of Generalised M-Vines	58
5.2	Central Theorems	62
6	Generalised Temporal Connection	66
6.1	Changing the Temporal Out-Vertex	67
6.1.1	Cross-Sectional D-Vine Structure	67
6.1.2	Cross-Sectional C-Vine Structure	71
6.1.3	More General Cross-Sectional R-Vine Structure and Tree Building	74
6.2	Changing the Temporal In-Vertex	77
6.2.1	General Influence on Matrix Representation	77
6.2.2	Example: General Regular Vine Structure	77
6.2.3	Explanation via Tree Building	78
6.3	Changing the Connecting Edges	82
6.4	Summary and Algorithm	84
6.4.1	Influence of Changing Connecting Edges on Generalised M-Vines	84
6.4.2	Generalisation of the Algorithm	86
7	R-T-Vine Copula Specification	89
7.1	Respective Copula Vine Specification	89
7.2	Reducing Complexity of the Copula Matrix	90

<i>CONTENTS</i>	1
7.2.1 Stationarity	90
7.2.2 Autoregressive Order p	92
7.2.3 Example Calculation: Reduction of Complexity	94
8 Maximum Likelihood Estimation	96
8.1 Introduction to MLE	96
8.2 Model Specification for R-T-Vine Copula Model	97
9 Application: Major Stock Indices	100
9.1 Dataset	100
9.1.1 Log>Returns	100
9.1.2 Data Transformation	101
9.2 Estimation of R-T-Vine Specification	103
9.2.1 Selecting Cross-Sectional Structure	104
9.2.2 Selecting Connecting Edges	107
9.2.3 R-T-Vine Estimation	109
9.2.4 Outperformance of Classical M-Vine	110
9.3 Simulation and Comparison of Results	114
10 Wrap-Up	119
A Vine Structure in Higher Sectional and Time Dimensions	121
B R Algorithms	123
List of Figures	125

Chapter 1

Introduction

The concept of copulas which was introduced by Sklar (1959) became a central tool for modelling multivariate dependence in recent years. The central theorem named after him proved that any n dimensional joint distribution function can be decomposed into its n marginal distributions and a copula, describing the joint dependence structure.

Copulas are even known in a broader public due to the appearance in mainstream media, where the copula formula is accused to be the instrument causing the financial crisis (see for instance the article "*The formula that felled Wall Street*" by Jones (2009)). Indeed, the (back then) widely used Gaussian copula does not allow for heavy tails and therefore do not capture extreme events as in the financial crisis accordingly.

A way of receiving more flexible multivariate copulas is the so called pair copula construction (PCC) developed by Aas et al. (2009). The idea of this construction goes back to Joe (1996). It uses a factorisation of the joint distribution into bivariate (conditional) copulas/distributions to create a more flexible multivariate distribution. However, there are multiple possible PCC's and one needs a convenient way to organise them. A graphical tool for organising are regular vines introduced by Bedford and Cooke (2001), which are a set of connected trees, being able to describe the marginal distributions and the coupling of those distributions. A convenient way to store all related information in so called regular vine matrices was invented by Dissman et al. (2012).

There are several models using PCC in combination with regular vines, so called regular vine specifications. We are especially interested in models, describing the serial, cross-serial and cross-sectional dependence among multivariate time series. Three models of interest are given by the Copula autoregressive (COPAR) (Brechman and Czado

(2015)), the long D-vine (Smith (2015)) and the M-vine (Beare and Seo (2015)) model.

We intend to develop a new and more general model based on these models within this thesis. Chapter 2 will cover all mathematical preliminaries of copulas and time series analysis. In Chapter 3, we will give an introduction into vine copula specification in general, i.e. the usage of regular vines to organise PCC and the properties of the regular vine matrices. In the following Chapters 4-6, we will review already existing models for multivariate time series and develop a more generalised regular vine model (R-T-Vine model) covering more cross-sectional as well as (cross-)serial possibilities in the vine structure, describing the overall dependence. These chapters are the core of this thesis, as we are able to come up with a new vine structure model better fitting to multivariate time series than the already existing ones. We will further describe the structure of the R-T-Vine and come up with an algorithm yielding the optimal R-T-Vine structure. Having this new structure, we formalise the vine specification with the help of this structure and point the possibilities of reducing the complexity by assuming stationarity or an autoregressive order p in Chapter 7. In Chapter 8, we will give a short introduction into (sequential) maximum likelihood estimation used for fitting the bivariate (conditional) copulas in our model. Finally, we fit our new model to the monthly returns of five stock indices and compare the model with the M-vine model (Beare and Seo (2015)) also with respect to prediction.

Chapter 2

Mathematical Preliminaries

2.1 Time Series Analysis

In areas such as economics and finance, a common problem is the analysis of multivariate time series, where each univariate time series is a component of a vector-valued time series. With these multivariate time series not only serial, but also (cross-) sectional dependence is modelled.

We start by giving an introduction into (multivariate) time series in general. This section is mainly referring to Brockwell, Davis et al. (2016).

Definition 2.1. Time Series

A time series is a stochastic process $(X_t)_{t \in T}$ for some index set T . However, the term is also used for the (partial) realisation, i.e. a set of observations x_t at a specific time $t \in T_0 \subset T$, of such a process.

Remark 2.2. If $(X_t)_{t \in T}$ respectively $(x_t)_{t \in T_0}$ are vector-valued we call such a process multivariate time series. To indicate when we are talking of a multivariate series we use the notation $(\mathbb{X}_t)_{t \in T}$.

In general, the time set T could take various values. In this thesis we will work with a discrete time set $T \subset \mathbb{R}$ of consecutive time points.

As already mentioned in the introductory chapter we are interested in analysing and modelling the dependence structure of a multivariate time series, e.g. asset returns, over time. The joint observation of those asset developments is an example for a multivariate time series.

Example 2.3. Monthly Closing Prices of Selected Indices

The figure below displays the monthly adjusted close prices of the DAX, FTSE, CAC, Nikkei and S&P500 from beginning of 2008 to mid year 2018.

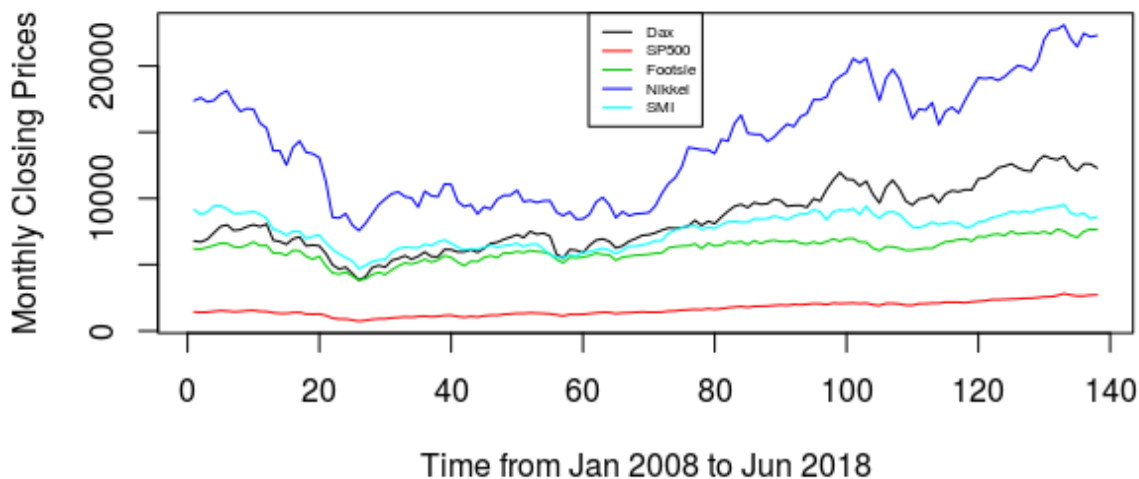


Figure 2.1: Time series of adjusted monthly prices for five indices from beginning of 2008 until June 2018.

2.1.1 Stationarity

Stationarity is one of the most important properties of stochastic processes in time series analysis. With stationarity, one obtains properties that are not only valid for individual times, but are invariant over time. In our generalised vine copula model, stationarity will significantly reduce the computational effort.

Before we define stationarity, we need to have a look at the second order structure of multivariate time series. Let us assume we have a d -variate time series, i.e. for

$t \in T: \mathbb{X}_t = \begin{pmatrix} X_{1t} \\ X_{2t} \\ \dots \\ X_{dt} \end{pmatrix}$, then the second order structure is given by

i) the mean vector $\mu_t := \mathbb{E}[\mathbb{X}_t] = \begin{pmatrix} \mathbb{E}[X_{1t}] \\ \mathbb{E}[X_{2t}] \\ \dots \\ \mathbb{E}[X_{dt}] \end{pmatrix}$ and

ii) the covariance matrix $\Gamma(t+h, t) := \begin{bmatrix} \gamma_{11}(t+h, t) & \dots & \gamma_{1d}(t+h, t) \\ \vdots & \ddots & \vdots \\ \gamma_{d1}(t+h, t) & \dots & \gamma_{dd}(t+h, t) \end{bmatrix}$,

where $\gamma_{ij}(t+h, t) := \text{Cov}[X_{t+h,i}, X_{t,j}]$.

Definition 2.4. (Weak) Stationarity

We call a (d -variate) time series $(\mathbb{X}_t)_{t \in T}$ weakly stationary if:

- i) $\mathbb{E}[|X_{it}|^2] < \infty \forall t$ and i ,
- ii) μ_t is independent of t and
- iii) $\Gamma(t+h, t)$ is independent of t for each fixed h .

Remark 2.5. As for the independence of $t \in T$ for stationary time series, the notation $\mu = \mu_t$ and $\Gamma(h) = \Gamma(t+h, t) \forall t \in T$ is often used.

There is another concept of stationarity, the so called strict stationarity.

Definition 2.6. Strict Stationarity

$(\mathbb{X}_t)_{t \in T}$ is strictly stationary if $(\mathbb{X}_{t_1}, \mathbb{X}_{t_2}, \dots, \mathbb{X}_{t_n}) \stackrel{d}{=} (\mathbb{X}_{t_1+h}, \mathbb{X}_{t_2+h}, \dots, \mathbb{X}_{t_n+h}) \forall t_1, \dots, t_n \in T, n \in \mathbb{N}$ and h such that $t_1+h, \dots, t_n+h \in T$.

Note that strict always implies weak stationarity. The reversed statement does not hold. In the case of multivariate normal distribution of all finite dimensional marginals (we call such a process Gaussian) also the reverse statement is true. In the following example, we show that for non-Gaussian time series weak does not imply strict stationarity.

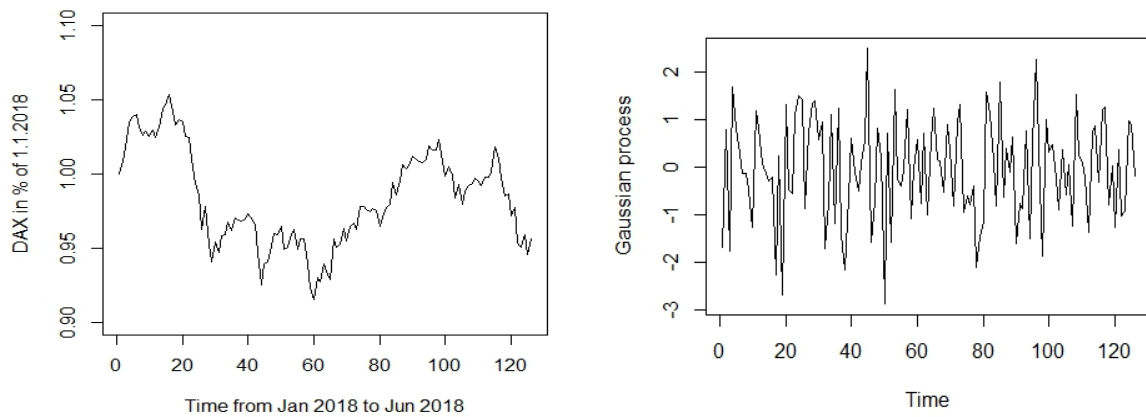
Example 2.7. Weak $\not\Rightarrow$ Strict Stationarity

Let $X_t = \begin{cases} \sim \text{expo}(1), & t \text{ odd} \\ \sim N(1, 1), & t \text{ even} \end{cases}$ be independent.

Then, $\mathbb{E}[X_t] = 1$ and $\Gamma(t+h, t) = \begin{cases} 1, & h = 0 \\ 0, & h > 0 \end{cases} \forall t, h : t \in T \text{ and } t+h \in T$, i.e. X_t is

weakly stationary. But, clearly, X_t is not strictly stationary as it has different distributions for t odd and even.

Example 2.8. The Figure 2.2 below presents two (one-dimensional) time series. The year-to-date daily closing prices of the DAX in percentage of the starting value and a Gaussian process with $X_t \sim \mathcal{N}(0, 1)$. The Gaussian process is an example for stationary time series (fluctuating around the mean 0) the DAX-price-series is non-stationary.



(a) DAX closing

(b) Gaussian process

Figure 2.2: Time series of (a) non-stationary DAX closing prices in percentage of the starting value and (b) stationary Gaussian process.

Returning to the indices developments in Example 2.2, we recognize that this time series does not look stationary. However, if we look at the respective log-returns, as shown in Figure 2.3, we can see a similar structure as in the Gaussian case mentioned in Example 2.8.

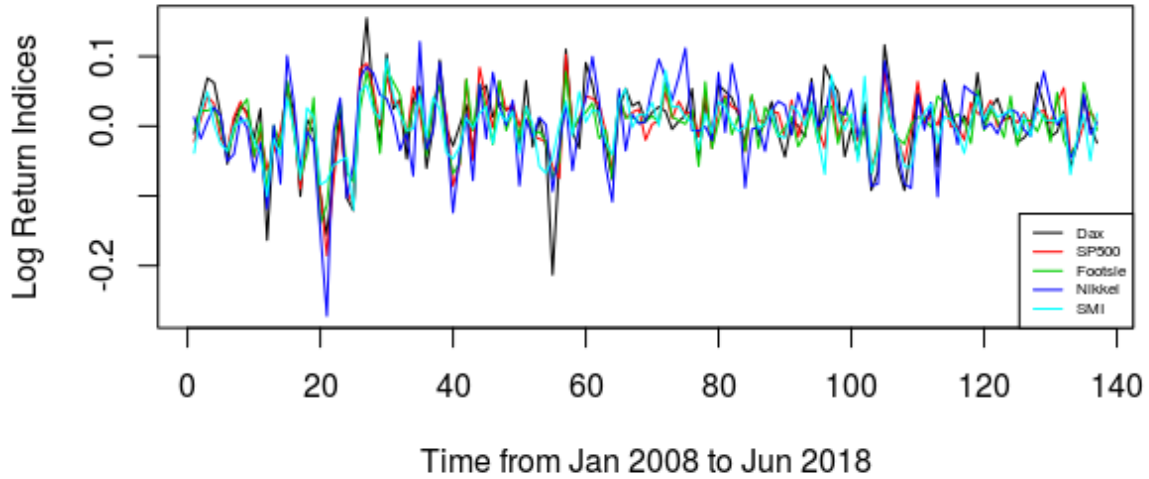


Figure 2.3: Monthly log-returns of the five indices shown Example 2.3 from beginning of 2008 up to June 2018.

2.1.2 Classical Time Series Models

Next, we will give a short overview on some widely known time series processes.

Autoregressive-Moving Average Processes

We will first define autoregressive (AR), moving average (MA) and their combination autoregressive-moving average (ARMA) processes.

Definition 2.9. AR/MA/ARMA processes

Let $(Z_t)_{t \in T}$ be i.i.d. $\mathcal{N}(0, \sigma^2)$ distributed. The time series $(X_t)_{t \in T}$ is called

MA(q) process, if $X_t = \sum_{j=1}^q \psi_j Z_{t-j}$,

AR(p) process, if $X_t = Z_t + \sum_{j=1}^p \phi_j X_{t-j}$ or

ARMA(p,q) process, if $X_t - \sum_{j=1}^p \phi_j X_{t-j} = Z_t + \sum_{j=1}^q \psi_j Z_{t-j}$.

Remark 2.10. In literature one often uses a short notation of the respective (characteristic) polynomials arising in the definition above, e.g. for the ARMA(p,q) process one writes $\Psi_q(x) = \sum_{j=1}^q \psi_j x_{t-j}$ and $\Phi_p(x) = \sum_{j=1}^p \phi_j x_{t-j}$.

GARCH Processes

The linear times series models such as ARMA models are not suited for reflecting the "stylised facts of financial data", which have been shown in several empirical studies of financial time series¹. The first model capturing those facts was the ARCH- (autoregressive conditionally heteroscedastic) model developed by Engle (1982). In contrast to the linear ARMA model for the series $(X_t)_{t \in T}$, where the conditional variance h_t of X_t given $\{Z_s, s < t\}$ is independent of t and of $\{Z_s, s < t\}$, Engle proposed to incorporate the sequence $(h_t)_{t \in T}$ into the model. He postulated that

$$X_t = \sqrt{h_t}e_t, \text{ where } e_t \sim N(0, 1) \text{ iid}$$

and h_t (known as the volatility) is related to the past values of X_t^2 via a relation of the form,

$$h_t = \alpha_0 + \sum_{i=1}^p \alpha_i X_{t-i}^2,$$

for some positive integer p , $\alpha_0 > 0$ and $\alpha_i \geq 0$ for $i = 1, \dots, p$.

This relation was later generalised by Bollerslev (1986) in the so called GARCH (generalised ARCH) model:

$$h_t = \alpha_0 + \sum_{i=1}^p \alpha_i X_{t-i}^2 + \sum_{i=1}^q \beta_i h_{t-i},$$

with $\alpha_0 > 0$, $\alpha_i \geq 0$ for $i = 1, \dots, p$ and $\beta_i \geq 0$ for $i = 1, \dots, q$.

Remark 2.11. For modelling empirical log-returns of financial data as seen in the previous section one neglects the normality assumption for e_t and uses $e_t \sim iid(0, 1)$ instead. Especially, one often uses heavier-tailed zero-mean distribution such as Students t-distribution.

Remark 2.12. In the paper of Brechmann and Czado (2015), the authors use a GARCH model with skewed Student t innovations to transform the data set of log-returns to standardised residuals. However, we decided to transform our data directly via a fitted skewed t-distribution as the conditional variance should also be covered by our general vine copula model.

Vector Autoregressive Model

After having reviewed the one-dimensional ARMA models we will now introduce the multidimensional autoregressive case, so called vector autoregressive (VAR) models, which

¹See, e.g., Mandelbrot (1963) and Fama (1965)

are used for modelling many problems arising in economics and finance. Such models became a standard tool in econometrics when Sims (1980) used them as an alternative to classical simultaneous equation models.

Definition 2.13. VAR(p) Model

Let $(\mathbb{X}_t)_{t \in T}$ be a stationary (d-variate) time series. We call such a time series VAR(p)-model if for all $t \in T$:

$$\mathbb{X}_t = c + \Phi_1 \mathbb{X}_{t-1} + \Phi_2 \mathbb{X}_{t-2} + \cdots + \Phi_p \mathbb{X}_{t-p} + \epsilon_t,$$

where c is a (d dimensional) vector of constants, the observation \mathbb{X}_{t-i} is called the i -th lag of \mathbb{X} , Φ_1, \dots, Φ_p are time-invariant $d \times d$ matrices and the d dimensional vector of error terms ϵ_t is satisfying:

- i) $\mathbb{E}[\epsilon_{i,t}] = 0 \forall t$ and i ,
- ii) $\text{Cov}[\epsilon_{i,t}, \epsilon_{j,t}] = \Sigma$, where Σ is a positive definite matrix and
- iii) $\text{Cov}[\epsilon_{i,t}, \epsilon_{j,t+h}] = 0 \forall h \neq 0$.

Normally, one assumes independent identical distributed $\epsilon_t \sim \mathcal{N}(0, \Sigma)$.

Remark 2.14. The probably most useful property of VAR(p)-models is its conditional independence of observations with lag greater than p .

For our full generalised vine copula model, we would need to calculate many parameters. This number greatly reduces when considering a model fitted to a VAR(p) time series and/or stationarity. We will discuss this in detail in Chapter 7.

2.2 Copulas

Because VAR models can only capture the linear dependence within the serial and sectional dependence structure, we are interested in extending those models, using copulas to describe the dependence structure, in this thesis. Therefore, we also need an introduction into copulas, which are the canonical statistical tool for statistical dependence modelling. We will be especially interested in copula density functions and using those as well as the representation of joint distribution function through their marginals and copulas to obtain an alternative expression for the joint density function.

Copulas were first introduced by Sklar (1959). One of the central theorems, which

links joint distribution functions to their marginal distributions via copulas, is named after him. A further interesting definition of copulas is given in the book *An introduction to copula*, published by Nelsen (2006). Besides the link given in Sklar's theorem, he defines copulas as multivariate distribution functions whose one-dimensional marginals are uniformly distributed. This section will be mainly referring to this book of Nelsen (2006). We will restrict ourselves to the two dimensional case as we can build higher dimensional dependence structures using the so called pair copula construction, discussed in Chapter 3.

2.2.1 Introduction to Copulas

First, let us formally define a copula.

Definition 2.15. Copula (Cf. Page 10 Nelsen (2006))

A copula is a function $C : [0, 1]^2 \rightarrow [0, 1]$ with the following properties:

i) $\forall u, v \in [0, 1]$:

$$C(u, 0) = C(0, v) = 0 \text{ and } C(u, 1) = u \text{ respectively } C(1, v) = v \quad (2.1)$$

ii) $\forall u_1, v_1, u_2, v_2 \in [0, 1]$ with $u_1 \leq u_2$ and $v_1 \leq v_2$:

$$C(u_2, v_2) - C(u_2, v_1) - C(u_1, v_2) + C(u_1, v_1) \geq 0. \quad (2.2)$$

To better understand this definition, we start by repeating the definitions of marginal and joint distribution functions.

Definition 2.16. Distribution Function

A distribution function is a function $F : \bar{\mathbb{R}} \rightarrow [0, 1]$, satisfying:

i) F is non decreasing and

ii) $\lim_{x \rightarrow -\infty} F(x) = 0$ and $\lim_{x \rightarrow \infty} F(x) = 1$.

An important distribution with regard to copulas is the uniform distribution on $[0, 1]$, that is why we briefly repeat the definition in the following example.

Example 2.17. Uniform Distribution on $[0, 1]$

$$U_{0,1}(x) = \begin{cases} 0 & \text{for } x \in [-\infty, 0) \\ x & \text{for } x \in [0, 1] \\ 1 & \text{for } x \in (1, \infty] \end{cases}$$

We will also have to deal with inverses of distribution function. For strictly increasing distribution functions we can directly compute the ordinary inverse, however, in the other cases we will need the concept of the quasi-inverse.

Definition 2.18. Quasi-Inverse

Let F be a continuous distribution function. We call

$$F^{-1}(t) := \inf\{x | F(x) \geq t\}$$

the quasi-inverse of F .

With the help of the following theorem we can link each distribution function to the uniform distribution and vice versa. It will be used (mostly implicit) in this thesis and was already introduced in 1952 by Rosenblatt.

Theorem 2.19. Probability Integral Transformation

Let X be a real random variable with continuous distribution function F and $U \sim U_{0,1}$.

Then

$$F(X) = U \text{ and } F^{-1}(U) \sim F.$$

Now, we turn to the multivariate (two-dimensional) case.

Definition 2.20. Joint Distribution Function (Cf. Page 17, Nelsen (2006))

A joint distribution function is a function $H : \bar{\mathbb{R}}^2 \rightarrow [0, 1]$ such that:

2-increasingness: $\forall x_1, y_1, x_2, y_2 \in \mathbb{R}$ with $x_1 \leq x_2$ and $y_1 \leq y_2$:

$$H(x_2, y_2) - H(x_2, y_1) - H(x_1, y_2) + H(x_1, y_1) \geq 0 \text{ and}$$

Groundness: $H(x, -\infty) = H(-\infty, y) = 0$ and $H(\infty, \infty) = 1$ holds.

Especially, H has marginal distributions given by $F(x) = H(x, \infty)$ and $G(y) = H(\infty, y)$.

Remark 2.21. Obviously, Equations (2.1) and (2.2) imply groundness respectively 2-increasingness in Definition 2.20.

Thus, a copula is a two-dimensional distribution function. Further, because of equation (2.1) both marginals are given by

$$U(x) = C(x, 1) = C(1, x) = x \text{ for } x \in [0, 1],$$

which can be interpreted (by extending the domain to $\bar{\mathbb{R}}$) as a uniform distribution function, given in Example 2.17.

Next, we will come to the interpretation of copulas given in Sklar's theorem.

Theorem 2.22. Sklar's Theorem

Let H be a joint distribution function with marginals F and G . Then there exists a copula C such that $\forall x, y \in \bar{\mathbb{R}}$:

$$H(x, y) = C(F(x), G(y)). \quad (2.3)$$

Proof. See Nelsen (2006) □

Remark 2.23. This theorem allows us to divide the handling of a multivariate probability law into the handling of the univariate marginals and the isolated dependence structure given by the copula.

By combining Sklar's theorem with the probability integral transform, we can even say more about the link of copulas and joint distribution functions:

Corollary 2.24. Let $(X, Y) \sim H$, where H is the joint distribution function with marginals F and G , then we have

$$(U_1, U_2) := (F(X), G(Y)) \sim C.$$

On the other hand, for all $(U_1, U_2) \sim C$ it holds

$$(X, Y) := (F^{-1}(U_1), G^{-1}(U_2)) \sim H$$

and

$$C(u_1, u_2) = H(F^{-1}(u_1), G^{-1}(u_2)) \quad \forall u_1, u_2 \in (0, 1).$$

When investigating the dependence measures in the next section, there will arise boundaries for copulas. Therefore, we will need the so called Fréchet-Hoeffding bounds. In the context of boundaries, the three special cases of copulas of interest are given in the following example.

Example 2.25. Commonotonicity

For u_1 and u_2 in $[0, 1]$ the copula given by

$$M_2(u_1, u_2) = \min\{u_1, u_2\} \quad \text{is called } \mathbf{comonotonicity\ copula},$$

$$W_2(u_1, u_2) = \max\{u_1 + u_2 - 1, 0\} \quad \text{is called } \mathbf{countercomonotonicity\ copula} \text{ and}$$

$$\Pi_2(u_1, u_2) = u_1 u_2 \quad \text{is called } \mathbf{product\ copula}.$$

Fréchet and Hoeffding independently obtained that the counter- and comonotonicity copula as defined above arise as natural boundaries for each two-dimensional copula.

Theorem 2.26. Fréchet-Hoeffding Bounds

Let $C : [0, 1]^2 \rightarrow [0, 1]$ be a copula. Then for u_1 and u_2 in $[0, 1]$, C is bounded by

$$W_2(u_1, u_2) \leq C(u_1, u_2) \leq M_2(u_1, u_2).$$

Proof. See Nelsen (2006) □

The product copula in the above example is also often called "independence copula" as it is equivalent to stochastic independence of the respective random variables, as stated in the following lemma.

Lemma 2.27. The random vector (X, Y) has stochastically independent components if and only if its distribution function can be split up into its marginals and the copula Π_2 .

Proof. Let H denote the joint distribution function of the random vector, F and G the respective marginals and x, y be in $\bar{\mathbb{R}}$. Then we get:

$$">\Rightarrow": F(X, Y) = \mathbb{P}(X \leq x, Y \leq y) \stackrel{\text{independence}}{=} \mathbb{P}(X \leq x)\mathbb{P}(Y \leq y) = \Pi_2(F(x), G(y))$$

$$">\Leftarrow": \mathbb{P}(X \leq x, Y \leq y) = F(X, Y) = \Pi_2(F(x), G(y)) = \mathbb{P}(X \leq x)\mathbb{P}(Y \leq y).$$

□

Finally, we have a look at the copula density and how we can use it to rewrite the joint density function.

Definition 2.28. Copula Density

Let C be a twice partially differentiable copula. The copula density is given by the function $c : [0, 1]^2 \rightarrow [0, 1]$ defined by

$$c(u_1, u_2) = \frac{\partial^2 C(u_1, u_2)}{\partial u_1 \partial u_2}.$$

To derive the representation of the joint density function we recall the expression from Sklar's theorem:

$$H(x, y) = C(F(x), G(y)).$$

Differentiating of both sides yields:

$$h(x, y) = \frac{\partial^2 C(F(x), G(y))}{\partial x \partial y} \stackrel{\text{chain rule}}{=} c(F(x), G(Y))f(x)g(y),$$

where the c, f and g indicate the densities to the distribution function C, F and G , respectively.

2.2.2 Copula Families

One can classify copulas in several classes. We will briefly introduce the two most important classes in the following: elliptical and Archimedian copulas.

Elliptical Copulas

Elliptical copulas inherit their properties from elliptical distributions. Therefore, we need to define elliptical distributions.

Definition 2.29. Elliptical Distribution(see Embrechts, Lindskog and McNeil (2001))

Let X be a n -dimensional random vector, $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$ a non-negative definite matrix and $\phi_{X-\mu}$ the characteristic function of $X - \mu$.

If $\phi_{X-\mu}(t)$ is a quadratic function of form $t'\Sigma t$, i.e. there exists a function Φ such that $\phi_{X-\mu}(t) = \Phi(t'\Sigma t)$, then X has an elliptical distribution with parameters μ , Σ and Φ .

Two well known examples for elliptical copulas are given by the bivariate Gaussian and t-copula.

Example 2.30. Bivariate Gaussian Copula

Let $X \sim N_2(\mu, \Sigma)$, then $X - \mu \sim N_2(0, \Sigma)$ with characteristic function $\phi_{X-\mu}(t) = \exp\{-\frac{1}{2}t^T\Sigma t\}$, which implies, that the multivariate normal distribution is an elliptical distribution. We denote the cumulative distribution function by Φ_Σ^2 .

Hence the bivariate Gaussian copula

$$C(u_1, u_2) = \Phi_\Sigma^2(\Phi^{-1}(u_1), \Phi^{-1}(u_2)),$$

where Φ^{-1} denotes the inverse of the one-dimensional cumulative normal distribution function, is an elliptical copula.

Example 2.31. t-Copula

Let $Z \sim N_2(0, \Sigma)$ and $S \sim \mathcal{X}_\nu^2$ be independent and $\mu \in \mathbb{R}^n$. Then $\mu + \sqrt{\frac{\nu}{S}}Z = X \sim t_\nu$, i.e. X is (elliptically) t-distributed.

The corresponding copula is defined for $\nu > 0$ and is given by:

$$C(u_1, u_2) = t_{\nu, R}(t_\nu^{-1}(u_1), t_\nu^{-1}(u_2)),$$

where $R_{ij} = \sigma_{ij}/\sigma_{ii}\sigma_{jj}$ with $i, j \in \{1, 2\}$.

Archimedean Copulas

Definition 2.32. Archimedean Copula(see Embrechts, Lindskog and McNeil (2001))

A copula $C_h : [0, 1]^2 \rightarrow [0, 1]$ is called Archimedean copula if it has the functional form:

$$C_h(u_1, u_2) = h(h^{-1}(u_1), h^{-1}(u_2)),$$

for a suitable non-increasing function $h : [0, \infty) \rightarrow [0, 1]$ with $h(0) = 1$ and $\lim_{x \rightarrow \infty} h(x) = 0$.

Example 2.33. Gumbel Copula

Let $h_\theta : [0, \infty) \rightarrow [0, 1]$ with $h(x) = \exp\{-x^{1/\theta}\}$, then $C_h(u_1, u_2) = h(h^{-1}(u_1), h^{-1}(u_2)) = \exp\{-[(-\ln(u_1))^\theta + (-\ln(u_2))^\theta]^{1/\theta}\}$ is the Gumbel copula.

Example 2.34. Clayton Copula

Let $h_\theta : [0, \infty) \rightarrow [0, 1]$ with $h(x) = (1 + x)^{1/\theta}$, then $C_h(u_1, u_2) = (u_1^{-\theta} + u_2^{-\theta} - 1)^{-1/\theta}$ is the Clayton copula.

2.3 Dependence Measures

The term (linear) dependence already arose in the previous sections. We now want to specify it further and highlight why the linear dependence, as covered by the VAR model, is not enough.

2.3.1 Linear Correlation

When talking about dependence in statistics most people think about linear correlation, which is indeed often used in practice as it is a widely known tool.

Definition 2.35. Linear correlation

Let X and Y be two random variables with non-zero and finite variances. The linear correlation is defined as

$$\rho(X, Y) := \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(Y)}},$$

where $\text{Cov}(\cdot, \cdot)$ and $\text{Var}(\cdot)$ denote the covariance and respectively the variance of the random variables.

However, by looking at the following example the shortcomings of correlation as a linear measure already become quite clear.

Example 2.36. Let X be a $\mathcal{N}(0, 1)$ -distributed random variable. We want to investigate the correlation between X and X^2 , which should clearly show dependence, but

$$\rho(X, X^2) = \frac{\text{Cov}(X, X^2)}{\sqrt{\text{Var}(X)}\sqrt{\text{Var}(X^2)}} = \frac{\mathbb{E}[X^3] - \mathbb{E}[X]\mathbb{E}[X^2]}{\sqrt{\mathbb{E}[X^2] - \mathbb{E}[X]^2}\sqrt{\mathbb{E}[X^4] - \mathbb{E}[X^2]^2}} = 0.$$

2.3.2 Kendall's τ

There are of course other important measures/concepts for dependence. We will just introduce one of those, called Kendall's τ , which was introduced by Kendall (1938). It belongs to the class of concordance measures.

Let us start by defining the term concordance, which describes a state in which the single parts do not contradict each other.

Definition 2.37. Concordant/Discordant pairs

We say the pairs (x_1, y_1) and (x_2, y_2) are concordant (discordant), if $(x_1 - x_2)(y_1 - y_2) > 0$ ($(x_1 - x_2)(y_1 - y_2) < 0$).

When describing the dependence structure of two random variables with copulas a convenient class of measures are the so called concordance measures. They are characterised by some useful properties.

Definition 2.38. Concordance measure

A concordance measure is a map I from the bivariate distributions with continuous marginals to the unit interval fulfilling the following conditions:

- i) Symmetry: For all random variables X, Y it holds $I(X, Y) = I(Y, X)$.
- ii) Coherence: Let $C_1 \prec C_2$ be two copulas, then $I(C_1) \leq I(C_2)$.
- iii) Independence: Let X, Y be independent random variables, then $I(X, Y) = 0$.
- iv) Sign change: For all random variable X, Y we have $I(X, -Y) = -I(X, Y)$.
- v) Convergence: Let $(X_n, Y_n)_{n \in \mathbb{N}}$ be a sequence of random vectors converging in distribution to (X, Y) , then $\lim_{n \rightarrow \infty} I(X_n, Y_n) = I(X, Y)$.

The probably best known concordance measures are Kendall's τ and Spearman's ρ . We will introduce and give the most important properties of the first.

Definition 2.39. Kendall's τ

Let (X, Y) be a random vector and (\tilde{X}, \tilde{Y}) be its independent copy. Kendall's τ is defined as

$$\tau(X, Y) = \mathbb{P}[(X - \tilde{X})(Y - \tilde{Y}) > 0] - \mathbb{P}[(X - \tilde{X})(Y - \tilde{Y}) < 0]. \quad (2.4)$$

One can read this definition as the difference of the probabilities of the random variables being concordant and discordant.

Theorem 2.40. Let X and Y be two continuous random variables with dependence structure given by the copula C . Then Kendall's τ for this two random variables is given by

$$\tau(X, Y) = 4 \int \int_{I^2} C(u, v) dC(u, v) - 1. \quad (2.5)$$

The integral in Equation (2.5) can be interpreted as the expected value of the copula with respect to the uniformly $[0, 1]$ distributed random variables U and V with joint distribution function C , i.e. $\tau(X, Y) = 4\mathbb{E}[C(U, V)] - 1$.

Further properties of Kendall's τ

Besides fulfilling the condition of a concordance measure Kendall's τ has further interesting properties:

1. For all random variables X it holds $\tau(X, X) = 1$.
2. τ is invariant under (almost surely) strict monotone increasing transformation.

Especially, the second property is of importance as copulas share the same property, but it does not hold for linear correlation.

Chapter 3

Vine Copula Specification

In this chapter, we will develop the graphical tool of regular vines. In combination with copulas, regular vines have proven to be useful tools in multi-dimensional dependence modelling. The initial idea was already developed by Joe (1996) and later Bedford and Cooke (2001) developed the graphical tool here presented. The presentation in this chapter is following the one in Beare and Seo (2015). The regular vines enable us to split the multivariate distribution of our time series into an expression only depending on the univariate distributions and the (conditional) bivariate copulas.

3.1 Regular Vines

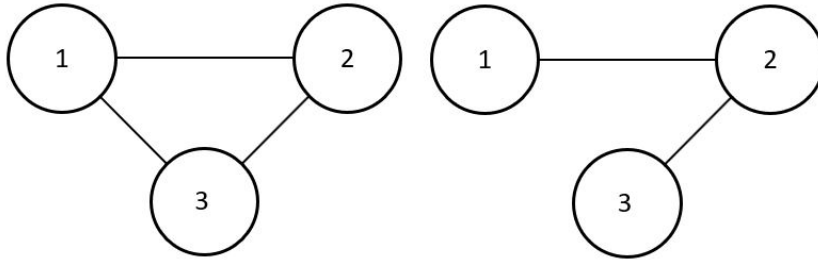
As a prerequisite for defining vines we need the concepts of graphs.

Definition 3.1. Graph

A graph G is a ordered pair (V, E) of vertices V and edges E consisting of two-elements subsets of V .

We call such a graph acyclic, if there exists no path forming a circle.

Example 3.2. Consider the two graphs on the vertices $\{1, 2, 3\}$ in Figure 3.1. The graph on the left is a cyclic graph and the other one without the edge $(3, 1)$ is acyclic.

Figure 3.1: Cyclic and acyclic graphs on $\{1, 2, 3\}$.**Definition 3.3. Tree**

A d -dimensional tree $T = (V, E)$ is a connected acyclic graph with d vertices V and $d - 1$ edges E , where $d \geq 2$.

After having reviewed these tools we are now able to define regular vines.

Definition 3.4. Vine

A collection of trees $\mathcal{V} = (T_1, \dots, T_{d-1})$ on a set V_1 with d -elements is called a vine if

- i) T_1 is a tree with vertices V_1 and edges E_1 and
- ii) for $i = 2, \dots, d - 1$, T_i is a tree with vertices E_{i-1} .

We call it a **regular vine** if in addition

- iii) (**proximity**) for $i = 2, \dots, d - 1$, a_1, a_2, b_1 and b_2 vertices in T_{i-1} :
 $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$ vertices in T_i connected by an edge $\implies a \cap b$ is a singleton

holds.

Remark 3.5. The definition of regular vines implies that two vertices $\{a_1, a_2\}$ and $\{b_1, b_2\}$ are connected, if the corresponding edges $\{a_1, a_2\}$ and $\{b_1, b_2\}$ in the previous tree share a vertex. However, not all should/can be connected, because this would result in a cyclic graph. This problem arises whenever the shared vertex is connected with three or more edges in the previous tree. In those cases one has to decide which connection to drop.

The most common examples for regular vines are the so called Drawable- (D -) and Canonical- (C -) vines. We call a regular vine D -vine if each vertex in T_1 is at most connected via two edges. Concerning C -vines every vertex is connected to the so called root vertex in each tree. For a better understanding of the concept of regular vines, we consider the following example of those two special cases.

Example 3.6. Let us consider a set of five vertices $\{1, 2, 3, 4, 5\}$. Figure 3.2 below displays the graphical representation of a D-vine (left) and C-vine (right).

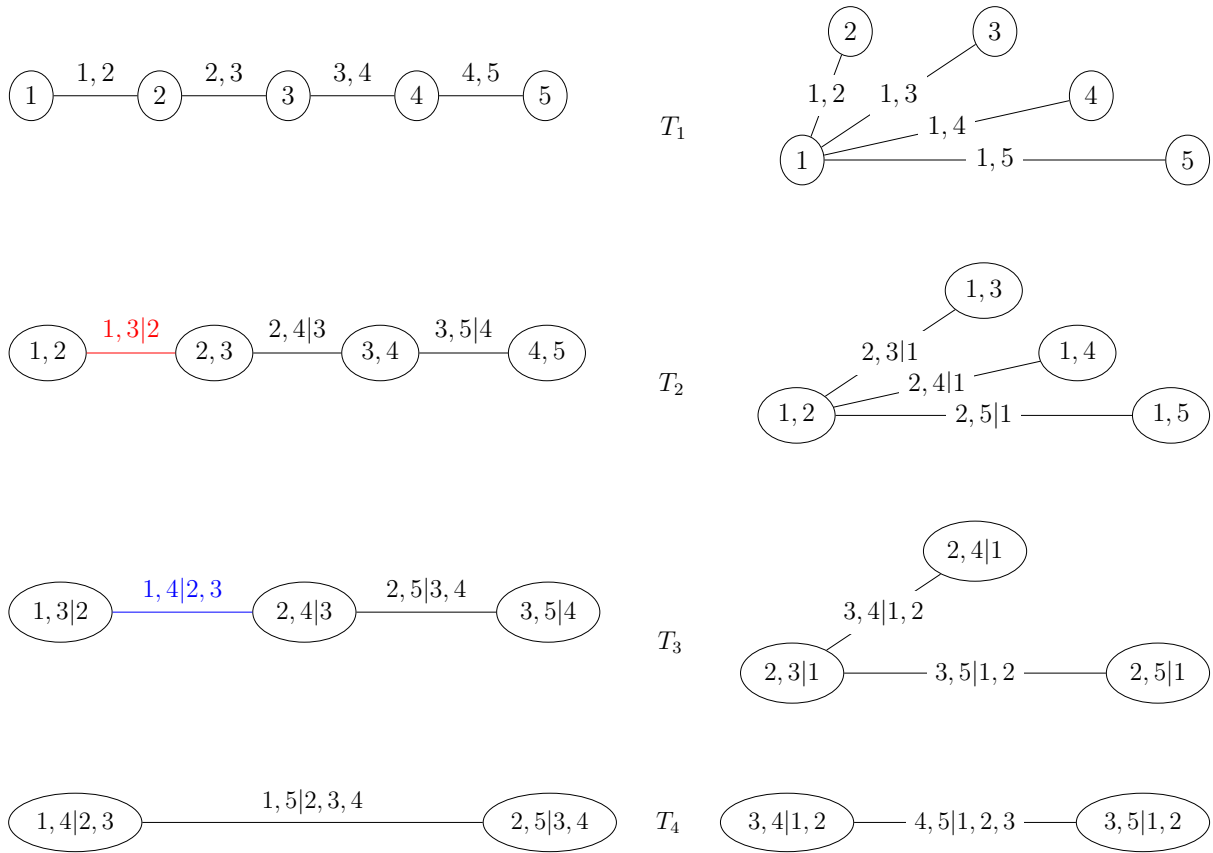


Figure 3.2: Trees of the five dimensional D- (left) and C-vine (right).

Note that by definition of the D-vine there will not arise the problem mentioned in Remark 3.5 and the trees are uniquely determined by the regular vine definition.

However, by just looking at the first tree of the C-vine case in Example 3.6. the building of the next tree just following the definition of regular vines is not unique. Besides the choice of the new central vertex (changing this would still lead to a C-vine) the problem mentioned in Remark 3.5 arises. The following Figure 3.3. illustrates all possible connection between the vertex of the second tree, with **green highlighted** chosen connections of the C-vine on the left and an alternative **orange highlighted** choice fulfilling the regular vine definition as well on the right (note, that this is actually a D-vine).



Figure 3.3: Two highlighted choices out of all possible connections in the second tree of regular vine with starting tree equal to the C-vine.

There are even more possible choices, defining a regular vine structure on the same starting tree, but we are not interested in further describing those structures here. At this point we only want to recall the two most important findings:

1. Given the starting tree of a C-vine the regular vine is not unique and
2. all other possible structures (without the central vertex in each tree) will be treated as more general regular vines in the rest of this thesis.

The possible choices of the vine structure depend on the number of elements in the set. On a set with strictly less than four vertices, all possible vines are *C* and *D* vines. On a set with cardinality four, all possible vines are C- or D-vines. For higher cardinality there can be more interesting structures like the generalised vines we are going to study in more detail within the next chapter of this thesis.

The labelling of vertices of regular vines can be best described by using the concepts of conditioned and conditioning sets (compare Beare and Seo (2015)). To define those concepts we need the so called complete union.

In the following let $\mathcal{V} = (T_1, \dots, T_{d-1})$ be a regular vine on V_1 and $e_k = (u_k, v_k)$ an edge of T_k for $1 \leq k \leq d - 1$.

Definition 3.7. Complete Union

The complete union of e_k is given by

$$U_{e_k} = \begin{cases} e_k & \text{for } k = 1 \\ \{i \in V_1 \mid i \in e_1 \in e_2 \in \dots \in e_k \text{ for some } (e_1, \dots, e_{k-1}) \in E_1 \times \dots \times E_{k-1}\} & \text{for } k \geq 2 \end{cases}$$

and for a singleton $i \in V_1$ it is given by the singleton, i.e. $U_i = \{i\}$.

Definition 3.8. Conditioning Set

The set $D_{e_k} = U_{u_k} \cap U_{v_k}$ is called conditioning set of e_k .

Definition 3.9. Conditioned Set

The respective conditioned set is then given by (a_k, b_k) , where $a_k = U_{u_k} \setminus D_{e_k}$ and $b_k = U_{v_k} \setminus D_{e_k}$.

The labelling of the edges (vertices) follows the same idea as in most literatures. The edge with conditioned set (a_e, b_e) and conditioning set D_e is labelled with $(a_e, b_e|D_e)$.

Example 3.6. continued

The edges in Figure 3.2 are already labelled. We want to explain the labelling of the edges based on the above defined principles. Therefore, we will go through all trees of the D-vine (Figure 3.2 (left)).

- T_1 : Obviously, the complete unions U_1, \dots, U_5 are given by actually different singletons. Therefore, the four edges have empty conditioning sets. Hence, the conditioned sets are equal to the edges and are given by $(U_i, U_{i+1})_{i=1, \dots, 4} = (1, 2), (2, 3), (3, 4), (4, 5)$.
- T_2 : Let us consider the first **red highlighted** edge $((1, 2)(2, 3))$ The complete union is given by $\{1, 2, 3\}$, the conditioning set by $2 = \{1, 2\} \cap \{2, 3\}$ and the conditioned set by $(1, 3) = \{1, 2, 3\} \setminus \{2\}$. Hence, the edge is labelled with **$(1, 3|2)$** .
- T_3 : Let e_3 be the **blue highlighted** edge connecting $(1, 3|2)$ and $(2, 4|3)$. This results in a conditioning set $D_{e_k} = U_{(1,3|2)} \cap U_{(2,4|3)} = \{1, 2, 3\} \cap \{2, 3, 4\} = \{2, 3\}$ and conditioned set $(U_{(1,3|2)} \setminus D_{e_k}, U_{(2,4|3)} \setminus D_{e_k}) = (\{1, 2, 3\} \setminus \{2, 3\}, \{2, 3, 4\} \setminus \{2, 3\}) = (1, 4)$. Yielding the label **$(1, 4|2, 3)$** for e_3 .
- T_4 : Following the very same steps as above, the edge is labelled with $(1, 5|2, 3, 4)$.

3.2 Pair Copula Construction

As already mentioned in the section about copulas we want to use the given vine structure, respective (bivariate) copulas and univariate distributions to express the corresponding multivariate distribution. We first focus on decomposing a multivariate density into a cascade of pair copulas. This general decomposition is mainly following Aas et al. (2009). Afterwards, we will specify these results for R-vine structures which is mainly following Kurowicka and Cooke (2006).

3.2.1 General Scheme

Assume the d -dimensional random vector $\mathbb{X} = (X_1, \dots, X_d)^T$ has the joint density given by the function $f(x) := f(x_1, \dots, x_m)$. Further, let us denote the respective marginal distributions by $f_i(x_i)$ and the conditional marginal distributions by $f_{i|j}(x_i|x_j)$. The multivariate density can be decomposed in the following way:

$$f(x) = f_d(x_d) \cdot f_{d-1|d}(x_{d-1}|x_d) \cdot f_{d-2|d-1,d}(x_{d-2}|x_{d-1}, x_d) \cdot \dots \cdot f_{1|2,\dots,d}(x_1|x_2, \dots, x_d). \quad (3.1)$$

As a consequence of Theorem 2.3 we already presented a representation of a multivariate distribution using the copula density for the two dimensional case. This can easily be extended to higher dimensional cases, yielding:

$$f(x) = c(F_1(x_1), \dots, F_d(x_d)) \prod_{i=1}^d f_i(x_i), \quad (3.2)$$

where the capital F is referring to the corresponding cumulative distribution function and c denotes the respective copula density.

In the next step we will combine the results (3.2) and (3.1), however, for readability reasons we only consider the case $d = 3$:

$$f(x) \stackrel{(3.1)}{=} f_3(x_3) \cdot \underbrace{f_{2|3}(x_2|x_3)}_{\star} \cdot \underbrace{f_{1|2,3}(x_1|x_2, x_3)}_{\star\star}, \quad (3.3)$$

where \star and $\star\star$ can be re-expressed by using (3.2) and $f_{i|j}(x_i, x_j) = \frac{f_{ij}(x_i, x_j)}{f_j(x_j)}$ as follows:

$$\star: f(x_3, x_2) \stackrel{(3.2)}{=} c_{23}(F_2(x_2), F_3(x_3)) \cdot f_2(x_2) f_3(x_3)$$

dividing both sides by $f_3(x_3)$ yields:

$$f_{2|3}(x_2|x_3) = c_{23}(F_2(x_2), F_3(x_3)) \cdot f_2(x_2) \text{ and similarly}$$

$$\star\star: f(x_1, x_2|x_3) \stackrel{(3.2)}{=} c_{12|3}(F_{2|3}(x_2|x_3), F_{1|3}(x_1|x_3)) \cdot f_{2|3}(x_2|x_3) f_{1|3}(x_1|x_3)$$

dividing both sides by $f_{2|3}(x_2|x_3)$ yields:

$$f_{1|2,3}(x_1|x_2, x_3) = c_{12|3}(F_{2|3}(x_2|x_3), F_{1|3}(x_1|x_3)) \cdot f_{1|3}(x_1|x_3).$$

The last term on the right-hand side can be again decomposed by using the same arguments as in \star , finally yielding

$$f_{1|2,3}(x_1|x_2, x_3) = c_{12|3}(F_{2|3}(x_2|x_3), F_{1|3}(x_1|x_3)) \cdot c_{13}(F_1(x_1), F_3(x_3)) \cdot f_1(x_1).$$

Hence, the complete pair copula decomposition in (3.3) is given by

$$\begin{aligned} & f_3(x_3) \cdot f_{2|3}(x_2|x_3) \cdot f_{1|2,3}(x_1|x_2, x_3) \\ = & f_3(x_3) \cdot \underbrace{c_{23}(F_2(x_2), F_3(x_3)) f_2(x_2)}_{\text{using } \star} \cdot \underbrace{c_{12|3}(F_{2|3}(x_2|x_3), F_{1|3}(x_1|x_3)) c_{13}(F_1(x_1), F_3(x_3)) f_1(x_1)}_{\text{using } \star\star} \\ = & f_1(x_1) f_2(x_2) f_3(x_3) \cdot c_{23}(F_2(x_2), F_3(x_3)) \cdot c_{12|3}(F_{2|3}(x_2|x_3), F_{1|3}(x_1|x_3)) \cdot c_{13}(F_1(x_1), F_3(x_3)). \end{aligned}$$

Remark 3.10. Note that the decomposition described before is not unique. There are several different possibilities of decompositions in (3.1), for example in the three dimensional case given in (3.3) there are six possibilities. Furthermore, the decomposition used for $\star\star$ is also not unique.

Having found the decomposition for $d = 3$, we can easily extend this approach to $d \in \mathbb{N}$. Iteratively we can decompose (3.1) by the general formula:

$$f_{i|\delta}(x_i|x_\delta) = c_{i,j|\delta_{-j}}(F(x_i|\delta_{-j}), F(x_j|\delta_{-j})) \cdot f_{i|\delta_{-j}}(x_i|x_{\delta_{-j}}),$$

where $i \in \{1, \dots, d\}$, δ a m -dimensional subset with $m \leq d$ and $\delta \subset \{x_1, \dots, x_d\}$ and δ_{-j} the $(m - 1)$ -dimensional set without the j -th element. Especially, this means that we are able to reduce the conditioning set by using pair copulas. But this approach also involves conditional cumulative distribution functions of the form $F_{i|\delta_{-j}}$, which can be evaluated by using the formula

$$F_{i|\delta}(x_i, x_\delta) = \frac{\partial \mathcal{C}_{i,j|\delta_{-j}}(F_{i|\delta_{-j}}(x_i|x_{\delta_{-j}}), F_{j|\delta_{-j}}(x_j|x_{\delta_{-j}}))}{\partial F_{j|\delta_{-j}}(x_j|x_{\delta_{-j}})},$$

proved by Joe (1996).

Applying these formulas iteratively to the conditional factorisation of an arbitrary density function f , given in (3.1), is called pair copula construction and yields a factorisation of f in its univariate margins and bivariate (conditional) copulas.

3.2.2 R-Vine Pair Copula Construction

A common way of arranging the density factorisation of a pair copula construction is to use the above mentioned regular vines. This was introduced by Bedford and Cooke (2001). There have been several fields of applications and as we are focusing on financial data within this thesis a good review of current research can be found in Aas (2016). In this section, we will denote the dimension by n instead of d as it is not directly related to our vine model.

Now, we will see how to arrange a pair copula construction using R-vines.

Definition 3.11. R-vine copula specification

The triplet $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ is a R-vine copula specification if

- i) $\mathbb{F} = (F_1, \dots, F_n)^T$ is a vector of continuous and strictly increasing univariate cumulative distribution functions,

- ii) $\mathcal{V} = (T_1, \dots, T_{n-1})$ is a regular vine and
- iii) $\mathcal{C} = \{C_e | e \in E(\mathcal{V})\}$ is the set of absolute continuous bivariate copulas corresponding to an edge in \mathcal{V} .

Assume we have an n -dimensional random vector $\mathbb{X} = (X_1, \dots, X_n)^T$ with distribution function F . We say, that F realises a R-vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ if and only if for all $i = 1, \dots, n$ $X_i \sim F_i$ and for all edges $i, j | \delta$ in $E(\mathcal{V})$, $C_{i,j | \delta}$ is the bivariate copula joining $X_i | \mathbb{X}_\delta$ and $X_j | \mathbb{X}_\delta$.

Theorem 3.12. A given R-vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ is realised by a unique distribution F given by the density

$$f = \prod_{i=1}^n f_i \prod_{i,j | \delta \in E(\mathcal{V})} c_{i,j | \delta}(F_{i | \delta}, F_{j | \delta}).$$

Proof. See Bedford & Cooke (2001). □

From this density we can also calculate the conditional cumulative distribution functions recursively:

For $(i, j) \in E(T_1)$ we have $\delta = \emptyset$ and hence $F_{i | \delta} = F_i$ and $F_{j | \delta} = F_j$ respectively. For $(i, j) \in E(T_k)$, $k \geq 2$ we have

$$F_{i | \delta} = \frac{\partial C_{i',j' | \delta'}(F_{i' | \delta'}, F_{j' | \delta'})}{\partial F_{j' | \delta'}} \text{ and } F_{j | \delta} = \frac{\partial C_{i'',j'' | \delta''}(F_{i'' | \delta''}, F_{j'' | \delta''})}{\partial F_{j'' | \delta''}},$$

where (i', j') and (i'', j'') are the edges in $E(T_{k-1})$, which are connected via (i, j) , with $i = i'$ and $j = j''$.

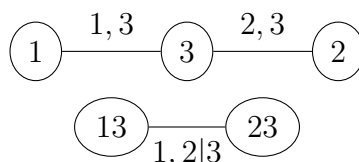
In Aas et al. (2009) we can find the simplified density of D- and C-vines, which are given by

$$f_D = \prod_{i=1}^n f_i \prod_{j=1}^{n-1} \prod_{k=1}^{n-j} c_{k,k+j | k+1, \dots, k+j-1}(F_{k | j+1, \dots, k+j-1}, F_{k+j | k+1, \dots, k+j-1})$$

and respectively

$$f_C = \prod_{i=1}^n f_i \prod_{j=1}^{n-1} \prod_{k=1}^{n-j} c_{j,j+k | 1, \dots, j-1}(F_{j | 1, \dots, j-1}, F_{j+k | 1, \dots, j-1}).$$

Remark 3.13. The complete pair copula decomposition of (3.3) presented in Subsection 3.2.1 is given by the 3-dimensional D-vine:



3.3 Matrix Notation of Regular Vines

For implementation, we need a convenient way to store the information of regular vines. Dissmann et al. (2012) proposed using so called regular vine matrices, which capture all relevant information on the respective vine within a matrix.

All information needed to describe a R-vine copula specification on V_1 , with $|V_1| = n$, can be stored in $(2 + x) n \times n$ -dimensional low triangular matrices, where x denotes the number of parameters of the allowed copula families. These matrices are given by:

- M , which stores all information on the tree structures,
- F , storing the copula families of the specification, and
- P_1, \dots, P_x , containing the respective parameters of the copula families.

Having M we should be able to reconstruct the trees of the regular vine and vice versa. Each edge $e_k = (a_k, b_k | D_{e_k})$ of T_k is described by the entries $(m_{i,j})_{i,j=1,\dots,d}$ of M in the following way: The conditioned set is given by $(m_{d-k+1,i}, m_{i,i})$ and the conditioning set by $\{m_{d-k+2,i}, \dots, m_{d,i}\}$ or the empty set if $k = 1$.

The easiest way to understand the structure of those matrices is by looking at an example. Hence, we will now have again a look at the five dimensional D-vine already

discussed in Example 3.5. The corresponding matrix is given by $M = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 2 & 1 & 3 & 0 & 0 \\ 3 & 2 & 1 & 2 & 0 \\ 4 & 3 & 2 & 1 & 1 \end{bmatrix}$.

For the explanation, we will again look at the two red/blue highlighted edges from T_2/T_3 in Figure 3.2.

- $(1, 3|2)$ is captured by the third column of M ($i = 3$). The conditioning set is given by $(m_{4,3}, m_{3,3}) = (1, 3)$ and the conditioning set by $\{m_{5,3}\} = \{2\}$.
- Taking a look at the second column we find for the conditioned set $(m_{3,2}, m_{2,2}) = (1, 4)$ the conditioning set $\{m_{4,2}, m_{5,2}\} = \{2, 3\}$, giving us the edge $(1, 4|2, 3)$ in T_3 .

Next, we want to formally define M .

Definition 3.14. R-Vine Matrix (Compare Dissmann et al. (2012))

A low triangular matrix $M = (m_{i,j})_{i,j=1,\dots,n}$ is a regular vine matrix if for $j = 1, \dots, n-1$ and for all $i = j+1, \dots, n-1$, there exists a $k \in \{i+1, \dots, n-1\}$ such that:

$$(m_{j,i}, \{m_{j+1,i}, \dots, m_{n,i}\}) \in B_M(j) \cup \tilde{B}_M(j),$$

where

$$B_M(j) = \{(m_{i,i}, D) | j = i+1, \dots, n; D = \{m_{j,i}, \dots, m_{n,i}\}\} \text{ and}$$

$$\tilde{B}_M(j) = \{(m_{j,i}, D) | j = i+1, \dots, n; D = \{m_{i,i}\} \cup \{m_{j+1,i}, \dots, m_{n,i}\}\}.$$

Remark 3.15. This definition assures several useful properties. First of all it covers the proximity condition in Definition 3.4. Further, it states that all entries in one column need to be different and deleting the first row and first column of a regular vine matrix yields again a regular vine matrix with dimension reduced by one.

Having defined and explained the regular vine matrix we now focus on the corresponding other matrices describing the copula vine specification. As we only want to give an idea we only give one reference matrix P instead of all possible parameters P_1, \dots, P_x in the following. Let the matrix M be given by $(m_{i,j})_{i,j=1,\dots,n}$, then the copula family matrix, assigning a copula family to every Copula $(C_e)_{e \in E(\mathcal{V})}$, with $e = (a_k, b_k | D_{e_k})$ is given by the $n \times n$ -matrix:

$$F = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ F_{C(m_{1,1}, m_{2,1} | m_{3,1} : m_{n,1})} & 0 & \dots & 0 & 0 \\ F_{C(m_{1,1}, m_{3,1} | m_{4,1} : m_{n,1})} & F_{C(m_{2,2}, m_{3,2} | m_{4,2} : m_{n,2})} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ F_{C(m_{1,1}, m_{n-1,1} | m_{n,1})} & F_{C(m_{1,2}, m_{n-1,2} | m_{n,1})} & \ddots & 0 & 0 \\ F_{C(m_{1,1}, m_{n,1} | \emptyset)} & F_{C(m_{2,2}, m_{n,2} | \emptyset)} & \dots & F_{C(m_{n-1, n-1}, m_{n, n-1} | \emptyset)} & 0 \end{bmatrix}$$

with parameters in the $n \times n$ -matrix:

$$P = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ P_{C(m_{1,1}, m_{2,1} | m_{3,1} : m_{n,1})} & 0 & \dots & 0 & 0 \\ P_{C(m_{1,1}, m_{3,1} | m_{4,1} : m_{n,1})} & P_{C(m_{2,2}, m_{3,2} | m_{4,2} : m_{n,2})} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{C(m_{1,1}, m_{n-1,1} | m_{n,1})} & P_{C(m_{1,2}, m_{n-1,2} | m_{n,1})} & \ddots & 0 & 0 \\ P_{C(m_{1,1}, m_{n,1} | \emptyset)} & P_{C(m_{2,2}, m_{n,2} | \emptyset)} & \dots & P_{C(m_{n-1, n-1}, m_{n, n-1} | \emptyset)} & 0 \end{bmatrix}$$

For our D-vine example we get the respective copula family matrix:

$$F = \begin{bmatrix} 0 & & & & \\ FC_{(1,5|2,3,4)} & 0 & & & \\ FC_{(2,5|3,4)} & FC_{(1,4|2,3)} & 0 & & \\ FC_{(3,5|4)} & FC_{(2,4|3)} & FC_{(1,3|2)} & 0 & \\ FC_{(4,5)} & FC_{(3,4)} & FC_{(2,3)} & FC_{(1,2)} & 0 \end{bmatrix} \text{ with parameters}$$

$$P = \begin{bmatrix} 0 & & & & \\ PC_{(1,5|2,3,4)} & 0 & & & \\ PC_{(2,5|3,4)} & PC_{(1,4|2,3)} & 0 & & \\ PC_{(3,5|4)} & PC_{(2,4|3)} & PC_{(1,3|2)} & 0 & \\ PC_{(4,5)} & PC_{(3,4)} & PC_{(2,3)} & PC_{(1,2)} & 0 \end{bmatrix} .$$

Chapter 4

General Regular Vine Matrices for Multivariate Time Series

In this chapter, we will have a look at different approaches for implementing regular vine structures to multivariate time series data, covering cross-sectional, serial and cross-serial (conditional) dependence. We will review the already existing approaches in the literature and based on the M-vine approach of Beare and Seo (2015), we will develop a new and more flexible model.

4.1 Introduction into Regular Vine Models for Multivariate Time Series

As prerequisite for a precise description of regular vine models for multivariate time series (RVMMVTS) we need the term cross-sectional (vine) structure.

Definition 4.1. Cross-sectional (Vine) Structure

We call the vine structure describing only the cross-sectional dependence within one (each) time step $t \in T$ cross-sectional structure (at time t). The respective cross-sectional matrix is given by the regular vine matrix describing this structure.

In general, all RVMMVTS's follow the same idea. In the following, we assume a multivariate time series with sectional dimension d and N time steps, i.e. $T = \{1, \dots, N\}$. The first tree of the RVMMVTS is given by the d -dimensional cross-sectional structures at time t and $t + 1$ for $t = 1, \dots, N - 1$ linked via one edge connecting a vertex from the structure at t and one from the one at $t + 1$. Even though, one could allow for different cross-sectional structures and links, we will assume a recurring structure in each

step as this enables us to use our model for prediction. The Figure 4.1 below visualises the scheme described above on the basis of a $d = 2$ dimensional M-vine (Beare and Seo (2015)) or COPAR (Copula Autoregressive by Brechmann and Czado (2015)) example. The **red highlighted** cross-sectional structures at each time point t are linked via the **blue highlighted** connecting edges.

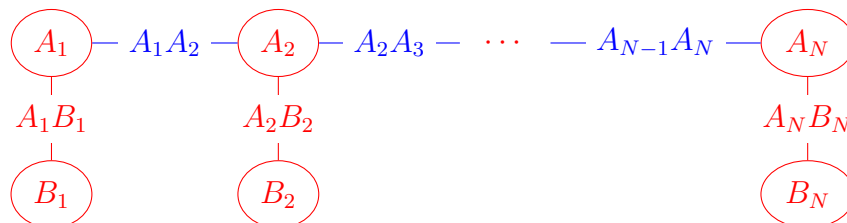


Figure 4.1: Example for first tree of RVMMVTS: M-vine/COPAR time series approach with sectional dimension two and N time steps.

Remark 4.2. The starting tree of COPAR and M-vine are the same in this case, but the succeeding trees will be different and therefore the matrix representations will deviate from each other. For more details we refer to Section 4.3.3.

In the following sections, we will especially look at the already existing M-vine (Beare and Seo (2015)), long D-vine (Smith (2015)) and the COPAR (Brechman and Czado (2015)) approaches. The first two approaches only allow for a description of the cross-sectional (conditional) dependence with D-vines, while the last approach is only applicable to cross-sectional C-vine structures. Our aim is to generalise the classical M-vine approach to build one model for all possible cross-sectional vine structures.

We will cover the above mentioned approaches by studying 5×3 -dimensional examples for cross-sectional D-, C- and more general R-vine structures in the following sections. Note that the $d = 5$ dimensional case is the smallest possible sectional dimension, in which a non-trivial (meaning non C- nor D-vine) representation of the cross-sectional vine structure in one time step arises. The time dimension $N = 3$ was chosen, because of a practical useful autoregressive order of two, which will be further analysed in Chapter 7 of this thesis.

Block Structure in Matrix Representation

For the interpretation of the (overall) structure we basically use two block types within the matrix representation. The later called **cross-sectional triangular structures** and the

parallelogramm structures. Note that those structures will not arise in the COPAR approach.

Definition 4.3. Cross-Sectional Triangular Structures

The triangles in the last $d - 1$ rows, given by the respective sub-diagonal part of the cross-sectional vine structure matrix at time $t \in T$, describing the cross-sectional dependence for each time step t , are called cross-sectional triangular structures.

Definition 4.4. Parallelogram Structures of Order p

The parallelograms in between the diagonal and cross-sectional triangular structures, describing the (conditional) dependence of the cross-sectional structure at time t to the cross-sectional structure at time $t - p$, where t denotes the time of the respective diagonal entries, are called parallelogram structures of order p .

The Figure 4.2 below visualises the two structures types within the sub-diagonal of the matrix representation. In general the copulas assigned to the cross-sectional triangular structures for each t and the parallelogram structures of one order will not be identical. However, they become identical when assuming stationarity. This will be discussed in more detail in Chapter 7.

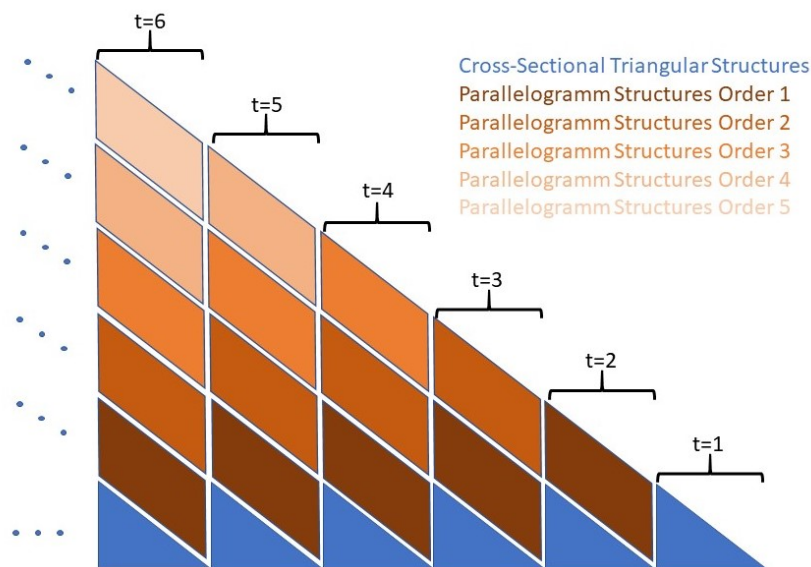


Figure 4.2: Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series.

Having developed all needed terminology to describe regular vine models for multivariate time series, we can now start with the examples for the respective cross-sectional structures.

4.2 Cross-Sectional D-Vine Structure

In this section, we assume that the cross-sectional dependence and therefore the multivariate distribution in each time step can be best covered by a D-vine structure. This especially means, that the matrix representation of the cross-sectional structure at each

time point t is given by:

$$\begin{bmatrix} E_t & & & & & \\ A_t & D_t & & & & 0 \\ B_t & A_t & C_t & & & \\ C_t & B_t & A_t & B_t & & \\ D_t & C_t & B_t & A_t & A_t & \end{bmatrix}.$$

4.2.1 Classical M-Vine Approach

A way of linking the D-vine structures is the so called M-vine approach presented by Beare and Seo (2015). In this approach the serial dependence is captured by one sectional time series over time, i.e. in our example A_t is connected to A_{t+1} , building a long D-vine connecting the cross-sectional structures.

The respective vine structure of this approach will result again in a D-vine after d steps, where d is equal to the dimension of the cross-sectional structure, as in each new tree within the vine specification the edges of the cross-sectional structures become the vertices of the connecting D-vine. The easiest way to understand this building of trees is by looking at a graphical example. In our case the cross-sectional dimension is five, therefore Figure 4.2 below only considers the first five trees of the whole M-vine structure. For a better understanding of the changes from tree to tree, we further highlight and explain the recurring changes in each step.

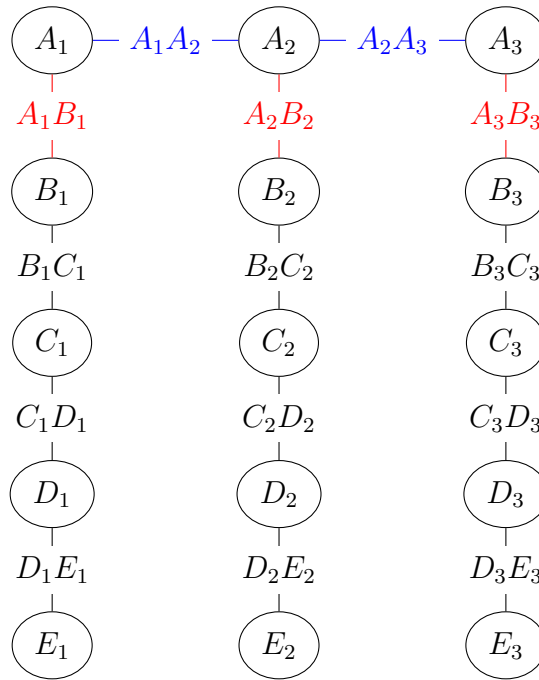
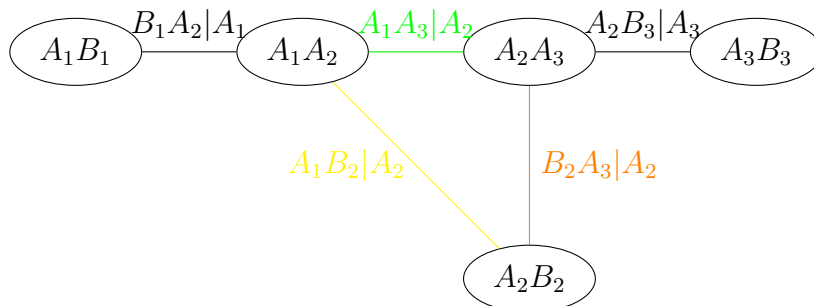


Figure 4.3: a) First tree of M-vine time series approach for three time steps and five sectional dimensions.

As we can see in the starting tree the connecting long D-vine is given by the edges A_1A_2 and A_2A_3 . In the second tree those two edges become vertices of the long D-Vine, i.e. stay as a part of the long D-vine.

Within the trees of the cross-sectional structure we only **highlighted in red** the edges $(A_tB_t)_{t=1,2,3}$. Those edges also become vertices of the connecting long D-vine while the other edges stay within the area of the cross-sectional structure. Especially, the dimension of the area of the cross-sectional structures reduces by one in every step while the connecting long D-vine is growing.

Note, that this choice is not uniquely determined by the regular vine definition. Looking at the edges $\{(A_iB_i), (A_iA_j)\}_{i=1,2,3 \text{ and } j=2,3}$, leaving all would lead to a cyclic graph, hence, we have to drop one of those three coloured edges in the following graph:



In the classical M-vine model, we drop the solely cross-serial connection $A_1A_3|A_2$. Instead one could also drop one of the cross-sectional and cross-serial connections $A_1B_2|A_2$ or $B_2A_3|A_2$.

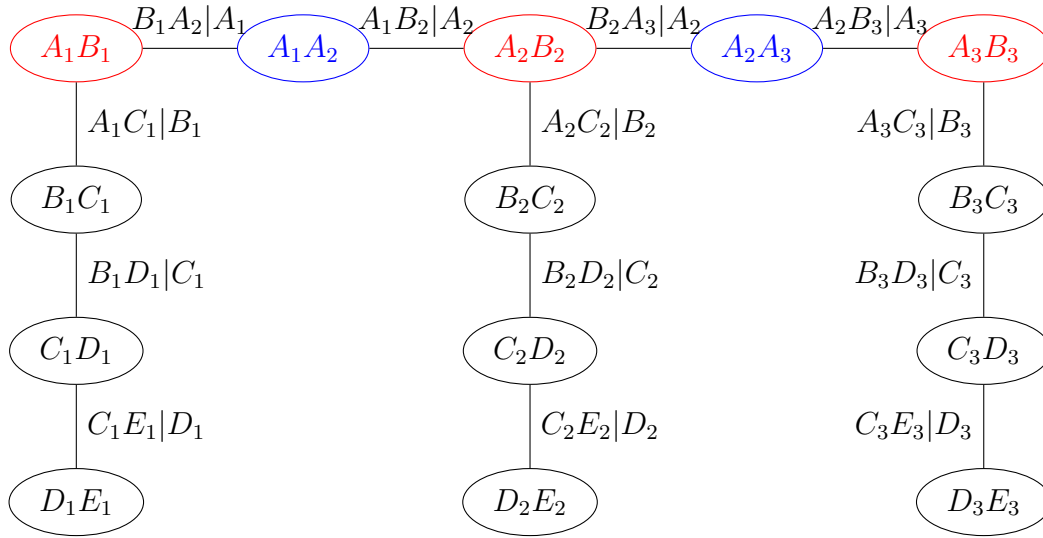


Figure 4.3: b) Second tree of M-vine time series approach for three time steps and five sectional dimensions.

In the next step we see the very same pattern as described above. That is why we do not again highlight and describe the changes in the following trees, i.e. the edges $(A_tC_t|B_t)_{t=1,2,3}$ in the second tree become vertices of the long D-vine in the third.

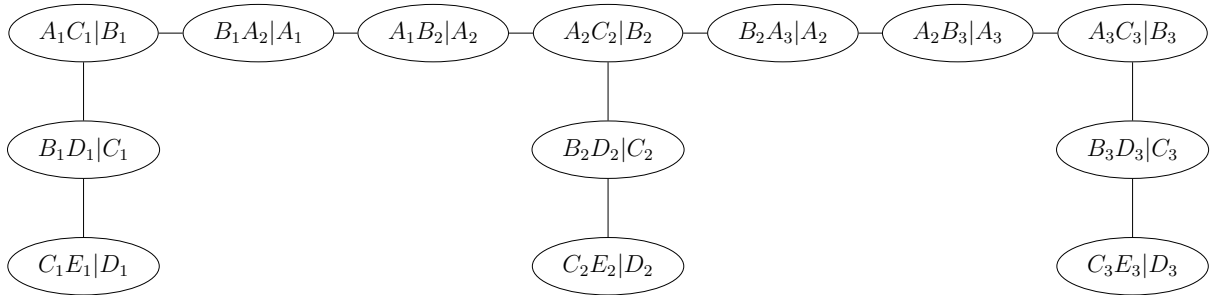


Figure 4.3: c) Third tree of M-vine time series approach for three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.

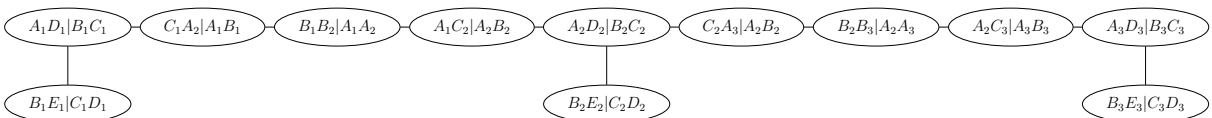


Figure 4.3: d) Fourth tree of M-vine time series approach for three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.

- **trinagular structure** reflecting the cross-sectional D-vine structure matrix, i.e. consisting of the sub-diagonal entries of the structure matrix, and
- the **sub-diagonal structure** in between (consisting of the above mentioned parallelogram structures indicated by the **yellow** for order one and **orange** for order two boxes) with descending sectional dimension E_{t-p} to A_{t-p} for each order $p = 1, 2$ and time point t of the respective diagonal entry.

Using these rules, the concrete example can easily be extended to higher dimensions both in time and in sectional dimension, compare Section 4.5. A more detailed overview of the M-vine approach and its properties can also be found in Teuma Manekeng (2017).

4.2.2 Comparison with D-Vine Approach

Another well-known approach is just aligning the cross-sectional D-vines to a new longer D-vine, i.e. linking the last vertex of the cross-sectional structure at time t (here E_t) to the first one at time $t + 1$ (here A_{t+1}). This model was described in Smith (2015) and a detailed review is given in Heuke (2016). Even though the influence of changing the connecting edges will be analysed in detail in Chapter 6, we already want to mention this example here.

The respective matrix for this approach with highlighted parallelogram structures of order one (**in yellow**) and order two (**in orange**) is given by the matrix (4.2).

Having a first look at the possible parallelogram structure of order one, brown surrounded in (4.3), it is the same as in the classical M-vine matrix (4.1), but only in the first two columns from the right. However, (4.3) does not have any recurring parallelogram structures nor the below triangular structure as already mentioned in the description of Figure 4.2. This may be a reason why using similar approaches for building the trees as in the COPAR approach for other cross-sectional vine structures, e.g. a D-vine structure, do not deliver a comparable COPAR structure in the corresponding matrix representation.

The respective trees of the vine structure for our 5×3 -dimensional example are displayed in Figure 4.5 below, using the *RVineMatrix()*-function from the *VineCopula*-package implemented in the statistical software **R**. For more details of the development of the trees from step to step we refer to Section 4.3.3. A detailed review of COPAR models can also be found in Ivanov (2016).

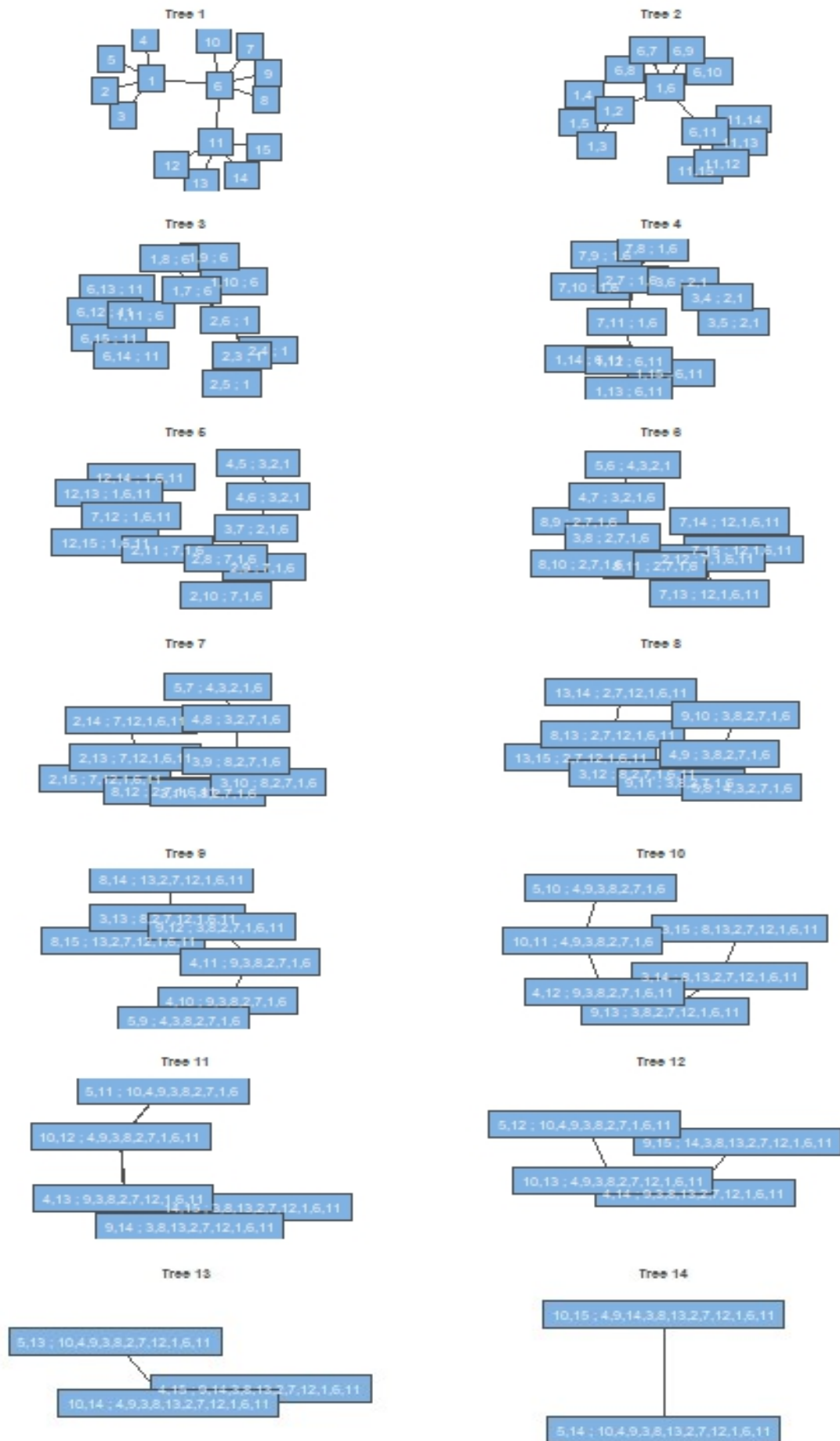


Figure 4.5: Copar time series approach for three time steps on five dimensions.

4.3.2 Generalised M-Vine Approach for Cross-Sectional C-Vine

Given the same starting tree, meaning cross-sectional C-vines connected via the time series of $(A)_{t=1,2,3}$ we are also able to build up a different vine structures fulfilling the regular vine definition. We focus on respective steps from tree to tree similar to the classical M-vine approach, discussed in Section 4.2.1, here.

Again the tree structure will result in a D-vine after d steps, where d denotes the dimension of the cross-sectional C-vine structure, here $d = 5$. The Figure 4.6 below displays the first five steps of the complete vine structure.

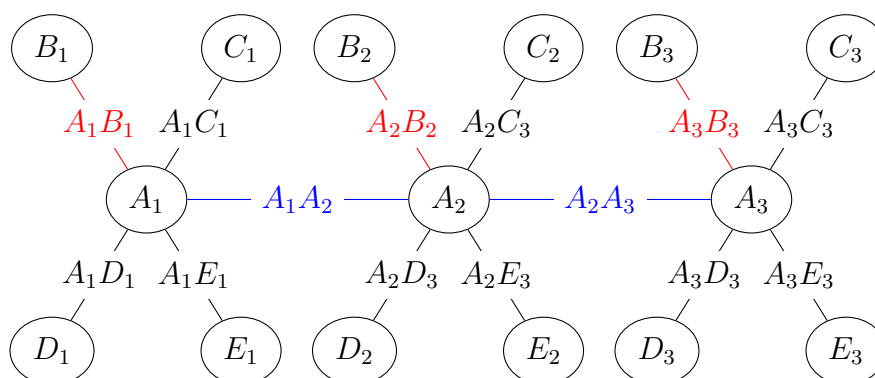


Figure 4.6: a) First tree of generalised M-vine time series approach for cross-sectional C-vine structure, three time steps and five sectional dimensions.

Again we see in the starting tree the connecting long D-vine is given by the edges A_1A_2 and A_2A_3 . In the second tree those two edges become vertices of the long D-Vine, i.e. stay as a part of the long D-vine.

Within the trees of the cross-sectional structure we only **highlighted in red** the edges $(A_tB_t)_{t=1,2,3}$. Those edges also become vertices of the connecting long D-vine while the other edges stay within the area of the cross-sectional structure. Especially, the dimension of the area of the cross-sectional structures reduces by one in every step while the connecting long D-vine is growing. To have a unique building of the successive trees, the new central vertex becoming part of the long D-vine is given by the new central-vertex of the respective cross-sectional C-vine. Here, we assume that those edges are ordered by the sectional dimension, i.e. from B to E . Note that this is no limitation, as we can easily obtain such a structure by relabelling the respective vertices in the cross-sectional structure. This choice leads to a connecting long D-vine only consisting of A_i 's and B_i 's in the conditioned sets/complete unions in the second tree of our vine structure.

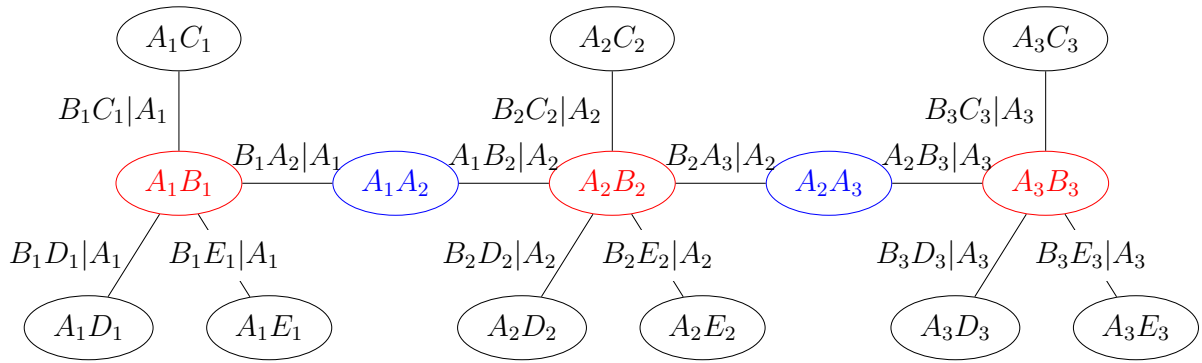


Figure 4.6: b) Second tree of generalised M-vine time series approach for cross-sectional C-vine structure, three time steps and five sectional dimensions.

In the next step we see the very same pattern as described in the first step. That is why we do not highlight and describe the changes in the following trees again, i.e. the edges $(B_t C_t | A_t)_{t=1,2,3}$ in the second tree become vertices of the long D-vine in the third. This results in a long connecting D-vine consisting of A_i 's, B_i 's and C_i 's in the complete unions of the next tree.

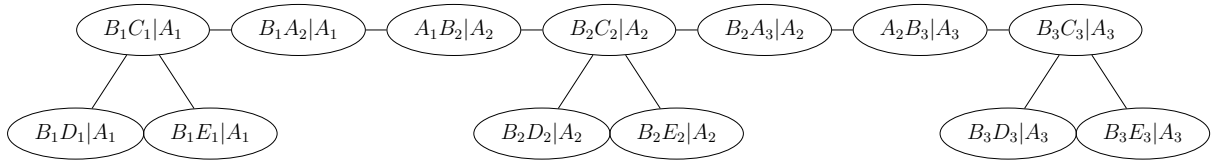


Figure 4.6: c) Third tree of generalised M-vine time series approach for cross-sectional C-vine structure, three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.

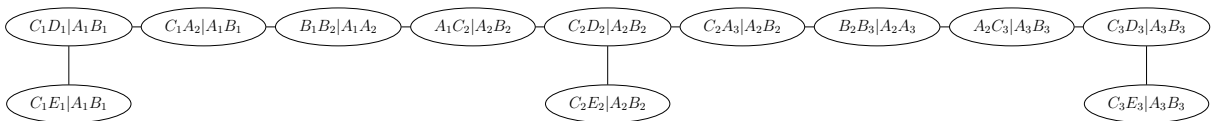


Figure 4.6: d) Fourth tree of generalised M-vine time series approach for cross-sectional C-vine structure, three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.

After the next step (Tree 5) the complete generalised M-Vine for the cross-sectional C-vine is now described by the connecting long D-vine. Printing all vertices is quite unreadable therefore the last part of Figure 4.5 below only gives an idea of the remaining structure.

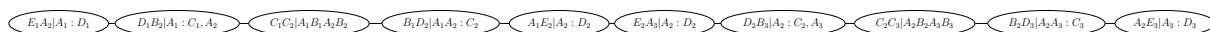


Figure 4.7: Sixth tree of classical **and** generalised M-vine in three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.

They are identical and a D-vine (i.e. the building of the next trees is unique) meaning that all following trees need to be identical, too. This of course results in an identical matrix representation as presented by the part above the separation line in (4.4) and the respective part in (4.1).

Now, we know that for the extreme case (D- and C-vine) the matrix representation of the (generalised) M-vine only differs with respect to the below triangular structure. This should also be true for all possible cross-sectional regular vines in between and will be further analysed in Section 4.4.

4.3.3 Deep-Dive Differences COPAR and (Generalised) M-Vine

Before turning to the more general structure we want to highlight the differences from the already existing approach for cross-sectional C-vines, the COPAR approach, to our newly created generalised M-vine approach.

As the new created approach significantly deviates from the existing approach we will especially take a closer look at how the respective trees of the vine specification are built. We will therefore go through the trees of a rather small example. Let us assume that we have $N = 3$ time steps and sectional dimension $d = 4$. The starting tree for both approaches is then given by:

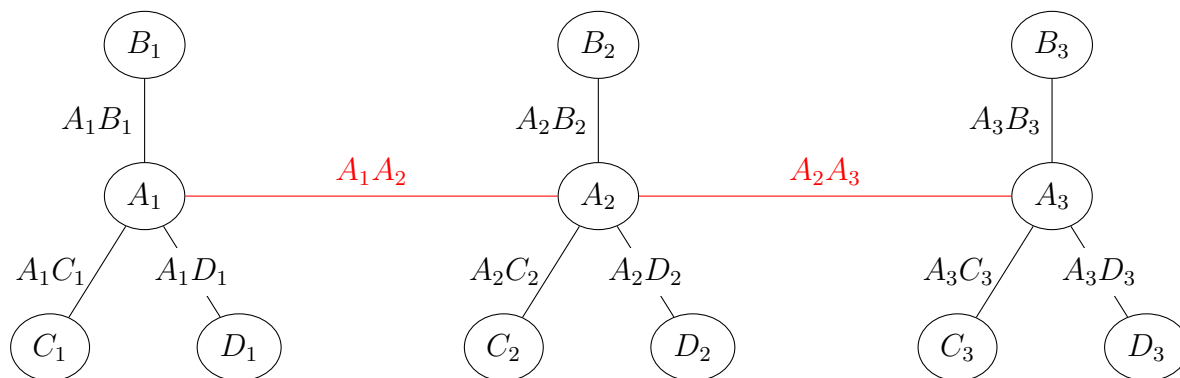
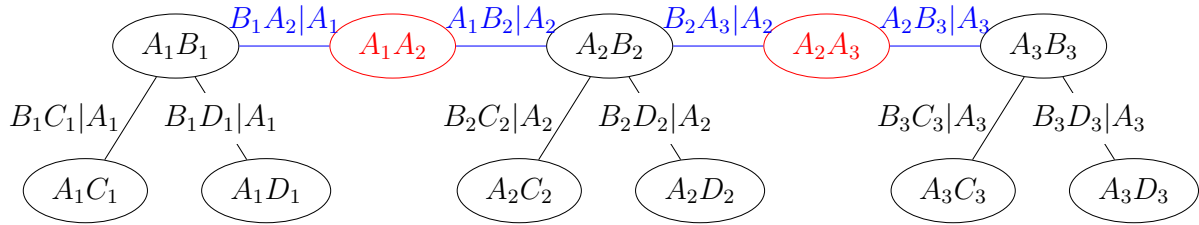


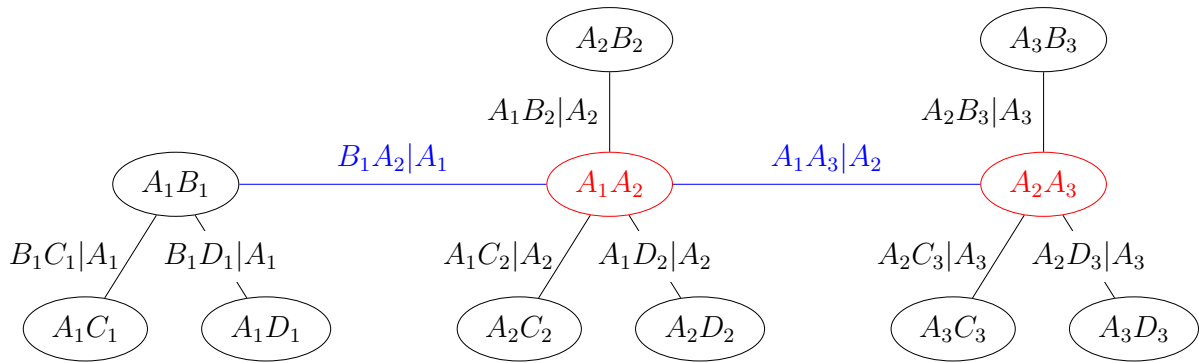
Figure 4.8: a) First tree of cross-sectional four dimensional C-vine.

First Difference

While in the COPAR approach the cross-serial edges (A_1A_2 and A_2A_3) become the new central vertices of the C-vines, in the generalised M-vine approach they are moved in between the cross-sectional structures.



i) Generalised M-vine tree 2



ii) COPAR tree 2

Figure 4.8: b) Second trees of i) generalised M-vine and ii) COPAR approach for cross-sectional four dimensional C-vine.

When looking again at the edges connecting the serial dependence, we find that for the COPAR approach the first edge ($B_1A_2|A_2$) is now moved in between the structures and the second ($A_1A_3|A_2$) becomes again the new central vertex.

In our new approach the **connecting edges in between** still become vertices in between as in the first step.

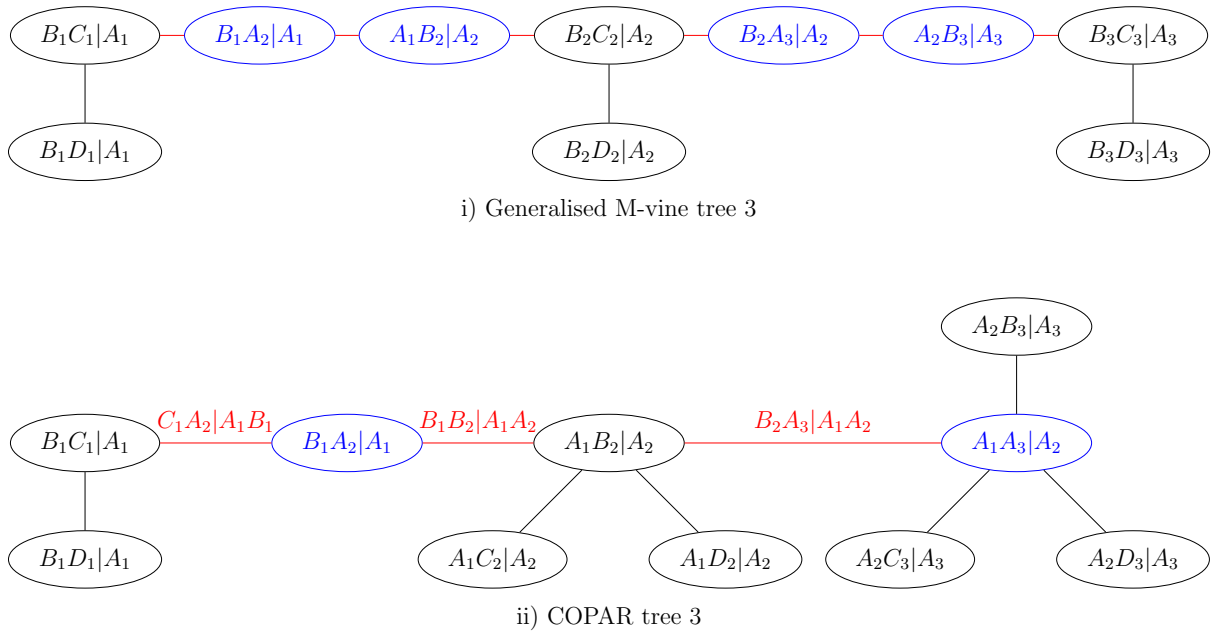


Figure 4.8: c) Third trees of i) generalised M-vine and ii) COPAR approach for cross-sectional four dimensional C-vine. Because of readability we dropped the labelling of the edges we are not referring to.

Second Difference

While the new approach already results in a D-vine within the next step, the classical COPAR approach still needs three further steps before resulting in a D-vine. For the **serial connecting edges** in the third COPAR tree we see a mixed picture the first ($C_1A_2|A_1B_1$) stays as a vertex in between the structures, the second ($B_1B_2|A_1A_2$) becomes a new central vertex and the last ($B_2A_3|A_1A_2$) becomes a vertex in between the C-vine structures.

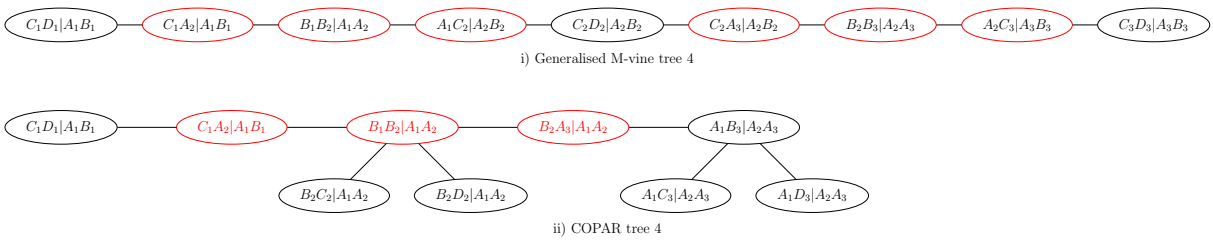


Figure 4.8: d) Fourth trees of i) generalised M-vine and ii) COPAR approach for cross-sectional four dimensional C-vine. Because of readability we dropped the labelling of the edges.

We will stop with the detailed comparison here and refer to the respective functions mentioned in the appendix for the further trees.

Summary of Differences

To sum up, besides the useful feature of resulting in a D-vine after d steps, where d denotes the sectional dimension, the rules for building the trees within the new approach is more straightforward. This is a huge advantage when trying to implement this approach for more general structures than C- or D-vines, covered in the next section. It also simplifies further generalisations discussed in Chapter 6. However, of course there was a reason for choosing COPAR models before. The COPAR approach supports modelling structures where one time series is driven by the time series of the other factors, e.g. so called factor models. This strong descriptiveness is therefore not covered within our generalised approach. As we do not want to restrict us, but allow for any possible connection between the single observations within one time step, this is no real shortfall for our model anyhow.

Case: Sectional Dimension $d = 2$

As already shortly mentioned in the introduction of this chapter, for sectional dimension two the starting tree of the COPAR model is equal to the tree of the classical M-Vine, which is by definition equal to our generalised M-Vine as the cross-sectional structure is identical.

For completeness of our comparison, we use this case with $N = 3$ to compare also the COPAR with the classical M-vine approach. The development of the trees is given in the Figure 4.8 below.

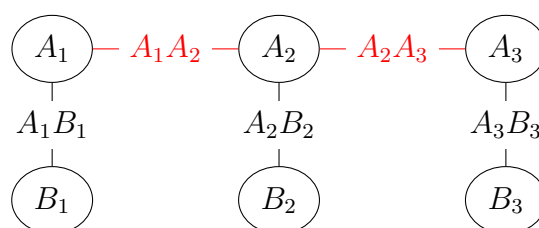


Figure 4.9: a) First tree of the M-vine/COPAR time series approach with sectional dimension two and $N = 3$ time steps.

We have the very same interpretation as in the case above: In the COPAR approach the cross-serial edges (A_1A_2 and A_2A_3) become the new central vertices of the C-vines. Especially the edge (A_2, B_2) prevents the structure from resulting in a D-vine already. In the M-vine approach the cross-serial edges become part in between the cross-serial edges of the resulting D-vine.

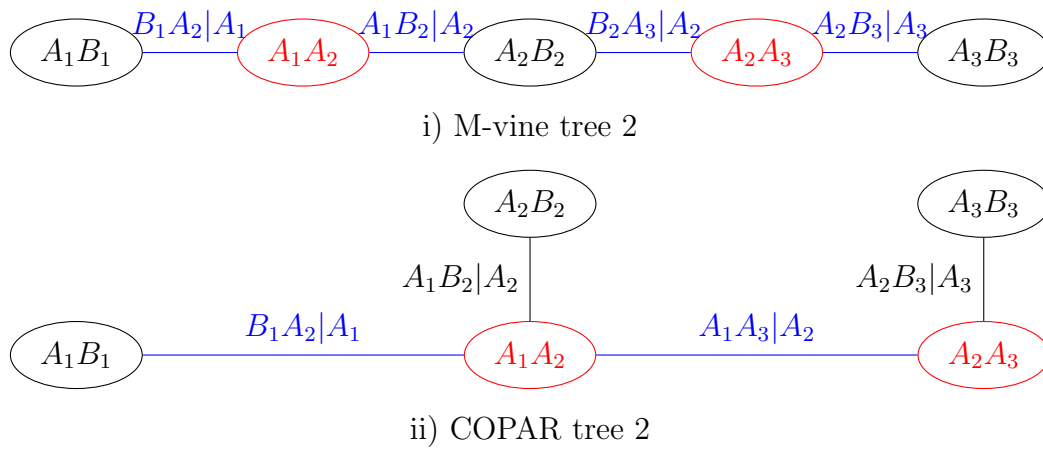


Figure 4.9: b) Second trees of i) M-vine and ii) COPAR approach for sectional dimensional two and $N = 3$ time steps.

When looking again at the edges connecting the serial dependence, we find that for the COPAR approach the edges $(B_1A_2|A_2)$ and $(A_1A_3|A_2)$ now become part of the long D-vine. Compared to the M-vine case this D-vine deviates in two vertices and this deviation will remain until the last tree.

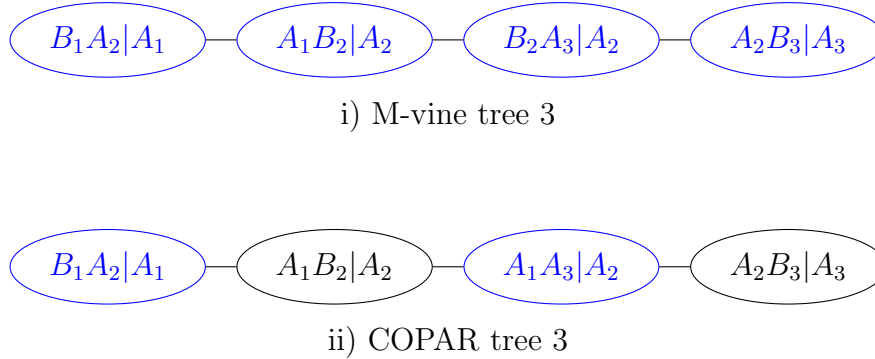


Figure 4.9: c) Third trees of i) M-vine and ii) COPAR approach for sectional dimensional two and $N = 3$ time steps.

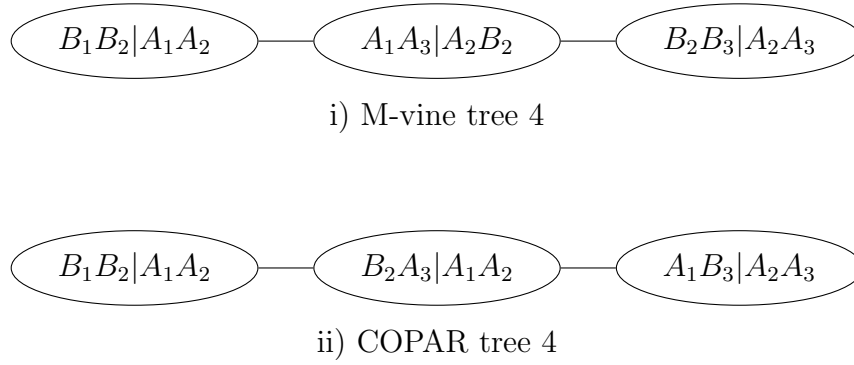


Figure 4.9: d) Fourth trees of i) M-vine and ii) COPAR approach for sectional dimensional two and $N = 3$ time steps.

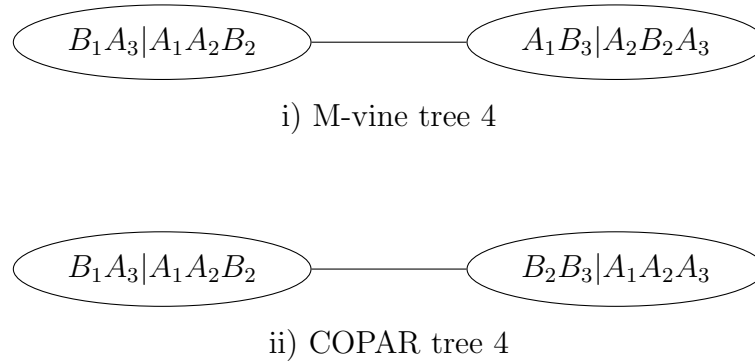


Figure 4.9: e) Last trees of i) M-vine and ii) COPAR approach for sectional dimensional two and $N = 3$ time steps.

4.4 More General Cross-Sectional R-Vine Structure

After having discussed the (generalised) M-vine approach for the cross-sectional D- and respectively C-vine in Subsections 4.2.1 and 4.3.2 we will transfer this method to general cross-sectional regular vine structures in this section. As the C- and D-vine can be seen as the extreme cases of the structures we expect a similar result for our overall matrix as already mentioned in 4.3.2, i.e. only a change in the cross-sectional triangular structure. In our five dimensional example there arises exactly one (neglecting possible relabelling of sectional dimension) non C- nor D-vine starting tree, which can be seen in Figure 4.10 below. Recall that there are other more general regular vine structures (as a result of choosing the higher order trees) in the five-dimensional case as already discussed in Section 3.1 of this thesis.

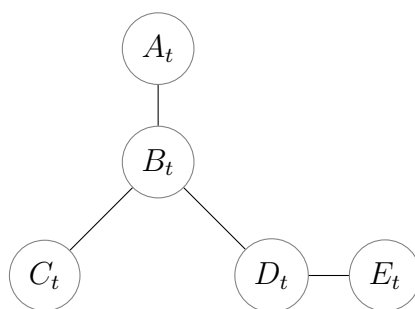


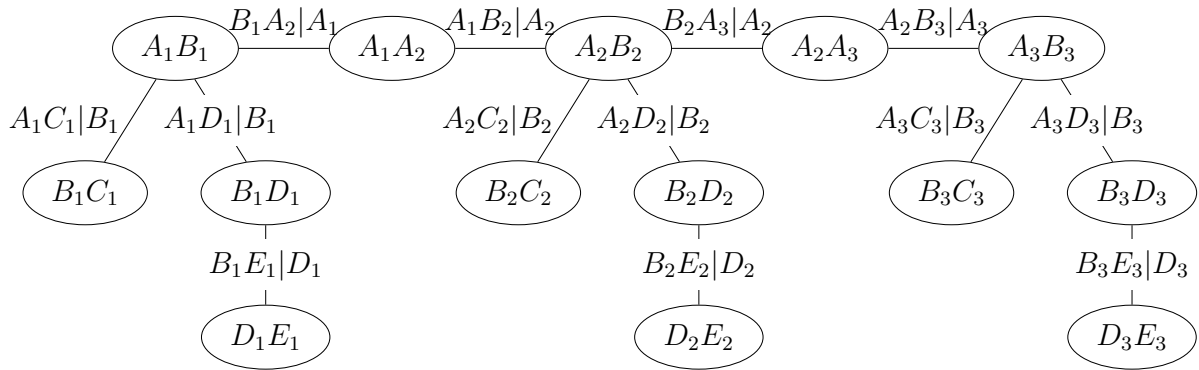
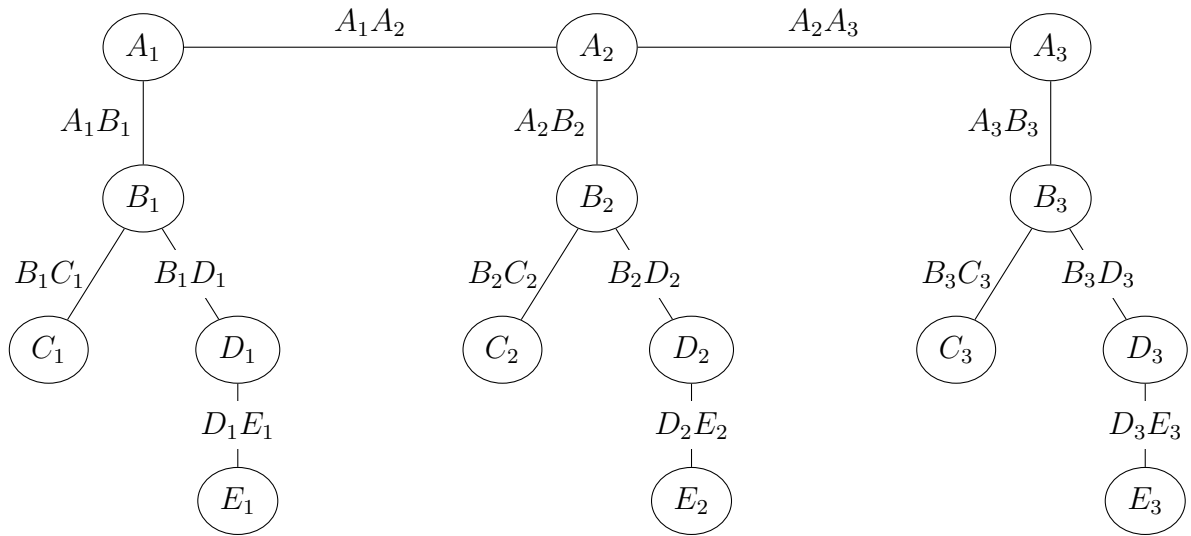
Figure 4.10: First tree of five dimensional regular vine.

The corresponding matrix is given by:

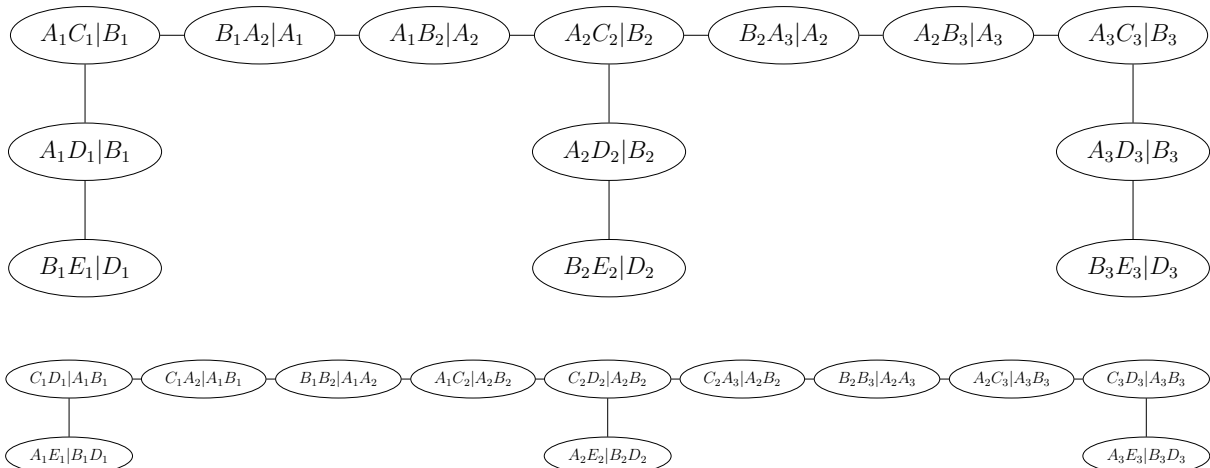
$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix}.$$

The serial dependence will again be captured by one sectional time series over time, namely A_t is connected to A_{t+1} .

As we use the same approach for building the trees as in the two (generalised) M-vine cases before, the structure will again result in a D-vine structure after $d = 5$ (dimension of the cross-sectional structure) steps. We further expect the sixth tree to be identical to sixth tree pictured in Figure 4.7. Figure 4.11 below therefore illustrates only the first six trees of the overall structure for the cross-sectional regular vine as above. We skip the detailed explanation of the tree building here, as it follows the same steps already presented for the D- and C-vine. We want to stress again that, whenever there arises a non-unique decision based on the regular vine definition, this is either clarified by the cross-sectional structure or by the fact that cross-serial edges become/stay a part of the long connecting D-vine.



Note here, that whenever there is more than one edge of the cross-sectional area connected to the long D-vine, we have to decide which edge is becoming a part of the long D-vine in the next step. However, this decision has already been taken in the cross-sectional structure. Especially, we use the common approach of ordering the diagonal of the cross-sectional matrix and therefore the edge with the next sectional dimension will be chosen (here: the edge containing C_t).



4.5 Summary and Algorithm for Generalised M-Vine

Considerations of the five times three dimensional examples, discussed above, can easily be extended to higher dimensions both in time and in the sectional dimension. The general form of each time series matrix can be determined by the cross-sectional regular vine matrix and the number of time points N , i.e. $T = \{1, \dots, N\}$.

As we want to implement an algorithm building such matrices in the statistical software **R**, we need a slightly different notation as before. We have to transform our data $(\mathbb{X}_t)_{t \in T} = (A_t, B_t, C_t, \dots)'_{t \in T}$ into the set of natural numbers $\{1, \dots, Nd\}$. The isomorphism used to map those two notations is explained in the following remark.

Remark 4.5. Isomorph Mapping of Notations

Let Θ be the restriction of $\{A, B, \dots, Z, AA, \dots\}$ to the first d elements, denoting the set of our previous sectional dimensions. We further need the following two functions:

- $f : \{1, \dots, d\} \rightarrow \Theta$, with $1 \mapsto A$, $2 \mapsto B$, ..., $26 \mapsto Z$, $27 \mapsto AA$, ... and
- $g : \mathbb{N} \rightarrow \{1, \dots, d\}$, with $g(x) = \mathbb{I}_{g^*(x) \neq 0} g^*(x) + \mathbb{I}_{g^*(x) = 0} d$, where $g^*(x) = x \bmod d$.

Then each element X_t in our previous notation (i.e. $X \in \Theta$ and $t \in T$) can be mapped to the set $\{1, \dots, Nd\}$ by the function

$$h = \begin{cases} (\Theta, T) & \rightarrow \{1, \dots, Nd\} \\ (X, t) & \mapsto td + f^{-1}(X) \end{cases} \text{ with inverse } h^{-1} = \begin{cases} \{1, \dots, Nd\} & \rightarrow (\Theta, T) \\ Y & \mapsto (f(g(Y)), \lfloor \frac{Y}{d} \rfloor) \end{cases}$$

To better understand this mapping, let us take a look at the following example.

Example 4.6. Let $N = 3$ and $d = 5$. Considering the vertex C_2 we get by the definition above:

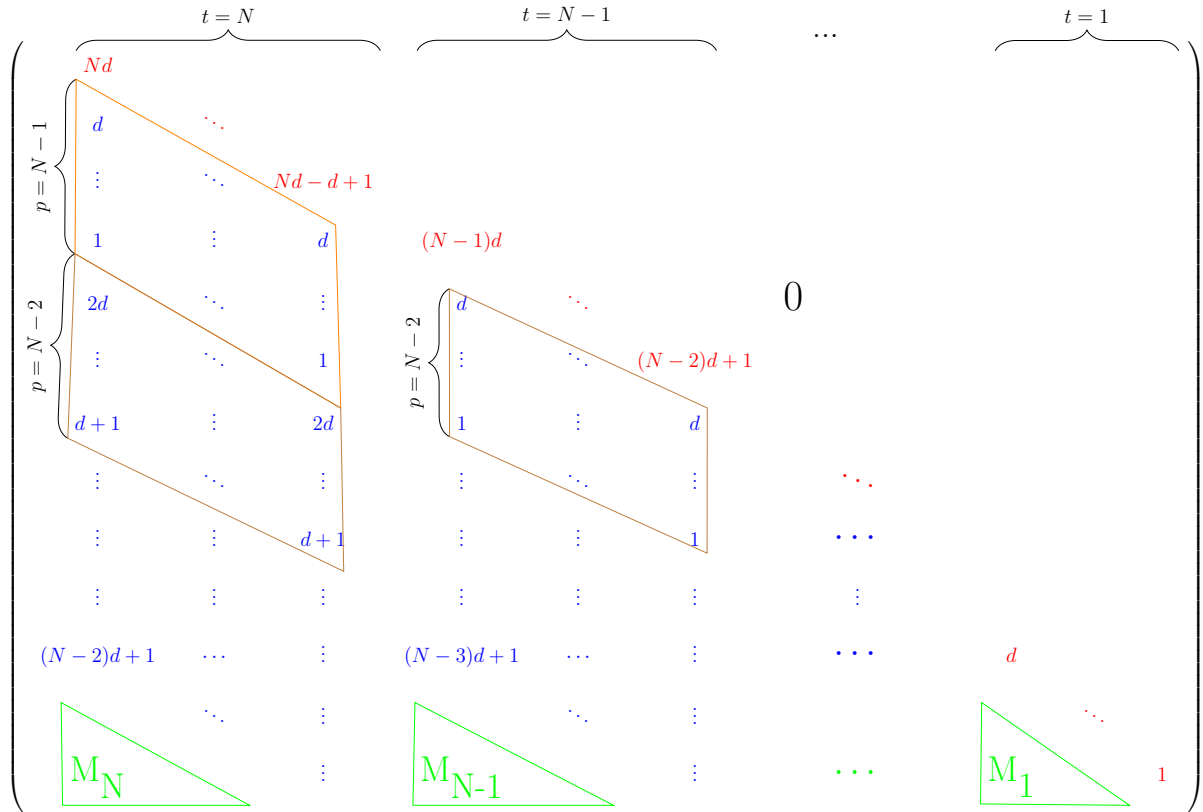
$$h(C, 2) = 2 \cdot 5 + f^{-1}(C) = 10 + 3 = 13 \text{ and } h^{-1}(13) = (f(g(13)), \lfloor \frac{13}{5} \rfloor) = (f(3), 2) = (C, 2).$$

Having the needed notation, we are now able to define the structure of our generalised M-vine matrices. The cross-sectional structure matrix can be described by

$$\begin{bmatrix} dt & & & & \\ & \ddots & & & \\ & & 0 & & \\ \mathbf{M}_t & & & d(t-1) + 1 & \end{bmatrix} \text{ with dimension } d.$$

The patterns of the three matrix parts always stay the same and are in line with the matrix representation in (4.1), (4.4) and (4.5), i.e. only deviating in the cross-sectional triangular structure for a given sectional and time dimensions. In the above developed notation they are given by:

- Descending **diagonal entries** starting from Nd going down to 1 ,
- **cross-sectional triangular structure** reflecting the changes in the cross-sectional dependence, i.e. the sub-diagonal M_t of the structure matrices at t , and
- the **sub-diagonal structure** in between, which is given by the columns of the respective parallelogram structures of order $p = t-1, \dots, 1$ given by $(t-p)d$ to $(t-p-1)d+1$, where t denotes the time of the diagonal entry of the column.



For the respective trees of the (generalised) M-vine structure, we find that the complete unions of the edges in each tree T_k for $k = 1, \dots, d$ are only consisting of the set $\{\cup_{i=1}^k i\}$.

To implement generalised M-vine matrices in \mathbf{R} we use the following algorithm:

Algorithm 1 Vine Structure Time Series Matrix

Input N : number of time steps

csStructure: cross-sectional structure matrix

Output M : generalised M-vine structure matrix

function GENERALISEDMVINE(csStructure, N)

$d \leftarrow$ sectional dimension

$m \leftarrow$ counter for each d columns starting at 1

Triangle \leftarrow sub-diagonal of csStructure

diagonal of $M \leftarrow$ decreasing sequence from Nd to 1

for each column from 1 to $(N - 1)d$ **do**

Sub-diagonal structure of $M \leftarrow (t - p)d$ to $(t - p - 1)d + 1$

with $t = N - m + 1$ and p from $t - 1$ to 1

After every d -th column

Triangle structure of the d columns $\leftarrow (N - m)d + \text{Triangle}$

$m \leftarrow m + 1$

First triangle structure $M_1 \leftarrow \text{Triangle}$

return M

Block Notation

Given the general structure matrix above we are further interest in using a block notation to better understand the sub-diagonal part. We will therefore identify all the diagonal entries at each time step t , i.e. $td : (t - 1)d + 1$, with $\mathbf{diag}(t)$, the parallelogram-structures or order p at t with $t - p$ and the cross-sectional triangular structure at t with M_t as before.

This results in a general representation given by (4.6) below.

$$(4.6) = \left(\begin{array}{cccc} \text{diag}(N) & & & \\ \mathbf{1} & \text{diag}(N-1) & & \\ \mathbf{2} & \mathbf{1} & \text{diag}(N-2) & \\ \vdots & \vdots & \ddots & \ddots \\ \mathbf{N-1} & \mathbf{N-2} & \mathbf{N-3} & \dots \\ M_N & M_{N-1} & M_{N-2} & \dots \end{array} \right) \dots \left(\begin{array}{c} \text{diag}(1) \\ M_1 \end{array} \right)$$

Ignoring the cross-sectional triangular structures $(M_t)_{t=1,\dots,N}$, (4.6) simplifies to

$$\begin{bmatrix} \text{diag}(N) & & & & & \\ \mathbf{1} & \text{diag}(N-1) & & & & \\ \mathbf{2} & \mathbf{1} & \text{diag}(N-2) & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ \mathbf{N-2} & \mathbf{N-3} & \mathbf{N-4} & \dots & \text{diag}(2) & \\ \mathbf{N-1} & \mathbf{N-2} & \mathbf{N-3} & \dots & \mathbf{2} & \text{diag}(1) \end{bmatrix},$$

which is actually just a block D-vine matrix.

This is probably due to the fact, that we connected the cross-sectional vines with the help of a growing long D-vine. Another interesting study would be using e.g. a C-vine for connecting the cross-sectional structures and investigate the effect on the block matrix as displayed above. So, instead of connecting A_i to A_{i+1} for $i = 1, \dots, N-1$, we would choose one central vertex, e.g. A_1 , and connect all other A_i 's to this central vertex.

Chapter 5

Generalised M-Vine

We now have an understanding of the generalised M-vine matrix structure and are even able to build the respective matrix by knowing only the number of time points and the cross-sectional structure matrix. However, we have not yet formally defined our new created structure. In this chapter, we will finally give a formal definition. Further, we will be able to proof the findings of the previous chapter. The formal definition and some of the theorems are generalisations of the theorems in Beare and Seo (2015) and the respective proofs are therefore only adapted to the generalised definition.

5.1 Formal Definition of Generalised M-Vines

Let $N, d \in \mathbb{N}$, where N again denotes the number of time steps and d the dimension of the cross-sectional structure. Recall that the overall aim is to model the multivariate distribution of $(\mathbb{X}_1, \dots, \mathbb{X}_N)$, where $\mathbb{X}_i \in \mathbb{R}^d$ by a regular vine structure, i.e. our generalised M-vine structure. Therefore, consider the regular vines on the vertices $V_1 = \{1, \dots, d\} \times \{1, \dots, N\}$.

Remark 5.1. In the previous chapter we identified the sectional components by letters (A,...,D,...) as this enhances the readability and highlighted the cross-sectional structure. For the formal definition it is more convenient to use also numbers for the sectional dimension. However, this is no problem as we can easily map those two sets via an isomorphism, given by the function f in Remark 4.5.

We further need the concept of restricted regular vines on a subset of those vertices $\tilde{V}_1 \subset V_1$.

Definition 5.2. Restriction of Regular Vines (Compare Beare and Seo (2015))

Let \mathcal{V} be a regular vine on V_1 with trees $T_k = (V_k, E_k) \forall k = 1, \dots, Nd - 1$ and $\tilde{V}_1 \subset V_1$

with $q := |\tilde{V}| \geq 2$. For $k = 1, \dots, q-1$ define $\tilde{E}_k := E_k \cap \binom{\tilde{V}_k}{2}$, $\tilde{T}_k := (\tilde{V}_k, \tilde{E}_k)$ and if $k \geq 2$, $\tilde{V}_k = \tilde{E}_{k-1}$. Then $\tilde{\mathcal{V}} = (\tilde{T}_1, \dots, \tilde{T}_{q-1})$ is called restriction of \mathcal{V} on \tilde{V}_1 .

The graphical interpretation of this definition is straightforward. In the first tree of the complete regular, one deletes the respective vertices not included in the subset. And for the higher trees the respective edges missing because of this deletion will be deleted. Note that a restriction is not necessarily a regular vine.

The last concepts needed to describe the structures of (generalised) M-vines are single and adjacent columns.

Definition 5.3. Single and Adjacent Columns

Let \mathcal{V} be a regular vine on $V_1 = \{(i, s)_{i=1, \dots, d \text{ and } s \in T}\}$, then

- i) $\mathbb{A}_t = \cup_{i=1}^d \{(i, t), (i, t+1)\}$ is called adjacent column and
- ii) $\mathbb{X}_t = \cup_{i=1}^d \{(i, t)\}$ is called single column

at time t .

Before defining the generalised M-vine let us give the definition of the classical M-vine first.

Definition 5.4. Classical M-Vine (Compare Beare and Seo (2015))

A regular Vine \mathcal{V} on V_1 with trees $(N_k, E_k)_{k=1, \dots, Nd-1}$ is called M-vine if and only if:

- i) $E_1 = \{\{(i, s), (j, t)\} \subset \binom{V_1}{2} | (i = j - 1 \text{ and } s = t) \text{ or } (i = j = 1 \text{ and } s = t - 1)\}$ and
- ii) for each pair of adjacent columns \mathbb{A}_t , $t = 1, \dots, N - 1$ the restriction of \mathcal{V} on \mathbb{A}_t is a D-vine.

Obviously, the second part of the definition is not true for the generalised M-vine anymore. Also the first restriction of the set in the first part of the definition does not necessarily hold for the new structure (e.g. if the cross-sectional structure is a C-vine). Therefore, we need to modify this definition covering the new features.

Definition 5.5. Generalised M-Vine

A regular vine \mathcal{V} on V_1 with trees $(N_k, E_k)_{k=1, \dots, Nd-1}$ is called generalised M-vine if and only if:

- i) $E_1 = \{\{(i, s), (j, t)\} \subset \binom{V_1}{2} | ((i, s), (j, t) \text{ are connected via an edge in the cross-sectional structure at time } t \text{ and } s = t) \text{ or } (i = j = 1 \text{ and } s = t - 1)\}$,

ii) for each single column \mathbb{X}_t , $t = 1, \dots, N$ the restriction of \mathcal{V} on \mathbb{X}_t has the same structure, namely the cross-sectional structure and

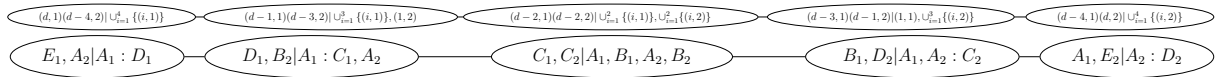
iii) for each restriction of \mathcal{V} on the respective adjacent columns $(\mathbb{A}_t)_{t=1, \dots, N-1}$ the d -th tree is a D-vine, given by the from $j = 1$ to d ordered edges:

$$((d-j+1, t), (j, t+1) | \bigcup_{i=1}^{d-j} \{(i, t)\} \cup \bigcup_{i=1}^{j-1} \{(i, t+1)\})_{j=1, \dots, d}.$$

Remark 5.6. Definition 5.5 is only an extension based on Definition 5.4. However, it is not easy to illustrate. Let us therefore give an interpretation of the single parts (i) to (iii).

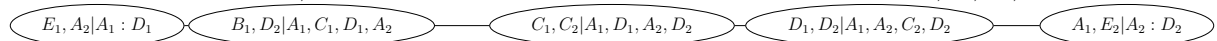
- The first part ensures the structure in the first tree, i.e. the cross-sectional structures connected via the edges connecting the sectional dimension 1 vertices of each time step.
- The second part only describes the fact that in the first d trees the cross-sectional structures for each time step are evolving like the chosen cross-sectional structure.
- The last part is more complex. Besides the fact that it ensures resulting in a D-vine, it also determines which edges to add first to the long D-vine, whenever there is more than one option, i.e. that the long D-vine is ordered according to the sectional dimension. This means the conditioned set in Tree $(d+1)$ is descending in sectional dimension for t and ascending for $t+1$ for all $t = 1, \dots, N-1$ (ensured by the ordering of the edges by j).

Example 5.7. Recall the equivalence of the sixth tree in Figures 4.7 and 4.11, i.e. the edges of the $d(=5)$ -th are the same even if the cross-sectional dependence is captured by different regular vines. Restricted on the adjacent columns \mathbb{A}_1 of any regular vine structure with fixed dimension d the edges are given by:



Recall: Edges of 5-th tree in a $d = 5$ and $N = 2$ generalised M-vine example in the notation of Chapter 4 and 5.

Remark 5.8. The structure of the D-vine in the example above is depending on the order of the edges added to the long D-vine. As already mentioned in Chapter 4 the ordered structure above can be obtained by relabelling the vertices V_1 . In the case, where we add the vertices in the order D, C, B, E we obtain:



i.e. we just have to change the role of B_i and D_i for $i = 1, 2$.

An alternative, but equivalent definition of the generalised M-vine structure, which is probably more intuitive or illustrative is given in the following definition.

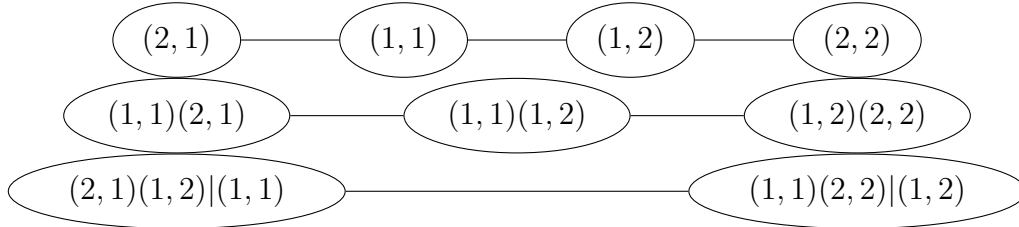
Definition 5.9. Generalised M-Vine (alternative)

A regular vine \mathcal{V} on V_1 with trees $(N_k, E_k)_{k=1, \dots, Nd-1}$ is called generalised M-vine if and only if:

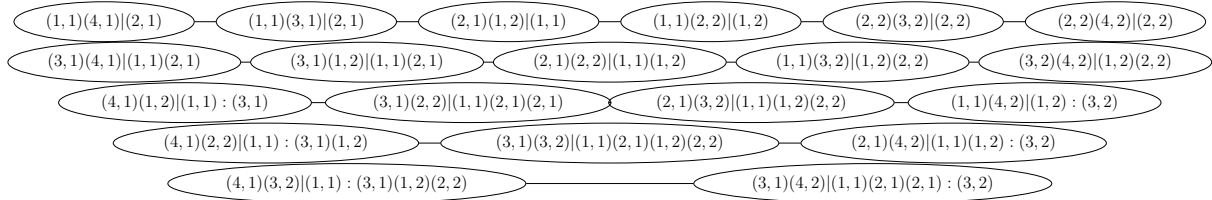
- i) $E_1 = \{ \{(i, s), (j, t)\} \subset \binom{V_1}{2} \mid (i, j) \text{ are connected via an edge in the cross-sectional structure at time } t \}$
- ii) for each single column \mathbb{X}_t , $t = 1, \dots, N$ the restriction of \mathcal{V} on \mathbb{X}_t has the same structure, namely the cross-sectional structure and
- iii) for each $k = 1, \dots, d - 1$ and each restricted pair of adjacent columns $\mathbb{A}_{(t,k)} = \cup_{i=1}^{k+1} \{(i, t), (i, t + 1)\}$, $t = 1, \dots, N - 1$ the restriction of $\tilde{\mathcal{V}}_{k:Nd-1}$ on $\mathbb{A}_{(t,k)}$ is a D-vine, where $\tilde{\mathcal{V}}_{k:Nd-1}$ is denoting the regular vine starting from the $k - th$ tree of \mathcal{V} .

Informally speaking this means that starting in each tree k of the regular vine \mathcal{V} the restriction of this vine on the sectional dimension 1 to $k + 1$ of any adjacent column of this tree and the following $(k + 1) \cdot 2 - k$ trees is a D-vine.

Example 5.10. Recall the trees from Figure 4.5 for $N = 2$. For the case $k = 1$ the D-vine from the alternative definition is given by:



And for the case $k = 3$ it is given by:



Having formally defined and visualised the definition we will next formalise the findings from the previous chapter following Beare and Seo (2015).

5.2 Central Theorems

We have already seen in the previous chapter that the generalised M-vine structure results in a D-vine after d steps. The following theorem formalises this assertion.

Theorem 5.11.

If \mathcal{V} is a generalised M-vine specification of a d -dimensional dataset on V_1 with N time steps, then the tree T_d is a D-vine.

Proof. (Following the steps Beare and Seo (2015))

The aim is to show that each vertex of T_d is connected to at most two vertices.

First note, that for $N \leq 2$ and $d > 0$ the tree T_m of the generalised M-vine is a D-vine due to (iii) in Definition 5.5 and for $d = 1$ and $N > 0$ the first tree of the generalised M-vine is already a D-vine. Thus, we will assume $d \geq 2$ and $N \geq 3$.

For $t = 1, \dots, N - 1$, we denote the restriction of \mathcal{V} on \mathbb{A}_t by \mathcal{V}_t . By the definition of generalised M-vine, we know that \mathcal{V}_t has d edges ($d + 1$ vertices) on tree T_d . Further, the intersection $\mathbb{A}_t \cap \mathbb{A}_{t+1}$ is equal to the single column $\mathbb{X}_{t+1} = \cup_{i=1}^m \{(i, t + 1)\}$. Hence, \mathbb{A}_t and \mathbb{A}_{t+1} share exactly the d vertices of the observations at $t + 1$. Therefore, \mathcal{V}_t and \mathcal{V}_{t+1} have exactly one common vertex on T_d . We will denote this common vertex by a_t .

Obviously, by the regular vine definition T_d has $Nd - d$ edges. Therefore, if we only consider \mathcal{V}_t and \mathcal{V}_{t+1} for $t = 1, \dots, N - 1$, each of those restrictions has $d + 1$ vertices on the d -th tree. However, \mathcal{V}_t and \mathcal{V}_{t+1} share the vertex a_t and the remaining d vertices belong to the tree of \mathcal{V}_t . As we already know from Definition 5.5 \mathcal{V}_t is a D-vine on T_d for all t , i.e. all vertices are connected to at most two vertices.

So, we need to show that also a_t is connected to at most two vertices. Indeed, when considering the generalised M-vine \mathcal{V}_t or \mathcal{V}_{t+1} , a_t has the element $(d, t + 1)$ in its conditioned set, which is of degree one. It follows, considering \mathcal{V} , that a_t has degree at most two, too. \square

We can even say more about the edges of the D-vine. The new important finding - the equality of the D-vine in the $d + 1$ -th tree of the generalised M-vine specification irrespective of the cross-sectional structure - is summarised in the next theorem.

Theorem 5.12. Equality of the D-Vine

The D-vine in tree T_{d+1} of a $(N \times d$ -dimensional) generalised M-vine does not depend on the chosen cross-sectional structure.

Proof. By Definition 5.5 (iii) T_d of \mathcal{V}_t (again denoting the restriction of \mathcal{V} on \mathbb{A}_t) is given by a D-vine with unique edges independent of the cross-sectional structure, i.e. \mathcal{V}_t on T_{d+1} is uniquely determined by the dimension d .

From the proof of Theorem 5.11 we know that T_d of \mathcal{V} is a D-vine and that \mathcal{V}_t and \mathcal{V}_{t+1} for $t = 1, \dots, N - 1$ only share one vertex on the d -th tree, which we denoted by a_t .

The vertex a_t is connected to \mathcal{V}_t by the edge $((1, t), (d, t + 1) | \cup_{i=1}^{d-1} \{(i, t + 1)\})$ and to \mathcal{V}_{t+1} by the edge $((d, t + 1), (1, t + 2) | \cup_{i=1}^{d-1} \{(i, t + 1)\})$. This especially implies that \mathcal{V}_t and \mathcal{V}_{t+1} have no common vertex on tree T_{d+1} , but are connected when looking at \mathcal{V} .

Hence, \mathcal{V} on T_{d+1} is given by just aligning $d + 1$ -th trees of $(\mathcal{V}_t)_{t=1, \dots, N}$. And as the $d + 1$ -th trees of each \mathcal{V}_t are unique and independent from the cross-sectional structure, the same is true for the overall tree. \square

We now know, that for fixed N and d the $(d + 1)$ -th and (as this a D-vine by Theorem 5.11) also the following trees of each $N \times d$ -dimensional generalised M-vine are equal. We further need to show that the vertices of the first d trees only differ from each other when the elements in the conditioned set have the same time index. This is equivalent to saying that the matrix representation of a $N \times d$ -dimensional generalised M-vine for different cross-sectional structures only differ in the cross-sectional triangular structure. This assertion is formalised in the following theorem.

Theorem 5.13. Conditional Cross-Serial Structure

The (conditional) inter-temporal vertices (meaning the elements of the conditioned set having different time indices) on the first d trees of a $N \times d$ -dimensional generalised M-Vine do not depend on the chosen cross-sectional structure.

Proof. Let us again denote the restriction of \mathcal{V} on any adjacent column $(\mathbb{A}_t)_{t=1, \dots, N-1}$ by \mathcal{V}_t . We will denote the k -th tree of this restriction by \tilde{T}_k .

Each tree \tilde{T}_k has $2d - k$ edges by the definition of the regular vine. Further, each pair of \mathcal{V}_t and \mathcal{V}_{t+1} share exactly $d - k$ edges on T_k , denoting the k -th tree of the unrestricted vine. Obviously, those shared edges cannot be (conditional) inter-temporal edges due to the requirement of different time indices in the conditioned set. Therefore, we will only consider \mathcal{V}_t for any $t = 1, \dots, N - 1$ in the following.

For the first tree \tilde{T}_1 there do not arise any inter-temporal vertex, on the second

tree we have exactly one inter-temporal vertex given by $((1, t), (1, t + 1))$ independent of the cross-sectional vine structure, we will therefore only look at trees $(\tilde{T}_k)_{k=3, \dots, d}$.

Note first, that in each of those trees \tilde{T}_k there arise exactly k inter-temporal edges connecting the vertices of the long connecting D-vine. Hence, $k - 2$ of the k inter-temporal vertices in \tilde{T}_k are given by the connecting edge of vertices already independent of the cross-sectional structure.

The two remaining critical vertices are given by the edge in the previous tree \tilde{T}_{k-1} connecting $v_1 = ((k - 1, t), (1, t + 1) | \cup_{i=1}^{k-2} (i, t))$ with the vertex in the intersection of the long D-vine and the $k - 1$ -th tree of the restriction of \mathcal{V} on the single column \mathbb{X}_t and $v_2 = ((1, t), (k - 1, t + 1) | \cup_{i=1}^{k-2} (i, t + 1))$ with the respective element when using the single column \mathbb{X}_{t+1} .

Let us denote the long D-vine in \tilde{T}_{k-1} by \mathcal{D}_{k-1} , the restriction on \mathbb{X}_t by \hat{V} and the vertex in the intersection by $e_{k-1} = \mathcal{D}_{k-1} \cap \hat{T}_{k-1}$. As a direct consequence of our ordered diagonal in the cross-sectional structure matrix, the complete union of the vertex is given by $U_{e_k} = \{(1, t), \dots, (k, t)\}$ with (k, t) in the conditioned set of e_{k-1} independent of the cross-sectional structure. The analogue statement replacing t by $t + 1$ is also true.

Having the complete union and the fact that (k, t) is in the conditioned set of e_{k-1} , we find that the edge connecting v_1 and e_{k-1} is given by $((k, t), (1, t + 1) | \cup_{i=1}^{k-1} (i, t))$ and respectively for v_2 by $((1, t), (k, t + 1) | \cup_{i=1}^{k-1} (i, t + 1))$.

This is especially independent of the cross-sectional structure and is true for all $k = 3, \dots, d$. \square

The last question arising on our new created specification is, if it is unique. This directly results of the theorems above.

Corollary 5.14. Uniqueness of Generalised M-Vines

For a given set of vertices V_1 with N time steps and a given d -dimensional cross-sectional structure there exists a unique generalised M-vine specification.

Before discussing further generalisation of our model in the next chapter, let us summarise our results so far.

- We have developed a new regular vine model for multivariate time series on basis of the M-vine approach from Beare and Seo (2015), which we called generalised M-vine

approach.

- The generalised M-vine approach is, in contrast to the already existing approaches in the literature, applicable for any cross-sectional regular vine structure.
- The trees (and therefore also the matrix representation) of a generalised M-vine are uniquely determined by the cross-sectional structure and the time horizon $T = \{1, \dots, N\}$.
- The d -th tree of each generalised M-vine is a D-vine with edges only depending on the final time N (independent of the cross-sectional structure).
- For fixed N and different cross-sectional structures the respective generalised M-vine matrices only differ from each other in the cross-sectional triangular structure (the vertices of the first d trees only differ in the conditioning set, if the elements in the conditioned set have the same time index).

Chapter 6

Generalised Temporal Connection

As already seen in the comparison of the D-vine and M-vine approach for cross-sectional D-vine structures, one can not only allow for more general cross-sectional structures, but also generalise the serial dependence model by allowing for different connecting edges between the time steps. In this chapter, we will analyse the influence of changing the respective vertices of those connecting edges to the overall matrix representation. The aim is to further develop the algorithm from Chapter 4, to capture also this generalisation.

For being able to describe those influences to the overall matrix we will use some further expressions, which need to be defined first. We will again use the letter notation for the sectional dimension here (i.e. the sectional dimension given by the set Θ).

Definition 6.1. Connecting Edge

We call the edges connecting the cross-sectional vine structures at time $t - 1$ with the one at time t (for each $t = 2, \dots, T$) in the first tree of our overall vine structure connecting edges. We denote those edges in the usual way, i.e. (Z_{t-1}, A_t) , where A and Z denote the respective sectional dimension in Θ .

Definition 6.2. (Temporal) Out/In Vertex

Let for each $t = 2, \dots, T$ the connecting edges be given by (Z_{t-1}, A_t) . Then we call Z_{t-1} the temporal out- and A_t the temporal in-vertex.

Definition 6.3. Order of the Sub-Diagonal Structure

The order of the sub-diagonal structure is given by the sectional dimensions of the columns of the parallelogram structures.

Let us have a look at the following example to better understand those definitions.

Example 6.4. Assume we are in 5×3 -dimensional classical M-vine approach as discussed in Section 4.2.1. The first tree is then given by Figure 4.2 a). Obviously, the connecting

edges are (A_1, A_2) and (A_2, A_3) . Therefore, the temporal in-vertices are given by A_2 and A_3 and the temporal out-vertex are A_1 and A_2 .

Having a look at the matrix representation (4.1) we see that for each parallelogram structure the sectional dimension in the columns is given decreasing by E, \dots, A . Hence, the order of the sub-diagonal structure is E, D, C, B, A .

6.1 Changing the Temporal Out-Vertex

Let us first consider changes of the temporal out-vertex. As in Chapter 4 we will take a look at the three different possibilities of the cross-sectional structure in the five dimensional case and will develop a general rule which explains the changes in the sub-diagonal structure of the overall matrix.

6.1.1 Cross-Sectional D-Vine Structure

We will start with the cross-sectional D-vine, where we have already seen a change of the out-vertex in the D-vine approach versus the classical M-vine approach (see Section 4.2). Further, we already mentioned the change of the sub-diagonal structure.

Let us recall the cross-sectional D-vine structure as well as both (the D-vine (4.2) and M-vine (4.1)) matrices with respective changes. The cross-sectional structure is given

by the matrix:

$$\begin{bmatrix} E_t & & & & \\ A_t & D_t & & & 0 \\ B_t & A_t & C_t & & \\ C_t & B_t & A_t & B_t & \\ D_t & C_t & B_t & A_t & A_t \end{bmatrix}.$$

Figure 6.2 below depicts the two matrices ((4.2) and (4.1)) and highlights the changes of the sub-diagonal structure, i.e. the reversion of the order of the sub-diagonal structure. To understand the changes better, recall from Example 6.4, that in the M-vine case the out-vertex is A_{t-1} , whereas in the D-vine case it is E_{t-1} . This deviating out-vertex is the only difference between the first trees of both approaches as illustrated in Figure 6.1.

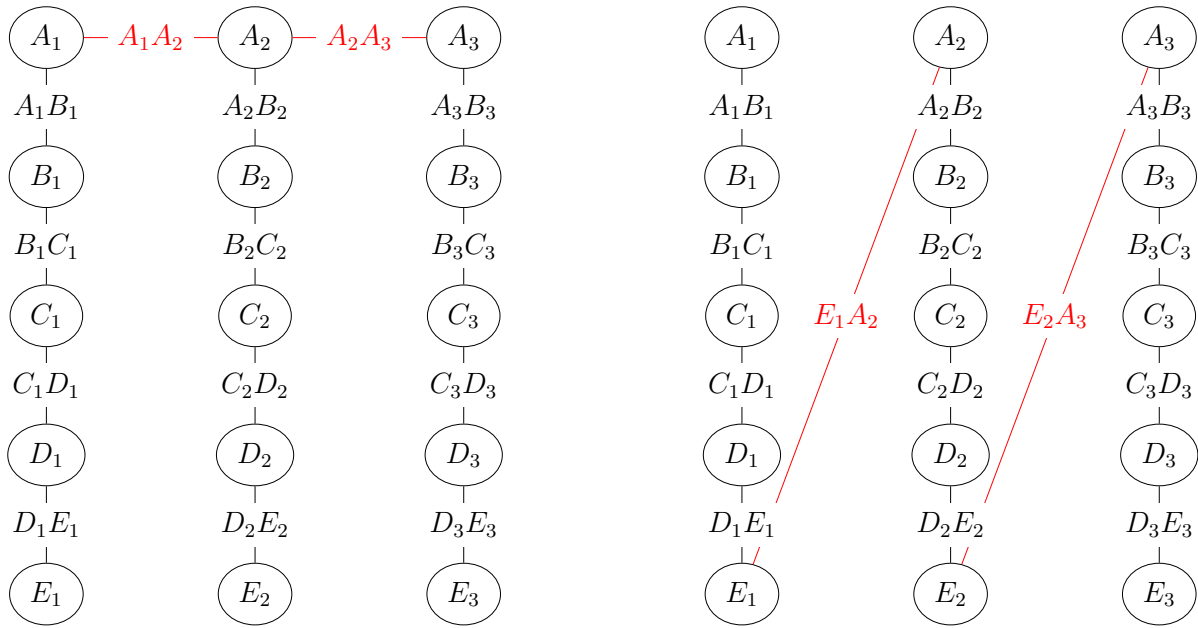


Figure 6.1: First tree of M-vine (left) and first tree of long D-vine (right) time series approach for three time steps and five sectional dimensions.

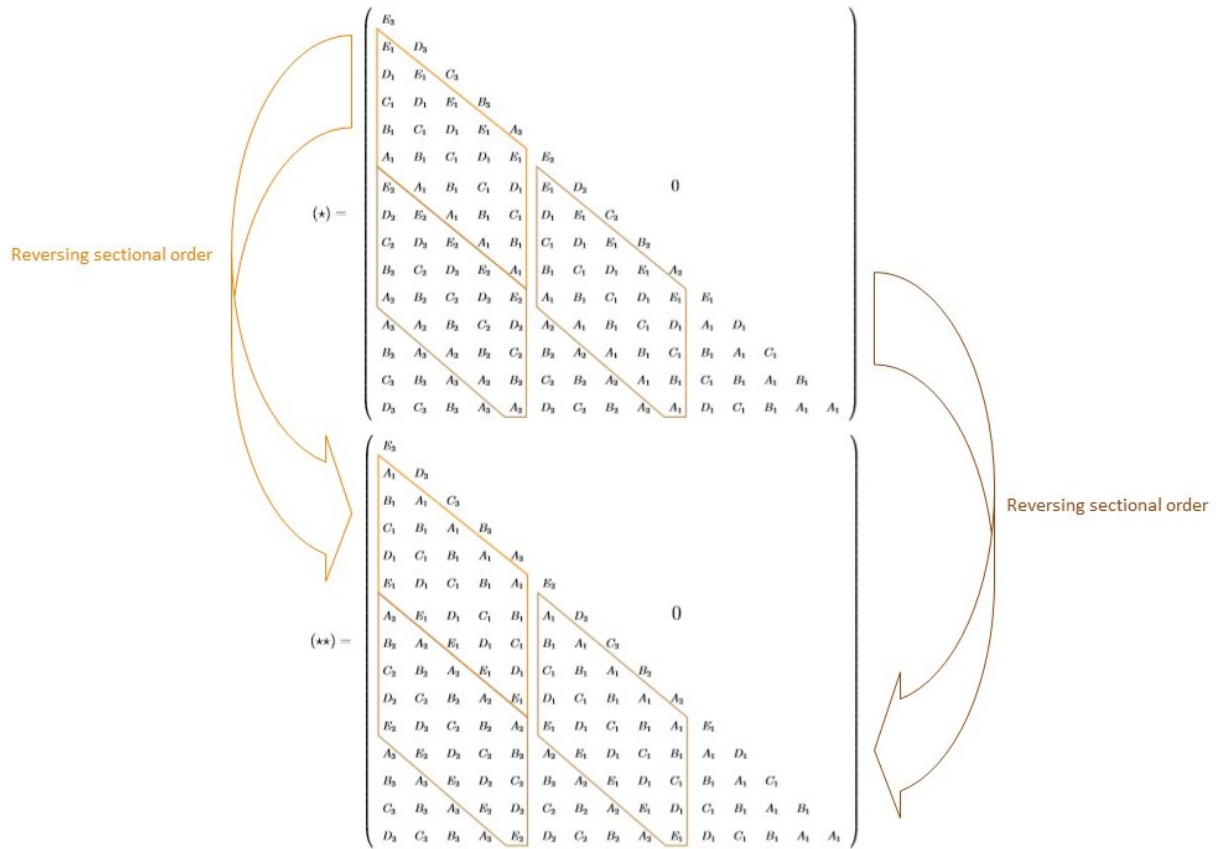


Figure 6.2: Comparison of classical M-vine (4.1) denoted by (★) and long D-vine (4.2) matrix denoted by (★★) in the 5×3 -dimensional case.

We will now change the connecting edges to (C_{t-1}, A_t) to better understand this reversion of the sub-diagonal order for cross-sectional D-vine structures. Figure 6.3 illustrates the corresponding starting tree and we can see that again the only difference to the one of the classical M-vine approach is given by the out-vertices.

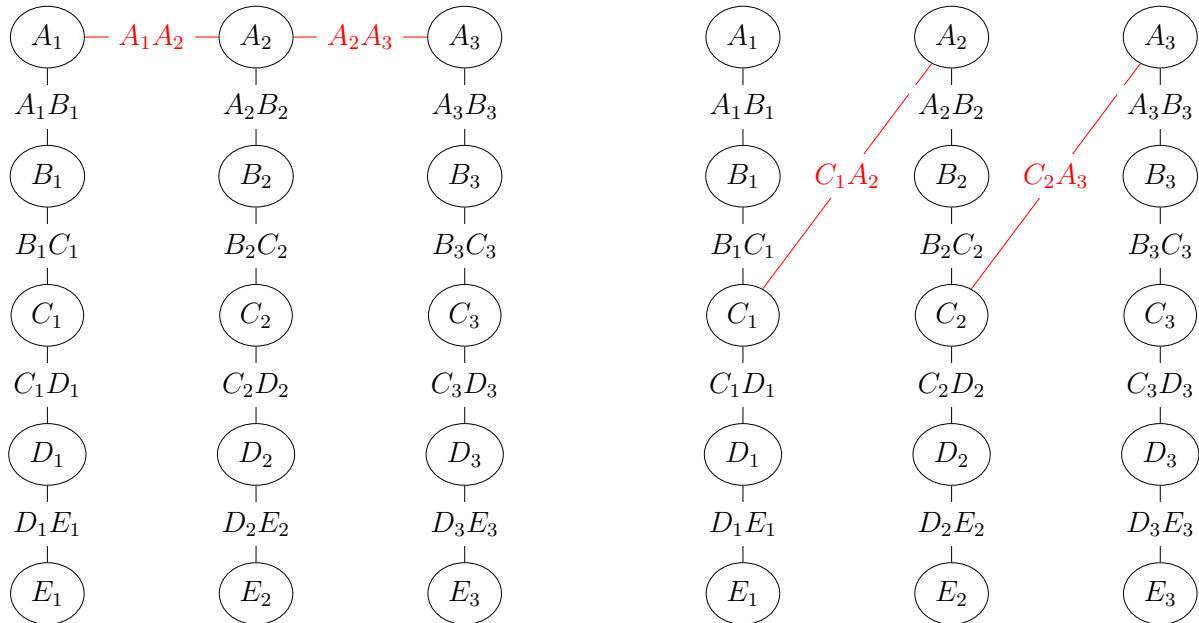


Figure 6.3: First trees of M-vine (left) and R-T-Vine with out-vertices C_{t-1} (right) time series approach for a five dimensional cross-sectional D-vine and three time steps.

The respective matrix in this case, with red highlighted area of changes versus the M-vine case is given by the matrix (6.1) below.

cross-sectional structure matrix:
$$\begin{bmatrix} E_t & & & & \\ A_t & D_t & & & 0 \\ B_t & A_t & C_t & & \\ C_t & B_t & A_t & B_t & \\ D_t & C_t & B_t & A_t & A_t \end{bmatrix} \Rightarrow E_t, D_t, C_t, B_t, A_t,$$

- for the out-vertex E the respective order is given by the first column, where

the diagonal entry E_t is switched to the last position:
$$\begin{bmatrix} E_t & & & & \\ A_t & D_t & & & 0 \\ B_t & A_t & C_t & & \\ C_t & B_t & A_t & B_t & \\ D_t & C_t & B_t & A_t & A_t \end{bmatrix} \Rightarrow$$

$$A_t, B_t, C_t, D_t, E_t$$

- and for the out-vertex C the order is given by the diagonal entries left of the C -column followed by the sub-diagonal of the respective column and C_t :

$$\begin{bmatrix} E_t & & & & \\ A_t & D_t & & & 0 \\ B_t & A_t & C_t & & \\ C_t & B_t & A_t & B_t & \\ D_t & C_t & B_t & A_t & A_t \end{bmatrix} \Rightarrow E_t, D_t, A_t, B_t, C_t.$$

We will later use these findings to create a general rule for the order of the sub-diagonal structure only depending on the cross-sectional structure matrix and the temporal out-vertex.

6.1.2 Cross-Sectional C-Vine Structure

Now, we change the cross-sectional structure to the five dimensional C-vine. We will again only change the out-vertices, i.e. we will look at the vertices $A_{t-1}, C_{t-1}, E_{t-1}$. Figure 6.4 depicts the starting trees of the respective "generalised M-vine" approaches. Recall that so far (in Chapter 4) the out-vertices were always given by A_{t-1} .

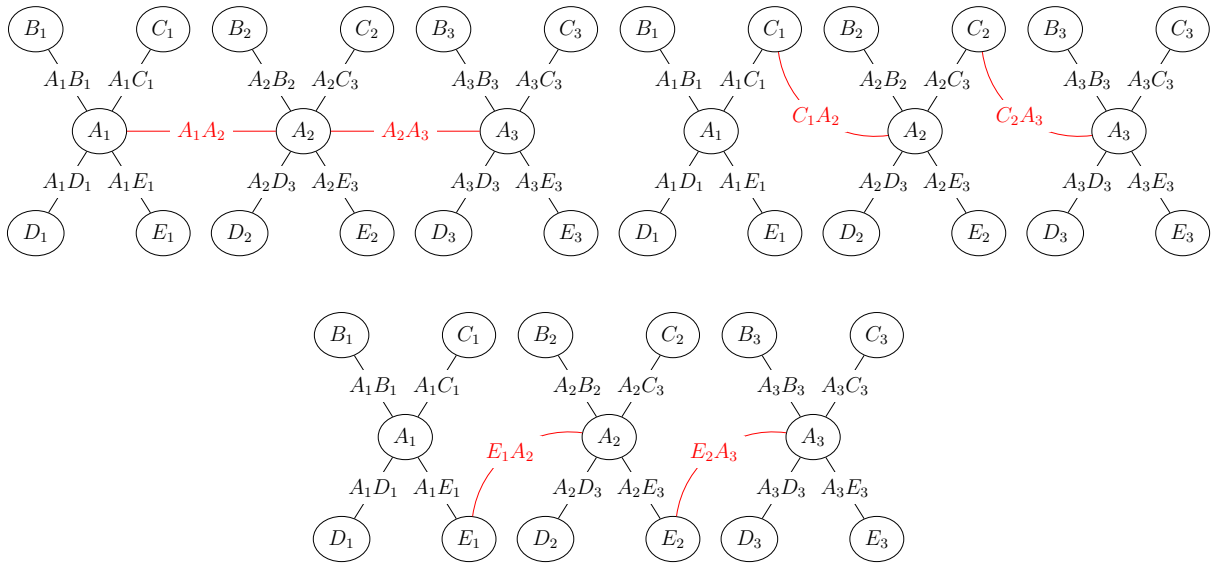


Figure 6.4: First trees of "generalised M-vine" time series approach for out-vertices A_{t-1} (top left), C_{t-1} (top right) and E_{t-1} (below) for a five dimensional cross-sectional C-vine and three time steps.

Having a closer look at the comparison of the respective matrices in Figures 6.5 and 6.6, we find that the **red highlighted** out-vertices in the cross-sectional C-vine case is just moved to the end of the order of the sub-diagonal structure compared to choosing A_{t-1} as out-vertices.

Obviously, this finding deviates from the reversing of the order mentioned in the previous section, where we studied the D-vine structure.

However, taking a look at the interpretation of the cross-sectional matrix as before, reveals the same pattern. For a detailed description we need the respective structure matrix as well as the orders for the sub-diagonal structure for each chosen out-vertex:

$\begin{bmatrix} E_t & & & & \\ D_t & D_t & & & 0 \\ C_t & C_t & C_t & & \\ B_t & B_t & B_t & B_t & \\ A_t & A_t & A_t & A_t & A_t \end{bmatrix}$	Out vertex	Order of sub-diagonal structure
	A	E, D, C, B, A
	E	D, C, B, A, E
	C	E, D, B, A, C

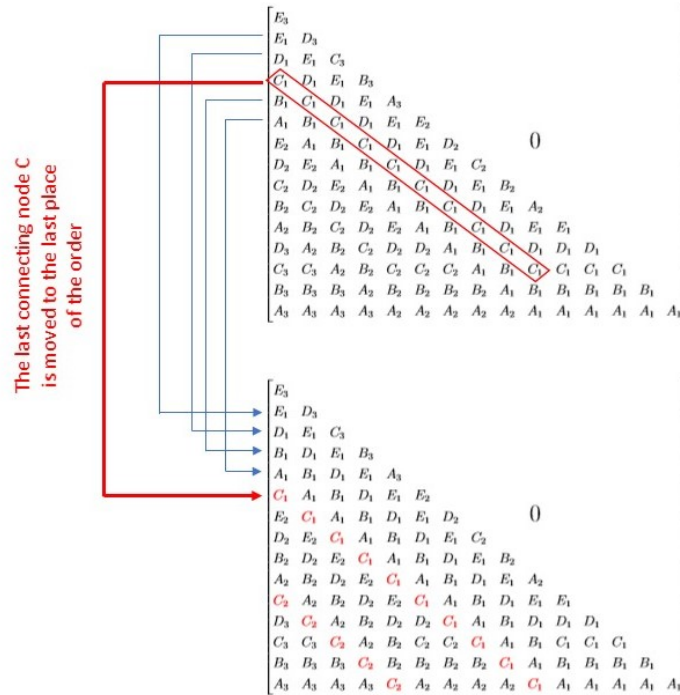


Figure 6.5: Comparison of "generalised M-Vine" matrix with out-vertex A_{t-1} (above) and C_{t-1} (below) in the 5×3 -dimensional case.

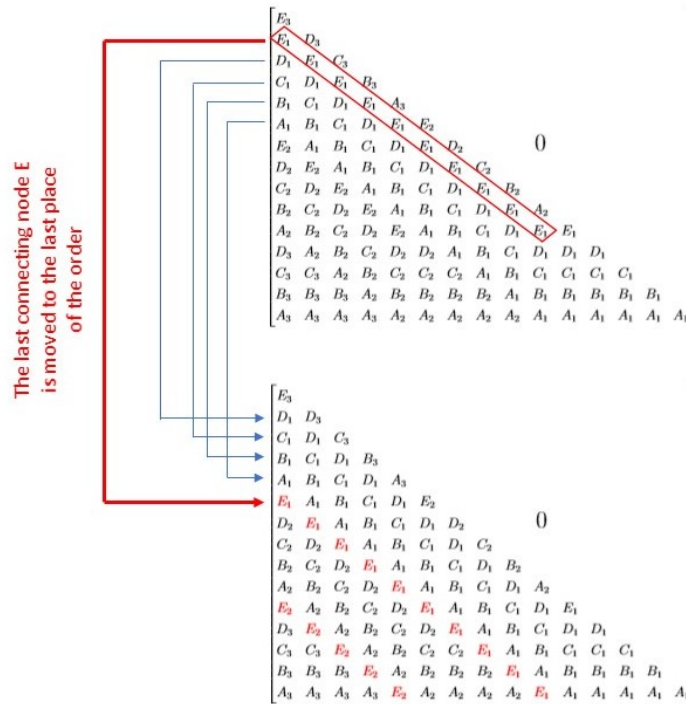


Figure 6.6: Comparison of "generalised M-Vine" matrix with out-vertex A_{t-1} (above) and E_{t-1} (below) in the 5×3 -dimensional case.

- For the out-vertices A , we see that the respective order matches the diagonal of the

$$\text{cross-sectional matrix: } \begin{bmatrix} E_t & & & & \\ D_t & D_t & & & 0 \\ C_t & C_t & C_t & & \\ B_t & B_t & B_t & B_t & \\ A_t & A_t & A_t & A_t & A_t \end{bmatrix} \Rightarrow E_t, D_t, C_t, B_t, A_t,$$

- for the out-vertex E the respective order is given by the first column, where

$$\text{the diagonal } E_t \text{ entry is switched to the last position: } \begin{bmatrix} E_t & & & & \\ D_t & D_t & & & 0 \\ C_t & C_t & C_t & & \\ B_t & B_t & B_t & B_t & \\ A_t & A_t & A_t & A_t & A_t \end{bmatrix} \Rightarrow$$

$$D_t, C_t, B_t, A_t, E_t$$

- and for the out-vertices C the order is given by the diagonal entries left of the C -column followed by the sub-diagonal entries of the C -column and C_t :

$$\begin{bmatrix} E_t & & & & \\ D_t & D_t & & & 0 \\ C_t & C_t & C_t & & \\ B_t & B_t & B_t & B_t & \\ A_t & A_t & A_t & A_t & A_t \end{bmatrix} \Rightarrow E_t, D_t, B_t, A_t, C_t.$$

6.1.3 More General Cross-Sectional R-Vine Structure and Tree Building

Finally, we will have a look at the non C- nor D-vine example of the cross-sectional structure. We skip the presentation of the matrices and only give the matrix and a table reflecting the dependence of out-vertex and the order of the sub-diagonal structure.

$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix} \quad \begin{array}{c|c} \text{Out vertex} & \text{Order of sub-diagonal structure} \\ \hline E & C, A, B, D, E \\ C & E, D, A, B, C \\ A & E, D, C, B, A \end{array}$$

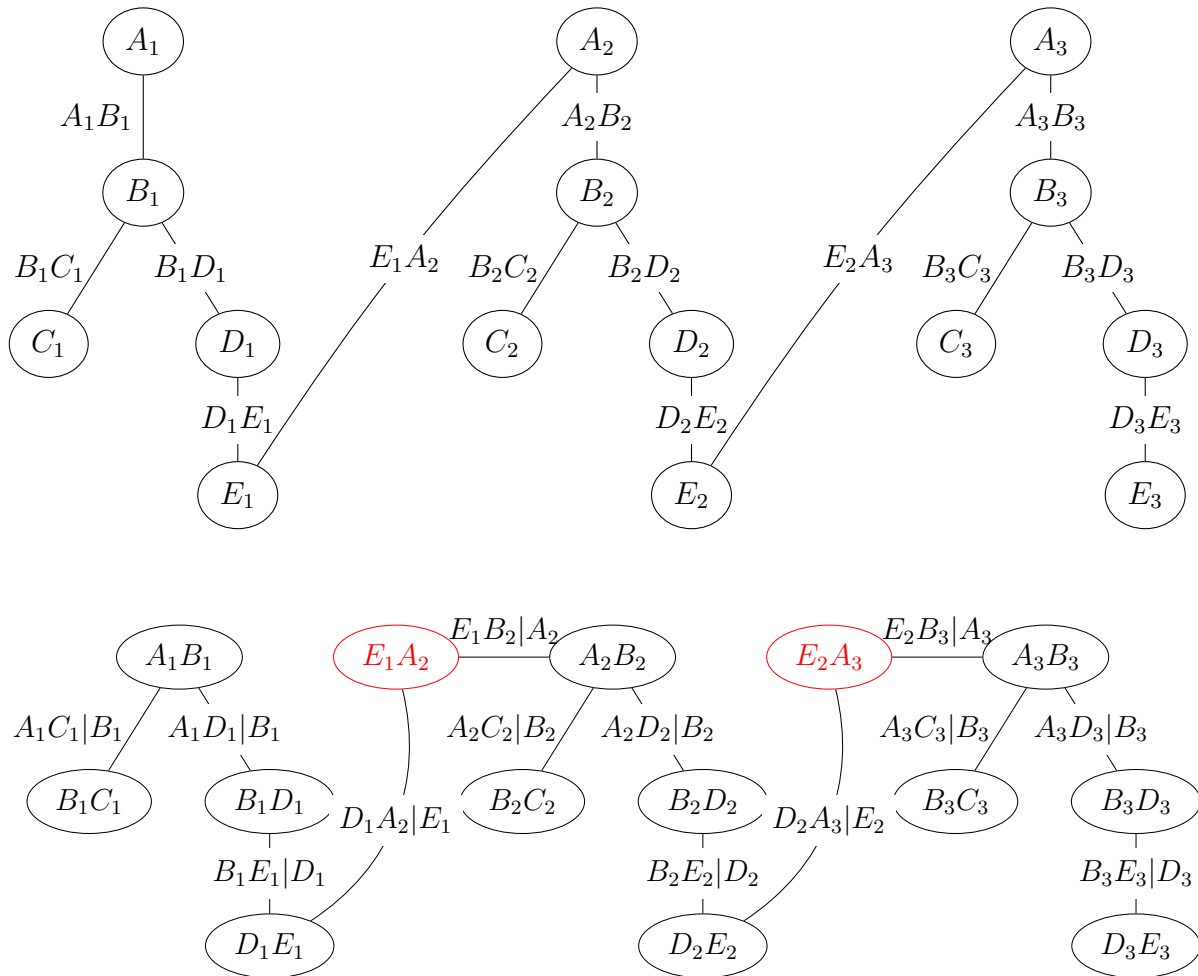
Again, we find the same interpretation as in the two cases before. The general rule extracted from the above examples is given by:

For each temporal out-vertex, the order of the sub-diagonal structure is given by

1. the diagonal entries to the left of the column, in which the out-vertex is the diagonal entry (from left to right),
2. followed by the sub-diagonal entries of this column
3. and the out-vertex itself.

This rule is valid for higher sectional as well as higher time dimensions. For a higher dimensional example, we refer to the respective part in the appendix.

We are further interested in how these changes are depicted in and can be explained by the respective trees. Therefore, we will compare the trees of our 5×3 -dimensional cross-sectional non C- nor D-vine example for the out-vertices A_{t-1} (in Figure 4.8) and E_{t-1} (Figure 6.7 with red highlighted deviating vertices).



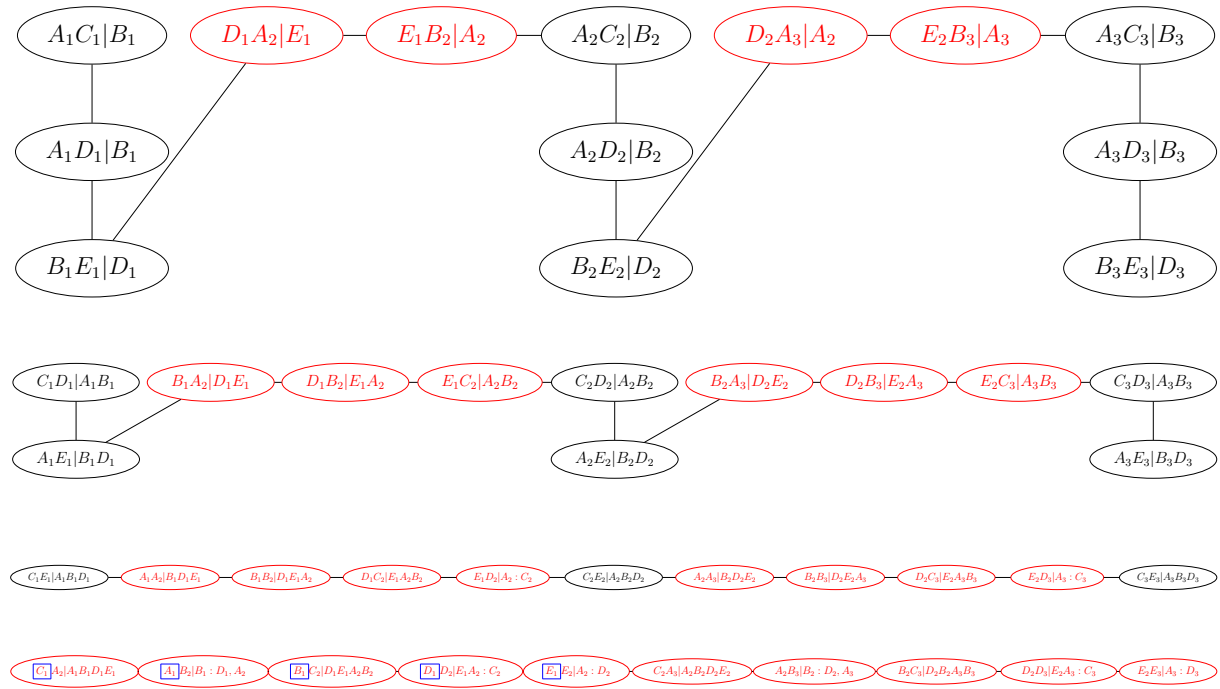


Figure 6.7: First six trees of the "generalised M-vine" time series approach with out-vertices E_{t-1} for a five dimensional cross-sectional regular vine structure and three time steps.

As we only change the cross-serial structure, but not the cross-sectional, it is not surprising, that the conditioning sets of the respective vertices in both figures are the same whenever the elements in the conditioned set have the same time index. However, when there arises a different time in the elements of the conditioned set the conditioning sets are not the same. This already explains the equality of below triangular structure and that there are changes of the sub-diagonal structure in the last five rows of the overall matrix. Further, we know that the trees (and therefore the sub-diagonal part) will be non-identical since the sixth (D-vine) tree is non-identical to the respective one in Figure 4.8 in every vertex.

In the example of the out-vertex E_t , the order of the sectional dimension becoming part of the root tree (prior called long connecting D-vine) is given by E, D, B, A, C . This results in a sub-diagonal order C, A, B, D, E , i.e. the sub-diagonal order is the reversed order of the sectional dimensions, becoming part of the root tree. To fully understand the order of the sub-diagonal structure it suffices to have a look at the conditioned sets of the first 5 (in general d) vertices of the sixth (in general $(d + 1)$ -th) tree, i.e. the blue surrounded elements, with time $t = 1$. The respective order of the sub-diagonal structure is then given by aligning the respective sectional dimensions from left to right.

6.2 Changing the Temporal In-Vertex

The next generalisation we want to discuss is changing the temporal in-vertices, i.e. instead of connecting $A_{t-1} \leftrightarrow A_t$, we now analyse the changes to the matrix when we connect $A_{t-1} \leftrightarrow X_t$, where X can be any other possible sectional dimension.

We will not go through all examples again, but just give the rule and explain it based on the five dimensional non C- nor D-vine example.

6.2.1 General Influence on Matrix Representation

An extensive study of changing the in-vertices revealed that it influences the order of the sectional dimensions on the diagonal structure directly as well as the cross-sectional triangular structure indirectly, implicitly linked to cross-temporal dependence with lag 1.

- The change of the order on the diagonal is completely analogue to the change of the order of the sub-diagonal structure, discussed in the previous section. The respective rule is given by:

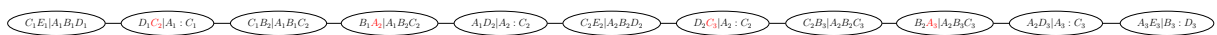
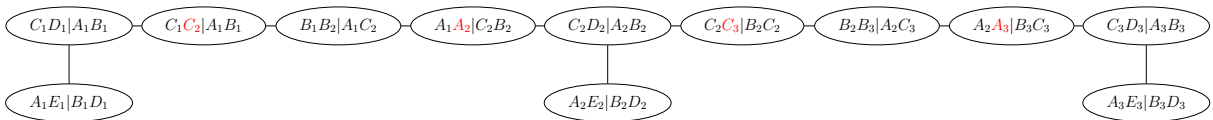
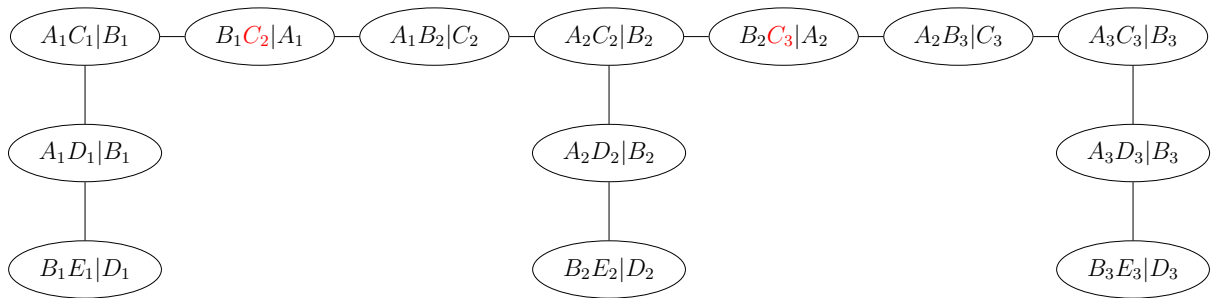
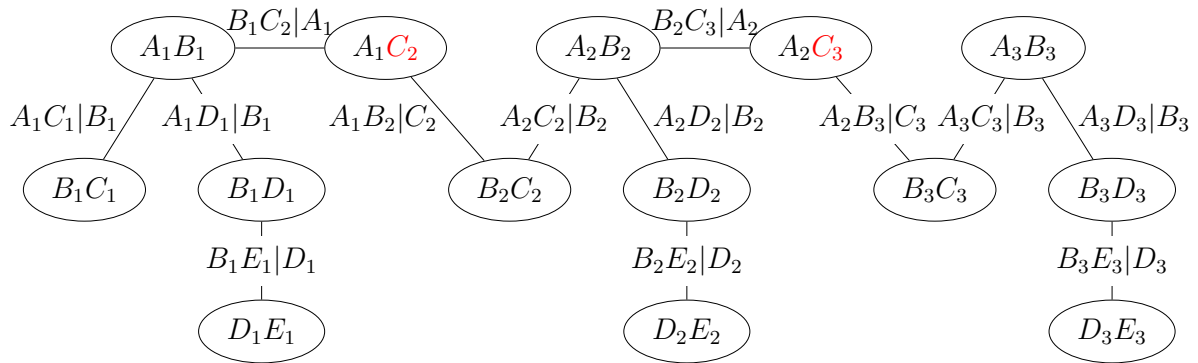
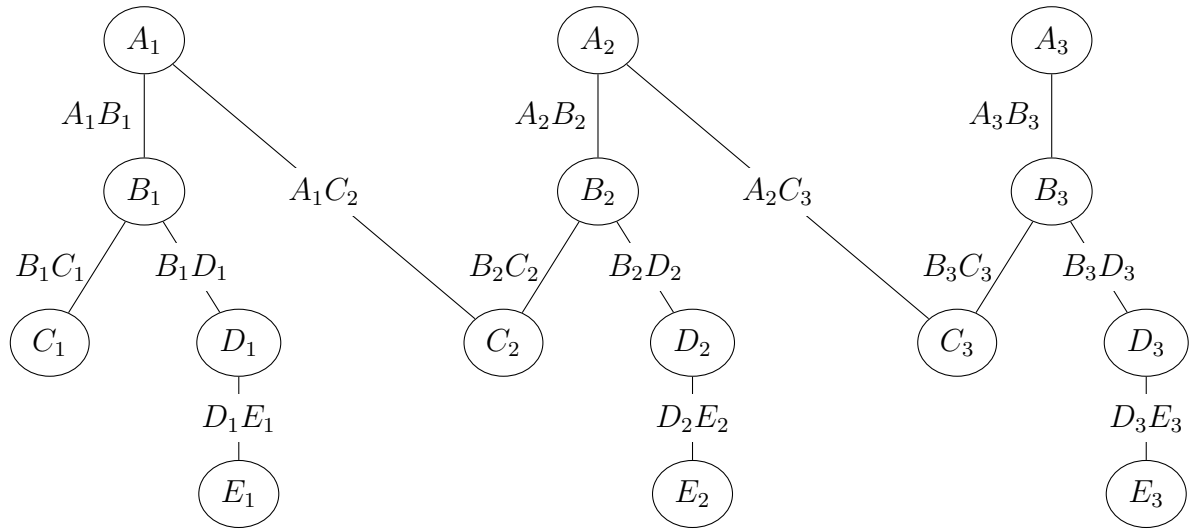
For each temporal in-vertex, the order of the sectional dimension on the diagonal of the overall matrix is given by

- 1. the diagonal entries to the left of the column of the cross-sectional matrix, in which the in-vertex is the diagonal entry (from left to right),**
 - 2. followed by the sub-diagonal entries of this column**
 - 3. and the in-vertex itself.**
- These changes of the diagonal structure also influence the cross-sectional triangular structure. The probably most simple way to picture this influence is, that now the diagonal of our cross-sectional matrix is given by the new order obtained by the rule above. Clearly, we have to adjust the the whole sub-diagonal part of the matrix as well.

6.2.2 Example: General Regular Vine Structure

Let us now use the five dimensional non C- nor D-vine example to visualise the changes described above. Recall that the matrix for the connection $A_{t-1} \leftrightarrow A_t$ is given by (4.5).

given by aligning the sectional dimension of the time $t = 2$ elements in the conditioned set of the first d vertices (from the left) in reversed order.



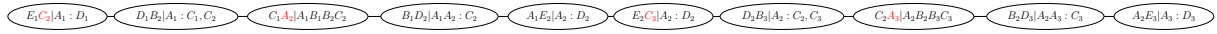


Figure 6.8: First six trees of the "generalised M-vine" time series approach for a five dimensional cross-sectional regular vine structure with in-vertex C and three time steps.

Next, we want to discuss the influence on the cross-sectional triangular structure. The trees of the respective structure are given in Figure 6.9.

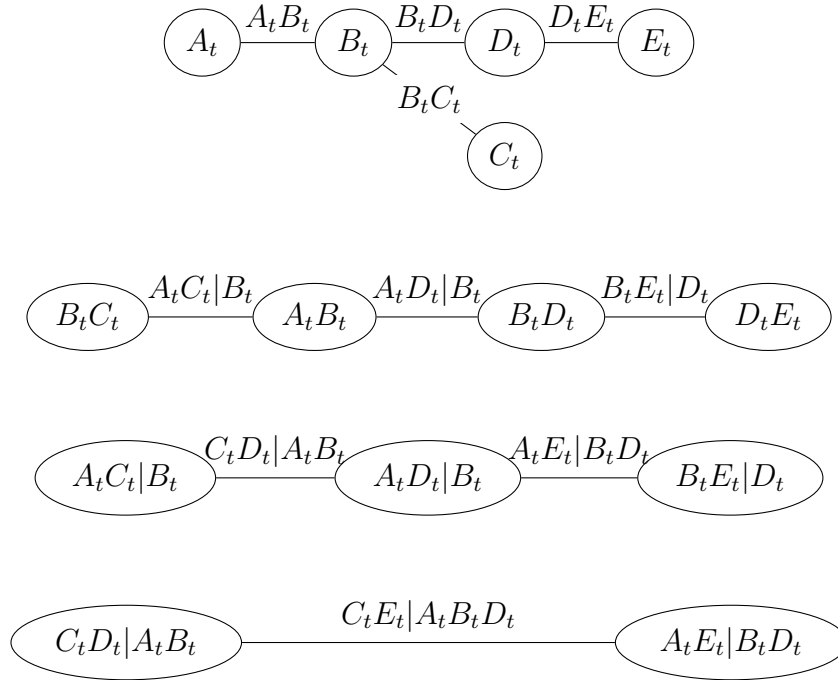


Figure 6.9: The trees of the non C- nor D-vine example.

As already seen several times we assigned the matrix representation

$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix}$$

to this vine specification, i.e. we have chosen the diagonal to be

descending from E to A . This is a common approach in practice of regular vine matrices.

When looking for example at the vertex $(C_t D_t | A_t B_t)$, which is reflected in the second column of the matrix, one could also replace the column by $(0, C_t, D_t, A_t, B_t)^T$ (and the third column respectively by $(0, 0, D_t, A_t, B_t)^T$) also reflecting the above trees accordingly.

As mentioned above the change of the diagonal of our overall matrix influences the triangular structure, given by the sub-diagonal of the cross-sectional structure matrix M . For getting the correct cross-sectional triangular structure we need to adjust the

diagonal of this matrix according to our developed rule and then we have to adjust the sub-diagonal, such that the "new cross-sectional structure matrix" M_{new} reflects the tree in Figure 6.9 correctly. This can be reached in the following way, given the new diagonal structure:

- For each entry $M_{new}(i, i)_{i=1, \dots, d}$ on the diagonal structure of the "new cross-sectional structure matrix" M_{new} , the default sub-diagonal is given by the entries $i + 1$ up to m of the order obtained, when $M_{new}(i, i)$ is the in/out-vertex. We denote the resulting default matrix by $M_{default}$.
- If there arises an element of the diagonal left to $M_{default}(i, i)$ in the sub-diagonal, this would be wrong. We have to adjust the whole sub-diagonal of this column i . Therefore, we go through the lines $i + 1$ up to d of the cross-sectional structure matrix M and search for $M_{default}(i, i)$.

If we can find this diagonal entry of the default matrix and the respective diagonal entry of the column in M is in the remaining part of our diagonal of the "new cross-sectional structure matrix", i.e. in the set $\{M_{new}(j, j)_{i < j \leq d}\}$, we add the diagonal entry to the sub-diagonal.

If not, we will add the entry with the coordinates of the respective and the column, where $M_{default}(i, i)$ is the diagonal entry, in M to our sub-diagonal.

Let us focus on the following example to better understand this building of the new matrix.

Example 6.5. Assume we are in the five dimensional non C- nor D-vine example. The in-vertex is given by D_t , resulting in a diagonal order of E, C, A, B, D .

In the first step, we have to calculate the orders for each diagonal entry and get the $i : d-1$ entries for the **default sub-diagonal**:

$$\begin{array}{l}
 \mathbf{E:} \quad C, A, B, D, E \\
 \mathbf{C:} \quad E, D, A, B, C \\
 \mathbf{A:} \quad E, D, C, B, A \\
 \mathbf{B:} \quad E, D, C, A, B
 \end{array}
 \quad
 M_{default} = \begin{bmatrix}
 E_t & & & & \\
 C_t & C_t & & & 0 \\
 A_t & D_t & A_t & & \\
 B_t & A_t & C_t & B_t & \\
 D_t & B_t & B_t & A_t & D_t
 \end{bmatrix}$$

In the next step, we see that obviously the **blue surrounded** entries are wrong. This means we have to search for $M_{default}(3, 3) = A_t$ in the fourth and fifth and for $M_{default}(4, 4) = B_t$ in the fifth line of the cross-sectional structure matrix.

- Searching for A_t in the fourth line, we find **two possible candidates**, with respec-

tive diagonal entries D_t and C_t :
$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix}$$
. Only D is arising in the remaining part of the diagonal order ..., B, D .

- In the fifth line, there arises only one **candidate** A_t :
$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix}$$
. The respective diagonal entry B_t is indeed in the remaining part of the diagonal order ..., B, D .

\implies The sub-diagonal of column 3 is given by $(D_t, B_t)^T$.

- Searching for B_t in the fifth line, revealed **two possible candidates** with respec-

tive diagonal entries D_t and C_t :
$$\begin{bmatrix} E_t & & & & \\ C_t & D_t & & & 0 \\ A_t & C_t & C_t & & \\ B_t & A_t & A_t & B_t & \\ D_t & B_t & B_t & A_t & A_t \end{bmatrix}$$
. The remaining diagonal of $M_{default}$ only contains D , implying the sub-diagonal element of column 4 is given by D_t .

Hence, the final "new cross-sectional structure matrix" M_{new} is given by:

$$M_{new} = \begin{bmatrix} E_t & & & & \\ C_t & C_t & & & 0 \\ A_t & D_t & A_t & & \\ B_t & A_t & D_t & B_t & \\ D_t & B_t & B_t & D_t & D_t \end{bmatrix}.$$

6.3 Changing the Connecting Edges

The change of the in-vertex has no influence on the sub-diagonal structure and the change of the out-vertex does not affect the diagonal nor the triangular structure. This turns out to be important when we look at the connection of e.g. $C_{t-1} \leftrightarrow C_t$, because we

section. Here, it is given by
$$\begin{bmatrix} E_1 & & & & \\ C_1 & D_1 & & & \\ A_1 & C_1 & A_1 & & \\ B_1 & A_1 & C_1 & B_1 & \\ D_1 & B_1 & B_1 & C_1 & C_1 \end{bmatrix}.$$

However, for algorithmic reasons it is easier to fit a cross-sectional structure first and adjust it in accordance to the order implied by the in-vertex.

6.4 Summary and Algorithm

To summarise the generalisation of the regular vine specification from this chapter, we give a formal definition of our obtained structure and implement those structures in an algorithm.

6.4.1 Influence of Changing Connecting Edges on Generalised M-Vines

Recalling Definition 5.5 (Generalised M-Vine), we notice that allowing for changes in the connecting edges, changes the second restriction within the first part of the definition, i.e. there can also arise an edge between vertices of different time steps when the sectional dimension is not 1 (respectively A). Furthermore, the changes of the sub-diagonal structure results in possible different edges on the d -th tree of the restriction of \mathcal{V} on the adjacent columns. Thus, the final definition of our general vine specification needs a further modification. We will call this R(egular)-T(emporal)-vine.

Definition 6.7. R-T-Vine

A regular vine \mathcal{V} on V_1 with trees $(N_k, E_k)_{k=1, \dots, Nd-1}$ is called R-T-Vine if and only if:

- i) $E_1 = \{\{(i, s), (j, t)\} \subset \binom{V_1}{2} \mid (i, s), (j, t) \text{ are connected via an edge in the cross-sectional structure at time } t \text{ and } s = t \text{ or } (i, s) \text{ is the out-vertex and } (j, t) \text{ is the in-vertex and } s = t - 1\}$,
- ii) for each single column $\mathbb{X}_t = \cup_{i=1}^d \{(i, t)\}$, $t = 1, \dots, N$ the restriction of \mathcal{V} on \mathbb{X}_t has the same structure, namely the cross-sectional structure and
- iii) for each restriction of \mathcal{V} on the respective adjacent columns $(\mathbb{A}_t)_{t=1, \dots, N-1}$ the d -th tree is a D-vine.

As this definition is quite similar to the one given in Chapter 5, Theorem 5.5 and 5.6 can be easily transferred to this general class of vine structures. Hence, also for R-T-Vines we have uniqueness (given a specific connection) and a D-vine structure on the d -th tree.

Theorem 6.8. Theorem 5.11 for R-T-Vine

If \mathcal{V} is a R-T-Vine specification of a d -dimensional dataset V_1 with N time steps, then the tree T_d is a D-vine.

Proof. Following the same steps as the proof of Theorem 5.11 except that the common vertex of the two restrictions does not necessarily contains the element $(d, t + 1)$, but still has degree one on the respective restrictions. \square

Theorem 6.9. Uniqueness of R-T-Vine

For a given set of vertices V_1 with N time steps, a given d -dimensional cross-sectional structure and given connecting edges, there exists a unique R-T-Vine.

Proof. We only need to show that for each given cross-sectional structure the first $d - 1$ trees are uniquely determined via the proximity condition of regular vines and the equality of the single columns \mathbb{X}_t and the rest follows from Theorem 6.7 above.

For $t = 1, \dots, N - 1$ taking a look at the first $d - 1$ trees of \mathcal{V}_t , denoting the restriction of \mathcal{V} on \mathbb{A}_t we find that they are uniquely determined by the cross-sectional structure (reflected in Definition 6.6 (ii)) and the fact that the additional vertices resulting from the edge connecting the single columns \mathbb{X}_t and \mathbb{X}_{t+1} are at most connected to two vertices (Note that this is a necessary and sufficient condition for Definition 6.6 (iii)).

The first tree is uniquely determined by (i) in the definition of the R-T-vine, so we will take a look at the trees $(T_k)_{k=2, \dots, d-1}$.

On tree T_k the restriction \mathcal{V}_t has $2d - k$ edges and each pair of \mathcal{V}_t and \mathcal{V}_{t+1} share exactly $d - k$ edges on T_k . Hence, the $N - 1$ uniquely determined restriction $(\mathcal{V}_t)_{t=1, \dots, N-1}$ explain $Nd - k$ edges of T_k . On the other hand, we know that \mathcal{V} on T_k has exactly $Nd - k$ edges. Thus, T_k is determined uniquely for $k = 2, \dots, d - 1$. Furthermore, T_k is connected as it is composed of the $N - 1$ overlapping regular vines $(\mathcal{V}_t)_{t=1, \dots, N-1}$. And the proximity condition must be satisfied, since each vertex in T_k is contained in at least one regular vine. \square

6.4.2 Generalisation of the Algorithm

We have seen that the influence of changing the out or in-vertex on the overall matrix can be fully described by a simple rule only knowing the cross-sectional matrix and the respective connecting edges. The rule to get the sectional order for the diagonal or sub-diagonal structure is summarised in the following algorithm.

Algorithm 2 Sectional order given specific vertex

Input csStructure: cross-sectional structure matrix
vertex: the vertex for which we want to get the new order
 d : sectional dimension

Output resultOrder: vector containing the sectional order of the (sub-)diagonal

function GETORDER(csStructure,vertex,d)
 $j \leftarrow$ row/column of diagonal entry in csStructure equal to vertex
resultOrder from 1 upto $j - 1 \leftarrow$ diagonal of csStructure from 1 upto $j - 1$
resultOrder from entry up to $d - 1 \leftarrow$ sub-diagonal of csStructure in column j
resultOrder at $d \leftarrow$ vertex
return structure

The change of the diagonal structure (changes of the in-vertex), however, influences the cross-sectional triangular structure. These changes are captured by the following two Algorithms 4 and 3.

Algorithm 3 Update default structure

Input csStructure: cross-sectional structure matrix
inOrder: diagonal order given by structure in Algorithm 2
startrow: row from which we start start searching for the diagonal entry
 d : sectional dimension

Output res: the vector containing updated sub-diagonal

function UPDATEDEFAULT(csStructure,inOrder,startrow,d)
for each row i from startrow to d **do**
 if searched element is in the i -th row **then**
 candidates \leftarrow diagonal entries to the columns containing the searched element
 res \leftarrow the element in candidates in remaining part of the inOrder
 else
 res \leftarrow the element in row i and column of searched element as diagonal entry
return res

Algorithm 4 Triangular structure given diagonal order

Input csStructure: cross-sectional structure matrix

inOrder: (sub-)diagonal order given by structure in Algorithm 2

 d : sectional dimension

Output M_{new} : new cross-sectional triangular structure

function GETTRIANGLE(csStructure,inOrder, d)

 diagonal of $M_{new} \leftarrow$ structure

 for each column j from 1 to $d - 1$ **do**

 sub-diagonal at j of $M_{new} \leftarrow$ getOrder(csStructure,inOrder[i], d) from j to $d - 1$

 if sub-diagonal of M_{new} in column $i \ni$ element of diagonal left to column i **then**

 sub-diagonal of $M_{new} \leftarrow$ updateDefault(csStructure,inOrder, $i+1,d$)

 return sub-diagonal of M_{new}

Having Algorithm 2,4 and 3 we are able to further generalise the Algorithm 1 for the overall matrix, including the feature of changing the connecting edges. Essential for this step is the independent combination of changing temporal in and out-vertices, which we have confirmed with the help of the changes in the matrix representation and based on the vine structure trees (i.e. the $m + 1$ -th tree).

Algorithm 5 Temporal vine matrix

Input N : number of time steps

csStructure: cross-sectional structure matrix

inVertex: the temporal in-vertex

outVertex: the temporal out-vertex

Output M : R-T-Vine structure matrix**function** TEMPORALVINEMATRIX(csStructure, N ,inVertex,outVertex) $d \leftarrow$ sectional dimension $m \leftarrow$ counter for each d columns starting at 1inOrder \leftarrow getOrder(csStructure,inVertex, d)outOrder \leftarrow getOrder(csStructure,outVertex, d)Triangle \leftarrow getTriangle(csStructure,inOrder, d)diagonal of $M \leftarrow$ decreasing sequence $d(N - m) + \text{inOrder}$ **for** each column from 1 to $(N - 1)d$ **do**Sub-diagonal structure of $M \leftarrow$ outOrder + $(t - p - 1)d$ with $t = N - m + 1$ and p from $t - 1$ to 1**After every** $d - \text{th}$ **column**Triangle structure of the d columns \leftarrow $(N - m)d + \text{Triangle}$ $m \leftarrow m + 1$ First triangle stucutre $M_1 \leftarrow$ Triangle**return** M

Those algorithms built the basis of the implemented temporal vine functions in the not yet published R-package *tvine* by Thomas Nagler, which will be used for the practical application study in Chapter 9.

Chapter 7

Regular-Temporal-Vine Copula Specification for Stationary Multivariate Time Series

Besides the developed R-T-Vine specification, we also have to take into account the respective matrices of copula families and relating parameters, already presented in Chapter 3, for a full general vine copula specification. We will recall the other matrices needed to fully describe the vine copula specification in this chapter. Further, we will discuss the influence of assuming stationarity or an autoregressive order of p to those matrices.

7.1 Respective Copula Vine Specification

Given a R-T-Vine matrix M (given by Algorithm 5) we further need to specify the respective copulas associated with this vine structure. Therefore, we first need to assign the respective copula families in the matrix F , with parameters stored in P_1, \dots, P_x , where x denotes the maximal number of parameters of the copula families assigned to our specification.

Recall that in the $N \times d$ -dimensional case with $M = (m_{ij})_{i,j=1,\dots,N*d}$ the respective matrices are given by

$$F = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ F_{C(m_{1,1}, m_{2,1} | m_{3,1} : m_{n,1})} & 0 & \cdots & 0 & 0 \\ F_{C(m_{1,1}, m_{3,1} | m_{4,1} : m_{n,1})} & F_{C(m_{2,2}, m_{3,2} | m_{4,2} : m_{n,2})} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ F_{C(m_{1,1}, m_{n-1,1} | m_{n,1})} & F_{C(m_{1,2}, m_{n-1,2} | m_{n,1})} & \ddots & 0 & 0 \\ F_{C(m_{1,1}, m_{n,1} | \emptyset)} & F_{C(m_{2,2}, m_{n,2} | \emptyset)} & \cdots & F_{C(m_{n-1, n-1}, m_{n, n-1} | \emptyset)} & 0 \end{bmatrix}$$

with parameters in the $n \times n$ -matrix:

$$(P_i)_{i=1, \dots, x} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ P_{C(m_{1,1}, m_{2,1} | m_{3,1} : m_{n,1})} & 0 & \cdots & 0 & 0 \\ P_{C(m_{1,1}, m_{3,1} | m_{4,1} : m_{n,1})} & P_{C(m_{2,2}, m_{3,2} | m_{4,2} : m_{n,2})} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ P_{C(m_{1,1}, m_{n-1,1} | m_{n,1})} & P_{C(m_{1,2}, m_{n-1,2} | m_{n,1})} & \ddots & 0 & 0 \\ P_{C(m_{1,1}, m_{n,1} | \emptyset)} & P_{C(m_{2,2}, m_{n,2} | \emptyset)} & \cdots & P_{C(m_{n-1, n-1}, m_{n, n-1} | \emptyset)} & 0 \end{bmatrix}$$

Therefore, in a completely general set-up of those matrices, one would have $\frac{(N * d - 1)^2 + N * d - 1}{2}$ different copulas with overall $\frac{x * (N * d - 1)^2 + N * d - 1}{2}$ parameters to specify. Obviously, this will be computationally costly for large N or d and needs to be reduced.

7.2 Reducing Complexity of the Copula Matrix

As already mentioned in Section 2.1, we can significantly reduce this number by assuming stationarity or an autoregressive order of p .

7.2.1 Stationarity

When assuming a strictly stationary times series we want the multivariate distribution of each arbitrary successive sample of time steps to be equal to the time shifted distribution of this sample. This directly implies that the cross-sectional structure will be time independent and therefore the triangular structures will be the same. Further, as the distribution is also described by the higher order trees within our vine specification we find that for each fixed order p the parallelogram structures are the same as well.

To formally describe this effect we will need the concept of translation invariance, which exactly describes the structures in our matrix. This part is mainly following the presentation in Beare and Seo (2015).

Definition 7.1. Translation Invariance

A vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ on N time steps and sectional order d fulfilling:

- i) $F_{t,i} = F_{t+1,i} \forall t, t+1 \in 1, \dots, N$ and $\forall i \in 1, \dots, d$ as well as
- ii) For any edge on \mathcal{V} , i.e. $e, e' \in \bigcup_{k=1}^{Nd-1} E_k$ such that $D_e = D_{e'} + (0, s)$ and $\{((a_e, b_e), t)\} = \{((a_{e'}, b_{e'}), t)\} + (0, s)$ for some $s \geq 1$, we have $C_{a_e, b_e | D_e} = C_{a_{e'}, b_{e'} | D_{e'}}$

is called translation invariant.

Theorem 7.2.

Let the joint distribution of $(\mathbb{X}_1, \dots, \mathbb{X}_N)$ realise the vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ on N time steps and sectional order d . If $(\mathbb{X}_1, \dots, \mathbb{X}_N)$ is strictly stationary, then $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ is translation invariant.

Proof. (Following Beare and Seo (2015))

□

Given that the parallelogram structures for each order as well as the triangular structure stays the same in each time point, the number of different copulas to be specified in the $N \times d$ case significantly reduces to $(N-1) * d^2 + \frac{(d-1)^2 + (d-1)}{2}$. This can be seen in Figure 7.1 in the following.

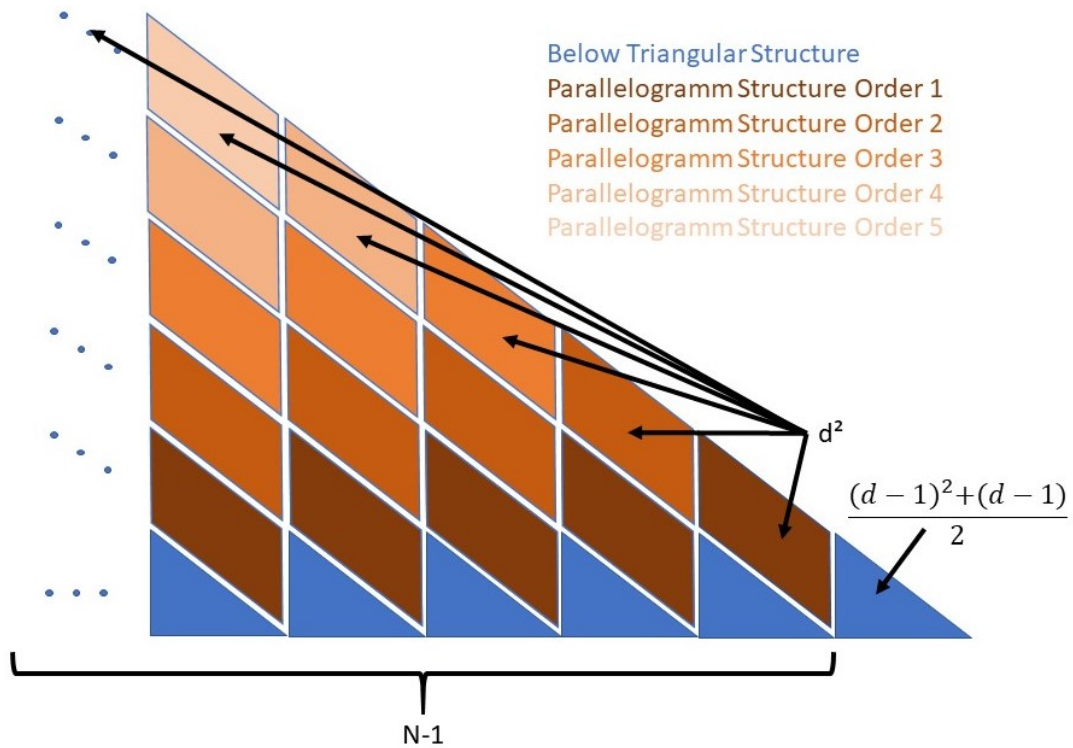


Figure 7.1: Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series with stationarity.

7.2.2 Autoregressive Order p

When assuming an autoregressive order of p in our multivariate time series we assume conditional independence of observations with lag greater than p . This results in assigning the independence copula to all entries in the parallelogram structures with order higher than p in our R-T-Vine specification.

This conditional independence of a strictly stationary process can also be defined via the so called p -Markov property.

Definition 7.3. p -Markov Property

The array $(\mathbb{X}_1, \dots, \mathbb{X}_N)$, with $\mathbb{X}_i \in \mathbb{R}^d$ is said to be p -Markov if for any $t = p + 1, \dots, n$, $\mathbb{P}(\mathbb{X}_t | \mathbb{X}_1, \dots, \mathbb{X}_{t-1}) = \mathbb{P}(\mathbb{X}_t | \mathbb{X}_{t-p}, \dots, \mathbb{X}_{t-1})$, i.e. the present state only depends on the p previous states.

To understand the influences on our copula vine specification we need the concept p -independence.

Definition 7.4. p -Independence (based on Beare and Seo (2015))

A R-T-Vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ for N observations of a d -dimensional dataset on the set of vertices V_1 is called p -independent if it assigns the independence copula to all edges in \mathcal{V} , which do not belong to the restriction of \mathcal{V} on some collection of $p + 1$ adjacent columns.

Indeed, this definition is equivalent to assigning the independence copula to all parallelogram structures with order larger than p , as shown in the following theorem.

Theorem 7.5.

Let \mathcal{V} be a p -independent R-T-Vine specification on the vertices V_1 . Then

- i) for each $k = 1, \dots, d - 1$ the tree T_{dp+k} has exactly $k(N - p - 1)$ independence copulas and
- ii) all edges of the trees $T_{d(p+1)}, \dots, T_{Nd-1}$ are only assigned independence copulas.

Proof. (Following Beare and Seo (2015))

□

Assigning independence copula to all parallelogram structures with order higher than p , yields a number of $(N - 1) * p * d^2 + \frac{(d - 1)^2 + (d - 1)}{2}$ copulas to be specified. For $p = 1$ this especially is the same reduction as in the stationary case. Combining this effect with stationarity even reduces the number to $p * d^2 + \frac{(d - 1)^2 + (d - 1)}{2}$. Again this can be seen in Figure 7.2 in the following.

Especially, all information concerning the assigned copula families can be stored in a $(d * (p + 1) - 1)^2$ -dimensional matrix instead of a $(N * d - 1)^2$ -dimensional one.

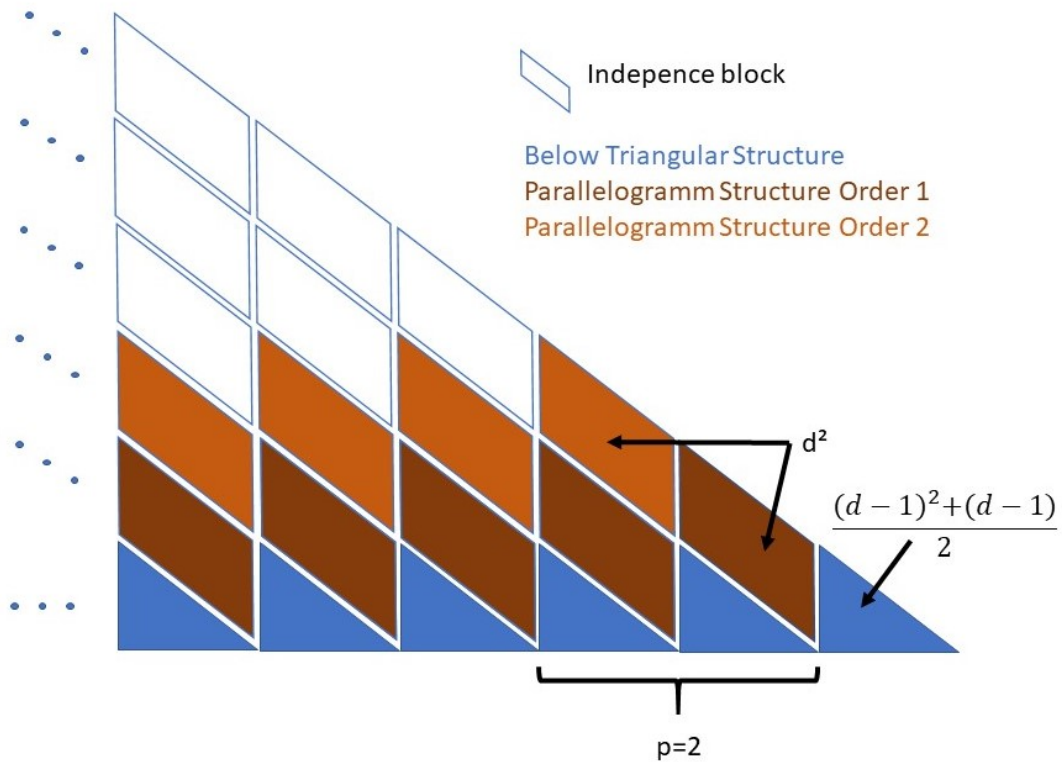


Figure 7.2: Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series with stationarity and autoregressive order $p = 2$.

So, now it is just left to show that indeed the p -Markov property implies the p independence.

Theorem 7.6.

Let the joint distribution of $(\mathbb{X}_1, \dots, \mathbb{X}_N)$ realise the vine copula specification $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ on N time steps and sectional order d . If $(\mathbb{X}_1, \dots, \mathbb{X}_N)$ is p -Markov then $(\mathbb{F}, \mathcal{V}, \mathcal{C})$ is p -independent.

Proof. (Following Beare and Seo (2015))

□

7.2.3 Example Calculation: Reduction of Complexity

In this section, we assume different numbers for time steps N and dimensions d . The following table summarises the number of different copulas to be estimated within the R-T-Vine specification, assuming no reducing factors, only stationarity, only autoregressive order one and two and stationarity combined with autoregressive order one and two.

Model/# Copulas specified for	N=100, d=5	N=5, d=100	N=100, d=100
Full Model	124,750	124,750	49,995,000
Stationary Model	2,485	44,950	994,950
AR(2) Model	4,960	89,900	1,989,900
AR(1) Model	2,485	44,950	994,950
AR(2)-stationary Model	60	24,950	24,950
AR(1)-stationary Model	35	14,950	14,950

We can see a significant reduction in all three cases. However, the effects become stronger for higher time dimension N and lower sectional dimension d . Especially, the total number of copulas for the combined effect does not depend on N , but only on the dimension d .

As the maximum likelihood estimation of the parameters for our vine specification, which will be described in more detail in the next chapter, is really time-consuming for the full model, we will use the stationary AR(2) model instead.

Chapter 8

Maximum Likelihood Estimation

8.1 Introduction to MLE

Before going into the details of the maximum likelihood estimation (MLE) of the R-T-Vine specification, we will give an overview on the MLE procedure following Millar (2011) and Shao (2007).

Let $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)^T$ be a given realisation of the sample $\mathbf{Y} = (\mathbb{Y}_1, \dots, \mathbb{Y}_n)^T$ with density function f . The value of the joint density function for our realisations is given by $f(\mathbf{Y} = \mathbf{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p), p \in \mathbb{N}$ is an unknown parameter vector we need to estimate. The estimator of $\boldsymbol{\theta}$ will be given by $\hat{\boldsymbol{\theta}}_n$, maximising the likelihood function given by

$$L(\boldsymbol{\theta}) = f(\mathbf{Y} = \mathbf{y}|\boldsymbol{\theta}),$$

with n denoting the sample size. For a twice differentiable $L(\boldsymbol{\theta})$, we only need to calculate the roots of the first derivative and check the second order condition. However, it is often easier to take the logarithm of the likelihood function, the log-likelihood $l(\boldsymbol{\theta}) = \ln(L(\boldsymbol{\theta}))$, instead. This is indeed admissible as the logarithm is a strictly monotone function.

Example 8.1. Let Y be a random variable with density $f(y|\boldsymbol{\theta})$ and \mathbf{Y} be an independent identical sample of Y with realisations given by \mathbf{y} . Then the likelihood function is given by:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n f(y_i|\boldsymbol{\theta})$$

and the log-likelihood by:

$$l(\boldsymbol{\theta}) = \sum_{i=1}^n \ln(f(y_i|\boldsymbol{\theta})).$$

Remark 8.2. The MLE estimator $\hat{\boldsymbol{\theta}}_n$ has some useful properties. The following two are taken out of Shao (2007) for independent identical distributed data.

Asymptotic Normality: The estimator $\hat{\boldsymbol{\theta}}_n$ is asymptotically normal distributed, i.e. $\sqrt{n}\hat{\boldsymbol{\theta}}_n \xrightarrow{d} \hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \mathcal{I}(\boldsymbol{\theta})^{-1})$, where $\mathcal{I}(\boldsymbol{\theta})$ is the Fisher information matrix.

Consistency: The estimator $\hat{\boldsymbol{\theta}}_n$ is consistent, i.e. $\hat{\boldsymbol{\theta}}_n \xrightarrow{P} \boldsymbol{\theta}$ as $n \rightarrow \infty$.

8.2 Model Specification for R-T-Vine Copula Model

We will now briefly review the (sequential) maximum likelihood estimation arising in our temporal vine specification.

Given the regular vine matrix of our R-T-Vine specification, we need to find the optimal copula families and parameters. This is usually done by optimising some information criteria, such as the Akaike or Bayesian Information Criteria (AIC/BIC).

Definition 8.3. AIC/BIC Let k be the number of estimated parameters and n the number of data points, then the AIC is given by

$$AIC = 2k - 2l(\hat{\boldsymbol{\theta}})$$

and the BIC by

$$BIC = k \ln(n) - 2l(\hat{\boldsymbol{\theta}}).$$

For finding the optimal copulas we need to minimise the information criteria. This corresponds to choosing the copula families on each edge in our vine structure \mathcal{V} , performing an MLE for the parameters and using the regular vine pair copula construction as described in Section 3.2.2 of this thesis to calculate the log-likelihood. Recall that the density is given by

$$f = \prod_{i=1}^d f_i \prod_{i,j|\delta \in E(\mathcal{V})} c_{i,j|\delta}(F_{i|\delta}, F_{j|\delta})$$

as stated in Theorem 3.12.

Remark 8.4. In general, a copula family to every edge in the selected vine specification needs to be assigned. However, when assuming stationarity and an autoregressive order of p within our R-T-Vine approach this optimisation is simplified, as we assign the independence or the same copulas to many edges, as already discussed in the previous chapter.

Sequential BIC and MLE in R-T-Vine Specifications

Remark 8.5. Within the implementation we determine the (conditional-) bivariate-copulas sequentially, i.e. tree by tree starting at the first tree. We start by fitting the copulas of the first tree using the $U_{0,1}$ -distributed dataset. For the following trees we have to form pseudo-observations using the (conditional-) joint distribution functions implied by the copulas obtained in the previous tree(s). With this pseudo-observations the selection is again straightforward.

To better understand this procedure of the copula fitting, let us discuss a $(n \times 2)$ -dimensional example here. We assume an autoregressive order of 1 and therefore only consider the last 4 rows and columns of the matrix representation below. The idea of sequentially estimating the copulas is mainly following Dissmann et al. (2012) slightly adapted to our model.

Let the excerpt from the R-T-vine be given by the matrix
$$\begin{bmatrix} B_2 & & & \\ B_1 & A_2 & & \\ A_1 & B_1 & B_1 & \\ A_2 & A_1 & A_1 & A_1 \end{bmatrix},$$
 i.e. we have a cross-sectional C-/D-vine connected via the series of sectional dimension A .

We denote the $(n \times 2)$ -dimensional dataset of $U_{0,1}$ distributed observations from time 1 to n by $\mathbb{D}_{(1,n)} = (A_i, B_i)_{i=1,\dots,n}$, the set of densities of the allowed copula families by \mathcal{F} and for each $c(\cdot, \cdot) \in \mathcal{F}$ the parameter-vector with k_c parameters of the copula by θ_c .

Following the BIC and MLE optimisation the optimal copula of the edge (A_1, B_1) (and due to our assumption in the R-T-vine model also of $(A_i, B_i)_{i=2,\dots,n}$) in the first tree is given by the following minimisation:

$$\min_{c \in \mathcal{F}} k_c \ln(n) - 2 \max_{\theta_c} \sum_{i=1}^n \ln(c(A_i, B_i | \theta_c)).$$

For the edge (A_1, A_2) (and $(A_i, A_{i+1})_{i=1,\dots,n-1}$) we need to have a look at the lagged data, i.e. we split the dataset into a dataset of dimension $((n-1) \times 4)$, given by $(\mathbb{D}_{(1,n-1)}, \mathbb{D}_{(2,n)})$, and the maximisation problem is given by:

$$\min_{c \in \mathcal{F}} k_c \ln(n-1) - 2 \max_{\theta_c} \sum_{i=1}^{n-1} \ln(c(A_i, A_{i+1} | \theta_c)).$$

Note that this only holds as we assume $U_{0,1}$ distributed observations (as also requested in most implementations), since the joint density can be decomposed into the

copula density and the marginal densities and here the marginals are constantly equal to 1.

Recall from Chapter 3 that we can recursively obtain the conditional distribution functions at the edge $(i, j) \in E(T_k)$ for $k \geq 2$ via

$$F_{i|\delta} = \frac{\partial C_{i',j'|\delta'}(F_{i'|\delta'}, F_{j'|\delta'})}{\partial F_{j'|\delta'}} \quad \text{and} \quad F_{j|\delta} = \frac{\partial C_{i'',j''|\delta''}(F_{i''|\delta''}, F_{j''|\delta''})}{\partial F_{j''|\delta''}}, \quad (8.1)$$

where (i', j') and (i'', j'') are the edges in $E(T_{k-1})$, which are connected via (i, j) , with $i = i'$ and $j = j''$. And for $(i, j) \in E(T_1)$ we have $\delta = \emptyset$ and hence $F_{i|\delta} = F_i$ and $F_{j|\delta} = F_j$ respectively.

Having determined all optimal copulas and the resulting conditional distribution functions in one tree, we can use them to obtain the optimal copulas in the following one.

In our example, for the edge $(A_2, B_1|A_1)$ (and again also for $(A_{i+1}, B_i|A_i)_{i=1, \dots, n-1}$) in the second tree, we need to minimise over all $c \in \mathcal{F}$:

$$k_c \ln(n-1) - 2 \max_{\theta_c} \sum_{i=1}^{n-1} \ln[c_{A_2, B_1|A_1}(F_{B_1|A_1}(A_i, B_i), F_{A_{i+1}|A_i}|\theta_c),$$

where we need to recursively calculate $F_{B_1|A_1}$ and $F_{A_2|A_1}$ via (8.1). For following trees and higher dimensional vine structures this procedure can easily be extended.

Chapter 9

Application: Major Stock Indices

9.1 Dataset

The considered data set consists of the monthly closing prices from beginning of 2007 up to June 2018 (138 observations) of five major stock indices: S&P500, DAX, Nikkei, FTSE and SMI. The data is retrieved from Yahoo finance¹.

9.1.1 Log>Returns

As already seen in Chapter 2, the closing prices of the major stock indices are not stationary. In finance, one assumes the asset price $(S_t)_{t \in T}$ to follow a continuous discounting, i.e.

$$S_t = S_{t-1}e^r,$$

where r is the constant continuous return over the time period $t - 1$ to t . In time series analysis, logarithmic differences of consecutive prices (log-returns) use this idea. We assume that those returns are stationary, as already seen in Chapter 2. The number of observations for each time series reduces by one as we are considering the log differences.

We transform our data set into the respective log-returns. Let us assign the stock indices to the sectional dimension from 1 to 5 as shown in the following table.

¹<https://de.finance.yahoo.com>

Sectional dimension	Stock Index
1	DAX
2	S&P500
3	FTSE
4	Nikkei
5	SMI

For each sectional dimension $X \in \{1, \dots, 5\}$ in our multivariate time series $((1, t), \dots, (5, t))_{t=1, \dots, 138}$, given by our initial data set, the log-returns $(\tilde{X}_t)_{t \in 1, \dots, 137}$ are given by

$$\tilde{X}_t = \log(X_{t+1}) - \log(X_t) = \log\left(\frac{X_{t+1}}{X_t}\right).$$

The Figure 9.1 below, shows the plots of the corresponding autocorrelation functions.

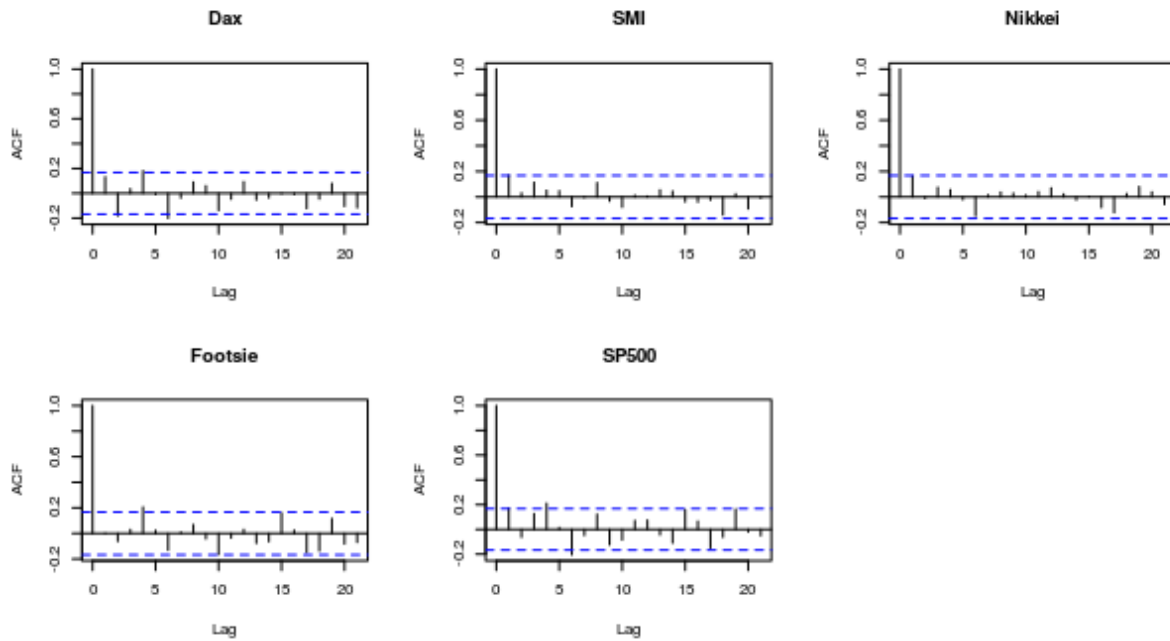


Figure 9.1: Plots of the autocorrelation functions for the log-returns of the considered data set of five major stock indices.

9.1.2 Data Transformation

So far, we have transformed the initial data into a stationary data set. In the next step, we need to normalise the data into the interval $[0, 1]$ in order to be able to apply copula models.

Recall the probability integral transformation (Theorem 2.18), i.e. $F(X) \sim U_{0,1}$.

Unfortunately, we do not have any information about the marginal distribution functions. That is why we will fit a skewed t distribution to each univariate time series to obtain the copula data of the monthly log-returns. The distribution function of a skewed t distribution at $x \in \mathbb{R}$ (Azzalini and Capitanio (2003)) is given by

$$ST(\xi, \omega^2, \alpha, \nu)(x) = \frac{2}{\omega} t(z, \nu) T\left(\alpha z \sqrt{\frac{\nu+1}{\nu+z^2}}, \nu+1\right), \text{ with } z = \frac{x-\xi}{\omega},$$

where $t(\cdot, \tau)$ and $T(\cdot, \tau)$ denote the density respectively the distribution function of the Student's t distribution.

Let $(\tilde{X}_t)_{t \in T}$ denote the univariate time series in one sectional dimension of our log-returns. We use the skewed student t innovations for all marginal series with parametrisation as in Azzalini and Capitanio (2003), i.e. the skewness is captured by only one parameter. This distribution type is well suited for typical heavy-tailed and skewed economic time series. Figure 9.2 below displays the respective histograms and fitted distribution functions, using the *sstd* functions from the **R**-package *fGarch*. We can see a good fitting of the fitted distributions, covering the negative skewness and the fat tails, to our data set.

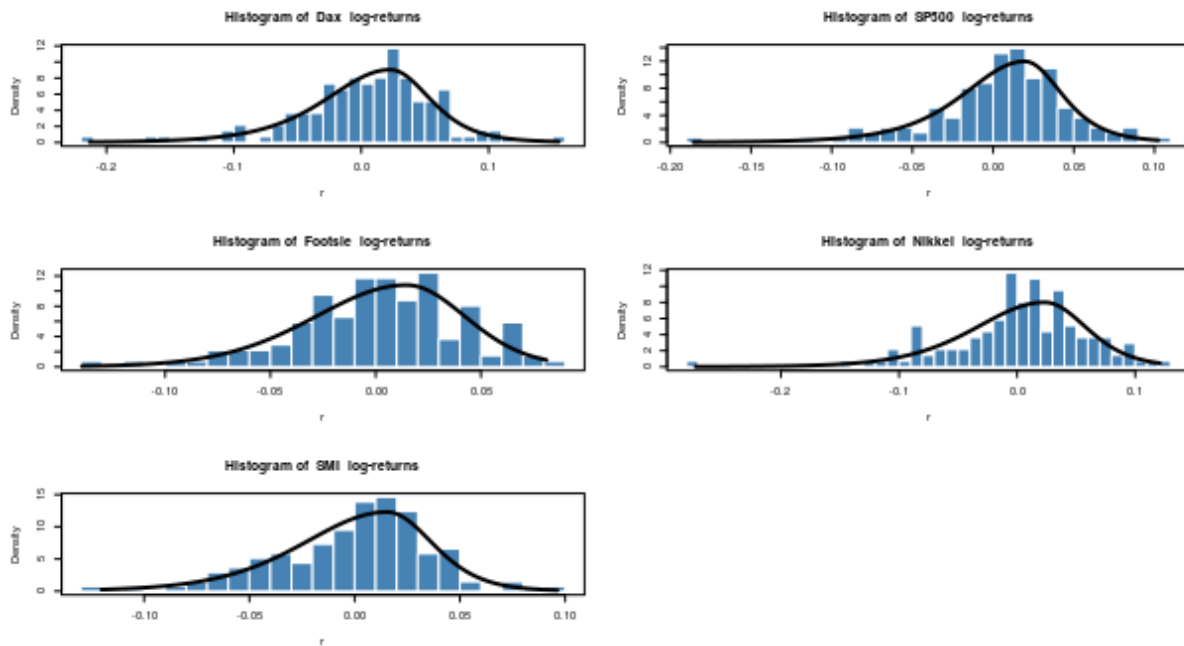


Figure 9.2: Histogramm of log-returns with fitted skewed t distributions for the five selected indices.

Finally, we apply the obtained skewed student t distribution functions $F_{\tilde{X}}$ to the series of the respective log-returns, i.e. $F_{\tilde{X}}(\tilde{X}_t) \sim U_{0,1}$, in all five sectional dimensions. The initial data set is now transformed into a stationary time series of uncorrelated uniform

distributed data-points. Furthermore, we computed the Kendall's τ in the following table and Figure 9.3 displays the pair-plots of all possible pairs.

	DAX	S&P500	FTSE	Nikkei	SMI
DAX	1.000	0.578	0.545	0.470	0.509
S&P500	0.578	1.000	0.587	0.432	0.543
FTSE	0.545	0.587	1.000	0.350	0.463
Nikkei	0.470	0.432	0.350	1.000	0.410
SMI	0.509	0.543	0.463	0.410	1.000

Unsurprisingly we find, that FTSE and S&P500 have the strongest dependence, indicated by the highest Kendall's τ in the table above. Further, the Nikkei is comparatively weakly dependent on the other indices as the respective row/column contains the lowest values.

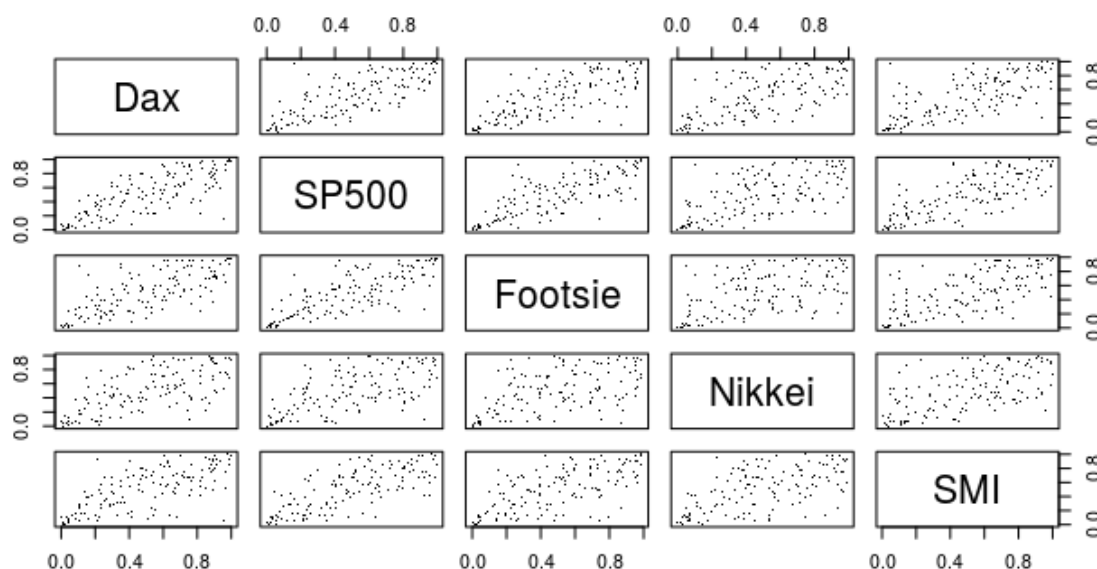


Figure 9.3: Pair-plots of transformed monthly log-returns of the five selected indices.

9.2 Estimation of R-T-Vine Specification

Using the transformed data from the previous section, we will estimate a regular temporal vine specification. This will be done in three steps: First we need to select the cross-sectional structure (either via the empirical Kendall's τ or some information criteria), second we determine the connecting edges (again using τ or some information criteria) and third using the obtained regular vine structure, we sequentially choose

the best copula families via BIC selection criteria and perform a maximum-likelihood estimation to obtain the R-T-vine specification, which then can be used for performing predictions.

Note that the selection of cross-sectional structure and connecting edges via BIC already requires the sequential fitting usually done in step three. Especially, we need to fit the copulas for all possible regular vine structures in the sectional dimension and respectively every R-T-vine structure implied by all possible connecting edges and the previously obtained cross-sectional structure.

9.2.1 Selecting Cross-Sectional Structure

Let $(\mathbb{D}_t)_{t=1,\dots,N}$ denote the time series of our complete transformed dataset with $d = 5$ sectional and $N = 137$ time dimensions. We use the optimal structure given by the implemented function *vinecop()* in the R-package *rvinecopulib* to automatically fit a regular vine describing the cross-sectional relation in our data.

Now, assuming $(\mathbb{D}_t)_{t=1,\dots,N}$ independent identically distributed, we fit the cross-sectional regular vine structure.

The implemented criteria for the tree selection is maximising the empirical Kendall's τ , i.e. the optimal vine is given by the solution of

$$\operatorname{argmax}_{\mathcal{V} \in \mathbb{V}(V_1)} \sum_{e_{ij} \in \mathcal{V}} |\tau_{ij}|, \quad (9.1)$$

where $\mathbb{V}(V_1)$ denotes the set of all possible vines on V_1 vertices.

For our transformed dataset $(\mathbb{D}_t)_{t=1,\dots,N}$ with sectional dimensions $\{1 = \text{DAX}, 2 = \text{S\&P500}, 3 = \text{FTSE}, 4 = \text{Nikkei}, 5 = \text{SMI}\}$ the optimal cross-sectional regular vine

matrix is given by $\begin{bmatrix} 3 \\ 4 & 5 \\ 5 & 4 & 4 \\ 1 & 1 & 2 & 2 \\ 2 & 2 & 1 & 1 & 1 \end{bmatrix}$. The corresponding trees are visualised in Figure 9.4

below.

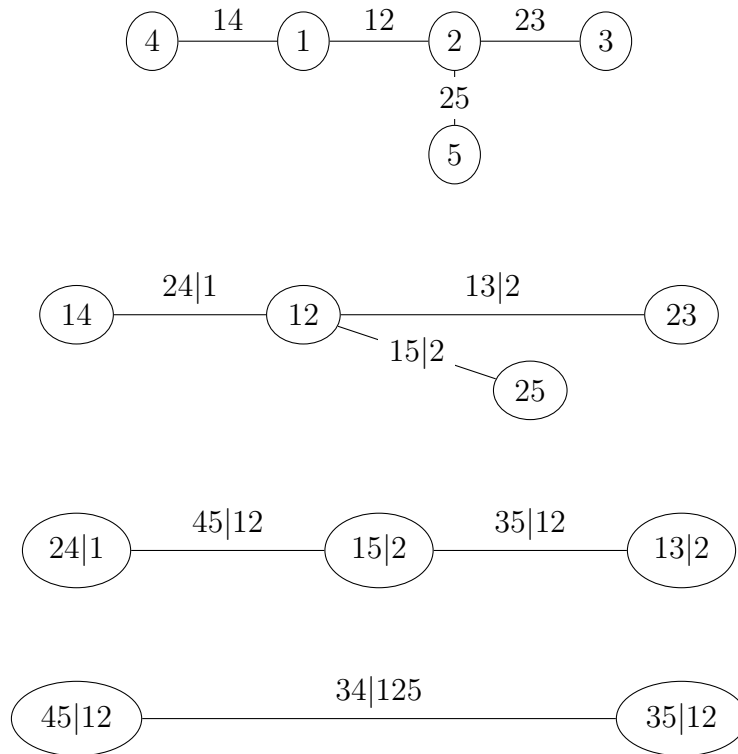


Figure 9.4: Fitted regular vine to our transformed data set.

This is indeed a more general regular vine structure, i.e. no C- nor D-vine.

In general, we just need this optimal vine to continue with our temporal vine estimation. However, as we would like to compare our new approach with the M-vine approach from Beare and Seo (2015) at this point, we need to further specify the respective copula families and parameters as described in Section 8.2 with respect to the BIC and the MLE.

For comparison with the classical M-vine model we need to fit a D-vine to the cross-sectional structure first. The optimal D-vine concerning the BIC is given by the

matrix $\begin{bmatrix} 4 \\ 3 & 2 \\ 1 & 3 & 5 \\ 5 & 1 & 3 & 1 \\ 2 & 5 & 1 & 3 & 3 \end{bmatrix}$. To obtain this optimal D-vine matrix we need to perform a full

model fitting for all $5!$ possible D-vine structures. We therefore perform a sequential BIC model selection based on parameter selected via MLE as shown in Chapter 8. For the cross-sectional structure we can use the complete dataset, i.e. we do not need to split up into the lagged data.

Comparing the information criteria for the non C- nor D-vine, obtained by maximising

the empirical Kendall's τ , with the D-vine model (as used in the classical M-vine approach), we find the following results:

Model/Criteria	Log-Likelihood	AIC	BIC
R-vine (τ)	261.44	-508.88	-488.44
D-vine (BIC)	273.58	-527.16	-497.96

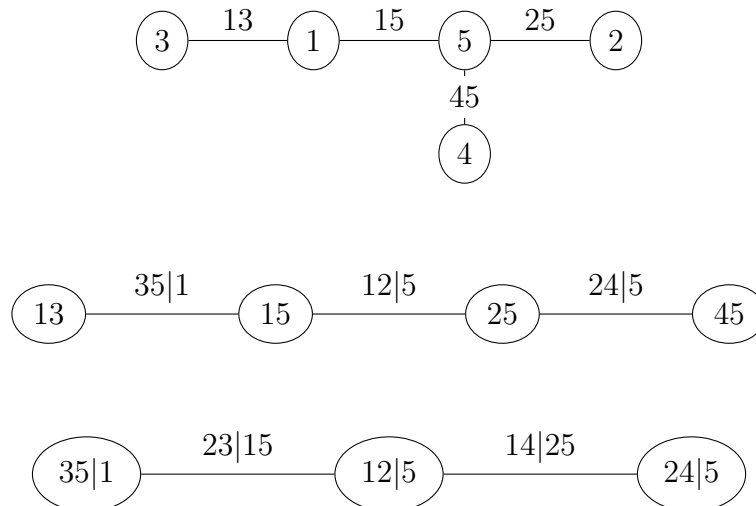
At least for the choice of the cross-sectional structure, the generalised M-vine (and the R-T-Vine) approach is outperformed by the model developed by Beare and Seo (2015) in all criteria.

Remark 9.1. However, given some more computational effort, we are able to further improve the R-T-vine model. Therefore, instead of maximising the empirical Kendall's τ , we need to optimise every possible vine structure model with respect to the selection criteria (e.g. maximising the BIC here).

This means for every possible vine structure on five vertices, we have to perform a full model fit. We therefore consider all $5! \cdot 4! \cdot 3! \cdot 2! \cdot 1! = 34,560$ possible below triangular matrices and neglect those matrices not full-filling the proximity condition. For a study on the number of possible vines and a more efficient algorithm we refer to Morales-N'apoles O. (2010). This time-consuming optimisation yields a slightly different optimal cross-

sectional structure given by
$$\begin{bmatrix} 3 & & & & \\ 4 & 4 & & & \\ 2 & 1 & 2 & & \\ 5 & 2 & 1 & 1 & \\ 1 & 5 & 5 & 5 & 5 \end{bmatrix}$$
. The corresponding trees are visualised in

Figure 9.5 below.



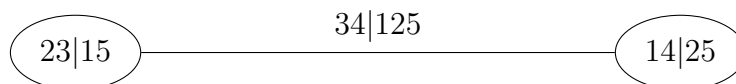


Figure 9.5: Fitted regular vine to our transformed data set using the maximisation of the BIC.

This model outperforms the two models above with respect to all selection criteria as summarised in the following table:

Model/Criteria	Log-Likelihood	AIC	BIC
R-vine (τ)	261.44	-508.88	-488.44
D-vine (BIC)	273.58	-527.16	-497.96
R-vine (BIC)	274.02	-528.04	-498.84

In the following we will continue working with both regular vine structures obtained by either the maximisation of Kendall's τ or the BIC.

Note that the *tvinecop()* function implemented in **R** is not performing the computational costly optimisation procedure.

9.2.2 Selecting Connecting Edges

Having determined the optimal cross-sectional structure, we are now interested in finding the optimal connecting edges. As in the implemented algorithm for finding the optimal trees in the R-packages *VineCopula*, *rvinecopulib* and *tvine* we use the edge, maximising the empirical Kendall's τ for our vine specification.

Recall that generally the optimal tree selection is performed by the optimisation problem in Equation (9.1). We are only interested in finding the optimal inter-temporal edges given the already recurring cross-sectional structure \mathcal{V} . Therefore, the optimisation problem simplifies to

$$\operatorname{argmax}_{\mathcal{V}^* \in \mathcal{M}(t, \mathcal{V})} \sum_{e_{ij} \in \mathcal{V}^*} |\tau_{ij}| = \operatorname{argmax}_{X, Y \in \{1, \dots, d\}} \tau((X_t)_{t=1, \dots, N-1}, (Y_s)_{s=2, \dots, N}) \quad (9.2)$$

where $\mathcal{M}(t, \mathcal{V})$ denotes the set of all possible R-T-Vines with time horizon t and cross-sectional structure \mathcal{V} . Using, the pre-defined algorithms in the not yet published R-package *tvine*, we see the maximal τ is given by the connection of FTSE at time t with itself at $t + 1$, i.e. the optimal connecting edge is given by (3, 3).

$$(9.4) = \begin{bmatrix} 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 3 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 3 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 5 & 3 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 5 & 3 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 2 & 4 & 5 & 3 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 8 & 1 & 2 & 4 & 5 & 3 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 10 & 8 & 1 & 2 & 4 & 5 & 3 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 9 & 10 & 8 & 1 & 2 & 4 & 5 & 3 & 6 & 0 & 0 & 0 & 0 & 0 \\ 6 & 7 & 9 & 10 & 8 & 1 & 2 & 4 & 5 & 3 & 3 & 0 & 0 & 0 & 0 \\ 14 & 6 & 7 & 9 & 10 & 9 & 1 & 2 & 4 & 5 & 4 & 5 & 0 & 0 & 0 \\ 15 & 14 & 6 & 7 & 9 & 10 & 9 & 1 & 2 & 4 & 5 & 4 & 4 & 0 & 0 \\ 11 & 11 & 12 & 6 & 7 & 6 & 6 & 7 & 1 & 2 & 1 & 1 & 2 & 2 & 0 \\ 12 & 12 & 11 & 11 & 6 & 7 & 7 & 6 & 6 & 1 & 2 & 2 & 1 & 1 & 1 \end{bmatrix}.$$

9.2.3 R-T-Vine Estimation

With the help of this matrix we now need to estimate the optimal families and the corresponding parameters for each edge in the temporal vine specification.

As we are dealing with stationary time series and assume an autoregressive order p (in our example $p = 2$ as this yields the lowest BIC for all $p \leq 2$ in the R-T-vine estimation using the empirical Kendall's τ) recall from Chapter 7, that we can describe the whole copula family matrix by the first $p + 1$ time steps in the matrix, containing the triangular structure and the parallelogram structures from 1 to p .

As already mentioned in the previous section the sequential fitting of the copulas in the **cross-sectional structure** can be performed on the complete dataset. For the estimation of the copulas in the **parallelogram structure** of order k in 1 to p , we need the lagged data with lag k . This is intuitively clear, however, let us take a look at the exemplary edge $(1, 7|6) = (6, 12|11) = (A_i, B_{i+1}|A_{i+1})_{i=1, \dots, N-1}$ in the parallelogram structure of order one in (9.4). For understanding the dependence structure we need to consider the data $(A_t, B_{t+1}|A_{t+1})_{t=1, \dots, N-1}$, i.e. we need the dataset $\tilde{\mathbb{D}} = ((\mathbb{D}_t)_{t=1, \dots, N-1}, (\mathbb{D}_t)_{t=2, \dots, N})$

In general, for the edges in the parallelogram structure of order k we need to split up the data into a dataset of dimension $(k + 1)d \times (N - k)$ for $k = 1, \dots, p$. The new

dataset $(\tilde{\mathbb{D}}_t)_{t=1,\dots,N-k}$ is given by

$$\tilde{\mathbb{D}} = ((\mathbb{D}_t)_{t=1,\dots,N-k}, \dots, (\mathbb{D}_t)_{t=k+1,\dots,N}).$$

As described in Chapter 8 the lagged data will be transformed to pseudo-observation to sequentially determine the optimal conditional bivariate copula at the respective edge using the BIC (and MLE). We always use the complete available data set for calculation (e.g. in the cross-sectional structure we will start at $t = 1$ even when estimating the edge in the time dimension $t > 1$) and ensure assigning the same copula to recurring edges in the next time step.

This whole procedure is captured by the function `twinecop()` in the not yet published **R**-package `twine`, developed by Thomas Nagler on basis of this thesis.

The contour plot of the fitted copulas for the R-T-vine specification given by (9.3) using this function can be found in Figure 9.6 below.

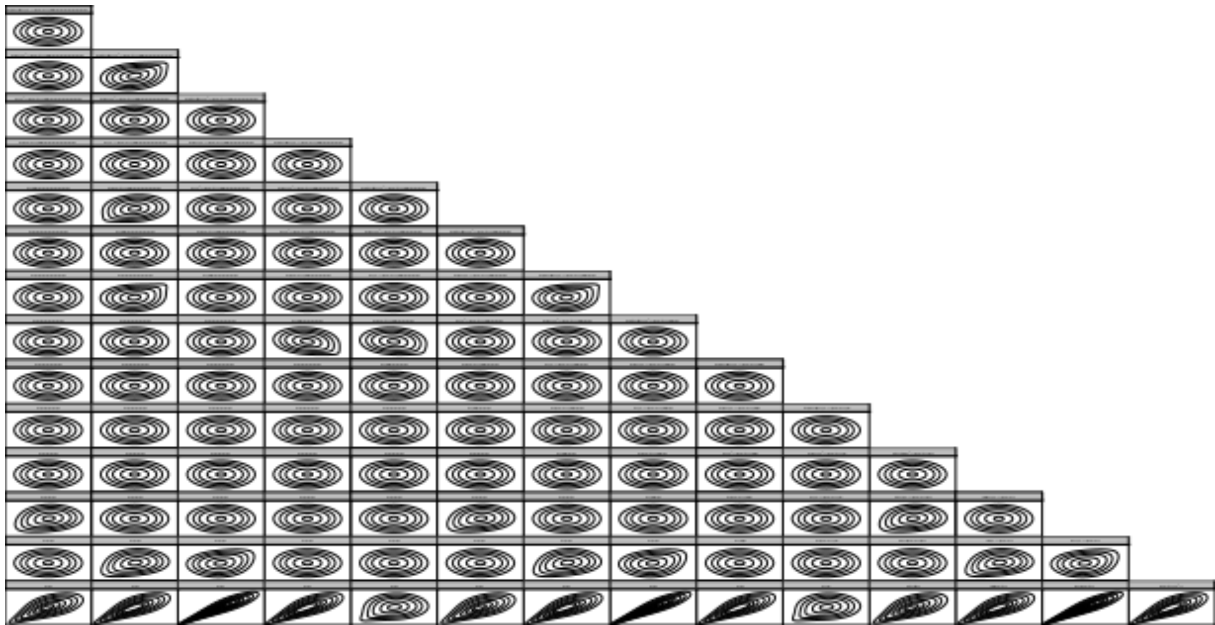


Figure 9.6: Contour plot of the (conditional) bivariate copulas obtained by `twinecop()` for our transformed data set.

9.2.4 Outperformance of Classical M-Vine

When using the empirical Kendall's τ for choosing the tree structure, we have already seen the better performance concerning the information criteria of the classical M-vine in the cross-sectional structure. Fitting the overall R-T-Vine model and comparing it with

the classical M-vine approach reveals the same result.

When considering the generalised M-vine (the R-T-Vine with connecting edges given by $(1, 1)$), the BIC is in between the one from classical M-vine and the one from the R-T-vine model. The results are summarised in the following table.

Model[cross-sectional structure, connecting edge]	Log-Likelihood	AIC	BIC
R-T-Vine [R-vine (τ) , $(3, 3)$] (9.3)	825.41	-1588.83	-1498.31
generalised M-vine [R-vine (τ) , $(1, 1)$] (9.4)	824.33	-1586.65	-1496.13
classical M-vine [D-vine (BIC), $(3, 3)$] (9.5)	867.41	-1648.81	-1523.26

The matrix representation of the classical M-vine for $N = 3$ is given by

$$(9.5) = \begin{bmatrix} 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 2 & 4 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 2 & 4 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 5 & 2 & 4 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 3 & 1 & 5 & 2 & 4 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 9 & 3 & 1 & 5 & 2 & 4 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 7 & 9 & 3 & 1 & 5 & 2 & 4 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 10 & 7 & 9 & 3 & 1 & 5 & 2 & 4 & 8 & 0 & 0 & 0 & 0 & 0 \\ 8 & 6 & 10 & 7 & 9 & 3 & 1 & 5 & 2 & 4 & 4 & 0 & 0 & 0 & 0 \\ 13 & 8 & 6 & 10 & 7 & 8 & 3 & 1 & 5 & 2 & 3 & 2 & 0 & 0 & 0 \\ 11 & 13 & 8 & 6 & 10 & 6 & 8 & 3 & 1 & 5 & 1 & 3 & 5 & 0 & 0 \\ 15 & 11 & 13 & 8 & 6 & 10 & 6 & 8 & 3 & 1 & 5 & 1 & 3 & 1 & 0 \\ 12 & 15 & 11 & 13 & 8 & 7 & 10 & 6 & 8 & 3 & 2 & 5 & 1 & 3 & 3 \end{bmatrix}.$$

Using the BIC Optimisation

Again, we might also use a different approach. Analogue to the selection of the cross-sectional structure we can perform an optimisation of the selection criteria (here BIC) instead of the empirical Kendall's τ for the connecting edges. Therefore, we fit a R-T-vine model based on the cross-sectional R-vine (BIC) for every possible connecting edge. Using the so obtained optimal cross-sectional structure we come up with the connecting edge $(1, 3)$. Hence, the sub-diagonal part, the diagonal as well as the triangular structure of the matrix will be modified compared to the respective generalised M-vine.

$\begin{bmatrix} 3 & & & & \\ 4 & 4 & & & 0 \\ 2 & 1 & 2 & & \\ 5 & 2 & 1 & 1 & \\ 1 & 5 & 5 & 5 & 5 \end{bmatrix}$	In vertex	Order of diagonal structure
	3	4, 2, 5, 1, 3
	Out vertex	Order of sub-diagonal structure
1	3, 4, 2, 5, 1	

The triangular structure given by this new diagonal order (still describing the same

regular vine structure) can be obtained with Algorithm 4 and yields:

$$\begin{bmatrix} 4 & & & & \\ 3 & 2 & & & 0 \\ 1 & 3 & 5 & & \\ 2 & 1 & 3 & 1 & \\ 5 & 5 & 1 & 3 & 3 \end{bmatrix}.$$

Here, we see a clear out-performance of the R-T-vine versus the generalised M-vine based on the cross-sectional structure optimised with respect to BIC and the other approaches, such as the classical M-vine, discussed above. The best three performing model are ranked and summarised in the following table.

Model[cross-sectional structure, connecting edge]	Log-Likelihood	AIC	BIC
R-T-Vine [R-vine (BIC), (1, 3)] (9.6)	994.77	-1909.55	-1792.75
classical M-vine [D-vine (BIC), (3, 3)] (9.5)	867.41	-1648.81	-1523.26
generalised M-vine [R-vine (BIC), (5, 5)] (9.7)	866.42	-1646.85	-1521.29

For completeness we will also present the matrix of the optimal R-T-vine (9.6) and the generalised M-vine (9.7) using the BIC optimisation. The respective contour plot of the R-T-vine is displayed in Figure 9.7 in the following.

$$(9.6) = \begin{bmatrix} 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 3 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 2 & 4 & 3 & 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 2 & 4 & 3 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 5 & 2 & 4 & 3 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 8 & 1 & 5 & 2 & 4 & 3 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 9 & 8 & 1 & 5 & 2 & 4 & 3 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 7 & 9 & 8 & 1 & 5 & 2 & 4 & 3 & 8 & 0 & 0 & 0 & 0 & 0 \\ 6 & 10 & 7 & 9 & 8 & 1 & 5 & 2 & 4 & 3 & 4 & 0 & 0 & 0 & 0 \\ 13 & 6 & 10 & 7 & 9 & 8 & 1 & 5 & 2 & 4 & 3 & 2 & 0 & 0 & 0 \\ 11 & 13 & 6 & 10 & 7 & 6 & 8 & 1 & 5 & 2 & 1 & 3 & 5 & 0 & 0 \\ 12 & 11 & 13 & 6 & 10 & 7 & 6 & 8 & 1 & 5 & 2 & 1 & 3 & 1 & 0 \\ 15 & 15 & 11 & 13 & 6 & 10 & 10 & 6 & 8 & 1 & 5 & 5 & 1 & 3 & 3 \end{bmatrix}$$

$$(9.7) = \begin{bmatrix} 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 3 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 4 & 3 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 1 & 2 & 4 & 3 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 5 & 1 & 2 & 4 & 3 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 8 & 5 & 1 & 2 & 4 & 3 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 9 & 8 & 5 & 1 & 2 & 4 & 3 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 7 & 9 & 8 & 5 & 1 & 2 & 4 & 3 & 10 & 0 & 0 & 0 & 0 & 0 \\ 10 & 6 & 7 & 9 & 8 & 5 & 1 & 2 & 4 & 3 & 3 & 0 & 0 & 0 & 0 \\ 14 & 10 & 6 & 7 & 9 & 9 & 5 & 1 & 2 & 4 & 4 & 4 & 0 & 0 & 0 \\ 12 & 11 & 10 & 6 & 7 & 7 & 6 & 5 & 1 & 2 & 2 & 1 & 2 & 0 & 0 \\ 15 & 12 & 11 & 10 & 6 & 10 & 7 & 6 & 5 & 1 & 5 & 2 & 1 & 1 & 0 \\ 11 & 15 & 15 & 15 & 10 & 6 & 10 & 10 & 10 & 5 & 1 & 5 & 5 & 5 & 5 \end{bmatrix}.$$

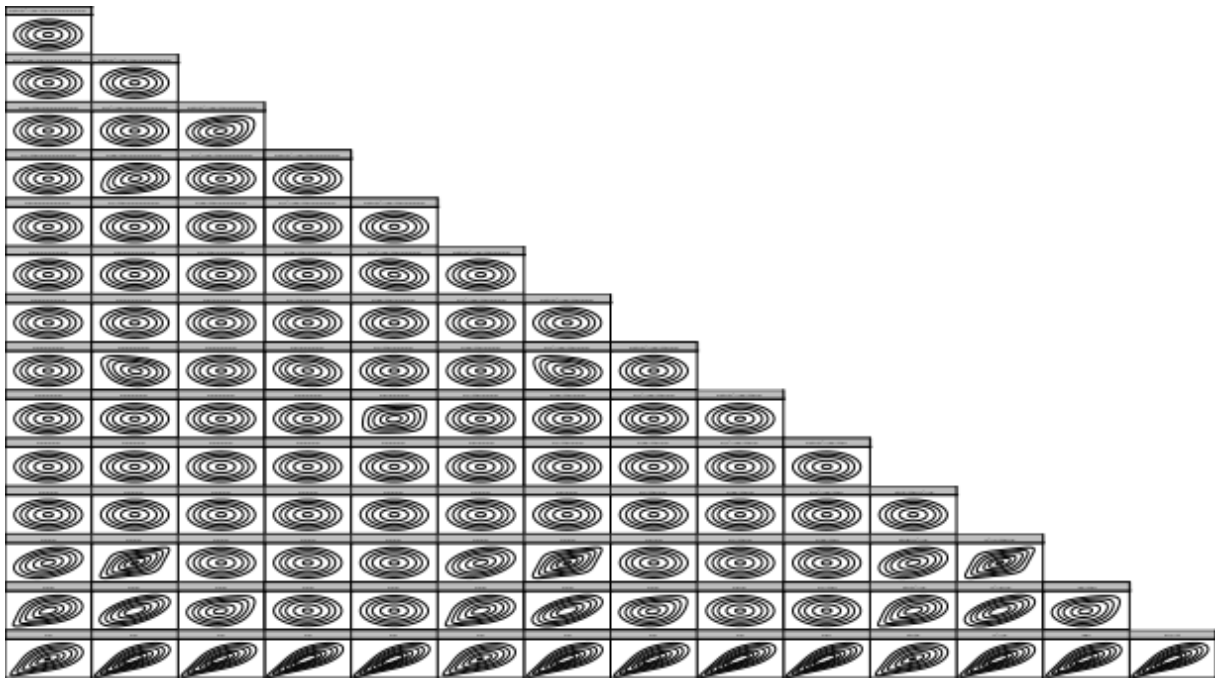


Figure 9.7: Contour plot of the (conditional) bivariate copulas obtained by the R-T-vine using the BIC for optimisation for our transformed data set.

9.3 Simulation and Comparison of Results

In the next step, we want to use our model for the prediction of monthly log-returns of our five chosen indices. We will compare the result of both R-T-Vine approaches (given by (9.3) and (9.6)) with the classical M-vine model (given by (9.5)) and a VAR(2)-Model.

The VAR(2)-Model will be fitted on the transformed copula dataset using only bivariate Gaussian copulas, which result in a multivariate normal distribution. For the conditional prediction with the help of copula models one can use the functions implemented in the not yet published **R**-package *tvine*. Let us just give a short introduction into the theory behind the simulation function implemented here.

Simulating from Regular Vine Models

For simulating the next time step in a model, one needs to specify the distribution to sample from. In regular vine models for multivariate time series we depict the distribution with the help of bivariate (conditional) copulas. A simple way to simulate every distribution was introduced by Rosenblatt (1952).

Definition 9.2. Rosenblatt Transform

The Rosenblatt transform $U = T(V)$ of a random vector $V = (V_1, \dots, V_d) \sim F$ is defined as

$$U_1 = F(V_1), U_2 = F(V_2|V_1), \dots, U_d = F(V_d|V_1, \dots, V_{d-1}),$$

where $F(v_k|v_1, \dots, v_{k-1})$ is the conditional distribution of V_k given V_1, \dots, V_{k-1} , $k = 2, \dots, d$. The vector U then consists of independent uniform variables.

Remark 9.3. For a given copula F , if U is a vector of independent random variables, $V = T^{-1}(U)$ has distribution F , i.e. the inverse of the Rosenblatt transformation

$$V_1 = F^{-1}(U_1), V_2 = F^{-1}(U_2|U_1), \dots, V_d = F^{-1}(U_d|U_1, \dots, U_{d-1}),$$

can be used to simulate from any distribution.

Hence, to simulate the next step, we first generate d independent uniform distributed realisation and then apply the Rosenblatt transformation conditioned on the d values of the previous p time steps.

Comparison - Preliminaries

The quality of the predictions will be assessed based on *continuous ranked probability score* (see Gneiting and Raftery (2007)). This is an often used tool to assess the quality of probabilistic forecasts, where the widely known tools such as the mean squared or absolute error can not directly be applied. Note that the CRPS simplifies to the mean absolute error if the observations are deterministic.

Definition 9.4. Continuous Ranked Probability Score (CRPS)

Let X be a random variable, F the respective cumulative distribution function and x be the observation related to an empirical probabilistic forecast associated to F . The CRPS of x and F is defined as:

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - \Theta(y - x))^2 dy,$$

where Θ denotes the heavy-side function, i.e. $\Theta : \mathbb{R} \rightarrow \{0, 1\}$, with $\Theta(x) = \mathbb{I}_{\mathbb{R}_0^+}(x)$.

The higher the CRPS, the better the predicted distribution is fitting to the observation, as a high CRPS indicates a strong increase in the cumulative distribution function around the observed value, i.e. assigns high probability to the interval around a observation.

To assess our models, we transfer the copula data obtained from the conditional

prediction in \mathbf{R} (via the function *tvinecop_sim_conditional*) into log-return data. Therefore, we use the inverse of the fitted skewed t-distributions from Section 9.1.2. For simplicity reasons we compare the actual log-return data with the empirical distribution function of this transformed predictions to calculate the CRPS's. The conditional monthly log-returns for July 2018 conditioned on June and May 2018 (i.e. the first out-of-sample prediction) can be seen in Figure 9.8, using the optimal R-T-vine (9.6).

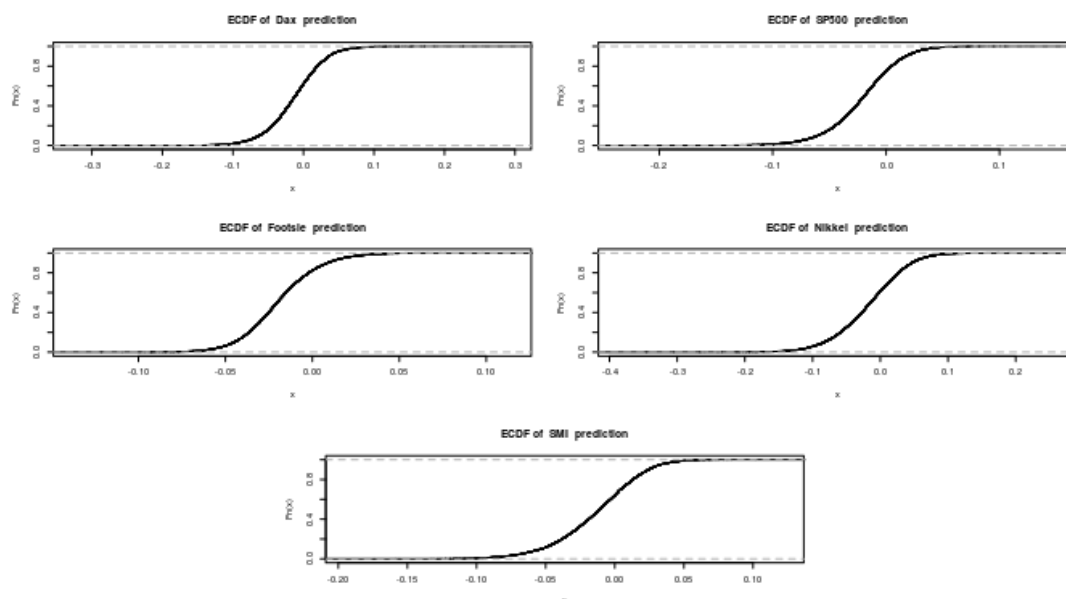


Figure 9.8: Empirical cumulative distribution functions of monthly log-returns for July 2018, based on the optimal R-T-vine (9.6).

We can see in Figure 9.8 that the obtained ECDF's of the FTSE and the S&P500 assign a higher probability to negative monthly log-returns compared to the other three indices for July 2018.

Comparison - CRPS Assesment

Next, we will asses our three models (9.6),(9.3) and (9.5) with the CRPS's of each empirical cumulative distribution function of the single indices. Therefore, we will calculate the CRPS's of the in-sample monthly log-returns from January 2018 to June 2018 and of the out-of-sample data from July 2018 to October 2018.

For each observation, we will especially simulate 100,000 one-step-ahead predictions (using *tvinecop_sim_conditional*) based on all previous observations, transform them into log-return data, calculate the empirical CRPS's with the \mathbf{R} -function *crps_sample*

from the package *scoringRules* and sum up the results.

For the in-sample set we obtain the following results:

Model/Index	DAX	S&P500	FTSE	Nikkei	SMI	Sum
R-T-vine (9.6)	0.157	0.091	0.142	0.129	0.148	0.667
R-T-vine (9.3)	0.130	0.102	0.112	0.121	0.127	0.592
classical M-vine (9.5)	0.115	0.086	0.102	0.109	0.126	0.538

Not surprisingly, we find that the computational costly model of maximising the BIC performs best. However, the second best model is the R-T-vine implemented in **R** even though we would expect it to be the worst, concerning the information criteria.

Let us now have a look at the CRPS's for the out-of-sample dataset:

Model/Index	DAX	S&P500	FTSE	Nikkei	SMI	Sum
R-T-vine (9.6)	0.131	0.127	0.108	0.147	0.113	0.626
classical M-vine (9.5)	0.116	0.112	0.089	0.142	0.099	0.558
R-T-vine (9.3)	0.117	0.088	0.093	0.125	0.103	0.526

Here, we again find that the BIC optimal R-T-vine performs best. However, the classical M-vine is out-performing the R-T-vine, using the empirical Kendall's τ , in the out-of-sample set.

Comparison with VAR(2)

As described in the beginning of this chapter, we fitted a VAR(2)-model (using only bivariate Gaussian copulas to capture the multivariate normal distribution) on the copula data obtained by transforming the log-returns via skewed t distributions. Following the very same procedure of conditional prediction discussed before, we obtained the following result concerning the CRPS's for the out-of-sample dataset:

Model/Index	DAX	S&P500	FTSE	Nikkei	SMI	Sum
VAR(2)	0.113	0.100	0.084	0.132	0.096	0.525

Hence, all copula models out-perform the VAR(2)-model with respect to the CRPS.

Conditional Prediction Quantile for DAX

We will close this section by using the best performing model, the R-T-vine model (9.6), for calculating the 95% confidence interval for the in- and out-of-sample data from January

to October 2018 of the DAX. We visualised the boundaries and the actual log-returns in Figure (9.9) below.

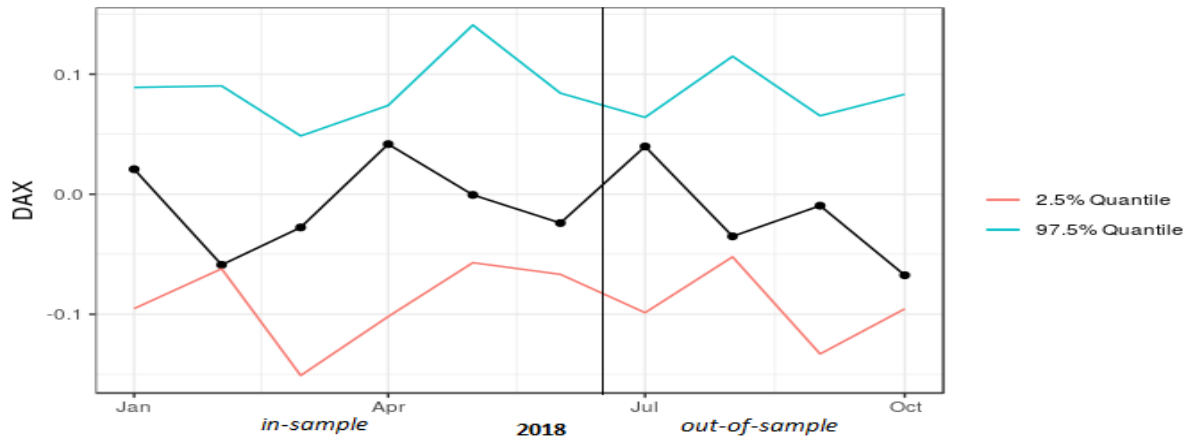


Figure 9.9: 95% confidence bounds for in- and out-of-sample monthly log-returns of the Dax index for 2018.

We can see in Figure 9.9 that all actual monthly log-returns of the selected time period lie in between the 2.5% and 97.5% quantile. This also implies a reasonable good fitting of our model.

To sum up, we found that the newly developed R-T-vine model captures more flexibility than the other models. However, when fitting the cross-sectional structure and the connecting edges to data, one has to be cautious in selecting the decision criteria for optimal tree selection. Of course the application of the new model to only one data set does not serve statistical relevance. One needs further applications to really assess the performance of the model.

Chapter 10

Wrap-Up

In this thesis, a new copula model, capturing serial, cross-serial and cross-sectional (conditional) dependence structures in stationary multivariate time series, has been developed. Therefore, we have given a short introduction into the mathematical preliminaries including the basics of time series analysis, the notion of copulas and the respective central theorems, the concept of regular vines and the application of regular vines within the pair copula construction. We have reviewed the already existing copula models such as the long D-vine by Smith (2015), the COPAR by Brechmann and Czado (2015) and the M-vine by Beare and Seo (2015). On basis of the latter, we have enhanced the model by first allowing for more general cross-sectional structures resulting in the so call generalised M-vine model. For this model, we have built an algorithm for coming up with a matrix representation for all possible cases and have been able to formalise and proof some nice features (for example: the selection of the cross-sectional structure only influences the triangular structure in the matrix representation). In the next generalisation step, we have allowed for different inter-temporal connection (changing of the connecting edges). We analysed the influence on the matrix structure and have further developed the algorithm of the generalised M-vine. We have called this new model R(egular)-T(emporal)-vine. On basis of the matrix representation and the corresponding algorithms the R-package *tvine* has been developed, including the new function *tvinecop()* by Thomas Nagler, performing a complete R-T-Vine specification. This function also captures the feature of copula selection for the R-T-Vine copula specification and the fitting of the parameters via maximum likelihood estimation, also both described within this thesis. Finally, we have applied this new implementation to monthly log-returns of five selected indices and have been able to prove the out-performance of the classical M-vine model.

Within our generalised M-vine approach, we have connected the cross-sectional

structures via a long D-vine, yielding also a D-vine in the block notation (compare Section 4.5), as this is the most promising approach for describing a serial structure. Another interesting vine structure might be given by other inter-temporal connections (i.e. the connections of the cross-sectional structures), e.g. via a C-vine. Figure 10.1 below shows the 5×3 -dimensional case, where the first time step of the first sectional dimension has been chosen as the central vertex.

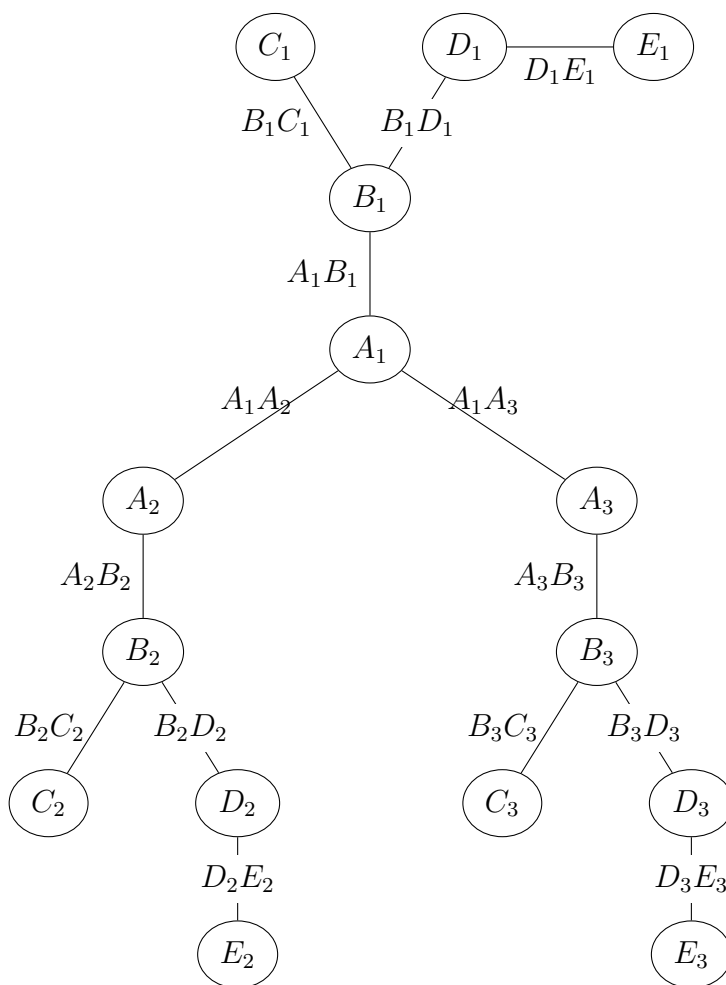


Figure 10.1: First tree of the alternative connection of the cross-sectional structures via a C-vine approach.

So far, we have only used sequential BIC selection and sequential MLE for selecting the optimal copula family and parameters. A computational costly joint MLE approach for the R-T-vine might improve the model and hence, might be interesting to consider.

and the respective starting tree, already indicating the complexity, is shown in Figure A.1 below.

Tree 1

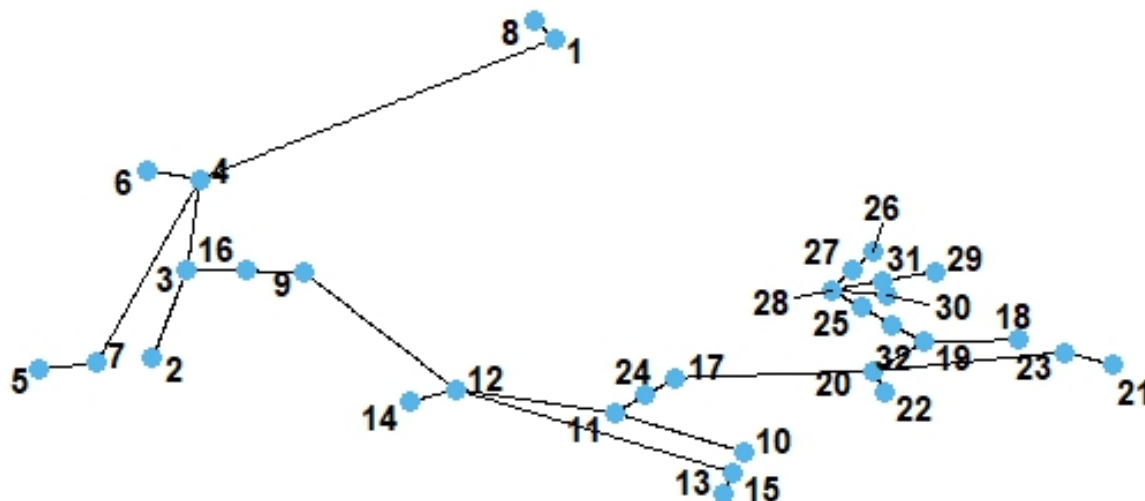


Figure A.1: First tree of the R-T-vine fitted to the daily log-returns of 8 selected stock indices [using the plot function in *rvinecopulib*].

Further, we compare the information criteria with the one of the classical and generalised M-vine in the following table:

Model/Criteria	Log-Likelihood	AIC	BIC
R-T-Vine	5655.71	-11083.41	-10668.64
generalised M-vine	5620.81	-11021.62	-10621.40
classical M-vine	5614.51	-10919.02	-10355.08

In this data set we can see a clear out-performance in all three criteria of the new model against the other models.

Appendix B

R Algorithms

Here, we provide a list of all functions and scripts programmed in **R** used in this thesis with a short description.

Besides the own programming, we heavily used the following new functions in the not yet published **R**-package *tvine* developed by Thomas Nagler on basis of this thesis:

- **tvinecop()**, fitting the R-T-vine either using the empirical Kendall's τ for tree selection or for pre-defined vines and
- **tvinecop_sim_conditional()**, predicting the next time step on basis of the last p data points and the fitted R-T-vine model.

Further we used the following **R**-packages:

- *rvinecopulib* (Nagler and Vatter 2018)
- *VineCopula* (Schepsmeier et al. 2018)
- *fGarch* (Wuertz et al. 2017)
- *scoringRules* (Jordan, Krueger, and Lerch 2018)

1. **RetrieveDataLogRet()**: It is retrieving the data for selected indices from Yahoo-finance. One can choose a start and end date, whether you want the closing prices or log-returns and the period length between each observation.
2. **myDataset()**: Wrapper of the `RetrieveDataLogRet()` function, retrieving the data used in the application in Chapter 9. Further the data is converted into copula

data using skewed t distributions and the respective parameters are stored.

3. **CoparTree**: This script builds the Copar regular vine matrix and produces the trees in Figure 4.5 using the functions implemented in *VineCopula*-package.
4. **getOrder()**: Function returning the sectional order for a given cross-sectional matrix and the in-/out-vertex. (Algorithm 2)
5. **getTriangle()**: Function returning the cross-sectional triangular structure for a new given diagonal. (Algorithm 3)
6. **updateDefault()**: This function is used in *getOrder()*, if there is a mistake in the default matrix. (Algorithm 4)
7. **TemporalVineMatrix()**: The function combines the functions 4-6 to come up with a complete R-T-vine matrix for a given cross-sectional structure and connecting edges. (Algorithm 5)
8. **TreePicture()**: Within Chapter 5 and 7 there are many figures containing only the first d trees of the R-T-vine structure, where d denotes the sectional dimension. To be able to analyse the full tree pictures, we created a function returning all 14 trees of the respective regular vine, covering the three cross-sectional cases and every possible connecting edge discussed in this thesis.
9. **DataAnalysis**: The script is used to produce the Figures and tables in Section 9.1.2 (histograms with fitted skewed t, table of Kendall's τ , pair-plot of transformed log-returns).
10. **optimalCS**: The script performs a full MLE (using the functions in the *tvine*-package) for every possible cross-sectional vine structure and returns the matrix of the structure with the highest log-likelihood. [Note: this is very time-consuming]
11. **Wrapper**: This script performs the R-T-vine estimation. It first returns the cross-sectional structure based on the empirical Kendall's τ and compares it with the D-vine. Then it fits the full model and compares the criteria of classical and

generalised M-vine with the R-T-vine model.

12. **FullOptimalTVine**: Using the structure obtained in `optimalCS`, this script performs the full MLE (again using the functions in `tvine`-package) for all possible connecting edges and returns the R-T-vine with the lowest BIC. Further it compares the criteria with the classical M-vine based on the same cross-sectional structure.
13. **CRPS**: The script compares the five models using the CRPS as discussed in Section 9.3.
14. **VAR**: Fitting a VAR(2) model to our copula dataset using only Gaussian copulas to restrict to a multivariate normal distribution.
15. **QuantilePlot**: The script conditionally simulates the monthly Dax log-returns for 2018, calculates the quantiles and plots them with the actual log-returns in Figure 9.9.

List of Figures

2.1	Time series of adjusted monthly prices for five indices from beginning of 2008 until June 2018.	5
2.2	Time series of (a) non-stationary DAX closing prices in percentage of the starting value and (b) stationary Gaussian process.	7
2.3	Monthly log-returns of the five indices shown Example 2.3 from beginning of 2008 up to June 2018.	8
3.1	Cyclic and acyclic graphs on $\{1, 2, 3\}$	20
3.2	Trees of the five dimensional D- (left) and C-vine (right).	21
3.3	Two highlighted choices out of all possible connections in the second tree of regular vine with starting tree equal to the C-vine.	22
4.1	Example for first tree of RVMMVTS: M-vine/COPAR time series approach with sectional dimension two and N time steps.	31
4.2	Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series.	32
4.3	First five trees of M-vine time series approach for three time steps and five sectional dimensions.	36
4.4	First three trees of long D-vine time series approach for three time steps and five sectional dimensions.	38
4.5	Copar time series approach for three time steps on five dimensions.	41
4.6	First five trees of generalised M-vine time series approach for cross-sectional C-vine structure, three time steps and five sectional dimensions.	44

4.7	Sixth tree of classical and generalised M-vine in three time steps and five sectional dimensions. Because of readability we dropped the labelling of the edges.	45
4.8	First four trees of i) generalised M-vine and ii) COPAR approach for cross-sectional four dimensional C-vine.	47
4.9	Comparison of the trees of i) M-vine and ii) COPAR approach for sectional dimensional two and $N = 3$ time steps.	50
4.10	First tree of five dimensional regular vine.	51
4.11	First six trees of the generalised M-vine time series approach for a five dimensional cross-sectional regular vine structure and three time steps. . .	53
6.1	First tree of M-vine (left) and first tree of long D-vine (right) time series approach for three time steps and five sectional dimensions.	68
6.2	Comparison of classical M-vine (4.1) denoted by (\star) and long D-vine (4.2) matrix denoted by ($\star\star$) in the 5×3 -dimensional case.	68
6.3	First trees of M-vine (left) and R-T-Vine with out-vertices C_{t-1} (right) time series approach for a five dimensional cross-sectional D-vine and three time steps.	69
6.4	First trees of "generalised M-vine" time series approach for out-vertices A_{t-1} (top left), C_{t-1} (top right) and E_{t-1} (below) for a five dimensional cross-sectional C-vine and three time steps.	72
6.5	Comparison of "generalised M-Vine" matrix with out-vertex A_{t-1} (above) and C_{t-1} (below) in the 5×3 -dimensional case.	73
6.6	Comparison of "generalised M-Vine" matrix with out-vertex A_{t-1} (above) and E_{t-1} (below) in the 5×3 -dimensional case.	73
6.7	First six trees of the "generalised M-vine" time series approach with out-vertices E_{t-1} for a five dimensional cross-sectional regular vine structure and three time steps.	76
6.8	First six trees of the "generalised M-vine" time series approach for a five dimensional cross-sectional regular vine structure with in-vertex C and three time steps.	80
6.9	The trees of the non C- nor D-vine example.	80

6.10	Sixth tree of the "generalised M-vine" time series approach for a five dimensional cross-sectional regular vine structure and three time steps with connecting edges $(C_{t-1}C_t)$	83
7.1	Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series with stationarity.	92
7.2	Highlighted structure types within the sub-diagonal of the matrix representation of a multivariate time series with stationarity and autoregressive order $p = 2$	94
9.1	Plots of the autocorrelation functions for the log-returns of the considered data set of five major stock indices.	101
9.2	Histogramm of log-returns with fitted skewed t distributions for the five selected indices.	102
9.3	Pair-plots of transformed monthly log-returns of the five selected indices.	103
9.4	Fitted regular vine to our transformed data set.	105
9.5	Fitted regular vine to our transformed data set using the maximisation of the BIC.	107
9.6	Contour plot of the (conditional) bivariate copulas obtained by <i>tvincop()</i> for our transformed data set.	110
9.7	Contour plot of the (conditional) bivariate copulas obtained by the R-T-vine using the BIC for optimisation for our transformed data set.	114
9.8	Empirical cumulative distribution functions of monthly log-returns for July 2018, based on the optimal R-T-vine (9.6).	116
9.9	95% confidence bounds for in- and out-of-sample monthly log-returns of the Dax index for 2018.	118
10.1	First tree of the alternative connection of the cross-sectional structures via a C-vine approach.	120
A.1	First tree of the R-T-vine fitted to the daily log-returns of 8 selected stock indices [using the plot function in <i>rvinecopulib</i>].	122

Bibliography

- Aas, Kjersti [2016]. “Pair-Copula Constructions for Financial Applications: A Review”. In: *Econometrics* 4[4], pp. 1–15.
- Aas, Kjersti et al. [2009]. “Pair-copula constructions of multiple dependence”. In: *Insurance Mathematics and Economics* 44[2], pp. 182–198.
- Azzalini, Adelchi and Antonella Capitanio [2003]. “Distributions Generated by Perturbation of Symmetry with Emphasis on a Multivariate Skew t-Distribution”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 65[2], pp. 367–389.
- Beare, Brendan K. and Juwon Seo [2015]. “Vine Copula Specifications for Stationary Multivariate Markov Chains”. In: *Journal of Time Series Analysis* 36[2], pp. 923–960.
- Bedford, Tim and Roger Cooke [2001]. “Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines”. In: *Annals of Mathematics and Artificial Intelligence* 32, pp. 245–268.
- Bollerslev, Tim [1986]. “Generalized autoregressive conditional heteroskedasticity”. In: *Journal of Econometrics* 31[3], pp. 307–327.
- Brechmann, E.C. and C. Czado [2015]. “COPAR - Multivariate time series modeling using the copula autoregressive model”. In: *Applied Stochastic Models in Business and Industry* 31[4], pp. 495–514.
- Brockwell Peter J. Richard A. Davis, et al. [2016]. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer International Publishing.
- Cooke, Roger and Dorota Kurowicka [2006]. *Uncertainty Analysis With High Dimensional Dependence Modelling*.
- Dissmann, Jeffrey et al. [2012]. “Selecting and Estimating Regular Vine Copulae and Application to Financial Returns”. In: *Computational Statistics & Data Analysis* 59, pp. 52–69.
- Embrechts, Paul, Filip Lindskog, and Er J. Mcneil [2001]. “Modelling Dependence With Copulas and Applications to Risk Management”. In: *Handbook of Heavy Tailed Distributions in Finance* 8.
- Engle, Robert [1982]. “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation”. In: *Econometrica* 50[4], pp. 987–1007.

- Fama, Eugene F. [1965]. “The Behavior of Stock-Market Prices”. In: *The Journal of Business* 38[1], pp. 34–105.
- Gneiting, Tilmann and Adrian E Raftery [2007]. “Strictly Proper Scoring Rules, Prediction, and Estimation”. In: *Journal of the American Statistical Association* 102[477], pp. 359–378.
- Heuke, Jakob [2016]. “Copula Modelling of Dependence in Multivariate Time Series”. Master thesis. TU Munich.
- Ivanov, Ievgen [2016]. “Copula Based Factor Models for Multivariate Asset Returns”. Master thesis. TU Munich.
- Joe, Harry [1996]. “Families of m-Variate Distributions with Given Margins and $m(m-1)/2$ Bivariate Dependence Parameters”. In: *Lecture Notes-Monograph Series* 28, pp. 120–141.
- Jones, Sam [2009]. “The formula that felled wall street”. In: *Financial Times* [0/23/2009].
- Jordan, Alexander, Fabian Krueger, and Sebastian Lerch [2018]. “Evaluating Probabilistic Forecasts with scoringRules”. In: *Journal of Statistical Software*. forthcoming.
- Kendall, M. G. [1938]. “A New Measure of Rank Correlation”. In: *Biometrika* 30[1/2], pp. 81–93.
- Mandelbrot, Benot [1963]. “The Variation of Certain Speculative Prices”. In: *The Journal of Business* 36.
- Millar, Russell B. [2011]. *Maximum Likelihood Estimation and Inference*.
- Morales-N’apoles O., Cooke R.M. & Kurowicka D. [2010]. “About the number of vines and regular vines on n nodes”. In:
- Nagler, Thomas and Thibault Vatter [2018]. *rvinecopulib: High Performance Algorithms for Vine Copula Modeling*. R package version 0.3.0.1.1.9000.
- Nelsen, Roger B [2006]. *An introduction to copulas*.
- Rosenblatt, Murray [1952]. “Remarks on a Multivariate Transformation”. In: *The Annals of Mathematical Statistics* 23[3], pp. 470–472.
- Schepsmeier, Ulf et al. [2018]. *VineCopula: Statistical Inference of Vine Copulas*. R package version 2.1.7. URL: <https://CRAN.R-project.org/package=VineCopula>.
- Shao, Jun [2007]. *Mathematical statistics*. 2. ed., corr. printing as of 4. printing.
- Sims, Christopher A. [1980]. “Macroeconomics and Reality”. In: *Econometrica* 48[1].
- Sklar, M. [1959]. *Fonctions de Repartition a N Dimensions et Leurs Marges*. Universite Paris 8, pp. 229–231.
- Smith, Michael Stanley [2015]. “Copula modelling of dependence in multivariate time series”. In: *International Journal of Forecasting* 31[3], pp. 815 –833.
- Teuma Mankeng, Stephanie [2017]. “Vine Copula Specifications for Stationary Multivariate Time Series”. Master thesis. TU Munich.

Wuertz, Diethelm et al. [2017]. *fGarch: Rmetrics - Autoregressive Conditional Heteroskedastic Modelling*. R package version 3042.83. URL: <https://CRAN.R-project.org/package=fGarch>.