Université catholique de Louvain

Faculty of Sciences

Institute of Statistics, Biostatistics and Actuarial Sciences

# High-dimensional dependence modeling using copulas

*Thesis submitted in partial fulfillment of the requirement for the degree of Docteur en Sciences by* ***Nathan Uyttendaele****.*

*Thesis Committee:*

***Johan Segers****, supervisor, Université catholique de Louvain*
*Christian Ritter, UCL*
*Philippe Lambert, Université de Liège*
*Ingrid Van Keilegom, chairwoman, UCL*
*Roel Braekers, Universiteit Hasselt*

November 25, 2016

*A thesis is not just about research and science. Behind a thesis, you will find sweat, joy, tears and hope. For me, this thesis was certainly all of that. But in its beginning, it was much more. Very few people knew it at the time, not even my supervisor, but this thesis was then, for many months, my main reason to wake up in the morning, giving me a purpose and offering me some stability after a very difficult series of events in my personal life.*

*In many ways, my supervisor, Johan, was the best supervisor I could have hoped for. Johan let me decide right from the start what this thesis would be about, pushing me and making suggestions when needed, always there to help, never letting me stuck too long, yet never imposing anything. I believe that no other supervisor would have left me with so much freedom and I'm grateful to Johan for that.*

*Johan is of course not the only person I wish to thank. The rest of my thesis committee — Christian, Ingrid, Philippe, Roel — deserve some credit for were I am now. I also wish to thank Gildas for his decisive contribution during the second part of my thesis, Lieven, Joris, Valérie and Céline for helping improve my Dutch during the last two years, Aleksandar, Anna, Anne, Aurélie, Benjamin, Cédric, Fabian, Florian, François, Germain, Hélène, Maïlis, Marco, Michal, Mickaël, Nathalie, Nicolas, Vincent, Samuel, Sylvie, the people at SMCS — Alain in particular —, The administrative staff here at ISBA — Nancy and Sophie in particular —, and many other persons that I hope will forgive me for not putting their name in here.*

*I also wish to thank my family and friends for their support during all this time. Dominique first of all, for is infinite patience with me and for always putting up with what I throw at him, Daniela, Julien, Kristel, Quentin, Sophie, and, last but not least, my brother and sisters.*

*Louvain-la-Neuve, November 2016.*

# Contents

# Introduction

In 2008, a significant event took place. The world economy faced its most dangerous financial crisis since 1930. This crisis has had many casualties and destroyed several million of jobs in the United States alone. Who is to blame? How did it all happen? We have now many answers to that question, but no absolute certainty. Experts around the world are still, at the moment, debating on what exactly went wrong.

In this context, the story of Gaussian copula models in finance is an interesting one. Felix Salmon, financial journalist, wrote, at the end of 2009:

> *For five years, David X. Li's formula, known as a Gaussian copula function, looked like an unambiguously positive breakthrough, a piece of financial technology that allowed hugely complex risks to be modeled with more ease and accuracy than ever before.*
>
> *Then the model fell apart. Cracks started appearing early on, when financial markets began behaving in ways that users of Li's formula hadn't expected. The cracks became full-fledged canyons in 2008.*
>
> *Today, as dazed bankers, politicians, regulators, and investors survey the wreckage of the biggest financial meltdown since the Great Depression, Li is probably thankful he still has a job in finance at all.*

The Gaussian copula did not cause the 2008 crisis. But the fact it had, by then, become so ubiquitous in finance only aggravated the collapse further.

High-dimensional dependence modeling cannot rely on Gaussian copulas only. Non-normal and more flexible copulas must also be used, something discussed by Embrechts (2009) after the crisis, but also by Patton (2006) or Embrechts et al. (2005) before.

But what are copulas? They have been introduced by Sklar (1959) more than half a century ago and represent a significant breakthrough in the study of dependencies between random variables and thus in the construction of multivariate distributions.

Let $(X_1, \ldots, X_d)$ be a vector of continuous random variables. The genius of Sklar was to realize that the joint multivariate distribution of this vector can be splitted in a set of $d$ marginal univariate distributions $\{F_1(x_1), \ldots, F_d(x_d)\}$ and a multivariate distribution $C(u_1, \ldots, u_d) : \mathbb{I}^d \to \mathbb{I}$, the latter being the joint distribution of the random vector $(U_1 = F_1(X_1), \ldots, U_d = F_d(X_d))$. This multivariate distribution $C$ is called a copula, and the joint distribution $J$ of the vector $(X_1, \ldots, X_d)$ can be built easily by injecting the univariate margins in the copula $C$:

$$J(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)).$$

By studying the copula $C$ rather than the joint distribution $J$, one can truly study the relationships between the random variables $(X_1, \ldots, X_d)$, free of any concern for the univariate margins, which, by definition, have nothing to do with the way the random variables interact with one another.

Thanks to Sklar, it is known that a function $C : \mathbb{I}^d \to \mathbb{I}$ qualifies as a copula as long as three properties are met:

(1) $C$ must have uniform univariate margins $\Leftrightarrow$ if all arguments we input in $C$ are equal to 1 except for one argument, the $C$ function will output this argument.

Example with $d = 6$:

$$C(1, 1, 1, u_4, 1, 1) = u_4.$$

It is easy to visualize this requirement by looking at an arbitrary bivariate copula, such as the one displayed in Figure 1. Any bivariate copula *must* exhibit the two red lines displayed in this figure.

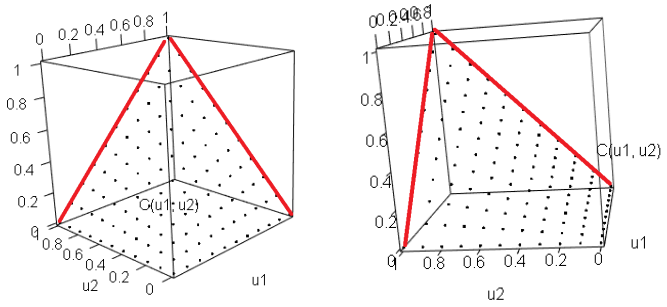



Figure 1: Two views of an arbitrary bivariate copula (the independence copula). The red lines highlight uniform margins.

(2) $C$ must also be grounded $\Leftrightarrow$ if any of the arguments we input in $C$ is 0, the function $C$ returns 0 as well. Examples:

$$C(u_1, u_2, 0, u_4, u_5, u_6) = 0,$$
$$C(u_1, 0, u_3, u_4, 0, u_6) = 0.$$

Again, it is easy to visualize this requirement on an arbitrary bivariate copula, such as the one in Figure 2. Any bivariate copula *must* exhibit the two green lines displayed in this figure.

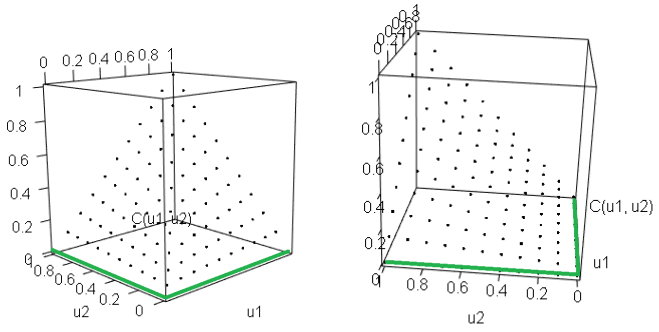



Figure 2: Two views of an arbitrary bivariate copula. The green lines hightlight grounding.

(3) $C$ must be $d$-increasing.

This last property is a more complicated one to introduce and requires first to define the notion of *C-volume* (Nelsen, 2006). Let $B$ be a hypercube in $[0,1]^d$, with $a = (a_1, \ldots, a_d)$ being a vertex of $B$ such that for any vertex $b_i = (b_{i1}, \ldots, b_{id})$ of $B$, $a_k \leq b_{ik}$ for all $k$. The C-volume of $B$ is defined as

$$V_C(B) = \sum_i \text{sgn}(b_i) C(b_i),$$

where the sum is taken over all vertices $b_i$ of $B$, and where

$$\text{sgn}(b_i) = 1$$

if $b_{ik} = a_k$ for an even number of $k$'s, and

$$\text{sgn}(b_i) = -1$$

if $b_{ik} = a_k$ for an odd number of $k$'s.

**Definition.** $C$ is $d$-increasing if $V_C(B) \geq 0$ for all $B \in [0,1]^d$.

Note that if a density $c(u_1, \ldots, u_d)$ exists, $d$-increasingness is the same as requesting that

$$c(u_1, \ldots, u_d) \geq 0$$

for any vector $u_1, \ldots, u_d$. The density $c$, if it exists, cannot be negative at any point in the domain of $C$.

In summary, the three properties any $d$-dimensional copula must meet are: (1) univariate margins, (2) being grounded and (3) being $d$-increasing.

Unfortunately, while this framework allows one to check if a given function $Y : \mathbb{I}^d \to \mathbb{I}$ could serve as a copula, the problem of finding "actual" copulas remains. Sklar and many other researchers where quick to come up with a large panel of bivariate copulas, a famous family of bivariate copulas being for instance the Archimedean family (Nelsen, 2006, pp. 109–155), which originally appeared not in statistics, but rather in the study of probabilistic metric spaces, where Archimedean copulas were studied as part of the development of a probabilistic version of the triangle inequality. For an account of this history, see Schweizer (1991).

But while the development of bivariate copulas toke off during the last decades, satisfying $d$-dimensional copulas, when $d > 2$, able to accurately model the relationships between a large number of random variables, are still lacking, with the possible exception of vine copulas, see for instance Bedford and Cooke (2001), Czado (2010), Czado et al. (2012), Brechmann et al. (2012) or Brechmann and Schepsmeier (2013). This is reflected in the literature across many areas of science, where applications involving copulas limit themselves to bivariate problems or, if they do not, often use the Gaussian copula or the t-copula (both are, for instance, formally defined by Kurowicka and Joe, 2011).

**The goal of this thesis** is to improve the understanding of $d$-dimensional random phenomena, more precisely to push the development of $d$-dimensional copulas further.

The thesis is written in three chapters. The first one is about a particular kind of graphs, called phylogenetic trees, and mainly deals with how a set of "small" trees can be used to build a larger tree containing them all. This first chapter is the only chapter of the thesis not related to copulas.

Chapter 2 is devoted to nested Archimedean copulas (NACs) and is built on top of Chapter 1. NACs are here formally defined, and their estimation, facilitated by the findings of Chapter 1, is discussed. Chapter 2, as well as Chapter 1, is largely based on papers from Segers and Uyttendaele (2014) and Uyttendaele (2016).

Finally, Chapter 3, which is mainly based on Mazo and Uyttendaele, deals with factor copulas, which are introduced and extended, in an effort to increase their level of flexibility. A general conclusion related to all three chapters follows, as well as an appendix chapter showcasing some R code allowing to work with the extended factor copulas from Chapter 3.

# Chapter 1

# On phylogenetic Trees

## 1.1 Introduction

The origin of graph theory started with the Swiss mathematician Leonhard Euler (1707-1783) and the method he used to solve the Königsberg Bridge problem, see Biggs et al. (1976). Graphs are mathematical structures used to model pairwise relations between objects and are nowadays successfully used to model many types of relations and processes in physical, biological, social and information systems. Many introductory books on graphs have been written during the last decades, see for instance Wilson (1970), Biggs (1993), Itô (1993), West et al. (2001), Chartrand (2006) or Chartrand et al. (2010). Hereafter is a brief summary of what graphs and phylogenetic trees, which are a particular kind of graphs (and the main subject of this chapter), are.

Formally, a graph $G$ is a set of vertices (or nodes) and a set of edges, and can be written

$$G = (V, E),$$

where $V = \{v_1, \ldots, v_h\}$ is the set of $h$ vertices and $E = \{e_1, \ldots, e_f\}$ is the set of $f$ edges, this last set being made of pairs of vertices from $V$ indicating what are the linked or connected vertices.

If one of the pairs from the set $\{(v_i, v_i)\}, i \in \{1, \ldots, h\}$, is allowed to be part of $E$, then the graph is said to contain a loop, that is, an edge that connects a vertex to itself.

A common way to represent a graph is through an *adjacency matrix* $A = \{a_{ij}\}$. An adjacency matrix is a squared matrix such that the number of rows/columns is equal to the number of nodes $h$ in the graph. The entries $\{a_{ij}\}$ of the matrix simply indicate if node $v_i$ is linked to node $v_j$. If yes, $a_{ij} = 1$. Otherwise, $a_{ij} = 0$. Loops in a graph are easy to spot by looking at the main diagonal of the matrix.

As an example, Table 1.1 displays the adjacency matrix for the loop-free graph in Figure 1.1.

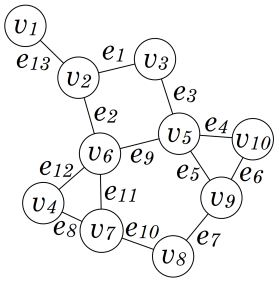



Figure 1.1: An undirected connected graph with 10 nodes and 13 edges.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $v_9$ | $v_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $v_1$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $v_2$ | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| $v_3$ | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_4$ | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| $v_5$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| $v_6$ | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| $v_7$ | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| $v_8$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| $v_9$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $v_{10}$ | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |

Table 1.1: The adjacency matrix corresponding to the graph in Figure 1.1.

Modern texts in graph theory usually define several types of graphs:

**An undirected graph** is a graph where edges have no orientation. That is, edges $(v_i, v_j)$ and $(v_j, v_i)$ are considered identical. The maximum number of edges in an undirected graph is $h(h-1)/2$. The adjacency matrix of an undirected graph is always a symmetric matrix.

**Directed graphs or digraphs** differ from an ordinary or undirected graph, in that the latter is defined in terms of unordered pairs of vertices, while the former is defined in terms of ordered pairs of vertices. The edges in a digraph are called *directed edges* or *arrows*. Entry $a_{ij}$ in the adjacency matrix indicates if there is a directed edge *from* node $v_i$ *towards* node $v_j$.

A **mixed graph** is a graph in which some edges may be directed and some may be undirected. Their usage is uncommon.

An **undirected graph** is said **connected** if, for every pair of vertices $(v_i, v_j)$ from this graph, there is a sequence of edges (called a path) that allows to go from one node to the other. A graph with just one vertex is considered connected. An edgeless graph ($E = \emptyset$) with two or more vertices is disconnected. An example of undirected connected graph is given in Figure 1.1.

Likewise, a **directed graph** is said **connected** if, after replacing all directed edges by undirected edges, there is a sequence of edges allowing to go from $v_i$ to $v_j$ for any pair of vertices $(v_i, v_j)$.

An **acyclic graph**, assumed undirected, is a graph having no graph cycles, where a graph cycle is a sequence of edges (the minimum size of this sequence being three), allowing to go from a starting node to itself with distinct nodes on the way. A graph containing no cycles of length three is called a triangle-free graph, and a graph containing no cycles of length four is called a square-free graph.

**Undirected trees** are undirected graphs that satisfy the following equivalent conditions:

- The graph is connected and has no cycles.
- The graph has no cycles and one is formed if any edge is added to the graph.
- The graph is connected, but is not connected if any single edge is removed from $E$.
- The graph is connected and has $h - 1$ edges.
- The graph has no cycles and has $h - 1$ edges.

**Directed trees** are directed graphs which would be undirected trees if the directions on the edges were ignored.

A **rooted tree** is a directed tree with a root, a special node such that all edges are directed away from that node. Any node in a rooted tree but the root has always one and only one parent and the root is the common ancestor to all other nodes.

An **unrooted tree** is defined in this thesis as an undirected tree for convenience. In an unrooted tree, it is not possible to identify the parent of an arbitrary node, as a neighbouring node could be about anything (child or parent), since the edges are undirected.

a **phylogenetic tree** is a rooted tree where all nodes but the end nodes have at least two children. The end nodes of such a tree are called *leaves*. These special rooted trees will be referred to as *NAC trees* or as *NAC structures* in Chapter 2.

Definition 1.1.1. *Let $D_0$ be a nonempty, finite set with $|D_0| = d$ elements. For concreteness, let $D_0 = \{U_1, \ldots, U_d\}$. Formally, a phylogenetic tree $\lambda$ on $D_0$ is a collection of nonempty subsets of $D_0$ such that*

*(i) $D_0 \in \lambda$;*

*(ii) $\{U_j\} \in \lambda$ for every $U_j \in D_0$;*

*(iii) if $A, B \in \lambda$, then either $A \subset B$, $B \subset A$, or $A \cap B = \varnothing$.*

*The elements of $\lambda$ are the nodes of the graph. The element $D_0$ of $\lambda$ is the root and the singleton elements $\{U_j\}$ of $\lambda$ are the leaves. The nodes of $\lambda$ that are not leaves are called* branching nodes *or* internal nodes. *If $A, B \in \lambda$ are such*

*that $A \subset B$, $A \neq B$, and there is no $C \in \lambda$ such that $A \subset C \subset B$ and $C \neq A$ and $C \neq B$, then $A$ is called a* child *of $B$ and conversely $B$ is called the* parent *of $A$. The set of children of $B$ in $\lambda$ is denoted $\mathcal{C}(B, \lambda)$.*

**Example**. Let $\lambda$ be the set

$$\big\{\{U_1, U_2, U_3\}, \{U_2, U_3\}, \{U_1\}, \{U_2\}, \{U_3\}\big\}.$$

The graphical representation of this phylogenetic tree is shown in Figure 1.2, where $D_{123}$ is a convenient label for the subset $\{U_1, U_2, U_3\}$ and $D_{23}$ for the subset $\{U_2, U_3\}$. Later in this thesis, sets such as $\{U_1, U_2, U_3\}$ will be referred to as 123 for even more convenience.

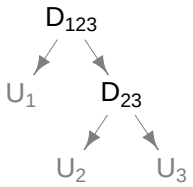



Figure 1.2: A phylogenetic tree. To ease the notation, the singletons $\{U_1\}$, $\{U_2\}$ and $\{U_3\}$ are denoted $U_1, U_2$ and $U_3$.

In this chapter, a novel way of storing the topographical information about such tree through a *lowest common ancestor matrix* (Section 1.3) and a novel way of calculating a distance between two phylogenetic trees (Section 1.6) are, among other things, given. However, the main contribution of this chapter is the developement of a new way of using "small" phylogenetic trees in order to build a larger phylogenetic tree containing them all, see Section 1.5. This chapter is largely based on Segers and Uyttendaele (2014).

## 1.2 Getting a marginal phylogenetic tree

Let $\lambda$ be a phylogenetic tree on $D_0$ and let $A$ be a nonempty subset of $D_0$. The set $A$ need not be a node of $\lambda$. The phylogenetic tree $\lambda$ induces a phylogenetic tree on $A$ by the following operation:

$$\lambda \sqcap A = \{A \cap B : B \in \lambda\} \setminus \{\varnothing\}.$$

That is, $\lambda \sqcap A$ is obtained by intersecting every node $B$ of $\lambda$ with $A$. Some of these intersections will be empty, and they are removed. Different nodes $B_1$ and $B_2$ of $\lambda$ may have identical intersections $B_1 \cap A$ and $B_2 \cap A$ with $A$; since $\lambda \sqcap A$ is the collection of all intersections, identical intersections are counted only once.

It is easy to verify that this construction produces a phylogenetic tree on $A$: verification of (i), (ii), and (iii) in Definition 1.1.1 is immediate.

## 1.3 Lowest Common Ancestor

Let $T$ be a subset of $D_0$ containing at least two elements, that is $|T| \geq 2$. The set $T$ does not need to be a node of $\lambda$. The *lowest common ancestor* (lca) in $\lambda$ of the elements of $T$ is given by the intersection of all the nodes $B$ in $\lambda$ that contain $T$, that is,

$$\mathrm{lca}(T, \lambda) = \bigcap_{B \in \lambda : T \subseteq B} B, \tag{1.3.1}$$

and it provides the lowest branching node (read: farthest from the root) through which the elements of $T$ are linked up in $\lambda$. As an example, looking at the tree in Figure 1.2, one can see that the lowest common ancestor between $U_2$ and $U_3$ is $D_{23}$, while

$$\mathrm{lca}(\{U_1, U_2\}, \lambda) = D_{123}$$

and

$$\mathrm{lca}(\{U_1, U_3\}, \lambda) = D_{123}.$$

As an alternative to an adjacency matrix, a phylogenetic tree can also be represented through a *lca matrix*.

A lca matrix $L = \{l_{ij}\}$ is a square upper triangular matrix with a number of rows/columns equal to the number of leaves in the tree. The matrix entries $\{l_{ij}\}, i < j$, indicate the furthest node from the root through which node $U_i$ and node $U_j$ are related, that is, their lowest common ancestor. The entries $\{l_{ij}\}, i = j$, weigh no particular meaning and are therefore conveniently used to store the labels of the leaves. Likewise, entries where $i > j$ can be used to store extra labels for the corresponding lowest common ancestors in the upper part of the matrix if needed. As an example, Table 1.2 displays the lca matrix for the phylogenetic tree shown in Figure 1.2.

| | | |
|---|---|---|
| $U_1$ | $D_{123}$ | $D_{123}$ |
| | $U_2$ | $D_{23}$ |
| | | $U_3$ |

Table 1.2: The lca matrix corresponding to the rooted tree in Figure 1.2.

Lowest common ancestor matrices offer several advantages compared with adjacency matrices. First, they are always more compact than adjacency matrices: a lca matrix is always a $d \times d$ matrix, while the corresponding adjacency matrix is of size $(d + I) \times (d + I)$, where $I \geq 1$ is the number of internal nodes

in the tree. Second, while the labels of the nodes, whether they are internal nodes or leaves, can easily be stored in the lca matrix itself, they must be stored apart with an adjacency matrix, which lacks convenience. But the most important advantage of lca matrices over adjacency matrices will become apparent in Chapter 2, where nested Archimedean copula are introduced.

Compared with adjacency matrices, lca matrices have however one practical disadvantage: it is hard to get an idea of what the tree behind a lca matrix looks like. Adjacency matrices are indeed much easier to read.

## 1.4 The 3-leaf case

Let $D_0 = \{U_1, U_2, U_3\}$. A phylogenetic tree built on $D_0$ when $D_0$ is made of three elements will be called a 3-leaf phylogenetic tree, and there are only four possible phylogenetic trees fulfilling Definition 1.1.1 in this case:

$$\begin{aligned}
\big\{\{U_1,U_2,U_3\},\{U_1\},\{U_2\},\{U_3\}\big\} &= \Lambda_{123},\\
\big\{\{U_1,U_2,U_3\},\{U_2,U_3\},\{U_1\},\{U_2\},\{U_3\}\big\} &= \lambda_{23},\\
\big\{\{U_1,U_2,U_3\},\{U_1,U_2\},\{U_1\},\{U_2\},\{U_3\}\big\} &= \lambda_{12},\\
\big\{\{U_1,U_2,U_3\},\{U_1,U_3\},\{U_1\},\{U_2\},\{U_3\}\big\} &= \lambda_{13}.
\end{aligned}$$

## 1.5 Recovering a phylogenetic tree from all its marginal 3-leaf phylogenetic trees

Let $\lambda$ be a phylogenetic tree on $D_0 = \{U_1, \ldots, U_d\}$, $d \geq 3$. Let $B$ be a branching node of $\lambda$. The set of all children of $B$ forms a partition of $B$, that is, taking the union of all children of a branching node $B$ allows to reconstruct that branching node. As a consequence, every branching node has at least two children.

Since the children of a branching node $B$ form a partition of $B$ and since each branching node has at least two children, it follows that each branching node can be reconstructed from the pairs of which it is the lowest common ancestor, that is, for every branching node $B$, we have

$$B = \bigcup \big\{\{U_i, U_j\} \subset D_0 : U_i \neq U_j,\ \mathrm{lca}(\{U_i, U_j\}, \lambda) = B\big\}. \tag{1.5.1}$$

The relation "... has the same lowest common ancestor as ..." is an equivalence relation on the set of pairs $\{U_i, U_j\}$ of $D_0$. This relation induces a partition of the set of pairs into equivalence classes: two pairs $\{U_i, U_j\}$ and $\{U_k, U_l\}$ belong to the same equivalence class if and only if they have the same lowest common ancestor in $\lambda$.

By Equation (1.5.1), the phylogenetic tree $\lambda$ can be reconstructed from the equivalence relation it induces on the set of pairs: every equivalence class

of pairs corresponds to a branching node, the branching node being given by the union of the pairs in that equivalence class. Put differently, the union of all pairs within an equivalence class yields the branching node that is the lowest common ancestor for each pair in that equivalence class. Hence, every phylogenetic tree $\lambda$ on $D_0$ can be represented as a partition on the set of pairs of $D_0$.

Let $d \geq 4$. Suppose that for any set $D_{ijk} = \{U_i, U_j, U_k\}$ with distinct $i, j, k \in \{1, \ldots, d\}$, the related marginal tree $\lambda \sqcap D_{ijk}$ is known. Define ${}^3(\lambda)$ as the set of these $\binom{d}{3}$ trees, that is, the set of all marginal 3-leaf phylogenetic trees one can built from $\lambda$.

In Proposition 1.5.1, it is shown that the phylogenetic tree $\lambda$ can be recovered from ${}^3(\lambda)$. Lemmas 1 and 2 contain some auxiliary results.

LEMMA 1. *Let $\lambda$ be a tree on $D_0$. For any nonempty subsets $T_1, T_2$ and $C$ of $D_0$ such that $(T_1 \cup T_2) \subset C$, we have*

$$\mathrm{lca}(T_1, \lambda) = \mathrm{lca}(T_2, \lambda) \iff \mathrm{lca}(T_1, \lambda \sqcap C) = \mathrm{lca}(T_2, \lambda \sqcap C).$$

*Proof.* The proof is built in two steps. First we need to prove that, for $\varnothing \neq A \subset C \subset D_0$, we have

$$\mathrm{lca}(A, \lambda \sqcap C) = \mathrm{lca}(A, \lambda) \cap C.$$

By definition, we have

$$\mathrm{lca}(A, \lambda) \cap C = \left( \bigcap_{B \in \lambda : A \subseteq B} B \right) \cap C = \bigcap_{B \in \lambda : A \subseteq B} (B \cap C).$$

Since $A$ is a subset of $C$ and since $A$ must be a subset of $B$, notice that requiring $A \subset B$ is equivalent to requiring $A \subset B \cap C$. Thus we can write

$$\mathrm{lca}(A, \lambda) \cap C = \bigcap_{B \in \lambda : A \subseteq B \cap C} (B \cap C).$$

On the other hand,

$$\mathrm{lca}(A, \lambda \sqcap C) = \bigcap_{B' \in \lambda \sqcap C : A \subseteq B'} B'.$$

Since $\lambda \sqcap C = \{B \cap C : B \in \lambda\} \setminus \{\varnothing\}$ by definition, we can rewrite the above expression as

$$\mathrm{lca}(A, \lambda \sqcap C) = \bigcap_{B \in \lambda : A \subseteq B \cap C, B \cap C \neq \varnothing} (B \cap C).$$

And because $A \subseteq B \cap C$ and $A \neq \varnothing$, the requirement $B \cap C \neq \varnothing$ can be dropped, thus

$$\mathrm{lca}(A, \lambda \sqcap C) = \bigcap_{B \in \lambda : A \subseteq B \cap C} (B \cap C) = \mathrm{lca}(A, \lambda) \cap C.$$

The second step of the proof of Lemma 1 begins by making use of the result from the first step. Indeed, we can now write:

$$\mathrm{lca}(T_j, \lambda \sqcap C) = \mathrm{lca}(T_j, \lambda) \cap C \text{ with } j = 1, 2.$$

Suppose to begin $\mathrm{lca}(T_1, \lambda) = \mathrm{lca}(T_2, \lambda)$. We therefore have

$$\begin{aligned} \mathrm{lca}(T_1, \lambda \sqcap C) &= \mathrm{lca}(T_1, \lambda) \cap C \\ &= \mathrm{lca}(T_2, \lambda) \cap C \\ &= \mathrm{lca}(T_2, \lambda \sqcap C). \end{aligned}$$

On the other hand, suppose that $\mathrm{lca}(T_1, \lambda \sqcap C) = \mathrm{lca}(T_2, \lambda \sqcap C)$. Obviously,

$$\mathrm{lca}(T_1, \lambda) \supset \mathrm{lca}(T_1, \lambda) \cap C,$$

and since $T_2$ is both a subset of $\mathrm{lca}(T_2, \lambda)$ and of $C$, we also have

$$\mathrm{lca}(T_2, \lambda) \cap C \supset T_2.$$

Because $\mathrm{lca}(T_1, \lambda \sqcap C) = \mathrm{lca}(T_2, \lambda \sqcap C)$ implies that $\mathrm{lca}(T_1, \lambda) \cap C = \mathrm{lca}(T_2, \lambda) \cap C$, we have

$$\mathrm{lca}(T_1, \lambda) \supset T_2,$$

which means that $\mathrm{lca}(T_1, \lambda)$ is an ancestor of $T_2$, but not necessarily the lowest. Therefore $\mathrm{lca}(T_1, \lambda) \supset \mathrm{lca}(T_2, \lambda)$. The converse inclusion holds as well, by symmetry of the argument. We conclude that the two sets $\mathrm{lca}(T_1, \lambda)$ and $\mathrm{lca}(T_2, \lambda)$ are in fact equal. □

Essentially this lemma states that if two subsets of $D_0$ have the same lowest common ancestor in $\lambda$, then they also have the same lowest common ancestor in any subtree of $\lambda$, provided the two subsets are included in the set of leaves of that subtree. For instance, consider the structure in Figure 1.3, where $D_0 = \{U_1, \ldots, U_7\}$. The set $\{U_5, U_6, U_7\}$ has the same lowest common ancestor as the set $\{U_5, U_7\}$ in $\lambda$, this lowest common ancestor being the node $D_{567}$. This holds true even if you consider only the subtree spanned on $\{U_4, U_5, U_6, U_7\}$, that is, even if you only consider the right branch of the structure in Figure 1.3.

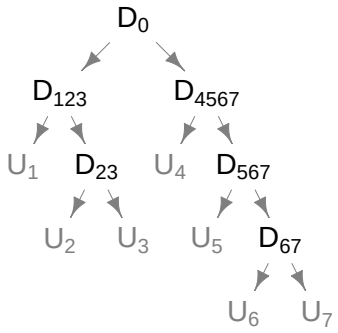



Figure 1.3: A phylogenetic tree that spans on 7 leaves, $D_0 = \{U_1, \ldots, U_7\}$.

LEMMA 2. *Let $\lambda$ be a tree on $D_0$ and let $A \in \lambda$. Let $B$ be a nonempty subset of $D_0$ with a least two elements. The lowest common ancestor of $B$ is equal to $A$ if and only if $B \subset A$ and there exist distinct children $B_1$ and $B_2$ of $A$ such that $B \cap B_1 \neq \varnothing$ and $B \cap B_2 \neq \varnothing$.*

*Proof.* Suppose first that $A$ is the lowest common ancestor of $B$. Clearly $B \subset A$. Let $B_1, \ldots, B_p$ be the children of $A$ and recall these children form a partition of $A$. Hence $B = B \cap A = \bigcup_{j=1}^{p} (B \cap B_j)$, and thus at least one of these intersections is not empty. However, if only one of these intersections would be nonempty, say $B \cap B_1$, then we would get $B = B \cap B_1$ and thus $B \subset B_1$, meaning that $B_1$ is also common ancestor of all elements of $B$. Since $B_1$ is a proper subset of $A$, this would be in contradiction with the assumption that $A$ is the lowest common ancestor of $B$. Therefore if $A$ is the sca of $B$, $B$ has a nonempty intersection with a least two children of $A$.

Conversely, suppose that $B \subset A$ and that there exist distinct children $B_1$ and $B_2$ of $A$ having nonempty intersections with $B$. Let $A'$ be a node in $\lambda$ such that $B \subset A'$. Then also $B \cap B_1 \subset A'$, and thus, as $B \cap B_1$ is nonempty, $A' \cap B_1$ is not empty. Similarly, $A' \cap B_2$ is not empty. Since $B_1$ and $B_2$ are disjoint, requirement (iii) in Definition 1.1.1 then forces $B_1$ and $B_2$ to be descendants of $A'$. As a consequence $A \subset A'$. We have obtained that $A$ is included in every node $A'$ containing $B$ as a subset. We conclude that $A$ is the lowest common ancestor of the elements of $B$, as required. □

The meaning of this second lemma is less straightforward. It states that if $B$ is a subset of $A$, $A$ being a node of $\lambda$, the only way $A$ is going to be the lowest common ancestor of $B$ is if $B$ has a nonempty intersection with two distinct children of $A$. Consider Figure 1.3 again. If $A = \{U_4, U_5, U_6, U_7\}$, then the lca of $B = \{U_4, U_5\}$ is $A$ because in that case, $B$ has a nonempty

intersection with the only two children of $A$, these two children being $\{U_4\}$ and $D_{567} = \{U_5, U_6, U_7\}$.

PROPOSITION 1.5.1. *The phylogenetic tree $\lambda$ can be recovered from the set ${}^3(\lambda)$, that is, it is possible to retrieve the partition of the set of pairs $\{U_i, U_j\}$ of $D_0$ into equivalence classes from the set ${}^3(\lambda)$.*

*Proof.* Let first $\{U_i, U_j\}$ and $\{U_i, U_k\}$ be two pairs with exactly one element, $U_i$, in common. To see whether they have the same lowest common ancestor in $\lambda$, it is sufficient to consider the tree induced by $\lambda$ on the set $\{U_i, U_j, U_k\}$: it is known from Lemma 1 that the pairs $\{U_i, U_j\}$ and $\{U_i, U_k\}$ have the same lowest common ancestor in $\lambda$ if and only if they have the same lowest common ancestor in $\lambda_{ijk}$.

On the other hand, if two pairs are disjoint, there exists no set with only three elements containing both pairs. Still, considering $\{\lambda_{ijk}\}$ with distinct $i, j, k \in \{1, \ldots, d\}$ turns out to be sufficient to verify the equivalence of the two disjoint pairs: the two pairs can only be equivalent if there is a third pair equivalent to both of them and having a nonempty intersection with each of them.

Indeed suppose first there exists a pair $\{U_i, U_j\}$ having the same lowest common ancestor as the pair $\{U_i, U_k\}$. Also suppose $\{U_i, U_k\}$ has the same lowest common ancestor as $\{U_k, U_l\}$. Then by transitivity $\{U_i, U_j\}$ has the same lowest common ancestor as $\{U_k, U_l\}$.

Conversely, suppose that $\{U_k, U_l\}$ and $\{U_i, U_j\}$ have the same lowest common ancestor, $A$. Recall Lemma 2. Let $B_i, B_j, B_k, B_l$ be the children of $A$, to which $U_i, U_j, U_k, U_l$ belong, respectively. Since the lca of $U_i$ and $U_j$ is $A$, the pair $\{U_i, U_j\}$ must have a non-empty intersection with two different children of $A$ (Lemma 2). Hence, $B_i$ and $B_j$ must be disjoint, $B_i \cap B_j = \varnothing$. Similarly $B_k \cap B_l = \varnothing$. Then $B_k$ and $B_l$ cannot both be equal to $B_i$.

- If $B_k$ is different from $B_i$, then $U_i$ and $U_k$ belong to two different children of $A$, and the lowest common ancestor of $\{U_i, U_k\}$ is $A$, too;
- If $B_l$ is different from $B_i$, then, similarly, the lowest common ancestor of $\{U_i, U_l\}$ is $A$, too.

In both cases, we have found a pair that is equivalent to $\{U_i, U_j\}$ and $\{U_k, U_l\}$ and that has a nonempty intersection with each of them. □

Given a phylogenetic tree $\lambda$, it is thus always possible to break it down into a set of 3-leaf structures, one 3-leaf structure for each combination of the elements of $D_0$, taken three at the time without repetition. Proposition 1.5.1 states that this set of 3-leaf trees is sufficient to recover $\lambda$.

The idea to decompose a structure into smaller pieces that uniquely determine the structure is however not new. Ng and Wormald (1996) show that a

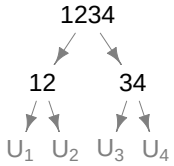



Figure 1.5: A phylogenetic tree that spans on 4 leaves with $D_0 = \{U_1, \ldots, U_4\}$.

given structure can be broken down into a set of triples and fans. A *triple* is a tree with three leaves and two internal vertices. A *fan* is a tree with only one internal vertex and at least three leaves (that is, a fan is a trivial tree with at least three leaves). A fan with $d$ leaves is called a $d$-fan.

Hereafter is a practical example on how to retrieve $\lambda$ from ${}^3(\lambda)$ when $d = 4$. Suppose indeed the $\binom{4}{3} = 4$ elements of ${}^3(\lambda)$ are as shown in Figure 1.4.

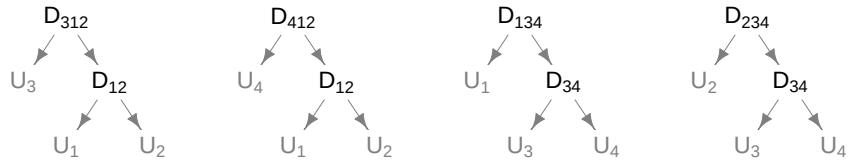



Figure 1.4: A set of 3-leaf structures that uniquely determines the 4-leaf structure in Figure 1.5. Note that the labels of the internal nodes are irrelevant. The lowest common ancestor of $U_1$ and $U_2$ could have been labeled $P$ instead of $D_{12}$ in the first structure and $Q$ in the second structure.

From Figure 1.4, we get that

- The lowest common ancestors of the pair $\{U_1, U_2\}$ are $\{D_{12}, D_{12}\}$;
- The lowest common ancestors of the pair $\{U_1, U_3\}$ are $\{D_{321}, D_{134}\}$;
- The lowest common ancestors of the pair $\{U_1, U_4\}$ are $\{D_{412}, D_{134}\}$;
- The lowest common ancestors of the pair $\{U_2, U_3\}$ are $\{D_{312}, D_{234}\}$;
- The lowest common ancestors of the pair $\{U_2, U_4\}$ are $\{D_{412}, D_{234}\}$;
- The lowest common ancestors of the pair $\{U_3, U_4\}$ are $\{D_{34}, D_{34}\}$.

It appears therefore that $\{U_1, U_3\}, \{U_1, U_4\}, \{U_2, U_3\}$ and $\{U_2, U_4\}$ belong to the same equivalence class, while $\{U_1, U_2\}$ is by itself, as well as $\{U_3, U_4\}$. The branching nodes of $\lambda$ in this case are therefore $\{U_1, U_2, U_3, U_4\}, \{U_1, U_2\}$ and $\{U_3, U_4\}$. The rooted tree structure $\lambda$ is thus as shown in Figure 1.5.

The general procedure for any $d \geq 4$ is as shown in Algorithm 1.

**Algorithm 1** How to retrieve $\lambda$ from ${}^3(\lambda)$

**for all** pairs $\{U_i, U_j\}$ such that $i < j$ **do**
Get from ${}^3(\lambda)$ the set of lowest common ancestors. That is, for each element of ${}^3(\lambda)$ where both $\{U_i, U_j\}$ are leaves, extract their common ancestor. There should be $d-2$ lowest common ancestors available for each pair;
**end for**
**for all** pairs $\{U_i, U_j\}$ such that $i < j$ **do**
Intersect the set of lowest common ancestors of the working pair with the other sets (one set for each other pair of variables). Any nonempty intersection means the two pairs are related, that is, belong to the same equivalence class. This also allows to determine the number of equivalence classes;
**end for**
**for all** equivalence classes **do**
Take the union of all pairs within each equivalence class to get the branching nodes of the structure. There are as many branching nodes as there are equivalence classes;
**end for**
Add the leaves to the branching nodes to get $\lambda$.

## 1.6 Distance between two phylogenetic trees

Given two phylogenetic trees $\lambda_1$ and $\lambda_2$, how to get a measure of similarity or dissimilarity between the two? Hereafter we consider two distances.

The first one is a 01-distance: if the adjacency or lca matrix of $\lambda_1$ is actually equal to the adjacency or lca matrix of $\lambda_2$, then the distance is 0. Otherwise, the distance is 1.

The second distance, called the tri-distance, is based on the comparison of the 3-leaf pieces of $\lambda_1$ and the 3-leaf pieces of $\lambda_2$, that is, based on the comparison of ${}^3(\lambda_1)$ and ${}^3(\lambda_2)$. If both trees are equal, all the 3-leaf pieces will be equal as well, and the distance is 0. If, among the $\binom{d}{3}$ trivariate pieces from $\lambda_1$ and the $\binom{d}{3}$ trivariate pieces from $\lambda_2$, $k$ pieces differ, then the distance is $k$. The maximum possible tri-distance is therefore $\binom{d}{3}$. Unlike the 01-distance, this last distance allows to assess how far from one another two phylogenetic trees are.

## 1.7 Supertree methods

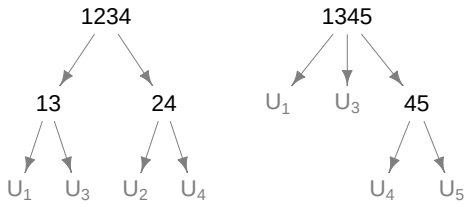
Supertree methods have been extensively studied in the field of phylogenetics. They are designed to output a tree (called the supertree) such that this tree will be the best match of an input set of smaller trees, this input set including

trees of various sizes, conflicting trees and also missing trees (that is, some information to build the supertree is actually lacking). Some interesting references to get started are Bininda-Emonds (2004), Wilkinson et al. (2005) or Swenson et al. (2012).

Many supertree methods take as input a set of unrooted trees and output an unrooted tree, see for instance the R package phytools (Revell, 2012), where two such methods are available. Hereafter is described how such methods can be used for the purpose of building a phylogenetic tree $\lambda$ based on an input set of marginal phylogenetic trees from $\lambda$.

phylogenetic trees are by definition a subset of rooted graphs. To start, all input phylogenetic trees must be unrooted. If the root of an input phylogenetic tree has two and only two children, the root is removed and the two children become directly connected. Otherwise, the root remains in the graph. In both cases, the edges become undirected.

Once the input trees are ready, all the topological information available across them is gathered as a matrix, called the character matrix. Let $\{e_1, \ldots, e_r\}$ be the set of $r$ internal edges (an internal edge is an edge between two internal nodes) across all trees in the set of unrooted input trees. Let $D_0 = \{U_1, \ldots, U_d\}$ be the set of leaves such that each of these leaves exists in at least one of the input trees. Elements of $D_0$ that are on one side of edge $e_i$, $i = \{1, \ldots, r\}$, receive the state 0, elements that are on the other side receive the state 1. Elements of $D_0$ that are on neither of the two sides of $e_i$ receive the unknown state for that edge. The $d \times r$ character matrix contains all of the states. A supertree is by definition a tree that spans on all of the elements of $D_0$. It has no particular properties or features a regular tree does not have: it is just a tree that has to span on all the elements of $D_0$. The goal of a supertree method is to find a supertree such that some loss function, this loss function comparing the topological information of the supertree to the $d \times r$ character matrix, is optimized.




| | | |
|---|---|---|
| $U_1$ | 0 | 0 |
| $U_2$ | 1 | ? |
| $U_3$ | 0 | 0 |
| $U_4$ | 1 | 1 |
| $U_5$ | ? | 1 |

| | | | |
|---|---|---|---|
| $U_1$ | 0 | 1 | 0 |
| $U_2$ | 1 | 0 | ? |
| $U_3$ | 0 | 1 | 0 |
| $U_4$ | 1 | 0 | 1 |
| $U_5$ | ? | ? | 1 |
| $O$ | 0 | 0 | 0 |

Figure 1.6: Two rooted input phylogenetic trees and the related character matrix built on the unrooted version of these two trees first without and second with an outgroup added. 1234 refer to the set $D_{1234} = \{U_1, \ldots, U_4\}$. Likewise, $12 = D_{12} = \{U_1, U_2\}$, $1345 = D_{1345} = \{U_1, U_3, U_4, U_5\}$, etc.

As an example, consider the two tree structures in Figure 1.6. Suppose one

wants to build a tree spanned on $D_0 = \{U_1, \ldots, U_5\}$ based on these two trees. Both input trees are first unrooted. The first rooted tree in Figure 1.6 loses one internal edge in the process and becomes a tree with structure of the form $>-<$, as shown on the top left side of Figure 1.7, where the only remaining internal edge (the $-$ in $>-<$) has nodes 13 and 24 at its tips.

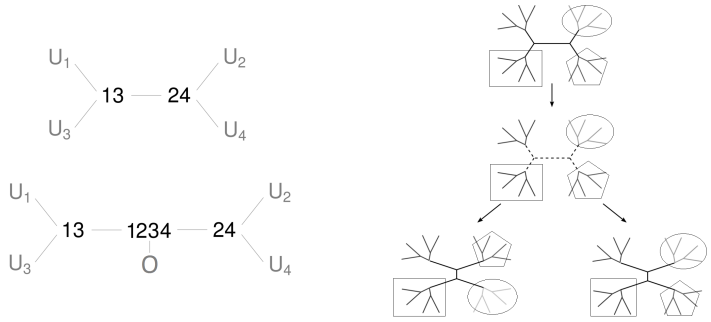



Figure 1.7: Top left: the unrooted version of the first rooted tree displayed in Figure 1.6. Bottom left: an outgroup $O$ has been attached to the root of the first rooted tree displayed in Figure 1.6 before the tree is unrooted. Thanks to this outgroup, we are able to keep track of the root, even though the tree has been unrooted. Right: an example of nearest neighbour interchange (NNI) from Wikipedia.

The second rooted tree in Figure 1.6, when unrooted, also becomes a tree with structure of the form $>-<$, where the internal edge has nodes 1345 and 45 at its tips. The character matrix that gathers all the topological information available based on these two unrooted trees is the $5 \times 2$ matrix displayed in Figure 1.6. Looking at the internal edge in the first unrooted tree (top left of Figure 1.7), we see that $U_1$ and $U_3$ are on one side of that edge and $U_2$ and $U_4$ are on the other side. We assign the label (or state) 0 to $U_1$ and $U_3$, and the label 1 to $U_4$ and $U_2$. The leaf $U_5$ does not appear in this input unrooted tree and it is therefore not known to which side this leaf belongs. We therefore assign an unknown state to $U_5$, thus ending the first column of the $5 \times 2$ character matrix. For the second unrooted tree, $U_1$ and $U_3$ are on one side of the internal edge and $U_4$ and $U_5$ are on the other side. We do not know to which side $U_2$ belongs. This makes up the second column of the $5 \times 2$ character matrix.

Once the character matrix has been built, the next step is to find an unrooted supertree that will agree as much as possible with the topological information available in the matrix. The strategy is to pick a starting supertree and then to apply topological rearrangements to that supertree in a recursive fashion. Rearrangements leading to a supertree closer to the character matrix are kept, otherwise they are not kept. The final supertree is one such that no further rearrangements of the supertree allows to come closer to the character matrix.

From here, many different supertree methods can be built by changing the

way the starting supertree is defined and the way the starting supertree is recursively modified. Later in this thesis,

- NJNNI will refer to the case where the starting supertree is built based on the neighbor joining (NJ) clustering method from Saitou and Nei (1987) and where the changes recursively applied on the tree are NNI rearrangements, see the right side of Figure 1.7 for an example and Felsenstein (2004, p. 39) for a detailed description ;
- RNix will refer to the case where the starting supertree is chosen at random and is then modified according to Nixon (1999).

In any case, these supertree methods output an unrooted supertree and, by unrooting input phylogenetic trees, destroy the topological information that could be used to root the outputted supertree in order to make it a phylogenetic tree.

This problem can be solved by using an extra leaf, called the outgroup. The outgroup is attached to the root of each input phylogenetic tree prior to the process of unrooting them. The set $D_0$ defined earlier becomes $\{U_1, \ldots, U_d, O\}$. As an example, the character matrix with such outgroup for the two input trees in Figure 1.6 is displayed on the most right part of the same figure. The unrooted version of the first tree in Figure 1.6 with this outgroup attached is displayed at the bottom left of Figure 1.7. Note that the character matrix, with outgroup, has now as many columns as there are internal edges across the original, rooted, input trees. A supertree based on this $6 \times 3$ character matrix will span on $\{U_1, \ldots, U_5\}$ and the outgroup $O$. Once a satisfying supertree is found, the final step is to use the outgroup to root the supertree before removing that outgroup from it.

## 1.8 Conclusion

In this chapter, what a phylogenetic tree is has been formally defined, and it was shown how such a tree can be stored as an adjacency matrix or as a lowest common ancestor matrix, the latter being a more compact and efficient way of storing the tree than the former. A novel way of calculating a distance between two phylogenetic trees has also been given, this new distance being based on the comparison of marginal 3-leaf parts of the two input trees one wishes to compare. In this chapter, how a set of "small" phylogenetic trees can be used to build a larger phylogenetic tree containing them all has also been explored in two different ways, including one novel.

While phylogenetic trees typically arise in evolutionary biology, where they are used to show the inferred evolutionary relationships among various biological species or other entities based upon similarities and differences in their physical or genetic characteristics, this chapter should also be of use for statisticians: indeed, phylogenetic trees are the backbones of nested Archimedean

copulas (Joe, 2001, pp. 87–89) and hierarchical Kendall copulas (Brechmann, 2014), where they are used to model the relationships between random variables as a hierarchy. In this context, leaves of the phylogenetic tree are random variables, while the other nodes represent stops through which these random variables are able to interact.

In the next chapter, the subject of which being nested Archimedean copulas, the content of the current chapter will be shown to be of huge interest when facing the issue of estimation of a phylogenetic tree that properly models the relationships between a given set of random variables, based on some data from the random phenomenon of interest.

# Chapter 2

# Nested Archimedean Copulas

## 2.1 Introduction

Nested Archimedean copulas (NACs), also called hierarchical Archimedean copulas (HACs), introduced by Joe (2001, pp. 87–89), are a natural generalization of Archimedean copulas. The key feature of an Archimedean copula is its generator, which are formally defined in Section 2.2.

Nested Archimedean copulas are made up of two parts: a NAC (or phylogenetic) tree $\lambda$, as defined in Chapter 1, and a set of generators $\Psi$. NACs offer more flexibility for modeling dependencies in a high-dimensional setting while still reducing to Archimedean copulas in simpler cases.

Estimation of nested Archimedean copulas has been the main topic of several papers, see for instance Okhrin et al. (2013a) or Górecki et al. (2015). In this chapter, largely based on papers from Segers and Uyttendaele (2014) and Uyttendaele (2016), the point of view according to which there are two main approaches to estimate a NAC is developed.

In the first approach to estimate a NAC, the target phylogenetic tree is estimated free of any concern for the sufficient nesting condition, this sufficient nesting condition being given in Section 2.2. With this first approach, estimation of the generators after the target phylogenetic tree has been estimated remains an open problem.

In the second approach, the target tree structure $\lambda$ is estimated in such a way that, should the generators from $\Psi$ all belong to a same parametric family, their parameters can be easily estimated afterwards so that the sufficient nesting condition is met.

For each of these two main approaches, several examples are given. Large

scale simulation is performed in Section 2.5, with special focus given to the ability of retrieving the target phylogenetic tree $\lambda$. In Section 2.6, estimation of a target phylogenetic tree based on the daily logreturns of 6 indices is performed using a NAC tree estimator from the first approach. The second approach is applied in Section 2.7, as well as in Section 2.8. In Section 2.7, the dependence structure of weekly logreturns is estimated through time around the 2008 crisis. In Section 2.8, before a discussion section, full estimation of a nested Archimedean copula based on exam results from 482 students is performed, as well as a naive attempt to check the fit using principal component analysis.

Before all this however, a not so short introduction to Archimedean copulas and nested Archimedean copulas is given.

## 2.2 Archimedean and Nested Archimedean copulas

Let $(X_1, \dots, X_d)$ be a vector of continuous random variables. The copula of this vector is defined as

$$C(u_1, \dots, u_d) = P(U_1 \leq u_1, \dots, U_d \leq u_d),$$

where $(U_1, \dots, U_d) = (F_{X_1}(X_1), \dots, F_{X_d}(X_d))$, and where $F_{X_1}, \dots, F_{X_d}$ are the marginal cumulative distribution functions (CDFs) of $X_1, \dots, X_d$.

Archimedean copulas (ACs) can always be written in closed form as

$$C(u_1, \dots, u_d) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d)), \tag{2.2.1}$$

with $\psi : [0, \infty) \to [0, 1]$, a continuous, nonincreasing function, however strictly decreasing on $[0, \inf\{x : \psi(x) = 0\})$ and such that $\psi(0) = 1$ and $\psi(\infty) = 0$. The $\psi$ function is called an Archimedean generator and $\psi^{-1}$ is its generalized inverse.

In order for $C$ in Equation (2.2.1) to be a proper copula when $d = 2$, $\psi$ is also required to a be convex function. This is however not enough for $d > 2$: in order for $C$ to be a proper $d$-dimensional copula, with $d > 2$, the generator is required to be $d$-monotone on $[0, \infty)$. That is,

$$(-1)^k \psi^{(k)}(t) \geq 0$$

must hold $\forall t \geq 0$ and $k \in \{0, 1, 2, \dots, d-2\}$, while $(-1)^{d-2}\psi^{(d-2)}(t)$ is nonincreasing and convex. See McNeil and Nešlehová (2009) for more details.

The generators in Table 2.1 (Hofert and Maechler, 2011) are among the most popular ones. All of them are completely monotone, that is, $d$-monotone for all integer $d \geq 2$. For the Frank family, $D(\theta) = \frac{1}{\theta}\int_0^\theta t/(\exp(t) - 1)\, dt$. For the five families that are given in this table, one can check that increasing

the value of the parameter $\theta$ leads to an increase of the Kendall's $\tau$ coefficient between any two variables of the related Archimedean copula. For instance, the minimum achievable Kendall's $\tau$ coefficient in the case of the AMH family is 0 when $\theta$ approaches 0 and 1/3 when $\theta$ approaches 1.

Table 2.1: Some popular generators of Archimedean copulas.

| name | generator $\psi(x)$ | $\theta$ | $\tau$ |
|---|---|---|---|
| AMH | $(1-\theta)/(e^x-\theta)$ | $\theta \in (0,1)$ | $1-2\left(\theta+(1-\theta)^2\log(1-\theta)\right)/(3\theta^2)$ |
| Clayton | $(1+x)^{-1/\theta}$ | $\theta \in (0,\infty)$ | $\theta/(\theta+2)$ |
| Frank | $-\log(1-(1-e^{-\theta})e^{-x})/\theta$ | $\theta \in (0,\infty)$ | $1+4(D(\theta)-1)/\theta$ |
| Gumbel | $\exp(-x^{1/\theta})$ | $\theta \in [1,\infty)$ | $(\theta-1)/\theta$ |
| Joe | $1-(1-e^{-x})^{1/\theta}$ | $\theta \in [1,\infty)$ | $1-4\sum_{k=1}^{\infty} 1/(k(\theta k+2)(\theta(k-1)+2))$ |

Estimation of an AC is usually performed either by assuming $\psi$ belongs to a parametric family or by not assuming anything about $\psi$, that is, the whole $\psi$ function has to be estimated in a nonparametric fashion. This last case as been studied in details by Genest et al. (2011b).

Consider a sample $(X_{11},\ldots,X_{1d}),\ldots,(X_{n1},\ldots,X_{nd})$, drawn from a multivariate distribution function whose copula $C$ is Archimedean. As originally recognized by Genest and Rivest (1993), inference for Archimedean copulas can be based on the probability integral transformation,

$$W = C(U_1,\ldots,U_d),$$

where $(U_1,\ldots,U_d)$ has distribution $C$, this distribution being the Archimedean copula of interest. The map

$$K(w) = P(W \leq w),$$

defined for all $w \in [0,1]$, is the Kendall distribution function (Barbe et al., 1996; Genest and Rivest, 2001; Nelsen et al., 2003), and is such that $K(w) \geq w$ for all $w \in [0,1]$ whatever the copula $C$. However, if the latter is Archimedean, then, for all $w \in [0,1]$,

$$\lim_{t \uparrow w} K(t) = K(w-) > w. \tag{2.2.2}$$

The nonparametric estimate of the target generator $\psi$ is obtained in several steps:

- Compute the pseudo sample $W_1,\ldots,W_n$ using

$$W_j = \frac{1}{n+1}\sum_{k=1}^{n} \mathbb{1}(X_{k1} < X_{j1},\ldots,X_{kd} < X_{jd}).$$

  Note: the empirical cumulative distribution function built on these pseudo observations can be shown to meet the condition expressed by Equation (2.2.2).

- Denote by $w_1, \ldots, w_m$ the distinct values of $W_1, \ldots, W_n$, in increasing order.
- For each $i \in \{1, \ldots, m\}$, set

$$p_i = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}(W_j = w_{m-i+1}).$$

- Find the values of $s_1, \ldots, s_{m-1}$ that solve the following system of equations:

$$\begin{aligned} w_2 &= (1 - s_{m-1})^{d-1} p_m, \\ w_3 &= (1 - s_{m-1}s_{m-2})^{d-1} p_m + (1 - s_{m-2})^{d-1} p_{m-1}, \\ &\vdots \\ w_m &= (1 - s_1 s_2 s_{m-1})^{d-1} p_m + \ldots + (1 - s_1)^{d-1} p_2. \end{aligned}$$

- Arbitrarily set $r_m = 1$ and retrieve $r_1, \ldots, r_{m-1}$ by using the fact that $r_i / r_{i+1} = s_i$ for $i \in \{1, \ldots, m-1\}$.
- The estimate of $\psi(x)$ is then

$$\hat{\psi}(x) = \sum_{i : r_i > x} \left(1 - \frac{x}{r_i}\right)^{d-1} p_i.$$

Once this nonparametric generator is available, the following algorithm will allow to draw observations from the nonparametric AC defined by this generator:

**Algorithm 2** Generating one observation from a nonparametric AC defined by $\hat{\psi}$.

1: draw one value $R$ from the vector $r_1, \ldots, r_m$ while taking into account the weights $p_1, \ldots, p_m$.
2: draw a vector $(S_1, \ldots, S_d)$ from the uniform distribution on the simplex.
3: set $U_1 = \hat{\psi}(RS_1), \ldots, U_d = \hat{\psi}(RS_d)$.

The second step in Algorithm 2 is always easy to carry out, as it suffices to generate independent exponential random variables $(E_1, \ldots, E_d)$ and to set, for every $i \in \{1, \ldots, d\}$,

$$S_i = \frac{E_i}{E_1 + \cdots + E_d}.$$

Note that while no ties are possible in the vector $(RS_1, \ldots, RS_d)$, the vector $(U_1, \ldots, U_d)$ might contain some, because $\hat{\psi}$ is a step function.

For more details on nonparametric estimation of Archimedean copulas, refer to Genest et al. (2011b).

In the case of Archimedean copulas, $C(u_1, \ldots, u_d)$ is a symmetric function in its arguments and this is why Archimedean copulas are sometimes called *exchangeable*. The result of this exchangeability property is easily seen by plotting a cloud of points generated from a bivariate Archimedean copula: the $y = x$ axis is a clear axis of reflection symmetry for the underlying distribution. For a cloud of points generated from a trivariate Archimedean copula, even more complex symmetries for the underlying trivariate distribution can be observed.

Moreover, given a $d$-variate Archimedean copula and $m \in \{2, \ldots, d-1\}$, any two $m$-variate margins from that Archimedean copula describe the same $m$-variate distribution. For instance, with $m = 3$ and assuming the joint distribution of $(U_1, \ldots, U_{10})$ is an Archimedean copula, the joint distribution of $(U_6, U_5, U_3)$ is equal to the joint distribution of $(U_3, U_{10}, U_2)$, $(U_1, U_4, U_8)$ or of any vector $(U_i, U_j, U_k)$ with distinct $i, j, k \in \{1, \ldots, 10\}$.

It is clear that, for modeling purposes, this exchangeability property becomes an increasingly strong assumption as the dimension $d$ grows. To relax it, nested Archimedean copulas were introduced. NACs are obtained by plugging in Archimedean copulas into each other (Joe, 2001, pp. 87–89). The following example shows how a bivariate Archimedean copula $C_{23}$ can be plugged into a bivariate Archimedean copula $C_{123}$:

$$C_{123}(u_1, \boldsymbol{C_{23}(u_2, u_3)}) = \psi_{123}\big(\psi_{123}^{-1}(u_1) + \psi_{123}^{-1}(\boldsymbol{\psi_{23}(\psi_{23}^{-1}(u_2) + \psi_{23}^{-1}(u_3))})\big) \quad (2.2.3)$$

The above trivariate copula, a nested Archimedean copula on $D_0 = \{U_1, U_2, U_3\}$, is still such that the $y = x$ axis remains an axis of reflection symmetry for all bivariate margins. However, while even more complex symmetries are observed on the trivariate level of an AC, part of these symmetries are lost on the trivariate level of the NAC described by (2.2.3). Moreover, while all bivariate margins are the same in an AC, the distribution of $(U_2, U_3)$ is not the same as the distribution of $(U_1, U_2)$ or $(U_1, U_3)$ in the NAC described by (2.2.3).

In a NAC, the way Archimedean copulas are nested corresponds to a special kind of tree $\lambda$ called a NAC tree or phylogenetic tree, formally defined in Chapter 1. Nested Archimedean copulas, such as the one in (2.2.3) and for which the tree can be seen on the left of Figure 2.1, are always defined through a NAC tree and through a collection of generators $\Psi$, one generator for each internal node in the NAC tree. Each generator fully describes the dependence between the random variables interacting through the related internal node. Archimedean copulas can be seen as a special case of NACs: they exhibit a trivial structure such as the one on the right of Figure 2.1 and have only one generator, the one related to the only internal node, the root. A trivial NAC tree of dimension $d$ is sometimes called a $d$-fan (Ng and Wormald, 1996).

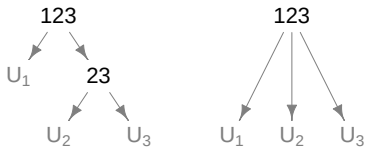



Figure 2.1: Left: the NAC tree implied by (2.2.3). The dependence between $U_2$ and $U_3$ is described by $\psi_{23}$ while the dependence between $U_1$ and $U_2$ or $U_1$ and $U_3$ is described by $\psi_{123}$. Right: the structure of a trivariate AC. The dependence between any two random variables is here described by $\psi_{123}$. Note that the labels of the internal nodes in both structures (123 and 23 for the left structure, 23 for the right structure) are actually irrelevant.

Let $\lambda$ be a phylogenetic tree on $D_0 = \{U_1, \ldots, U_d\}$, that is, a d-leaf tree. Define the related set of indices as $i(D_0) = \{1, \ldots, d\}$. Suppose that for each $B \in \lambda$ with $|B| \geqslant 2$, an Archimedean generator $\psi_B$ is given, that is, a generator is given for each branching node in the structure and $\Psi$ is the set of all these generators.

Next, recursively define the functions $C_B : [0,1]^{|B|} \to [0,1]$, with $B \in \lambda$, $|B| \geqslant 1$, by

$$C_B(\{u_b : b \in i(B)\}) = \begin{cases} u_b & \text{if } B = \{U_b\} \\ \psi_B\Big(\sum_{A \in \mathcal{C}(B,\lambda)} \psi_B^{-1}\big(C_A(u_a : a \in i(A))\big)\Big) & \text{if } |B| \geqslant 2 \end{cases}, \tag{2.2.4}$$

where the set of children of $B$ in $\lambda$ is denoted $\mathcal{C}(B, \lambda)$ and $i(B)$ is the set of indices related to $B$ (example : if $B = \{U_1, U_3, U_4\}$, then $i(B) = \{1, 3, 4\}$).

Definition 2.2.1. *A d-variate copula $C_{D_0}$ is a* nested Archimedean copula (NAC) *if it is of the form $C_B$ in* (2.2.4)*, with $B = D_0$.*

For convenience, a NAC $C_{D_0}$ is often written as $C_{i(D_0)}$. For instance, with

$$D_0 = \{U_1, U_2, U_3\},$$

the notation becomes $C_{123}$.

Definition 2.2.1 is however not free from identifiability issues. Recall that a parameter $\mu$ (possibly infinite-dimensional) in a statistical model $(P_\mu : \mu \in M)$, with $P_\mu$ a probability measure on a fixed space, is identifiable if $\mu_1 \neq \mu_2$ implies that $P_{\mu_1} \neq P_{\mu_2}$, that is, different parameters yield different distributions of the observable. For $d$-variate nested Archimedean copulas, the parameter $\mu$ consists of the pair

$$\big(\lambda, \Psi = \{\psi_B : B \in \lambda, |B| \geqslant 2\}\big).$$

In this parametrization, the parameter $\mu$ is not identifiable, since replacing a generator function $\psi_B(x)$ by the function $\psi_B(ax)$, with $0 < a < \infty$, yields the same copula ; that is, the generator functions are identifiable up to scaling

only. This issue can be solved easily in different ways, for instance by requiring that $\psi_B(1) = 1/2$.

A more fundamental identifiability issue arises if some generator functions are the same. Consider for instance the tree $\lambda$ implied by Equation (2.2.3), shown on the left in Figure 2.1. If the generators $\psi_{123}$ and $\psi_{23}$ are the same, say $\boldsymbol{\psi}$, then the nested Archimedean copula with parameter $(\lambda; \{\psi_{123}, \psi_{23}\})$ actually describes an exchangeable Archimedean copula with generator $\boldsymbol{\psi}$, that is, a nested Archimedean copula with trivial tree structure and single generator $\boldsymbol{\psi}$.

To ensure identifiability of the structure, we must require that for any two nodes $A$ and $B$ such that $A \subset B$ and $A \neq B$, meaning $A$ is a descendant of $B$, the bivariate Archimedean copulas generated by the generator functions $\psi_A$ and $\psi_B$ are different, prohibiting the tree structure to collapse at this level. If this condition holds, then the structure $\lambda$ can be identified. This weak restriction on the generators will be assumed to hold throughout this Chapter.

Note that some generator functions can still be identical. Consider for instance the NAC structure in Figure 2.2.

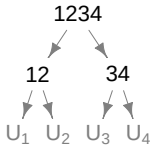



Figure 2.2: A four-variate NAC tree.

The generators associated to the nodes 12 and 34 can be identical, without simplification of the tree being possible.

**Sufficient nesting condition.** Even if this last identifiability condition on the generators holds, a set $\Psi$ of arbitrary generators does not ensure that the resulting NAC defined through $(\lambda, \Psi)$ will be a proper copula, meeting the three properties any copula must mean (refer to the introduction section of the thesis). In case one is working with fully nested Archimedean copula structures, that is, structures where each branching node has either two leaves as children, or one leaf and another branching node, a sufficient nesting condition ensuring this NAC is a proper copula was however developed by Joe (2001) and McNeil (2008): $\forall I, J \in \lambda$ such that $|I|, |J| \geq 2$ (that is, such that $I, J$ are internal nodes of the phylogenetic tree $\lambda$) and such that $J$ is a child of $I$, $\psi_I^{-1} \circ \psi_J$ is required to be a Bernstein function.

A Bernstein function $f$ is such that $f : [0, \infty) \to [0, \infty)$ and

$$(-1)^k f^{(k)}(x) \leq 0,$$

for all $x > 0$ and $k \in \mathbb{N}$.

If all elements of $\Psi$ belong to the same parametric family (if not, see Hofert, 2011), something assumed throughout most of this chapter and most paper related to NACs, and if, moreover, this family is one of the families given in Table 2.1, this sufficient nesting condition simply states that the parameters $\theta_I$ and $\theta_J$ related to $\psi_I$ and $\psi_J$ must be such that $\theta_I < \theta_J$ for every pair of internal nodes $I$ and $J$ in the NAC tree structure such that $J$ is a child of $I$. That is, the Kendall's correlation coefficient $\tau$ between two random variables increases as these variables are able to interact through a node farther away from the root node.

It was later proved that this sufficient nesting condition holds for all NACs, not only fully nested ones, see Holeňa et al. (2015) Also note that a recent paper by Rezapour (2015) gave weak sufficient and *necessary* conditions concerning the generators, guaranteeing that the constructed function is a copula.

To conclude this section, we come back to lowest common ancestor matrices, that were developed in Chapter 1, Section 1.3, and that were said to be of particular interest when working with nested Archimedean copulas. Given a nested Archimedean copula $(\lambda, \Psi)$, the lowest common ancestor matrix built on $\lambda$ will naturally match any scatterplot matrix drawn based on iid observations from that NAC. For instance, assume that $(\lambda, \Psi)$ is a nine-dimensional copula of the form

$$C_L(u_3, u_6, u_1, C_M(u_9, u_2, u_7, u_5, C_W(u_8, u4; \psi_W); \psi_M); \psi_L),$$

where the three generators $\psi_L, \psi_M$ and $\psi_W$ are all Clayton generators with Kendall's $\tau$ correlation coefficients set to 0.2, 0.5 and 0.8, respectively.

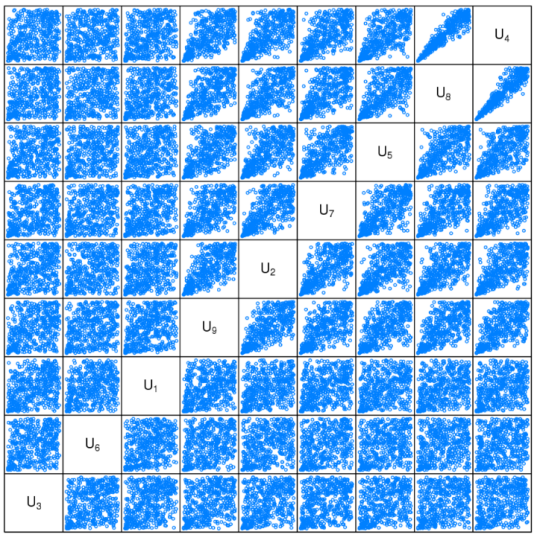
Figure 2.3 displays on its top a scatterplot matrix based on 500 observations from that NAC (source: Hofert and Maechler, 2011), while the bottom of Figure 2.3 displays the lca matrix built on the NAC of interest.

The theoretical match between the two is of interest for estimation purposes: indeed, if one wants to check if an estimated tree $\hat{\lambda}$ fits the data, one can compare the lca matrix resulting from $\hat{\lambda}$ with a scatterplot matrix of the data, offering a quick, visual goodness-of-fit diagnostic tool.

Some suggested papers for the readers eager to learn more about NACs are: McNeil (2008), Hofert and Pham (2013), Okhrin et al. (2013b) or Matsypura et al. (2016).

## 2.3 A first approach to estimate a NAC

In this first approach, the NAC structure $\lambda$ from a target NAC must be estimated free of any constraints, including the sufficient nesting condition. How to get towards $\hat{\lambda}$ in such case?




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| L | L | L | M | M | M | M | W | $U_4$ |
| L | L | L | M | M | M | M | $U_8$ | |
| L | L | L | M | M | M | $U_5$ | | |
| L | L | L | M | M | $U_7$ | | | |
| L | L | L | M | $U_2$ | | | | |
| L | L | L | $U_9$ | | | | | |
| L | L | $U_1$ | | | | | | |
| L | $U_6$ | | | | | | | |
| $U_3$ | | | | | | | | |

Figure 2.3: Top: scatterplot matrix of a nine-dimensional NAC. Bottom: the lca matrix for the corresponding tree $\lambda$.

### 2.3.1 Estimation of a NAC tree, $d = 3$

Let $(X_1, X_2, X_3)$ be a vector of continuous random variables such that the joint distribution of $(U_1, U_2, U_3) = (F_{X_1}(X_1), F_{X_2}(X_2), F_{X_3}(X_3))$ is a nested Archimedean copula. We are interested in estimating the NAC structure based on $n$ observations $(x_{l1}, x_{l2}, x_{l3})$ from $(X_1, X_2, X_3)$, $l = 1, \ldots, n$. We know from Section 1.4 in Chapter 1 that there are only four structures fulfilling Definition 1.4 in the trivariate case: $\Lambda_{123}, \lambda_{23}, \lambda_{12}$ and $\lambda_{13}$.

In the trivial structure (tree $\Lambda_{123}$), all bivariate marginal distributions of the nested Archimedean copula are the same, while in structures $\lambda_{23}$, $\lambda_{12}$ and $\lambda_{13}$, two bivariate marginal distributions are the same and one is different. Moreover, if the bivariate marginal distributions are not all the same, being able to determine which one is different is enough to select the proper nested Archimedean copula structure $\lambda_{23}$, $\lambda_{12}$ or $\lambda_{13}$.

It is known from Genest and Rivest (1993) that the Kendall distribution of a pair of variables $(X_j, X_k)$ fully determines the copula of that pair provided the copula is Archimedean, which is always true if both variables are part of a NAC. Thus, rather than working directly with bivariate distributions, Kendall distributions are going to be used instead, since they are univariate and therefore easier to handle. The Kendall distribution of the pair $(X_j, X_k)$ is defined as the distribution of the variable

$$W_{jk} = C_{jk}(U_j, U_k) = F_{jk}(X_j, X_k),$$

where $C_{jk}(u_j, u_k) = P(U_j \leq u_j, U_k \leq u_k)$ is the joint CDF of $(U_j, U_k)$, and where $F_{jk}(x_j, x_k) = P(X_j \leq x_j, X_k \leq x_k)$ is the joint CDF of $(X_j, X_k)$. The map defined for all $w \in [0, 1]$ by

$$K_{jk}(w) = P(W_{jk} \leq w),$$

is the Kendall distribution function (Barbe et al. 1996; Nelsen et al. 2003; Genest and Rivest 2001).

The Kendall distribution function of a pair of variables $(X_j, X_k)$ can be estimated (Genest, Nešlehová, and Ziegel, 2011b) by first computing the pseudo-observations $w_{1,jk}, \ldots, w_{n,jk}$ and then the empirical distribution function of these pseudo-observations:

$$w_{m,jk} = \frac{1}{n+1} \sum_{l=1}^{n} 1(x_{lj} < x_{mj}, x_{lk} < x_{mk});$$

$$K_{n,jk}(w) = \frac{1}{n} \sum_{m=1}^{n} 1(w_{m,jk} \leq w), \text{ with } 0 < w < 1.$$

Since there are three possible pairs in our case, namely $(X_1, X_2), (X_1, X_3)$ and $(X_2, X_3)$, three empirical Kendall distribution functions need to be estimated. The distance between the empirical Kendall distribution functions of $(X_i, X_j)$ and $(X_i, X_k)$ is defined as

$$\int_0^1 |K_{n,ij}(x) - K_{n,ik}(x)| \, dx = \frac{1}{n} \sum_{m=1}^{n} |w_{(m),ij} - w_{(m),ik}| = \delta_{ij,ik},$$

where $w_{(1),ij}, \ldots, w_{(n),ij}$ are the ordered pseudo-observations related to the variables $(X_i, X_j)$ and $w_{(1),ik}, \ldots, w_{(n),ik}$ are those related to the variables $(X_i, X_k)$.

In general, a trivial NAC structure on $\{U_1, U_2, U_3\}$ will result in three distances that are all about the same, while trees such as $\lambda_{12}$, $\lambda_{13}$ or $\lambda_{23}$ will result in one small distance relative to two other distances that are bigger and about the same. For any three variables $(X_i, X_j, X_k)$, if, for instance, $\delta_{ij,ik}$ is the minimum among the three distances, it seems reasonable to assume that the tree spanned on $(X_i, X_j, X_k)$ is either the trivial structure or the structure $\lambda_{jk}$ where $(X_i, X_j)$ and $(X_i, X_k)$ have the same Kendall distribution.

The problem of determining the structure of $(X_1, X_2, X_3)$ can thus be rewritten as a hypothesis test:

$$\begin{aligned} H_0 &: \quad \text{the true structure is the trivial structure,} \\ H_1 &: \quad \text{the true structure is not the trivial structure,} \end{aligned}$$

and if $H_0$ gets rejected, the true structure must be $\lambda_{12}$, $\lambda_{13}$ or $\lambda_{23}$, depending on which pair of Kendall functions were the closest.

As a test statistic, the absolute difference between the minimum distance and the average of the two remaining distances is used. The null hypothesis is rejected when the test statistic is observed in the upper tail of its $H_0$ distribution.

As the $H_0$ distribution of the test statistic is unknown, p-values are calculated thanks to some bootstrapping. How exactly this is done is described hereafter.

Under $H_0$, the original sample is assumed to come from an unknown trivariate Archimedean copula. Using the work of Genest et al. (2011b), also refer to Section 2.2, it is possible to estimate that Archimedean copula nonparametrically and to resample from that estimated Archimedean copula, thanks to Algorithm 2. For each new sample, the three empirical Kendall distributions, the three distances, and the related test statistic are calculated. The p-value of the observed test statistic, the one from the original sample, is then estimated by the proportion of test statistics obtained from the new samples that are greater than or equal to the value of the observed test statistic from the original sample. Should this estimated p-value be lower than a significance level $\alpha$,

for instance 10%, the null hypothesis is rejected.

Since the estimator for the Kendall distribution depends on the data only through the ranks and since the test statistic only depends on this estimator, the NAC structure estimator for trivariate structures is rank-based, too.

There are two key points in the procedure presented above:

- First, choose between a trivial structure (= $H_0$) and $H_1$.
- Second, if $H_0$ gets rejected, determine what should be the estimated structure: $\lambda_{12}$, $\lambda_{13}$ or $\lambda_{23}$?

Possible errors are:

- if the true structure is the trivial structure, rejecting it and therefore committing a type I error ;
- if the true structure is structure $\lambda_{12}$, $\lambda_{13}$ or $\lambda_{23}$, failing to reject $H_0$ (type II error) ;
- if the true structure is for instance structure $\lambda_{12}$, successfully rejecting $H_0$, yet picking the wrong structure among $\lambda_{12}$, $\lambda_{13}$ or $\lambda_{23}$ afterwards. This will be referred to later on as a type III error.

The main difficulty with the test developed in this section is encountered when the true structure is the trivial trivariate structure, that is, the structure one gets when the nested Archimedean copula is actually an exchangeable Archimedean copula. Indeed, if the probability of committing a type I error is fixed to $\alpha = 0.10$, the trivial structure will be rejected 10% of the time regardless of the input sample size $n$. The estimator is therefore not a consistent estimator for the trivial trivariate structure, unless $\alpha$ tends to 0 as $n$ increases, so that type I errors are asymptotically impossible. Practically speaking however, this rule has little significance as it does not help to select $\alpha$ given a value of $n$. In the simulations, $\alpha$ will be set to a fixed value of 10% for all $n$, and these simulations will confirm that a fixed value for $\alpha$ does not seem to prevent consistent estimation for the target structure when this target structure is made of 3-fans only, but well when this is not the case.

### 2.3.2 Estimation of a NAC tree structure, $d \geq 4$

Let $\lambda$ be a NAC structure on a finite set $D_0 = \{U_1, \ldots, U_d\}, d \geq 4$. It is known from Proposition 1.5.1 in Subsection 1.5 that if the NAC tree spanned on any three distinct elements of $D_0$ is known (that is, each element of ${}^3(\lambda)$ is known), then $\lambda$ can be uniquely recovered, for instance using Algorithm 1.

A first suggestion for estimating $\lambda$ is to estimate each element of ${}^3(\lambda)$ using the procedure developed in Subsection 2.3.1, thus effectively getting $\widehat{{}^3(\lambda)}$, which can then be used to build $\hat{\lambda}$.

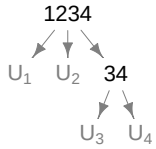



Figure 2.4: A four-variate NAC tree.

However, if each element of ${}^3(\lambda)$ is estimated, the problem of reconstructing a tree from that set of estimated trivariate trees is a bit different than what was considered in Section 1.5. Indeed, it is not guaranteed that a proper nested Archimedean copula structure can be recovered from a given set of estimated trivariate structures, that is, $\widehat{{}^3(\lambda)}$ does not necessarily lead to a proper NAC tree. When $\hat{\lambda}$ retrieved from $\widehat{{}^3(\lambda)}$ is not a proper nested Archimedean copula structure, meaning it does not fulfill Definition 1.1.1, or is such that ${}^3(\hat{\lambda}) \neq \widehat{{}^3(\lambda)}$, we call $\widehat{{}^3(\lambda)}$ a *faulty* set.

With a value of $\alpha$ equal to 0.00 for all tests required to estimate ${}^3(\lambda)$, we fail to reject the null hypothesis everywhere and we therefore get a set of estimated trivariate structures each describing a 3-fan. Such a set is never a faulty set, and $\hat{\lambda}$, the estimated NAC structure retrieved from it, will always be a trivial structure of dimension $d$, a $d$-fan. Of course, if the true structure is not a $d$-fan, a value of $\alpha$ equal to 0.00 means you are sure to commit type II errors.

With a value of $\alpha$ equal to 1.00 for all tests, all null hypotheses are rejected and we end up with a set where each estimated trivariate structure describes a triple. Such a set can be a faulty set and usually is.

Assuming the copula of the vector $(X_1, \ldots, X_d)$ is a nested Archimedean copula, a faulty set of estimated trivariate structures means at least one error (type I, type II or type III) has been committed. Notice the converse is not true: even when at least one type I, type II or type III error has been committed, $\widehat{{}^3(\lambda)}$ might lead to a structure $\hat{\lambda}$ meeting Definition 1.1.1 and such that ${}^3(\hat{\lambda}) = \widehat{{}^3(\lambda)}$. However $\hat{\lambda}$ is not equal to $\lambda$, the target structure. Suppose indeed the target structure is the structure on Figure 2.2. This structure is uniquely determined by the four trivariate structures in Figure 1.4. Suppose however that the first two triples in Figure 1.4 are replaced by two 3-fans, that is, two type II errors have been committed during the estimation process. Yet, this leads to a proper four-variate structure, shown in Figure 2.4, however unequal to the target structure.

A faulty set is essentially a red flag that should be viewed as an opportunity for correction. What kind of corrective measure should be applied to such a set is however not straightforward. As done in the simulation study, we simply suggest to decrease the value of $\alpha$ for all tests until the resulting set of estimated

trivariate structures is not a faulty set anymore. At worst, $\alpha$ is to be decreased down to 0.00, we end up with a set of 3-fans, and $\hat{\lambda}$ is then a $d$-fan. This last strategy is certainly not the best one can imagine, but is a very convenient one to apply and ensures that $\hat{\lambda}$ will always be a proper tree.

Since the estimator developed in Subsection 2.3.1 is unable to consistently estimate a 3-fan if we keep the same value of $\alpha > 0$ for all $n$, it also means consistent estimation of any $\lambda$ that has at least one trivial trivariate component is impossible, as will be seen in the performance section of this chapter.

Another suggestion in order to estimate $\lambda$ in the case $d \geq 4$ is to use **supertree methods** (refer to Chapter 1).

In order to use a supertree method in this case, it is first assumed that the target structure $\lambda$ spanned on the vector $(U_1, \ldots, U_d)$ is a binary tree, that is, a NAC structure where each internal node has two and only two children (this assumption will be later relaxed). Example of a binary tree: the tree on the left of Figure 2.1 or in Figure 2.2. Not a binary tree: the tree in Figure 2.4 or the second tree displayed in Figure 1.6.

As input set of trees for the supertree method, $\widehat{{}^3(\lambda)}$ is used. But since the target tree is now assumed to be a binary tree, it is known that $\widehat{{}^3(\lambda)}$ must itself be made of binary trees only. Given a vector of three variables $(X_i, X_j, X_k)$, the hypothesis test from Subsection 2.3.1 can be skipped: just estimate the Kendall distribution for each of the tree pairs $(X_i, X_j)$, $(X_i, X_k)$ and $(X_j, X_k)$, and find out which are the two estimated Kendall distributions that are the closest according to some distance. If, for instance, the estimated Kendall distributions of $(X_i, X_j)$ and $(X_i, X_k)$ are the closest, then the trivariate target tree structure must be a structure where $U_i$ is left apart while $U_j$ and $U_k$ are together.

The result of the supertree method fed with a set $\widehat{{}^3(\lambda)}$ of estimated binary trees will be a binary supertree $\hat{\lambda}_b$. Even if $\widehat{{}^3(\lambda)}$ is a faulty set, a proper binary NAC tree $\hat{\lambda}_b$ will be produced. This is because a supertree method resumes all the information from the input set $\widehat{{}^3(\lambda)}$ as a character matrix and then try to built in a recursive fashion a tree that agrees as much as possible with that character matrix according to some criterion. Again, refer to Chapter 1 for details.

But what if the target tree $\lambda$ was in fact not a binary tree? To allow for a more general estimation, suggestion is to collapse, if necessary, one or several parts of the estimated binary structure $\hat{\lambda}_b$ produced by the supertree method used. Many ways to properly collapse such a binary tree are explored in Section 2.4.

### 2.3.3 Estimation of the generators

What to do next, once $\lambda$ has been estimated free of any concern for the sufficient nesting condition remains an open problem. If the generators across $\lambda$ are assumed to belong to a same parametric family $F$, estimation of the related set of parameters $\Theta$ can be performed by simple inversion of the estimated Kendall correlation coefficients, the same way it is done using equation (2.4.5) in Section 2.4. In most cases, this is expected to lead to a proper NAC $(\hat{\lambda}, \hat{\Psi})$. But in some cases, it will unfortunately not be true. This is because estimated Kendall correlation coefficients along trees built using this first approach are not always ordered, making the sufficient nesting condition hard to satisfy. A stage-wise estimation of the parameters based on a quasi-maximum likelihood approach (from the lowest to the highest nesting level), as done by Okhrin and Ristig (2014), with subsequent shortening of the feasible parameter space, could however help ensure that the final object is a well defined NAC.

## 2.4 A second approach to estimate a NAC

In this section, a second approach to estimate a nested Archimedean copula $(\lambda, \Psi)$ is presented. With this approach, a binary tree is built using hierarchical clustering and then is collapsed if necessary, according to some strategy. Then, if the target generators are assumed to belong to a same known parametric family $F$, estimation of the set of parameters $\Theta$ related to these generators so that the sufficient nesting condition (see Section 2.2) is met can easily be performed. Okhrin et al. (2013a) and Górecki et al. (2015) are both particular cases of this second approach. By design, this second approach will always lead to a proper NAC, meeting the sufficient nesting condition, in contrast to the first approach developed earlier. But if the target NAC is not a NAC meeting the sufficient nesting condition, the estimated NAC one gets here will always be biased.

**Estimation of the target tree structure** $\lambda$. Based on an iid sample of size $n$ from $(X_1, \ldots, X_d)$, a distance for each couple $(X_i, X_j)$ with distinct $i, j \in \{1, \ldots, d\}$ is first estimated. The random variables are then clustered, using hierarchical clustering, one at the time, according to the estimated distances. Getting an estimated binary tree structure this way however makes sense only if the estimated distance for each couple $(X_i, X_j)$ is a function of a measure of association $\bowtie$ such that highly related random variables will be considered close and therefore the distance between the two will be small ; if the tree is built from leaves to the root using the smallest distances first ; and if the target NAC is such that the dependence between any two random variables in the structure increases as the variables are able to interact through nodes that are farther away from the root, that is, if the target NAC $(\lambda, \Psi)$ is assumed to meet the sufficient nesting condition.

A measure of association $\bowtie$ between two random variables can be obtained, for instance, through

- Kendall's $\tau$,
- a distance between the (theoretical) Kendall distribution of two independent variables and the empirical Kendall distribution of the two variables under study,
- Hoeffding's $\mathbb{D}$ statistic (see Hoeffding, 1948),
- or Spearman's rho.

Note that these measures of association are all such that $\bowtie (X_i, X_j) = \bowtie (U_i, U_j)$, so that the binary tree estimated on $(X_1, \ldots, X_d)$ is actually the binary tree estimated on $(U_1, \ldots, U_d)$.

For the linkage criteria, single, complete or average linkage can be used (and where experimentally tested by Górecki et al., 2015). In the rest of this section, only average linkage will be considered.

If somehow the target tree structure $\lambda$ is known to be a binary tree, it is possible to directly move on to the problem of estimation of the generators from here. However, a target NAC structure is not always a binary structure and thus $\hat{\lambda}_b$, the result from the hierarchical clustering, must be transformed.

Let $V_r = \{v_1, \ldots, v_r\}$ be the set of internal nodes in $\hat{\lambda}_b$. For any two nodes $\{v_l, v_m\}$ from $V_r$ with $l, m \in \{1, \ldots, r\}$ such that $v_l$ is a parent of $v_m$, collapse the two nodes if you suspect the generator $\psi_{v_l}$ to be too similar to the generator $\psi_{v_m}$. The right side of Figure 2.5 shows the collapsing of nodes 234 and 34 into a new node. Notice the resulting tree is not a binary tree anymore.

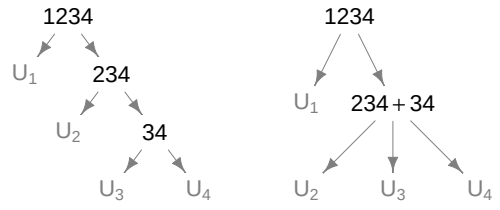



Figure 2.5: Left: a binary structure spanned on four random variables. Right: nodes 234 and 34 have been collapsed into one node. The resulting tree is not a binary tree anymore.

To compare two successive generators $\psi_{v_l}$ and $\psi_{v_m}$, one needs first to estimate these generators, as they are unknown. If they are known to belong to a same given parametric family $F$, the problem is then to estimate $\theta_{v_l}$ and $\theta_{v_m}$ and to collapse if $\left|\hat{\theta}_{v_l} - \hat{\theta}_{v_m}\right| < \theta_c$, where $\theta_c$ is some threshold value. Note that at this point, we are not concerned by the sufficient nesting condition: the goal here is only to decide if two successive nodes in the binary tree should be collapsed, based on a rough estimation of the related generators $\psi_{v_l}$ and

$\psi_{v_m}$. Formal estimation of the generators is done later. To make the problem of whether or not two successive nodes should be collapsed easier, each of the two related generators can also be summarized as a scalar measure $s$ reflecting the dependence between the random variables interacting through the related node, and the two nodes are collapsed if the absolute difference between their respective estimated scalar measure is lower than a chosen threshold, that is, if

$$|\hat{s}_{v_l} - \hat{s}_{v_m}| < s_c.$$

For instance, looking back at Figure 2.5, one can estimate a summary of the dependence between the random variables interacting through node 234 by averaging the estimated Kendall correlation coefficients, calculated within the random pairs $(U_2, U_3)$ and $(U_2, U_4)$. The estimated summary of the generator related to node 34 is equal to the estimated Kendall's $\tau$ between the random variables $(U_3, U_4)$. The two nodes are collapsed if the inequality

$$\left| \left( \frac{\hat{\tau}_{23} + \hat{\tau}_{24}}{2} \right) - \hat{\tau}_{34} \right| < t_c \tag{2.4.1}$$

holds, where $t_c$ is the critical threshold for collapsing.

Using the fact that a vector of estimated Kendall's taus is asymptotically distributed according to a multivariate normal distribution (Ehrenberg, 1952; Genest et al., 2011a; Rublík, 2016), a hypothesis test based on Equation (2.4.1) can be built, offering a meaningful choice for the critical threshold $\tau_c$. This novel hypothesis test is hereafter formalized.

First, let's mark the two nodes with the symbols $\uparrow$ and $\downarrow$, the node marked with $\uparrow$ being the one closer to the root node, while the one marked with $\downarrow$ is the one further down the tree, thus further from the root node. Second, let's generalize the left side of Equation (2.4.1) as

$$T = \frac{\sum_{i=1}^{n_\uparrow} \hat{\tau}_i}{n_\uparrow} - \frac{\sum_{j=1}^{n_\downarrow} \hat{\tau}_j}{n_\downarrow}, \tag{2.4.2}$$

where $\{\hat{\tau}_i\}, i \in \{1, \ldots, n_\uparrow\}$, is the set of all estimated Kendall's taus that are such that, for each Kendall's tau in that set, the two related random variables interact through the node marked with $\uparrow$. Similarly, $\{\hat{\tau}_j\}, j \in \{1, \ldots, n_\downarrow\}$ is the set of Kendall's taus that are such that, for each Kendall's tau in that set, the two related random variables interact through the node marked with $\downarrow$.

Thanks to Rublík (2016), we know that the vector $\hat{\boldsymbol{\tau}}$ made of the union of $\{\hat{\tau}_i\}$ and $\{\hat{\tau}_j\}$ is such that

$$\sqrt{n}(\hat{\boldsymbol{\tau}} - \boldsymbol{\tau}) \to N_{n_\uparrow + n_\downarrow}(\mathbf{0}, \boldsymbol{V}),$$

where $n$ is the sample size (not to be confused with $n_\uparrow$, $n_\downarrow$, or $n_\uparrow + n_\downarrow$) and $\boldsymbol{V}$ is the asymptotic covariance matrix, for which an estimator $\hat{\boldsymbol{V}}$ can also be found in Rublík (2016), the expression of it being given hereafter.

Let $\{(l_1, u_1), \ldots, (l_{n_\uparrow+n_\downarrow}, u_{n_\uparrow+n_\downarrow})\}$ be the set of $n_\uparrow + n_\downarrow$ pairs of coordinate indices of the $d$-dimensional random vector $\boldsymbol{U} = (U_1, \ldots, U_d)$. To each pair in that set, there corresponds one and only one estimated Kendall's tau in either $\{\hat{\tau_i}\}$ or $\{\hat{\tau_j}\}$.

Define $\boldsymbol{u}_i$ as observation $i$ of $\boldsymbol{U}$ and $\boldsymbol{u}_i(l_1)$ as element $l_1$ of the vector $\boldsymbol{u}_i$. Then,

$$\hat{\boldsymbol{V}} = 4(\boldsymbol{\Gamma}_n - \boldsymbol{\Upsilon}_n \boldsymbol{\Upsilon}_n^T),$$

where

$$\boldsymbol{\Gamma}_n = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\Gamma}_i^{(n)} \left( \boldsymbol{\Gamma}_i^{(n)} \right)^T,$$

$$\boldsymbol{\Gamma}_i^{(n)} = \frac{1}{n-1} \sum_{j \in \{1,\ldots,n\},\ j \neq i} \boldsymbol{\Psi}(\boldsymbol{u}_i, \boldsymbol{u}_j),$$

$$\boldsymbol{\Psi}(\boldsymbol{u}_i, \boldsymbol{u}_j) = \begin{bmatrix} \operatorname{sign}(\boldsymbol{u}_j(l_1) - \boldsymbol{u}_i(l_1)) \operatorname{sign}(\boldsymbol{u}_j(u_1) - \boldsymbol{u}_i(u_1)) \\ \vdots \\ \operatorname{sign}(\boldsymbol{u}_j(l_{n_\uparrow+n_\downarrow}) - \boldsymbol{u}_i(l_{n_\uparrow+n_\downarrow})) \operatorname{sign}(\boldsymbol{u}_j(u_{n_\uparrow+n_\downarrow}) - \boldsymbol{u}_i(u_{n_\uparrow+n_\downarrow})) \end{bmatrix},$$

and, finally,

$$\boldsymbol{\Upsilon}_n = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \boldsymbol{\Psi}(\boldsymbol{u}_i, \boldsymbol{u}_j).$$

Under the null hypothesis, nodes $\uparrow$ and $\downarrow$ are the same, and one would expect $T$ in (2.4.2) to have a centered normal distribution with variance $V_T$. Straightforward calculations lead to

$$V_T = \left( \frac{1}{n_\uparrow n_\downarrow} \right)^2 \left[ n_\uparrow^2 \operatorname{Var} \left( \sum_{j=1}^{n_\downarrow} \hat{\tau}_j \right) + n_\downarrow^2 \operatorname{Var} \left( \sum_{i=1}^{n_\uparrow} \hat{\tau}_i \right) - 2 n_\uparrow n_\downarrow \operatorname{Cov} \left( \sum_{i=1}^{n_\uparrow} \hat{\tau}_i, \sum_{j=1}^{n_\downarrow} \hat{\tau}_j \right) \right],$$

where one can easily extract an estimate of

$$\operatorname{Var} \left( \sum_{j=1}^{n_\downarrow} \hat{\tau}_j \right),$$

$$\operatorname{Var} \left( \sum_{i=1}^{n_\uparrow} \hat{\tau}_i \right),$$

and

$$\operatorname{Cov} \left( \sum_{i=1}^{n_\uparrow} \hat{\tau}_i, \sum_{j=1}^{n_\downarrow} \hat{\tau}_j \right)$$

from $\hat{\boldsymbol{V}}/n$, leading to $\hat{V}_T$.

The rule becomes: collapse nodes $\uparrow$ and $\downarrow$ if

$$\frac{|T|}{\sqrt{\hat{V}_T}} < z_{1-\alpha/2} \tag{2.4.3}$$

holds, that is, the threshold for collapsing is $t_c = z_{1-\alpha/2}\sqrt{\hat{V}_T}$.

Note that instead of using Kendall's $\tau$ in (2.4.1) or (2.4.2), other measures of association can be used, for instance Spearman's $\rho$, Hoeffding's $\mathbb{D}$ statistic, etc. Should a vector of estimated Spearman's $\rho$s or Hoeffding's $\mathbb{D}$ statistics be normally distributed (at least asymptotically), then a meaningful threshold for collapsing could be found using the same reasoning as above.

Another strategy to decide if two successive nodes should be collapsed or not is to break down both structures before and after collapsing of two given nodes into their respective set of trivariate pieces (refer to Chapter 1). Since the structures before and after collapsing are different, some of the trivariate pieces will be different as well. Let $P$ be the set of trivariate pieces before collapsing and $P_+$ the set of pieces after collapsing. Let $P_\Delta$ be the set of vectors of the form $(U_i, U_j, U_k)$ such that the tree structure spanned on $(U_i, U_j, U_k)$ is not the same in $P$ and $P_+$. In $P_+$, the tree structure spanned on $(U_i, U_j, U_k)$ is a 3-fan, while in $P$, it is not the case. The decision to collapse or not is made by performing the hypothesis test developed in Subsection 2.3.1, since this test precisely aims at deciding whether or not the tree structure spanned on three random variables $(U_i, U_j, U_k)$ is a trivial tree structure.

Looking back at Figure 2.5, one can observe that, in the left structure, the tree spanned on $(U_2, U_3, U_4)$ is not a 3-fan, while in the right structure it is. If the p-value of the test is lower than or equal to a threshold $\alpha$, the nodes 234 and 34 should not be collapsed into one.

When there is more than one vector of the form $(U_i, U_j, U_k)$ in $P_\Delta$, the hypothesis test developed in Subsection 2.3.1 must be applied several times, and the end result is a set of p-values. For such cases, a rule of thumb such as the one hereafter can be used (do not collapse if the inequality holds):

$$\text{average p-value} \leq \alpha \tag{2.4.4}$$

**Estimation of the set of generators $\Psi$.** Once the target tree structure $\lambda$ has been estimated, one needs to estimate the generators. Usually, as done in Okhrin et al. (2013a) and Górecki et al. (2015), it is assumed that all these generators belong to a same known parametric family $F$. In such case, the only thing left to estimate is a set of parameters $\Theta$.

An interesting by-product of the hierarchical clustering used to build $\hat{\lambda}_b$ is a set of non-increasing measures of association from leaves to the root of $\hat{\lambda}_b$. Since the sufficient nesting condition requires the $\theta$s in $\Theta$ to decrease from

leaves to the root of the NAC tree, any monotonically increasing map $\delta$:

$$\begin{array}{l} \text{space of the chosen measure of association used to build } \hat{\lambda}_b \\ \rightarrow \text{ parameter's space for the parametric family } F \text{ chosen for the generators,} \end{array}$$

applied to these non-increasing measures of association, will lead to a set $\hat{\Theta}$ allowing to get an estimated NAC $(\hat{\lambda}, \hat{\Psi})$ meeting the sufficient nesting condition.

In Okhrin et al. (2013a), the parametric family for the generators is specified prior to the hierarchical clustering, allowing them to directly use, as if it was a measure of association, the $\theta$ parameter related to that parametric family. The by-product of the hierarchical clustering allows therefore to directly get $(\hat{\lambda}, \hat{\Psi})$ meeting the sufficient nesting condition without any extra effort. In other words, $\delta(\bullet) = \bullet$.

In Górecki et al. (2015), the measure of association used for the hierarchical clustering is Kendall's $\tau$. The main advantage of using Kendall's $\tau$ is that there is a direct, well known, relationship between Kendall's $\tau$ and $\theta$ for most parametric family of generators. That is, a possible map $\delta(\bullet)$ is easy to find. For instance, if the parametric family for the generators is assumed to be the Clayton family, a possible map between Kendall's $\tau$ and $\theta$ is

$$\delta(\tau) = \frac{2\tau}{1-\tau} = \theta. \tag{2.4.5}$$

See Hofert and Maechler (2011) or Table 2.1 for the map using other families.

Using other measures of association, such as Hoeffding's $\mathbb{D}$ statistic for instance, a suitable map $\delta$ is not known. However, as long as $\delta$ is a monotonically increasing function, $(\hat{\lambda}, \hat{\Psi})$ will be a proper copula, but the resulting estimator could be biased or with a very high mean square error.

To conclude this section, the second approach to estimate a target NAC using hierarchical clustering is summarized in Algorithm 3 hereafter. Note that the major difference between Algorithm 3 and Algorithm 3 in Górecki et al. (2015) is the collapsing step, starting at line 6 and ending at line 22 of Algorithm 3. This collapsing step is missing in Górecki et al. (2015).

**Input.** An iid sample from $(X_1, \ldots, X_d)$ such that the related copula is assumed to be a NAC, a measure of association $\bowtie$ and a distance based on $\bowtie$ (a large value for $\bowtie$ leading to a small distance), a linkage criteria $L$, a strategy $S$ for how to collapse successive nodes in the binary tree resulting from the hierarchical clustering, a generator's family $F$ and a map $\delta$.

**Output.** An estimated NAC $(\hat{\lambda}, \hat{\Psi})$ meeting the sufficient nesting condition.

Although this algorithm might seem long, its implementation in practice is straightforward.

**Algorithm 3** Estimating a NAC using hierarchical clustering.

**for all** pairs $(X_i, X_j)$ such that $i, j \in \{1, \ldots, d\}$ **do**
    get $distance(X_i, X_j)$ and store it in a distance matrix.
**end for**
using hierarchical clustering with $L$ and the distance matrix just built, get a binary tree $\hat{\lambda}_b$.
Also store the set of ordered measures of association for later use as $A$, this set being a by-product of the hierarchical clustering. Note that the cardinality of $A$ is $d - 1$.
set $\hat{\lambda}_t = \hat{\lambda}_b$.
**repeat**
    set $\hat{\lambda} = \hat{\lambda}_t$.
    let $V_r = \{v_1, \ldots, v_r\}$ be the set of internal nodes in $\hat{\lambda}_t$.
    **for all** pairs $(v_l, v_m)$ in $V_r$ such that $v_l$ is the parent of $v_m$ **do**
        try to collapse $v_l$ and $v_m$ using strategy $S$.
        store the decision, do not apply it yet.
    **end for**
    update $\hat{\lambda}_t$ by collapsing what needs to be collapsed.
    do this based on the set of decisions one gets from the loop just above.
**until** $\hat{\lambda} == \hat{\lambda}_t$.
the repeat until loop automatically stops as soon as no further collapsing of $\hat{\lambda}_t$ is possible.
next, update the set of ordered measures of association $A$ by comparing $\hat{\lambda}_b$ to $\hat{\lambda}$ as described hereafter.
**for all** sets of nodes in $\hat{\lambda}_b$ that have been collapsed into one node in $\hat{\lambda}$ **do**
    get the minimum of the related ordered measures of association in $A$.
    replace the related ordered measures of association in $A$ by this minimum.
**end for**
apply the map $\delta$ on each element of $A$ to get a set of estimated parameters $\hat{\Theta}$, which can then be used with $F$ to get $\hat{\Psi}$.
**return** $(\hat{\lambda}, \hat{\Psi})$

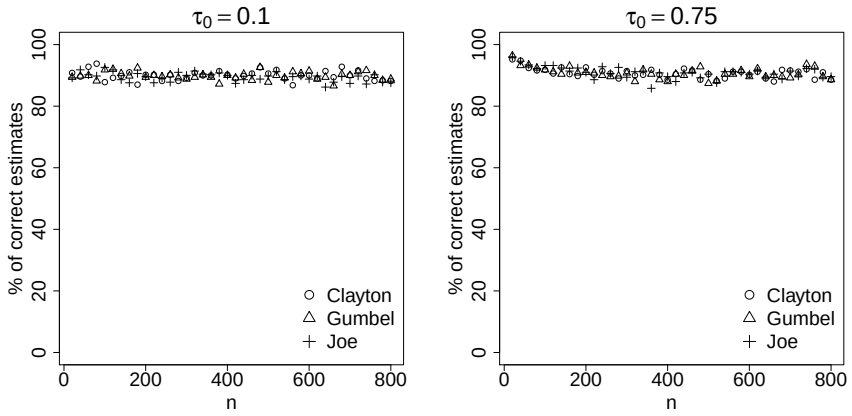



Figure 2.6: Percentage of correct estimates when the true structure is the trivial trivariate structure.

## 2.5 Performance study

This section is mainly concerned with the ability of estimating $\lambda$ based on a sample from a NAC $(\lambda, \Psi)$.

In this section, the method based on the estimation of each element of ${}^3(\lambda)$ using the hypothesis test from Subsection 2.3.1 in order to built $\hat{\lambda}$ will be referred to as the S&U method.

Subsection 2.5.1 starts by comparing the S&U method with the method from Okhrin et al. (2013a), while Subsection 2.5.2 shows the result of using Equation (2.4.3) in order to decide whether or not two succesive nodes should be collapsed. Finally, further comparisons between various methods are performed in Subsection 2.5.3.

### 2.5.1 S&U vs Okhrin et al. (2013a)

Let $(X_1, X_2, X_3)$ be a vector of random variables, the copula of which is Archimedean. Five hundred samples of size $n$ are generated from $(X_1, X_2, X_3)$ with the help of the R package **nacopula**[1] (Hofert and Maechler, 2011). With $\alpha$ arbitrarily set to 0.10, how many times among the 500 samples can one retrieve the 3-fan using the S&U method? Figure 2.6 shows the percentage of correct estimates for various values of $n$, various generator families and two different values of the related parameter $\theta$, expressed as Kendall's $\tau$ coefficient for convenience according to Table 2.1.

As expected, the percentage of correct estimates does not converge towards

[1]Please note that this package has since been merged with the **copula** package.

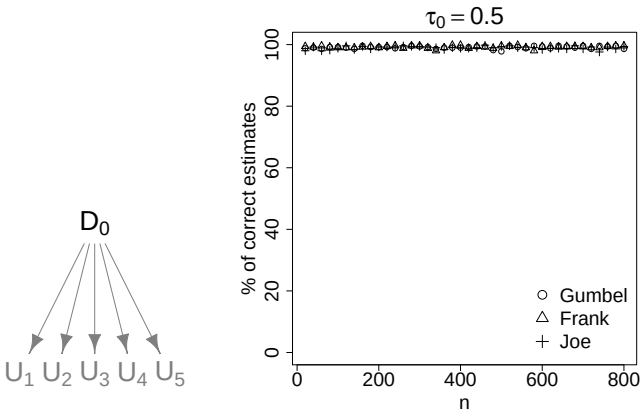



Figure 2.7: Percentage of correct estimates for the trivial five-variate case.

100% but oscillates around 90%, that is, $(1-\alpha)\times 100\%$. Should a value of 2.5% or 0.1% be used for $\alpha$ for all $n$, then the percentage of correct estimates would likewise oscillate around 97.5% or 99.9% respectively.

If samples are generated from a 5-fan structure (left of Figure 2.7), with $\tau_0$ arbitrarily set to 0.5 for all tested generator families and the same arbitrary value of $\alpha$ as before, the same lack of consistency can be observed for the S&U method, as shown on the right of Figure 2.7. Notice that the percentage of correct estimates in this case is near 100%, even though $\alpha = 0.1$. This excellent performance can be explained by the way faulty structures are handled: recall that the strategy is to decrease $\alpha$ until a valid structure emerges. At worst, $\alpha$ is decreased to 0% and the estimated structure is then the trivial five-variate structure which happens to be the target structure in this case, meaning this rule of lowering $\alpha$ not only ensures that $\hat{\lambda}$ is always a proper tree but also improves the performance of the estimator for this particular case.

Finally, if samples are generated from the structure on the left of Figure 2.8, with $\tau_{1234} = 0.3$ and $\tau_{34} = 0.7$ for all tested generator families (note that the same generator family is always used across all nodes of a given structure in this performance section), a lack of consistency can again be observed, as the percentage of correct estimates eventually oscillates around 97% (right-hand side of Figure 2.8). Since two of the trivariate components of this structure are 3-fans, namely the structure of $(U_1, U_2, U_3)$ and the structure of $(U_1, U_2, U_4)$, this lack of consistency was, again, expected.

In all these cases, it seems consistency can be achieved only if one lets $\alpha$ tend to 0 as $n$ increases, in order to ensure type I errors are asymptotically impossible.

In order to estimate a NAC structure, Okhrin et al. (2013a) advise to use what they call the binary aggregated grouping with recursive estimation

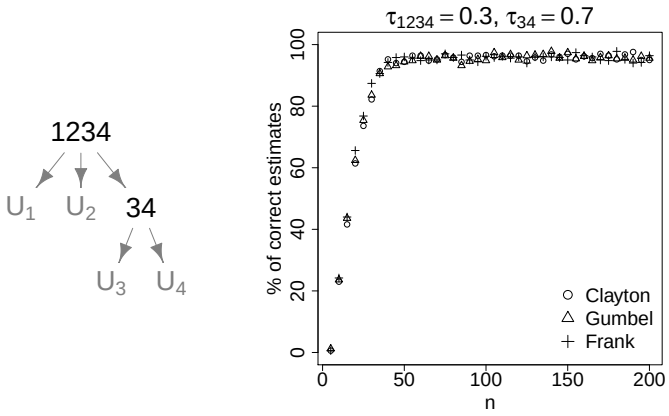



Figure 2.8: Percentage of correct estimates for a four-variate case.

method, or RML method in short. In a nutshell, this method consists in building a fully nested tree from bottom to top and then to aggregate some of the nodes of the resulting tree according to some criterion, so that the final estimated structure can possibly be something else than a fully nested tree. This method is a particular case of the second approach developed in Section 2.4.

To apply the RML method throughout this performance section, the function **estimate.copula** of the R package HAC (Okhrin and Ristig, 2014) has been used, this package being related to the work of Okhrin et al. (2013a). Since only the Clayton and Gumbel generator families are currently implemented in the HAC package, assessment of the RML method performance for other generator families is not possible at the time of writing.

For the aggregation step in their approach, Okhrin et al. (2013a) suggest several criteria. The one used in this performance section, the only one currently implemented in the HAC package, is the following: for any two successives nodes with estimated parameters $\hat{\theta}_I$ and $\hat{\theta}_J$ in the structure, the nodes have to be aggregated if $|\hat{\theta}_I - \hat{\theta}_J| < \epsilon$, where $\epsilon$ has to be chosen by the user.

With a value for $\epsilon$ arbitrarily set to 0.30 for all $n$, Figure 2.9 displays the performance of their method for the estimation of a 3-fan.

Increasing the value of $\epsilon$ for all $n$ typically improves the performance of their estimator in the case of a 3-fan, as it increases the chances of aggregation. Lowering the value of $\epsilon$ typically deteriorates the performance of their estimator for this case. At the limit, with $\epsilon$ set to 0.00, no aggregation is done at all, and their estimator is unable to estimate correctly the trivial trivariate structure studied here. These remarks hold if the samples are generated from a 5-fan.

The case of the structure on the left of Figure 2.8 is a little more complex: their estimator turned out to be able to consistently estimate this structure when $\epsilon$ is set to 0.15, 0.30 or 0.60, but not for a value of $\epsilon$ set to 5.00 for

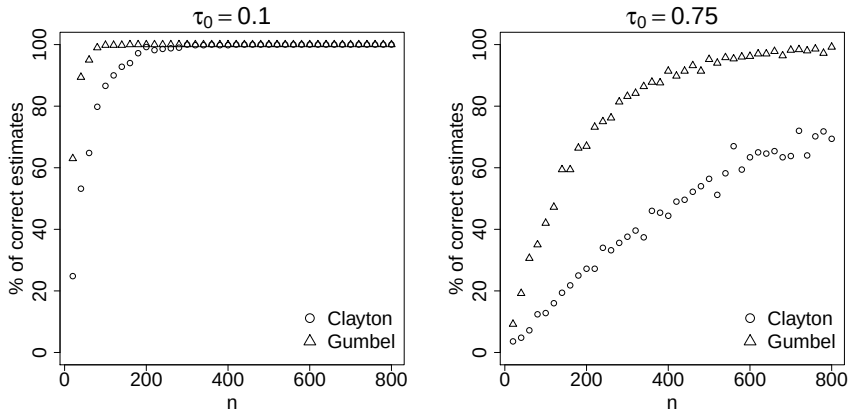



Figure 2.9: Performance of the RML method by Okhrin et al. (2013a) for a 3-fan, with $\epsilon = 0.30$ as threshold for aggregation.

instance.

What happens with the S&U method and the RML method from Okhrin et al. (2013a) in case the target structure is a non-trivial trivariate structure (a triple) or a structure made up only of triples?

Given 500 samples of size $n$ from a triple such as the one in the left of Figure 2.1 and $\alpha = 0.10$, how many times among the 500 samples is the S&U method able to retrieve this triple? Figure 2.10 shows the percentage of correct estimates for various values of $n$ and various generator families (again, note that the same generator family is always used across all nodes of a given structure in this performance section). The parameters $\theta_0$ (root node, labelled 123 or $D_0$) and $\theta_{23}$ (the other branching node, labelled 23 or $D_{23}$) are expressed as Kendall's $\tau$ coefficients for convenience.

As the sample size increases, there is a clear convergence towards 100% of correct estimates. The more apart $\tau_0$ and $\tau_{23}$, the faster the convergence towards 100% of correct estimates (compare the two horizontal axes in Figure 2.10). These results strongly suggest that the S&U estimator, at least when $\alpha = 10\%$, is a consistent estimator for any non-trivial trivariate NAC structure and thus for any larger NAC structure made up only of triples. Indeed, if the samples are generated from the seven-variate structure such as the one on the left of Figure 2.11, with $\tau_0 = 0.1$, $\tau_{123} = 0.3$, $\tau_{23} = 0.6$, $\tau_{4567} = 0.3$, $\tau_{567} = 0.5$, and $\tau_{67} = 0.8$ for all tested generator families, the proportion of correct estimates grows to 100% as $n$ increases (right-hand side of Figure 2.11).

Increasing the value of $\alpha$ for all $n$ actually further improves the performance of the S&U estimator for both NAC structures, the best performance possible being delivered when $\alpha$ is set to its upper limit, that is, $\alpha = 100\%$. If $\alpha$ is set to its lower limit, that is $\alpha = 0\%$, the estimator becomes unable to correctly

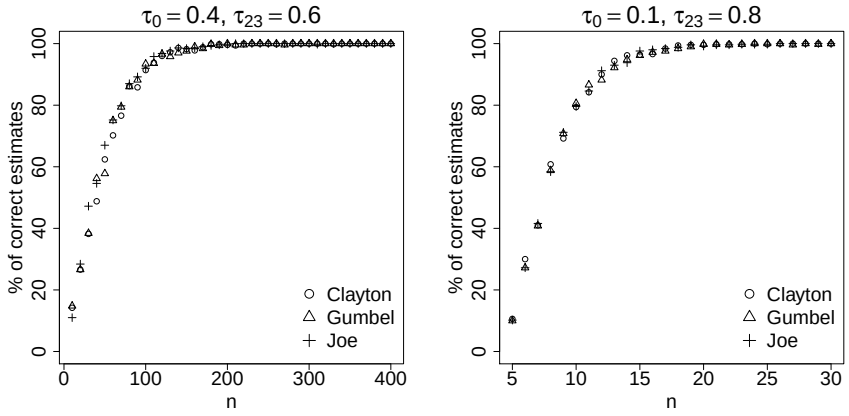



Figure 2.10: Percentage of correct estimates when $d = 3$ and true structure is $\lambda_{23}$.

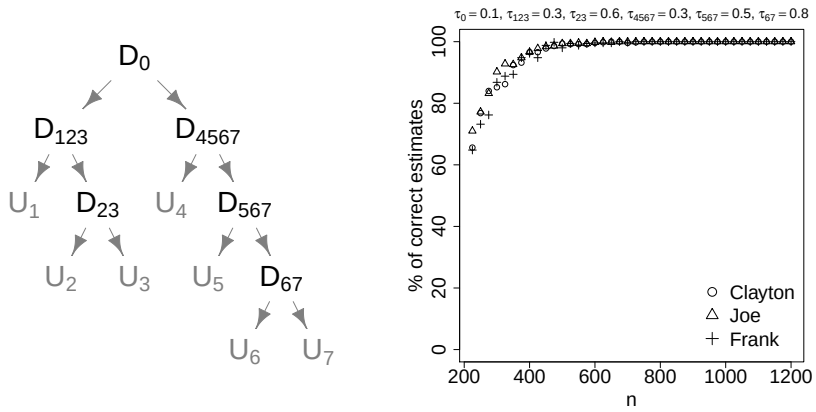



Figure 2.11: Percentage of correct estimates (right) for a seven-variate structure (shown on the left).

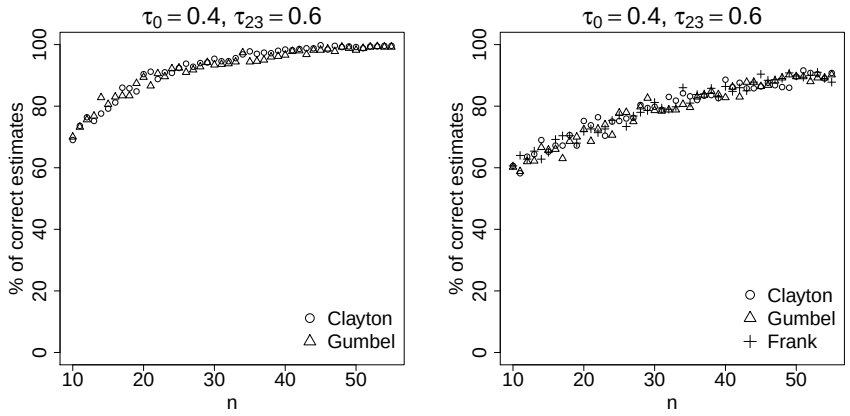



Figure 2.12: Percentage of correct estimates when $d = 3$ and the true structure is $\lambda_{23}$. Left is the RML method for Clayton and Gumbel generators, right is the method described in this paper for Clayton, Gumbel and Frank generators, the latter generator being an arbitrary choice. Both methods were pushed to their respective limit in order to deliver the best performance possible for structure $\lambda_{23}$.

estimate any of the two structures studied here.

When the target structure is a triple, the percentage of correct estimates was also found to converges towards 100% by using the RML method from Okhrin et al. (2013a), provided the value of $\epsilon$ is small enough. In fact, as any aggregation should be avoided in case the target structure is a triple, the performance of their estimator typically improves by lowering $\epsilon$ for all $n$, the best performance being delivered when $\epsilon = 0$, that is, when the aggregation step is completely skipped. Should the value of $\epsilon$ be too large, then their estimator will fail to be a consistent estimator for the non-trivial trivariate structure.

In case the target structure is a triple, Figure 2.12 allows for a direct comparison between the S&U method and the RML method when both are pushed to their favorable respective limit, thus with $\alpha = 100\%$ for the S&U method and with $\epsilon = 0$ for the method of Okhrin et al. (2013a).

There is a performance gap between the two methods. Recall however that the RML method was applied with the prior knowledge that the generators were Clayton and Gumbel generators while the S&U method does not require such prior knowledge.

### 2.5.2 Equation (2.4.3) to collapse nodes

Let $(X_1, X_2, X_3)$ be a vector of random variables, the copula of which is Archimedean. Five hundred samples of size $n$ are generated from $(X_1, X_2, X_3)$, that is, the setup is here the same as in the beginning of the previous subsection. Using Equation (2.4.3), the null hypothesis assumes a trivial tree structure for

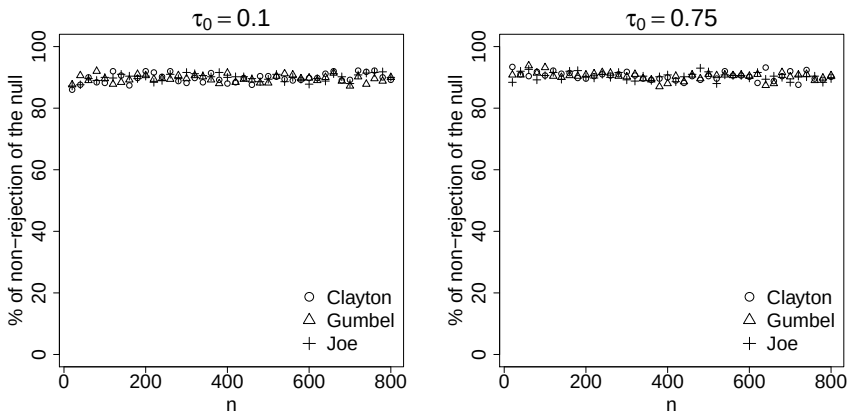



Figure 2.13: Failure to reject the null when the true structure is a trivial trivariate structure (that is, the null is true) as a function of the sample size.

the data (which is the true one in this case), while the alternative hypothesis assumes that, say, the tree structure is a structure of the form 1(23), where $U_2$ and $U_3$ are clustered together, while $U_1$ is left apart, also see the left of Figure 2.1. Does one successfully fail to reject the trivial tree structure $1-\alpha$ % of the time? The simulation results displayed in Figure 2.13 where conducted with $\alpha = 0.10$, various values of $n$, various generator families and two different values of the related parameter $\theta$, expressed as Kendall's $\tau$ coefficient for convenience according to Table 2.1.

Given 500 samples of size $n$ from a triple such as the one in the left of Figure 2.1 and the same $\alpha$ as before, how many times among the 500 samples does Equation (2.4.3) lead to rejection of the null? Figure 2.14 shows the percentage of rejection of the null for various values of $n$ and various generator families.

As the sample size increases, there is a clear convergence towards 100 %. The more apart $\tau_0$ and $\tau_{23}$, the faster the convergence towards a percentage of rejection equal to 100 % (compare the two horizontal axes in Figure 2.14).

These results from Figure 2.14 can be compared with those in Figure 2.10, and one can notice the approaches in Figure 2.10 and Figure 2.14 lead to similar results. In both cases, increasing the value of $\alpha$ will lead to a higher rejection rate. A notable difference between the tests used in Figure 2.10 and Figure 2.14 is however that the one in Figure 2.10 relies on bootstrap to get the $H_0$ distribution of the related test statistic, while the approach based on Equation (2.4.3) does not. Another notable difference is that Equation (2.4.3) is built using Kendall's taus only, while the test statistic used in Figure 2.10 is based on a comparison of empirical Kendall distribution functions (refer to Subsection 2.3.1 for details).

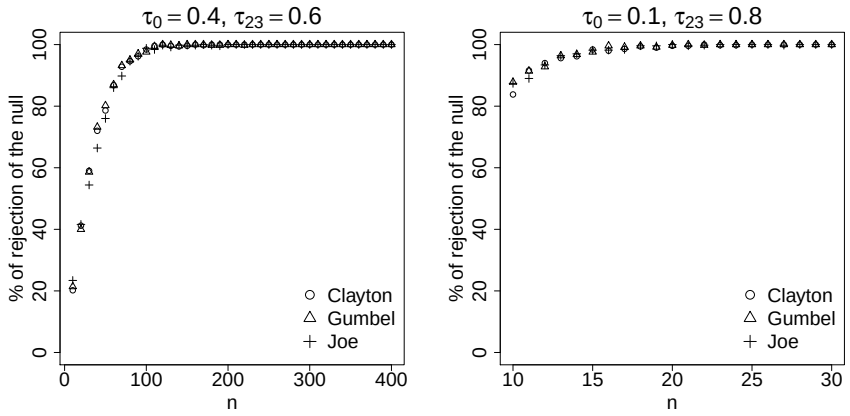



Figure 2.14: Rejection of the null as a function of $n$ when the true structure is $\lambda_{23}$.

### 2.5.3 Further comparisons between NAC tree estimators

When the sample size is $n$, the methodology used in this subsection to estimate the performance of a NAC structure estimator is described through the following steps:

- generate $N = 100$ samples of size $n$ from the NAC $(\lambda, \Psi)$;
- apply the NAC tree structure estimator on each of these $N$ samples while considering the univariate margins unknown (that is, work on ranks) and get $N$ estimates of $\lambda$;
- calculate a distance between each estimate of $\lambda$ and $\lambda$ itself;
- get the average of the $N$ resulting distances or some other descriptive measure of these distances, such as:

$$(\text{average of the distances})^2 + \text{variance of the distances}; \tag{2.5.1}$$

- the lower the average of the distances or the lower (2.5.1) is, the better the performance of the estimator for the NAC defined through $(\lambda, \Psi)$ is, given a fixed sample size $n$.

Two distances between a given estimate of $\lambda$ and $\lambda$ itself are considered: the 01-distance and the tri-distance, refer to Section 1.6.

Since the estimator S&U is based on a hypothesis test, it is required to choose a threshold $\alpha$ prior to the estimation of a target tree structure.

Regarding estimators based on hierarchical clustering (Section 2.4), the choice of a threshold to decide if any two successive nodes in the estimated binary tree structure should be collapsed is also required prior to the estimation of a target tree structure, as seen in Equations (2.4.4) or (2.4.1).

Comparison of the performance of different tree structure estimators is

therefore a challenge, as the performance of a given estimator depends on the chosen threshold for that estimator. Given a sample size $n$, a target NAC ($\lambda$, $\Psi$) and a tree structure estimator, the suggestion is to use a threshold ensuring that $P(\hat{\lambda}_n = \lambda)$ is maximized. This particular threshold will be called the optimal threshold. By making use of the related optimal threshold for each estimator, one only takes into account the best performance of each estimator, which should allow for "fair" comparisons between estimators.

In the particular case where $\lambda$ is a binary structure,

- the estimator S&U, which is based on $\binom{d}{3}$ hypothesis tests, should always reject the nulls. If one null is not rejected, it means the final estimated structure contains at least a 3-fan, and therefore the estimate of $\lambda$ cannot be equal to $\lambda$ itself. Thus the threshold $\alpha$ should be set to 100% or more so that all nulls are always rejected.
- Regarding estimators based on hierarchical clustering or supertree methods, in case $\lambda$ is a binary structure, $P(\hat{\lambda}_n = \lambda)$ is maximized if the collapsing step is skipped. This can be achieved by setting $\alpha$ to 100% or more in (2.4.4), and $\tau_c$ to 0 or less in (2.4.1).

Although unknown when $\lambda$ is not a binary structure, the optimal threshold for an estimator can be estimated. Indeed, given $N = 100$ samples of size $n$, a target NAC ($\lambda$, $\Psi$) and a tree structure estimator, the estimated optimal threshold is the value such that the average of the $N = 100$ 01-distances between each estimate of $\lambda$ and $\lambda$ itself, is minimal. Note that on real data, since the target structure is by definition unknown, the optimal threshold cannot be estimated as it is done here.

In this subsection, all tree structure estimators but the S&U estimator rely on two steps: a binary tree is first built, and then parts of it are collapsed. For the first step, RNix and NJNNI refer to the two supertree methods described in Section 2.3 while kind, hD and kt refer to distance matrices related to measures of deviation from the independent bivariate Kendall distribution, Hoeffding's $\mathbb{D}$ statistics or Kendall correlation coefficients, respectively (a large measure of deviation, a large $\mathbb{D}$ or a large Kendall's $\tau$ leading to a small distance). For the second step, kb and kagg refer to Equations (2.4.4) and (2.4.1), respectively.

Target structures are given in Figures 2.15 and 2.16.

Figure 2.17 through 2.21 give the simulation results for these target structures. The generators used across each structure and the related parameters, expressed as Kendall correlation coefficients for convenience, are specified below the figures.

Notice the average of the 01-distances is actually equal to the percentage of estimates that are unequal to the target structure, so that a value of 1 means not a single estimate of $\lambda$ among the $N = 100$ available was equal to the target $\lambda$, while a value of 0 means all estimates of $\lambda$ were equal to $\lambda$.

Some comments are:

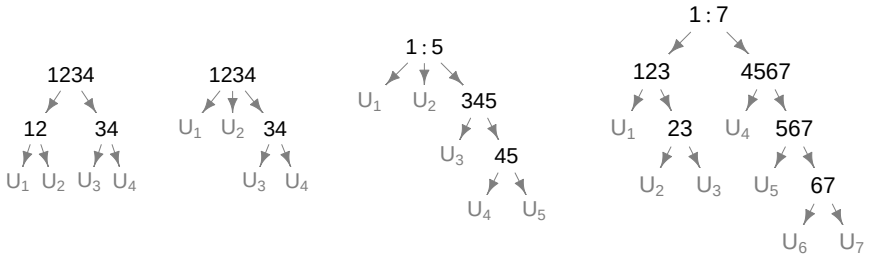



Figure 2.15: two four-variate structures, a five-variate structure and a seven-variate structure. 1:5 means 12345, while 1:7 means 1234567.

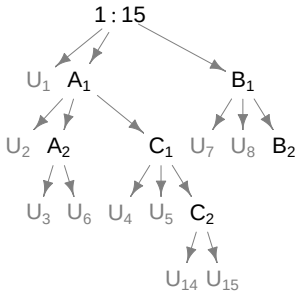



Figure 2.16: A fifteen-variate structure, where $B_2$ refer to an Archimedean copula spanned on $U_9$ through $U_{13}$.

- As the sample size increases, all estimators perform better.
- The more the target NAC is resolved, that is, the more generators of successive nodes in the target structure are different, the better the performance, as seen in Figure 2.17 through 2.20. The tree structure of a poorly resolved NAC is, in general, harder to estimate.
- The group of estimators used in this section are clearly not a homogeneous group regarding performances. For instance, at the top left of Figure 2.17, one can see that the NJNNI_kb estimator performs slightly better than the S&U estimator in terms of mean 01-distance, while the kt_kagg and hD_kagg estimators both significantly perform better.
- Several estimators beat the S&U estimator by a large amount in case of binary target structures (Figure 2.17 and 2.20). For instance, top right of Figure 2.20, the kt_kagg estimator can be seen to get the target structure wrong only 20% of the time when the sample size is 30, while the S&U estimator gets it wrong more than 80% of the time on the same samples. Moreover, the bottom right part of the same figure shows that, when the kt_kagg gets the target structure wrong, it is only by a very small, almost negligible, amount. When the S&U estimator gets it wrong, the resulting estimate of $\lambda$ is usually far away from $\lambda$ itself.
- As seen in Figure 2.21, the kind_kagg and kt_kagg estimators seem to offer the same ability to retrieve the target structure. Note however that the kt_kagg estimator is easier to compute.
- No matter what the generators of the target NAC are, notice that kt_kagg remains the top runner in almost all figures where it is used.
- In the simulations for the four-variate binary structure, the kt_kagg estimator turned out to be the one with the smallest execution times, producing estimates up to a 100 times faster than the S&U estimator. For many target NACs, estimators using hierarchical clustering exhibit smaller execution times than the S&U estimator on the same data.

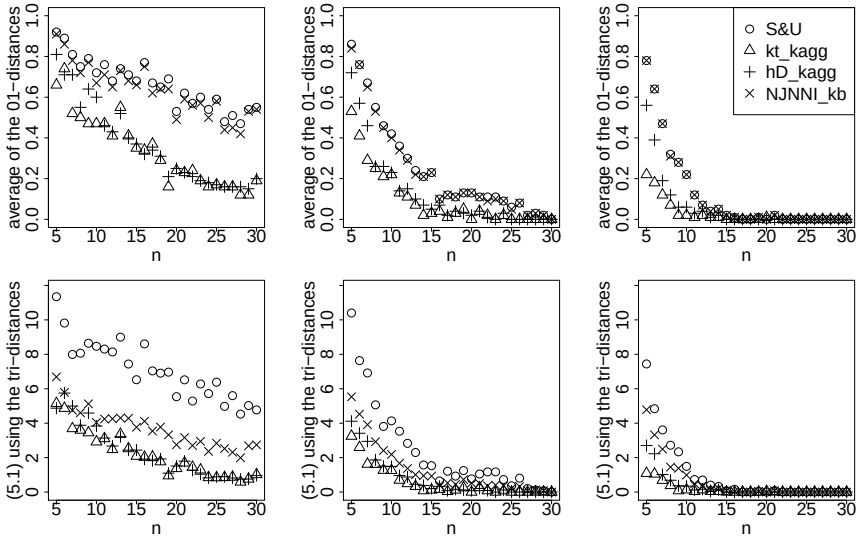



Figure 2.17: results for the four-variate binary structure. The generators used across the structure are all Clayton generators. The related sets of parameters are ($\tau_{1234} = 0.4, \tau_{12} = 0.6, \tau_{34} = 0.6$) for the left-hand side of the figure, ($\tau_{1234} = 0.3, \tau_{12} = 0.7, \tau_{34} = 0.7$) in the middle, and ($\tau_{1234} = 0.2, \tau_{12} = 0.8, \tau_{34} = 0.8$) for the right-hand side of the figure.

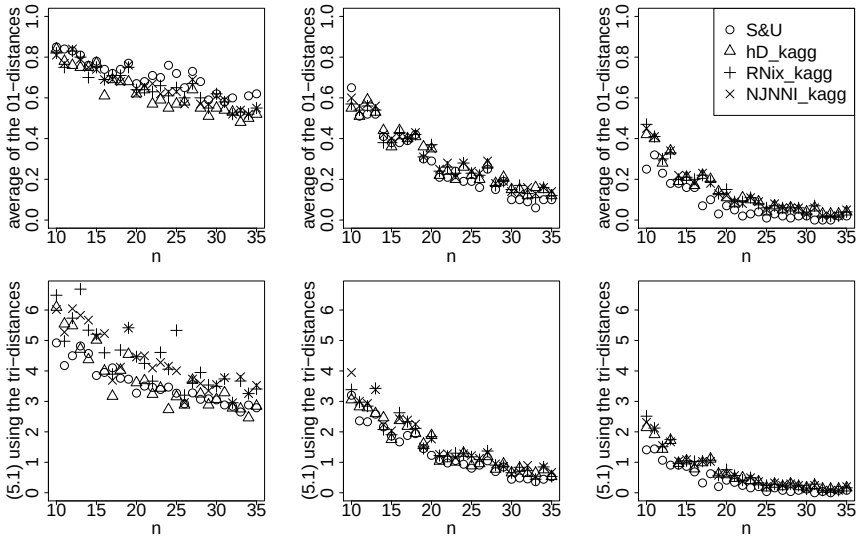



Figure 2.18: results for the second four-variate structure. The generators used across the structure are, again, all Clayton generators. The related sets of parameters are ($\tau_{1234} = 0.4, \tau_{34} = 0.6$) for the hand-left side of the figure, ($\tau_{1234} = 0.3, \tau_{34} = 0.7$) in the middle, and ($\tau_{1234} = 0.2, \tau_{34} = 0.8$) for the hand-right side of the figure.

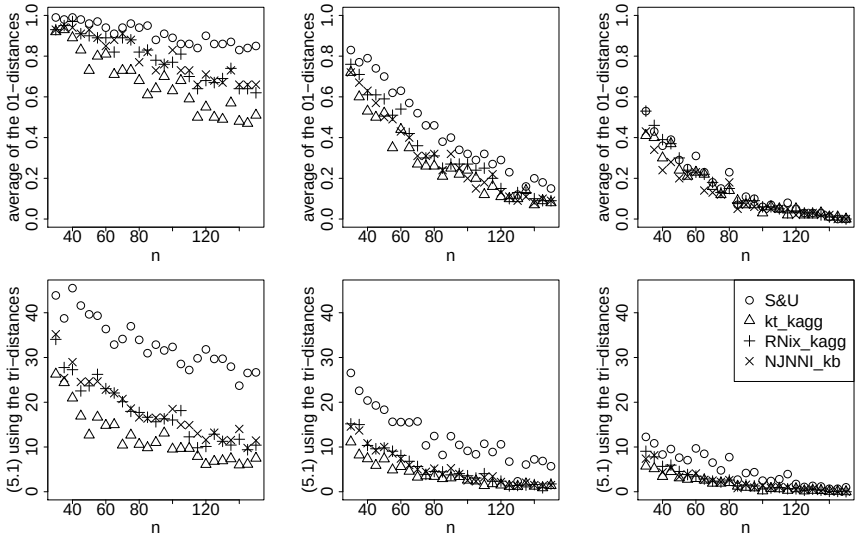



Figure 2.19: results for the five-variate structure. The generators used across the structure are all Gumbel generators. The related sets of parameters are $(\tau_{1:5} = 0.4, \tau_{345} = 0.5, \tau_{34} = 0.6)$ for the left-hand side of the figure, $(\tau_{1:5} = 0.3, \tau_{345} = 0.5, \tau_{34} = 0.7)$ in the middle, and $(\tau_{1:5} = 0.2, \tau_{345} = 0.5, \tau_{34} = 0.8)$ for the right-hand side of the figure.

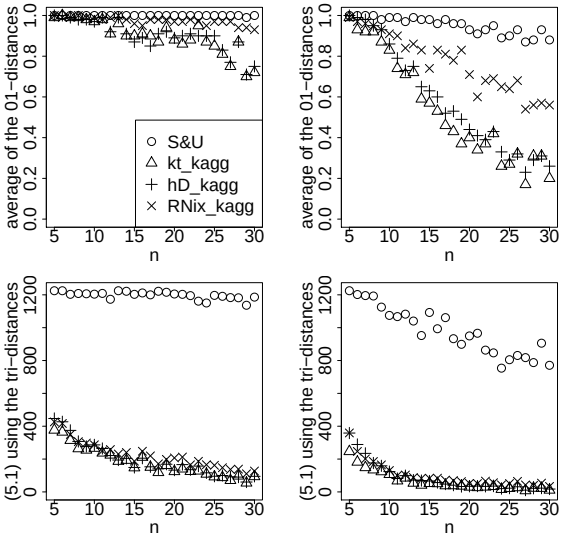



Figure 2.20: results for the seven-variate structure. The generators used across the structure are all Frank generators. The related sets of parameters are $(\tau_{1:7} = 0.35, \tau_{123} = 0.5, \tau_{23} = 0.65, \tau_{4:7} = 0.45, \tau_{567} = 0.55, \tau_{67} = 0.65)$ for the left-hand side of the figure and $(\tau_{1:7} = 0.2, \tau_{123} = 0.5, \tau_{23} = 0.8, \tau_{4:7} = 0.4, \tau_{567} = 0.6, \tau_{67} = 0.8)$ for the right-hand side of the figure.

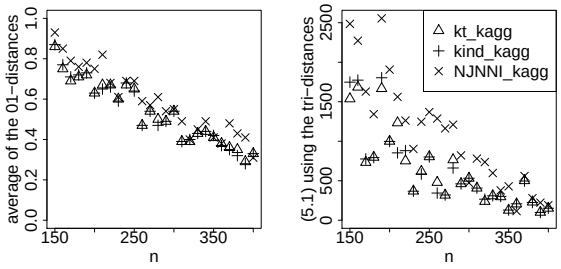



Figure 2.21: results for the fifteen-variate structure. The generators used across the structure are all Joe generators. The set of parameters is ($\tau_{1:15} = 0.1, \tau_{A_1} = 0.25, \tau_{A_2} = 0.5, \tau_{B_1} = 0.5, \tau_{B_2} = 0.75, \tau_{C_1} = 0.35, \tau_{C_2} = 0.45$).

## 2.6 Daily log returns

Daily log returns from January 2010 to December 2012 of the following indices were gathered with the help of Yahoo! Finance:

- Abercrombie & Fitch Co. (ANF), traded in New York;
- Amazon.com Inc. (AMZN), traded in New York;
- China Mobile Limited (ChM), traded in Hong Kong;
- PetroChina (PCh), traded in Hong Kong;
- Groupe Bruxelles Lambert (GBLB), traded in Brussels;
- and KBC Group (KBC), traded in Brussels.

For each of these six time series, a GARCH(1,1) model with generalized error distribution was fitted and the residuals were extracted, that is, each of the six original time series was divided by the related estimated standard deviations, leading to a table of $n = 740$ observations and $d = 6$ columns. These new six time series will be called the GARCH(1,1)-standardized log returns. A Ljung-Box test (lag 20) on each of the six GARCH(1,1)-standardized log returns was performed, as well as on each of the six squared GARCH(1,1)-standardized log returns, and failed to reject the null hypothesis of zero serial correlation every time. A chi-squared test was also performed on each of the six GARCH(1,1)-standardized log returns to check if the generalized error distribution assumed for the residuals in the GARCH(1,1) model is appropriate. Again, $H_0$ was never rejected.

Figure 2.22 shows the estimated NAC structure, using the S&U estimator, based on the GARCH(1,1)-standardized log returns of ANF, AMZN, ChM and PCh, the estimated NAC structure for ANF, AMZN, GBLB and KBC, and the estimated NAC structure for ChM, PCh, GBLB and KBC.

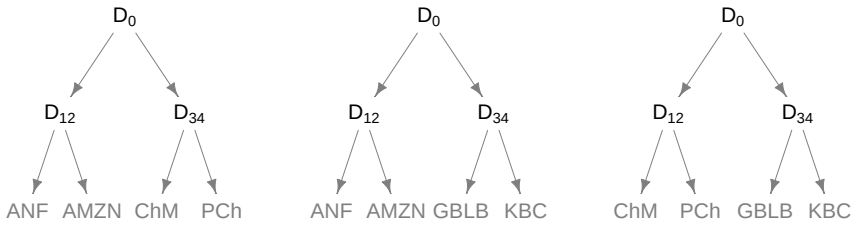



Figure 2.22: Given two series of GARCH(1,1)-standardized log returns from one geographical area and two from another area, a natural clustering by area arises. The above structures are all strongly supported by the data, as the 12 related p-values are less than 10e-04.

In order to build a six-variate structure, eight extra trivariate structures must be estimated. The left of Figure 2.23 shows a reasonable guess for the six-variate structure in which the eight extra trivariate structures all are 3-fans.

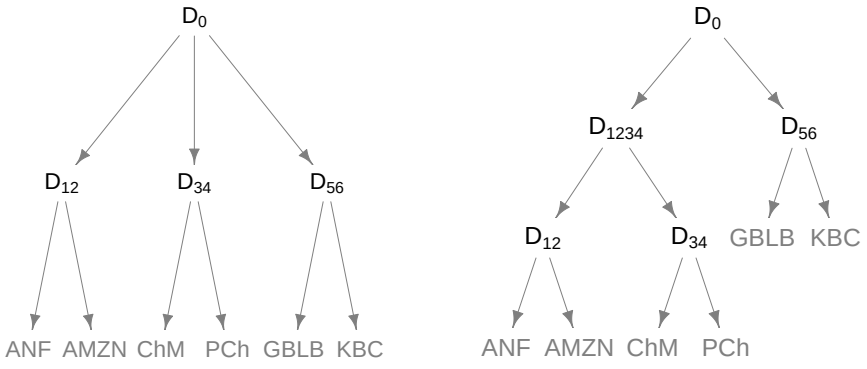



Figure 2.23: Possible six-variate structures for the data.

However, the 3-fan in four of the eight extra trivariate structures is strongly rejected by the data, which rather suggest the structure on the right of Figure 2.23. Unfortunately, this last structure implies rejection of the 3-fan for all eight extra trivariate structures and not only for half of them, making the estimation of a six-variate structure quite uncertain. Since both PetroChina and China Mobile are not only traded in Hong Kong but also in New York, their log returns in Hong Kong is expected to be more related to the log returns of some companies in New York (for instance ANF and AMZN) than to the log returns of two companies in Belgium. The structure on the right of Figure 2.23 seems therefore more appropriate.

In general, difficulties can appear when the S&U estimator is applied to real data for which the true copula is not a NAC. For instance, one can end up with a subset of estimated triples each strongly supported by the data (that is, very small p-values, meaning type I or type III errors are unlikely) and yet $\widehat{{}^3(\lambda)}$ is a faulty set. Such faulty sets cannot be fixed because they do not result from estimation errors, well from the fact that the target copula is actually not a NAC.

## 2.7 The Belgian stock market, 2006-2010

Using Yahoo! Finance, the weekly logreturns between mid 2006 and mid 2010 for $d = 7$ arbitrary components from the BEL20 index have been extracted.

The components are:

- Ackermans & van Haaren (Ack) - diversified industrials,
- Ageas - life insurance,
- Bekaert (Bek) - diversified industrials,
- Cofinimmo (Cof) - industrial and office REITs (real estate),
- KBC Group (KBC) - banks,
- Solvay - specialty chemicals,
- Umicore (Umi) - specialty chemicals.

The weekly logreturn of component $i$ at the $t$-th week (week 1 being the last week of June 2006) is

$$X_i(t) = \log\left(\frac{R_i(t+1)}{R_i(t)}\right),$$

where $R_i(t)$ stands for the raw price the $t$-th week.

Using a moving window covering a two-year period, a tree structure was estimated every every year between 2006 and 2010, that is, a tree structure was estimated on the weekly logreturns for the following periods:

- mid 2006 to mid 2008,
- mid 2007 to mid 2009,
- mid 2008 to mid 2010.

The estimator used is a special case of Algorithm 3, where a binary tree is first built based on the matrix of Kendall's taus and then collapsed using the hypothesis test developed in Subsection 2.3.1, also see Section 2.4 for details. Note: the number of observations used to build each tree is about 104, as a result of the two-year large moving window. The estimated trees are displayed in Figure 2.24, where the indicated year on each tree corresponds to the middle of the two-year large window.

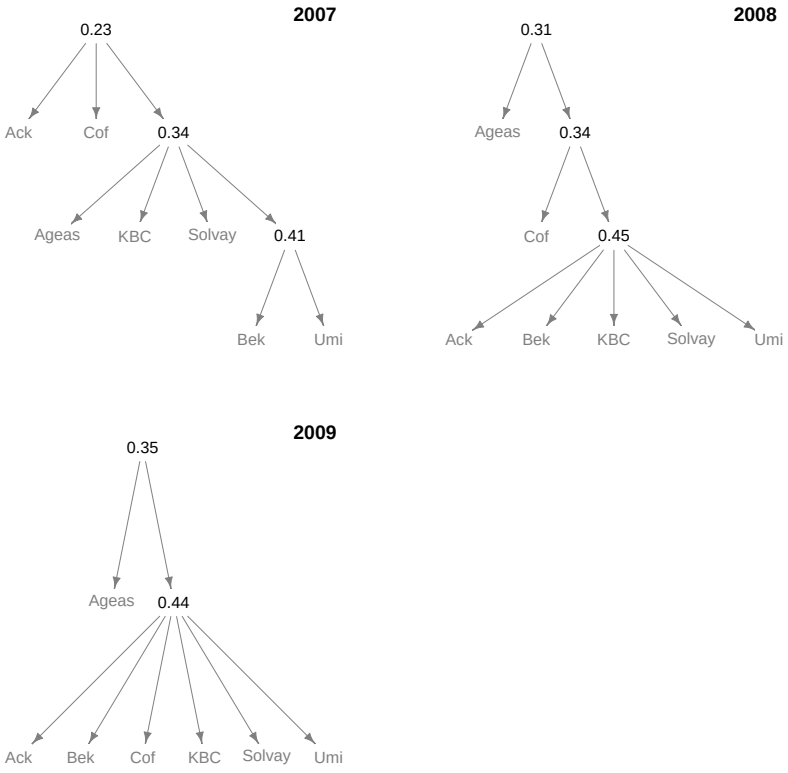



Figure 2.24: The dependence structure between 7 components of the BEL20 through time.

To help interpret the estimated structures, an estimated summary of the generator at each internal node of a given structure has been calculated. Given a node and a structure, the estimated Kendall correlation coefficients of all the pairs of random variables interacting through that node are averaged. So, for instance, the value 0.34 in the estimated tree for 2007 is the result of averaging the Kendall's correlation coefficients of the following pairs: (Solvay, Bek), (Solvay, Umi), (KBC, Bek), (KBC, Umi), (Ageas, Bek), (Ageas, Umi).

A few comments one can make are:

- Cof becomes increasingly related to other components through time, going down the three remarkably from 2007 to 2009.
- Bek and Umi remain related through a rather strong correlation through time, the minimum being 0.41 in 2007. KBC and Solvay also enjoy a rather strong correlation through time.
- The position of Ageas is stable in the tree, related to most components

throuhg a correlation of $\sim 0.35$.

- The minimum dependence (root node) in the tree increases as one goes from 2007 to 2009, while the maximum dependence remains stable.

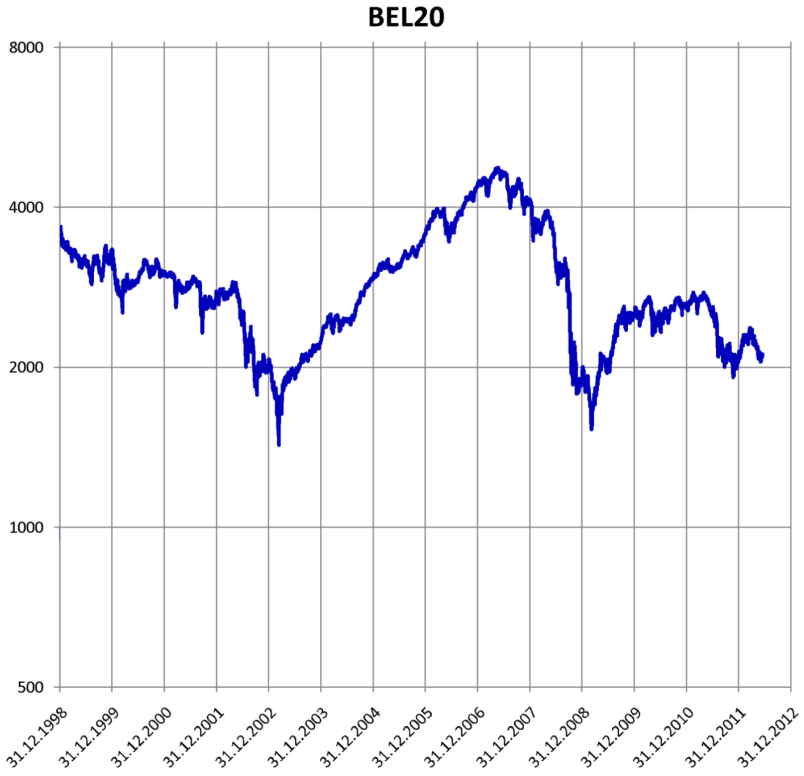



Figure 2.25: The BEL20 index through time.

## 2.8 The story of 482 students

In this section, a NAC tree is estimated based on the results of 482 students to their exams, using the kt_kagg tree structure estimator with $\tau_c$ set to 0.075. The main reason the kt_kagg tree structure estimator is used rather than another is because it was the best performing one in Section 2.5.3, as well as the fastest.

The 482 students are in their first year, studying economics and management in a Belgian university. They took 14 exams, that is, there are 14 random variables: Private Law, Psychology and Management, Sociology, Chemistry, Geography, English, Dutch, History, Mathematics, Physics, Statistics, a

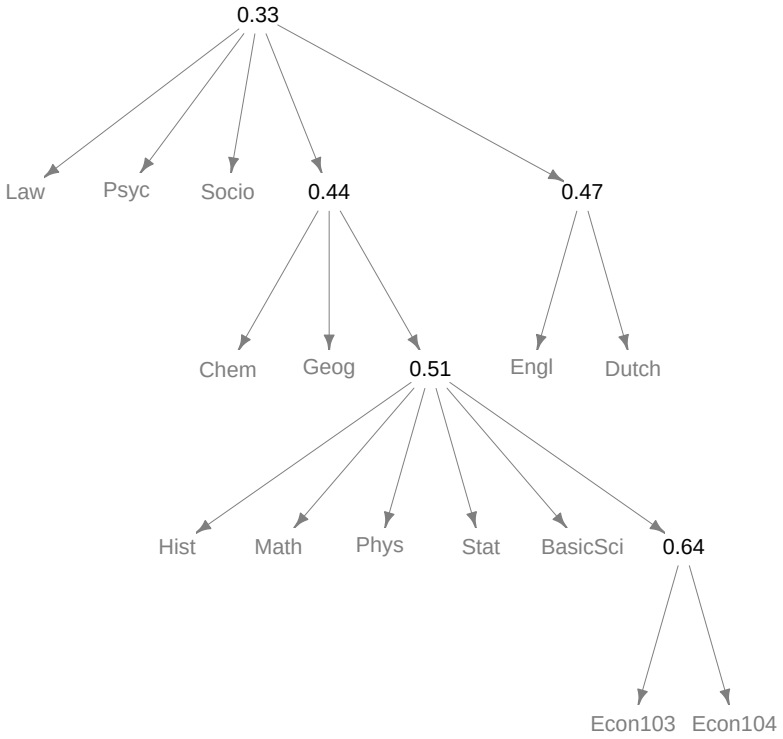



Figure 2.26: NAC tree built on the grades of 482 students.

course designed to make sure students have the required prerequisites in sciences (BasicSci), and, finally, micro and macro economics (Econ103) and a related course (Econ104). Figure 2.26 shows the estimated tree.

To help interpret the estimated structure, an estimated summary of the generator at each internal node of the structure are obtained by averaging the estimated Kendall correlation coefficients of all the pairs of random variables interacting through that node. For instance, 0.51 was obtained by averaging the Kendall correlation coefficients of the pairs (Hist, Math), (Hist, Phys), (Hist, Stat), (Hist, BasicSci), (Hist, Econ103), (Hist, Econ104), (Math, Phys), (Math, Stat), (Math, BasicSci), (Math, Econ103), (Math, Econ104), (Phys, Stat), (Phys, BasicSci), (Phys, Econ103), (Phys, Econ104), (Stat, BasicSci), (Stat, Econ103), (Stat, Econ104), (BasicSci, Econ103) and (BasicSci, Econ104).

The estimated average Kendall correlation coefficient at the root is 0.33, suggesting that students tend to have good grades everywhere or bad grades everywhere, and less often a mix of good and bad grades. The strongest estimated Kendall's $\tau$ can be observed between the courses Econ103 and Econ104. Merging both courses in one exam could be a time-saving idea for the teachers in charge. English and Dutch courses are apart from the rest of the tree, with an estimated Kendall's $\tau$ of 0.47. Natural sciences such as Mathematics, Physics or Statistics are related through a rather strong estimated mean Kendall's $\tau$ (0.51), while courses such as Psychology or Sociology are both directly connected to the root where the dependence is the weakest.

If one is to assume that the five generators of the tree shown in Figure 2.26 all belong to a same parametric family $F$, say the Clayton family, estimation of the related set of parameter $\Theta$ can be easily performed by inverting the mean Kendall correlation coefficients displayed in Figure 2.26 according to Equation (2.4.5).

To check the fit, one could compare the empirical correlation matrix, built on the raw data, versus the correlation matrix one gets from $(\hat{\lambda}, \hat{\Psi})$. A good match between the two matrices does not mean that the true copula underlying the dataset is a NAC, but is nonetheless a key requirement.

Since comparison of two $14 \times 14$ correlation matrices is quite obnoxious to perform, suggestion is to use principal component analysis to help perform this check (Wold et al., 1987; Jolliffe, 2002; Abdi and Williams, 2010). The 482 students, projected on the first two components, can be seen in the top left of Figure 2.27. 5000 points from $(\hat{\lambda}, \hat{\Psi})$ can be seen in the top right of Figure 2.27.

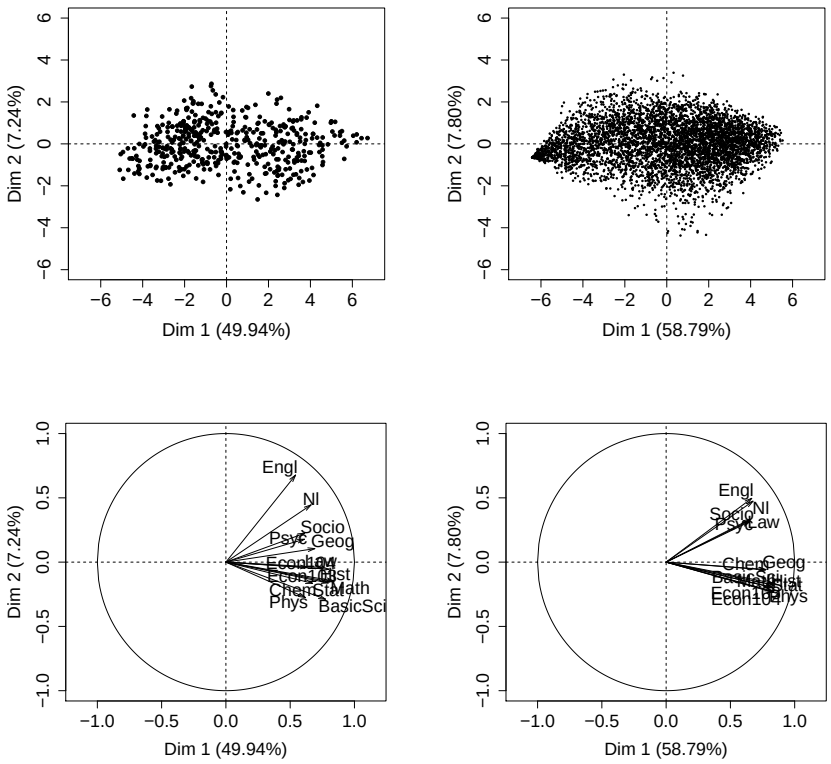



Figure 2.27: Top left: the 482 students projected on the first two components. Top right: the 5000 generated points projected on the first two components. Bottom: corresponding correlation circles.

The circle of correlations based on the 482 students can be seen in the bottom left panel of Figure 2.27. The circle of correlations based on 5000 observations from $(\hat{\lambda}, \hat{\Psi})$ can be seen in the bottom right panel of Figure 2.27. By comparing the two correlation circles, one can quickly and effortlessly check for large discrepancies between the dependence structure of the original data and the dependence structure of the data generated from $(\hat{\lambda}, \hat{\Psi})$. While there are differences, one can observe that all variables point on the right in both circles, and given a variable, such as Engl, one can see that the difference in position between this particular variable on the left correlation circle and on the right correlation circle never exceeds 20 degrees.

In most papers related to NACs, a generator with the same functional form is used throughout the tree structure. Different generators can however be used, as discussed by Hofert (2011), offering extra flexibility. In his paper,

Hofert (2011) gives several examples of different Archimedean families one can use while meeting the sufficient nesting condition. For instance, Hofert (2011) tells us that if a parent node is defined through a Clayton generator, then the child node can be defined through generator

$$\psi_h(t) = (1 + t^{1/\theta_h})^{-1},$$

as long as the parameter's value of the Clayton generator $\in (0, 1]$. Translated as a Kendall correlation coefficient, it means one cannot exceed $1/3$ for the correlation coefficient on the parent node.

Looking back at Figure 2.26, at least one opportunity for the generator $\psi_h$ can be seen: node 0.47. Indeed, the parent node has a Kendall correlation coefficient just below $1/3$. The only remaining task is to revert the value 0.47 in order to get $\hat{\theta}_h$. Straighforward calculations can show that $\theta_h = \frac{4}{6(1-\tau)}$, leading to $\hat{\theta}_h = 1.258$. The resulting estimated NAC now uses different generators across its tree structure, and, thanks to Hofert (2011), it is known that this estimated NAC is a proper copula.

Of course, one has no reason to use different generators unless the resulting NAC offers a better fit. To check if using $\psi_h(t)$ at node 0.47 allows to improve the fit, comparison of the log-likelihood of the Archimedean copula on (Engl, Dutch) using a Clayton generator first and then using $\psi_h(t)$ is performed. Straighforward calculations can show that the copula density using $\psi_h(t)$ as generator is

$$c(u, w; \theta) = \frac{2(-1 + 1/u)^{\theta}(-1 + 1/w)^{\theta}}{D},$$

where

$$D = u^2 w^2 (-1 + 1/u)(-1 + 1/w)\big(1 + 1/\theta((-1 + 1/u)^{\theta} + (-1 + 1/w)^{\theta})\big)^3,$$

allowing to calculate the related log-likelihood. The log-likehood using a Clayton generator is 74.00. Using $\psi_h(t)$, the log-likelihood is 50.10. So in this case, it seems one is better off with a Clayton generator at node 0.47.

## 2.9 Discussion

Based on the performance sections of this chapter, it seems that the tree structure estimator kt_kagg, introduced in this thesis as an extension of the work from Górecki et al. (2015), is most likely the best NAC tree estimator available at the moment, not only because it is able to retrieve the target structure more often than other estimators, but also because it is one of the fastest tree structure estimator there is. The new tree structure estimator based on supertree methods has failed to beat the kt_kagg estimator in our study, but nonetheless

outperforms the S&U method, which itself is able to outperform (but not always) the method from Okhrin et al. (2013a). Can a better NAC tree estimator than kt_kagg be found? Hopefully, further research on the matter by someone having read this thesis should be facilitated, thanks to the proposed framework from this chapter, as well as by the novel way of comparing tree estimators, using the tri-distance introduced in Section 1.6.

# Chapter 3

# Extending One-Factor Copulas

## 3.1 Introduction

Factor copulas refer to copulas which can be expressed by means of unobserved variables, the factors. Most of the time, only one univariate (and continuous) factor $X_0$ is used, and thus one talks about one-factor copulas. They are a hot topic of research at the moment, being written about by Krupskii and Joe (2013), Joe (2014), Krupskii and Joe (2015), Mazo et al. (2015a) or Oh and Patton (2015). Nonetheless, the scope of current one-factor copulas is still limited when it comes to the construction of parametric models.

First, only factors with a uniform distribution are currently considered to the literature. Yet, in applications, the identification of a factor may implicitely assume estimating its distribution. Second, studying the factor's impact on the dependence structure is not allowed, either. Indeed, in current one-factor copulas, only conditional independence is allowed. That is, the variables $U_1, \ldots, U_d$, whose joint distribution is a copula, are independent conditionally on the factor $X_0 = x_0$. This means that, for all $u_1, \ldots, u_d \in [0, 1]$,

$$P(U_1 \leq u_1, \ldots, U_d \leq u_d | X_0 = x_0) = \prod_{i=1}^{d} P(U_i \leq u_i | X_0 = x_0).$$

As a result, current one-factor copulas are written (Krupskii and Joe, 2013) as

$$
\begin{aligned}
C(u_1, \ldots, u_d) &= \int_0^1 \prod_{i=1}^{d} P(U_i \leq u_i | X_0 = x_0)\, dx_0 \qquad (3.1.1)\\
&= \int_0^1 \prod_{i=1}^{d} \frac{\partial C_i(u_i, x_0)}{\partial x_0}\, dx_0\\
&= \int_0^1 \prod_{i=1}^{d} C_{i|0}(u_i|x_0)\, dx_0,
\end{aligned}
$$

where $C_i$ is some bivariate copula and $X_0$ has a uniform distribution.

As (3.1.1) allows to see, the task of modeling only amounts to choose a parametric form for $P(U_i \leq u_i | X_0 = x_0), i \in \{1, \ldots, d\}$. What if the practitioner assumes that the dependence grows with the factor's value? Or, what if the dependence structure remains the same, but is not conditional independence?

This chapter is about overcoming these limitations and is largely written based on Mazo and Uyttendaele. It introduces new representations for one-factor copulas, allowing one to construct novel parametric models. These representations cover many models of the literature in a nontrivial way, as seen in Section 3.5. Section 3.3 explores various properties of these representations, while Section 3.6 explores estimation and also proposes a novel test to assess whether conditional independence may hold or not, without assuming any parametric form for the dependence structure. Section 3.7 presents the numerical experiments used to illustrate the testing procedure as well as a real data analysis. An example from Chapter 2 is also revisited using a newly developed model from Section 3.5, see the beginning of Section 3.7.

## 3.2 Extended one-factor copulas and nested extended one-factor copulas

Consider the *the law of total probability*,

$$
\begin{aligned}
C(u_1, \ldots, u_d) &= P(U_1 \leq u_1, \ldots, U_d \leq u_d)\\
&= \int P(U_1 \leq u_1, \ldots, U_d \leq u_d | X_0 = x_0) f_0(x_0)\, dx_0, \qquad (3.2.1)
\end{aligned}
$$

where the integral is taken over the support of $X_0$ and $f_0$ denotes its density. Equation (3.2.1) is the one from which the formula of current one-factor copulas originated, given by (3.1.1). One can easily see that one-factor copulas are a reformulation of the law of total probability in which the factor $X_0$ is uniformly distributed on $[0, 1]$ and the variables $U_1, \ldots, U_d$ are assumed to be independent conditionally on the factor.

To extend one-factor copulas, in addition to letting the density of $X_0$, $f_0$, be unspecified, it is proposed to reconsider the decomposition of

$$P(U_1 \leq u_1, \ldots, U_d \leq u_d | X_0 = x_0).$$

Given $X_0 = x_0$, certainly the vector $(U_1, \ldots, U_d)$ has a distribution function, but it is not, in general, a copula, because $U_i|X_0 = x_0$ is not, in general, uniformly distributed. By Sklar's theorem (Sklar, 1959; Nelsen, 2006),

$$P(U_1 \leq u_1, \ldots, U_d \leq u_d | X_0 = x_0)$$

can be decomposed as a copula and a set of marginal distributions, that is,

$$P(U_1 \leq u_1, \ldots, U_d \leq u_d | X_0 = x_0) = \\ C_{x_0}(P(U_1 \leq u_1 | X_0 = x_0), \ldots, P(U_d \leq u_d | X_0 = x_0)). \quad (3.2.2)$$

If we let $x_0$ vary, both the copula $C_{x_0}$ and the margins $P(U_i \leq u_i|X_0 = x_0)$, $i = 1, \ldots, d$, will be, in fact, conditional distributions.

*Example* 3.2.1. Consider (3.2.1) with $X_0$ following an exponential distribution, as

$$f_0(x_0) = e^{-x_0}, \qquad x_0 > 0. \quad (3.2.3)$$

Moreover, in (3.2.2), assume that

$$P(U_i \leq u_i | X_0 = x_0) = \int_0^{u_i} \frac{\Gamma(1 + x_0)}{\Gamma(x_0)} (1 - t)^{x_0 - 1} \, dt,$$

where

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} \, dt, \qquad z > 0, \quad (3.2.4)$$

is the well known Gamma function. Finally, assume that the density of $C_{x_0}$, $c_{x_0}$ is

$$c_{x_0}(u_1, \ldots, u_d) = (\det R(x_0))^{-1/2} \exp\left[-\frac{1}{2} z^{\top} ([R(x_0)]^{-1} - I) z\right], \quad (3.2.5)$$

where $z = (z_1, \ldots, z_d)$, $z_i$ is the quantile of order $u_i$ of the standard normal distribution, $I$ is the $d \times d$ identity matrix, and

$$R(x_0) = \begin{pmatrix} 1 & & & \\ & \ddots & \beta(x_0) & \\ & \beta(x_0) & \ddots & \\ & & & 1 \end{pmatrix}, \quad (3.2.6)$$

where

$$\beta(x_0) = e^{-x_0}.$$

In Example 3.2.1, for a fixed $x_0$, the copula $C_{x_0}$ is a multivariate Gaussian copula with an exchangeable correlation matrix with parameter $\beta(x_0) = e^{-x_0}$. Likewise, the distribution of $U_i$ given $X_0 = x_0$ is a beta distribution with parameters 1 and $x_0$. By Sklar's theorem, $P(U_i \leq u_i | X_0 = x_0)$ and $C_{x_0}$ can be set independently.

*Example* 3.2.2. Consider (3.2.1) with $X_0$ following a Pareto distribution, as

$$f_0(x_0) = x_0^{-2}, \qquad x_0 > 1. \tag{3.2.7}$$

Moreover, in (3.2.2), assume that

$$C_{x_0}(u_1, \ldots, u_d) = \exp\left[ -\left( (-\log u_1)^{x_0} + \ldots + (-\log u_d)^{x_0} \right)^{1/x_0} \right].$$

In Example 3.2.2, for a fixed $x_0$, the copula $C_{x_0}$ is recognized to be a Gumbel-Hougaard copula with parameter $x_0$, see Nelsen (2006, p. 153). The margins $P(U_i \leq u_i | X_0 = x_0)$, $i = 1, \ldots, d$ were not specified.

While examples such as Example 3.2.1 and Example 3.2.2 could be multiplied endlessly, there is a representation, presented below, which permits to get them all, and build general parametric one-factor copulas quite easily. So, in view of both the law of total probability (3.2.1) and the "conditional Sklar's theorem" (3.2.2), every one-factor copula can be written as

$$\begin{aligned} &C(u_1, \ldots, u_d) \\ &= \int C_{x_0}[P(U_1 \leq u_1 | X_0 = x_0), \ldots, P(U_d \leq u_d | X_0 = x_0)] f_0(x_0)\, dx_0, \end{aligned} \tag{3.2.8}$$

where, as in Examples 3.2.1 and 3.2.2, $C_{x_0}$ is to be understood as a collection, running over $x_0$, of well defined $d$-variate copulas and where the integral is taken over the support of $X_0$. In representation (3.2.8), as well as in Example 3.2.1 and Example 3.2.2, letting $x_0$ vary induces a collection of copulas $\{C_{x_0}\}$ which reflects the change in the dependence structure as the factor varies. For instance, in the former example, $C_{x_0} \to \Pi$ (pointwise) as $x_0 \to \infty$, where $\Pi$ denotes the independence copula, that is, $\Pi(u_1, \ldots, u_d) = u_1 \cdots u_d$ for all $u_1, \ldots, u_d \in [0, 1]$. On the other hand, if $x_0 \to 0$, then $C_{x_0} \to M$, where $M$ denotes the Fréchet-Hoeffding bound for copulas, that is, $M$ represents the complete positive dependence structure, with $M(u_1, \ldots, u_d) = \min(u_1, \ldots, u_d)$ for all $u_1, \ldots, u_d \in [0, 1]$. As the factor's value varies, the dependence between the variables $X_1, \ldots, X_d$ varies as well, ranging from independence to complete positive dependence. The opposite happens in Example 3.2.2. We have that $C_{x_0} \to M$ whenever $x_0 \to 0$ and $C_{x_0} \to \Pi$ whenever $x_0 \to 1$.

Representation (3.2.8) can be recast in terms of standard uniform variables only. Let $Q_0 = F_0^{-1}$ be the inverse of the factor's distribution function $F_0$. By

the change of variables $u_0 = F_0(x_0)$ in (3.2.8), we have

$$\begin{aligned}
&C(u_1, \ldots, u_d) \\
&= \int C_{x_0}[P(U_1 \le u_1 | U_0 = F_0(x_0)), \ldots, P(U_d \le u_d | U_0 = F_0(x_0))] f_0(x_0)\, dx_0 \\
&= \int_0^1 C^{\diamond}_{Q_0(u_0)}[P(U_1 \le u_1 | U_0 = u_0), \ldots, P(U_d \le u_d | U_0 = u_0)]\, du_0 \\
&= \int_0^1 C^{\diamond}_{Q_0(u_0)}[C_{1|0}(u_1|u_0), \ldots, C_{d|0}(u_d|u_0)]\, du_0 \\
&= \int_0^1 C^{\diamond}_{Q_0(u_0)}\left[\frac{\partial C_1(u_1, u_0)}{\partial u_0}, \ldots, \frac{\partial C_d(u_d, u_0)}{\partial u_0}\right] du_0, \qquad (3.2.9)
\end{aligned}$$

where $(U_i, U_0) \sim C_i$, $i = 1, \ldots, d$.

Examples 3.2.1 and 3.2.2 can be recast in view of (3.2.9).

*Example* 3.2.3 (continuation of Example 3.2.1). From (3.2.3), we have $Q_0(u_0) = -\log(1-u_0)$, hence, for a fixed $u_0 \in (0,1)$, $C^{\diamond}_{Q_0(u_0)}$ is a multivariate Gaussian copula with correlation matrix given by

$$R(u_0) = \begin{pmatrix} 1 & & & \\ & \ddots & \beta(u_0) & \\ & \beta(u_0) & \ddots & \\ & & & 1 \end{pmatrix}, \qquad \beta(u_0) = e^{-Q_0(u_0)} = 1 - u_0.$$

Furthermore, since $P(U_i \le u_i | U_0 = u_0) = P(U_i \le u_i | X_0 = Q_0(u_0))$, we have

$$C_{i|0}(u_i|u_0) = \int_0^{u_i} \frac{\Gamma(1 + Q_0(u_0))}{\Gamma(Q_0(u_0))} (1-t)^{Q_0(u_0)-1}\, dt,$$

so that the underlying bivariate copula is

$$C_i(u_i, u_0) = \int_0^{u_0} \int_0^{u_i} \frac{\Gamma(1 + Q_0(t))}{\Gamma(Q_0(t))} (1-y)^{Q_0(t)-1}\, dt\, dy,$$

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

In Example 3.2.3, note that $u_0 = 0$ implies $\beta(u_0) = 1$, and thus $C^{\diamond}_{Q_0(u_0)}$ is the Fréchet-Hoeffding bound $M$. Likewise, $u_0 = 1$ implies $\beta(u_0) = 0$ (by continuity), and thus $C^{\diamond}_{Q_0(u_0)}$ is the independence copula.

*Example* 3.2.4 (continuation of Example 3.2.2). From (3.2.7), we have $Q_0(u_0) = 1/(1-u_0)$, hence, for a fixed $u_0 \in (0,1)$,

$$\begin{aligned}
&C^{\diamond}_{Q_0(u_0)}(u_1, \ldots, u_d) = \exp\left[-\left((-\log u_1)^{\beta(u_0)} + \ldots + (-\log u_d)^{\beta(u_0)}\right)^{1/\beta(u_0)}\right], \\
&\beta(u_0) = Q_0(u_0) = \frac{1}{1-u_0},
\end{aligned}$$

that is, $C^{\diamond}_{Q_0(u_0)}$ is a multivariate Gumbel-Hougaard copula with parameter given by

$$\beta(u_0) = Q_0(u_0) = 1/(1-u_0).$$

In Example 3.2.4, $u_0 = 0$ implies $\beta(u_0) = 1$, and thus $C^{\diamond}_{Q_0(u_0)}$ is the independence copula. Likewise, $u_0 = 1$ implies $\beta(u_0) = \infty$ (by continuity), and thus $C^{\diamond}_{Q_0(u_0)}$ is the Fréchet-Hoeffding bound. In short, we simply replaced the $x_0$'s of Example 3.2.1 and Example 3.2.2 by $Q_0(u_0)$.

Mathematically, both representations (3.2.8) and (3.2.9) are of course equivalent. It is worth stressing that these representations are better not to be taken as plain mathematical results, but rather as a convenient way to generate new parametric one-factor copula models, as was shown in the above examples. The advantage of the representation in (3.2.9) is that it involves copulas only and allows an easy comparison with (3.1.1). This last equation indeed corresponds to (3.2.9) with $C^{\diamond}_{Q_0(u_0)} = \Pi$, $x_0 = u_0$ and $X_0 = U_0$. Representation (3.2.8) is however more convenient when one adopts a point of view centered on the factor itself.

Both representations (3.2.8) and (3.2.9) contain two $d$-dimensional copulas: one, $C_{x_0}$ or $C^{\diamond}_{Q_0(u_0)}$, that's part of the integrand, and one, $C(u_1, \ldots, u_d)$, that's the result of the integral. In the rest of this chapter, the latter will often be referred to as the *outer* copula, while the former will often be referred to as the *inner* copula. In the rest of this chapter, the copulas $C_1, \ldots, C_d$ in (3.2.9) will be referred to as the bivariate copulas or *linking* copulas.

The inner copula in both representations is usually assumed to be a copula with one parameter, this parameter being explicitly related to $u_0$ through $Q_0$ in representation (3.2.9). However, many $d$-dimensional copula have more than one parameter. In such cases, each parameter can be assumed to be related to $u_0$ through a different mapping $Q_l$, $l \in \{1, \ldots, p\}$, where $p$ is the number of parameters of the related inner copula, and one can write:

$$C(u_1, \ldots, u_d) = \int_0^1 C^{\diamond}_{\{Q_l(u_0)\}} \left[ \frac{\partial C_1(u_1, u_0)}{\partial u_0}, \ldots, \frac{\partial C_d(u_d, u_0)}{\partial u_0} \right] du_0.$$

In general however, we will just write $C^{\diamond}_{\{Q_l(u_0)\}}$ as $C^{\diamond}_{Q_0(u_0)}$ or even $C^{\diamond}_{u_0}$.

Since any $d$-dimensional copula can be used in representations (3.2.8) and (3.2.9) as inner copula, one can also consider using an outer copula as inner copula, nesting the construction in itself.

Let us rewrite expression (3.2.9) as

$$C(u_1, \ldots, u_d) = \int_0^1 C^{\diamond_1}_{t_1} \Big[ H_{11}(u_1, t_1), \ldots, H_{d1}(u_d, t_1) \Big] dt_1, \tag{3.2.10}$$

that is, we rewrite $C^{\diamond}_{Q_0(u_0)}$ as $C^{\diamond_1}_{t_1}$, replace $\frac{\partial C_i(u_i, u_0)}{\partial u_0}$ by $H_{i1}(u_i, t_1)$ and $u_0$ by $t_1$.

Next, define $C^{\diamond_1}_{t_1}$ as

$$C_{t_1}^{\diamond_1}(v_1,\ldots,v_d)=\int_0^1 C_{t_2}^{\diamond_2}\big(H_{12}(v_1,t_2),\ldots,H_{d2}(v_d,t_2)\big)dt_2,$$

where $C_{t_2}^{\diamond_2}$ is some $d$-variate copula, the parameters of which being each related to $t_2$ through a different mapping, and where $H_{i2}(v_i,t_2)=\frac{\partial C_{i2}(v_i,t_2)}{\partial t_2}$, $\{C_{i2}\}$ being a set of bivariate copulas, with $i\in\{1,\ldots,d\}$.

The outer copula $C(u_1,\ldots,u_d)$ in Equation (3.2.10) is then

$$\begin{aligned}&C(u_1,\ldots,u_d)\\&=\int_0^1\int_0^1 C_{t_2}^{\diamond_2}\big(H_{12}(H_{11}(u_1,t_1),t_2),\ldots,H_{d2}(H_{d1}(u_d,t_1),t_2)\big)\,dt_2\,dt_1.\end{aligned}$$

Notice that in the above nesting scheme, the $d$-variate copula $C_{t_1}^{\diamond_1}$ actually does not depend on $t_1$. In general, we will always assume that only the last $d$-variate copula $C_{t_w}^{\diamond_w}$ in a nesting chain actually depends on its related variable $t_w$, in order to avoid an extra layer of complexity. While working with this last assumption, a nested extended one-factor copula (NEOFC), hereafter denoted as $C_w^{\odot}$, can be written as

$$\begin{aligned}&C_w^{\odot}(u_1,\ldots,u_d)\\&=\int_{[0,1]^w} C_{t_w}^{\diamond_w}\big(G_1(u_1;t_w,\ldots,t_1),\ldots,G_d(u_d;t_w,\ldots,t_1)\big)\,dt_w\ldots dt_1,\end{aligned}\tag{3.2.11}$$

where $w,d\geq 2$, $G_i(u_i;t_w,\ldots,t_1)$ is

$$G_i(u_i;t_w,\ldots,t_1)=H_{iw}^{t_w}\circ\cdots\circ H_{i1}^{t_1}(u_i),\tag{3.2.12}$$

and $H_{ij}^{t_j}(\bullet)\equiv H_{ij}(\bullet,t_j)$, with

$$\begin{aligned}H_{i1}(\bullet,t_1)=&\frac{\partial C_{i1}(\bullet,t_1)}{\partial t_1},\\H_{i2}(\bullet,t_2)=&\frac{\partial C_{i2}(\bullet,t_2)}{\partial t_2},\\&\vdots\\H_{iw}(\bullet,t_w)=&\frac{\partial C_{iw}(\bullet,t_w)}{\partial t_w}.\end{aligned}\tag{3.2.13}$$

As an example, if $w = 3$, we have

$$\begin{aligned} G_i(u_i; t_3, t_2, t_1) =& H_{i3}^{t_3} \circ H_{i2}^{t_2} \circ H_{i1}^{t_1}(u_i) \\ =& \frac{\partial C_{i3}(H_{i2}^{t_2} \circ H_{i1}^{t_1}(u_i), t_3)}{\partial t_3} \\ =& \frac{\partial C_{i3}(\frac{\partial C_{i2}(H_{i1}^{t_1}(u_i), t_2)}{\partial t_2}, t_3)}{\partial t_3} \\ =& \frac{\partial C_{i3}(\frac{\partial C_{i2}(\frac{\partial C_{i1}(u_i, t_1)}{\partial t_1}, t_2)}{\partial t_2}, t_3)}{\partial t_3}. \end{aligned}$$

The block of equations (3.2.13) reveals that there is a set of $w$ bivariate copulas underlying $G_i(u_i; t_w, \ldots, t_1)$. Since $i \in \{1, \ldots, d\}$, this means that there is always a set of $d \times w$ bivariate copulas underlying any NEOFC.

To summarize, a one-factor copula (OFC) can be written as

$$C(u_1, \ldots, u_d) = \int_0^1 \prod_{i=1}^{d} \left( \frac{\partial C_i(u_i, u_0)}{\partial u_0} \right) du_0, \tag{3.2.14}$$

an extended one-factor copula (EOFC) as

$$C(u_1, \ldots, u_d) = \int_0^1 C_{u_0}^{\diamond} \left( \frac{\partial C_1(u_1, u_0)}{\partial u_0}, \ldots, \frac{\partial C_d(u_d, u_0)}{\partial u_0} \right) du_0, \tag{3.2.15}$$

and a nested extended one-factor copual (NEOFC) as

$$\begin{aligned} & C_w^{\odot}(u_1, \ldots, u_d) \\ &= \int_{[0,1]^w} C_{t_w}^{\diamond_w} \big( G_1(u_1; t_w, \ldots, t_1), \ldots, G_d(u_d; t_w, \ldots, t_1) \big) \, dt_w \ldots dt_1. \end{aligned} \tag{3.2.16}$$

Note that EOFCs are a special case of NEOFCs: setting $w = 1$ in the last expression above means falling back to extended one-factor copulas. Also note that, while the inner copula of a NEOFC is assumed to depend on only one factor, $T_w$, a nested extended one-factor copula actually encompasses more than one factor: from $T_1$ to $T_w$.

## 3.3 Properties of EOFCs and NEOFCs

In this section, various properties of EOFCs and NEOFCs are given. Identifiability issues are also addressed.

### 3.3.1 On the inner copula

In order to generate new parametric families of one-factor copulas using (3.2.8) or (3.2.9), one can act on 3 components: the bivariate or linking copulas $C_i$, $i \in \{1, \dots, d\}$, the factor's distribution, represented either by its density $f_0$ or by its quantile function $Q_0$, and the set of multivariate copulas $\{C_{x_0}\} = \{C^{\diamond}_{Q_0(u_0)}\}$. Depending on the choice for the inner copula $C_{x_0}$, three different forms of factor copulas can be made. This is illustrated by the following example.

*Example* 3.3.1. Let $X_0$ follow an exponential distribution with parameter $\lambda > 0$,

$$f_0(x_0) = \lambda e^{-\lambda x_0}, \qquad x_0 > 0.$$

For $i \in \{1, \dots, d\}$, let $C_i$ be a Clayton copula so that

$$C_i(u_i, u_0) = [(u_i^{-\alpha_i} + u_0^{-\alpha_i} - 1)]^{-1/\alpha_i} \qquad \alpha_i \geqslant 0. \tag{3.3.1}$$

Finally, let $c_{x_0}$, the density of $C_{x_0}$, be as in (3.2.5) where $R(x_0)$ is as in (3.2.6) and where

$$\beta(x_0) = e^{-\beta_0 - \beta_1 x_0}, \qquad \beta_0, \beta_1 \geqslant 0. \tag{3.3.2}$$

In Example 3.3.1, we have built a parametric model for one-factor copulas which allow for different features. First, the number of parameters, $d + 3$, is linear in $d$, the dimension. Second, as will be seen in Section 3.6, assuming a parametric form for the factor's distribution, one can estimate the related parameter $\lambda$ by maximum pseudo-likelihood, thus learning valuable information about a variable which, by definition, is never directly observed. Finally, one can control the growth rate of the dependence structure, relative to the change of the factor's value. For instance, in (3.3.2), a decrease in $\beta_1$ leads to an increase in $\beta(x_0), \forall x_0$, $\beta(x_0)$ being the correlation parameter. If one set $\beta_1 = 0$, then $\beta(x_0) = \exp(-\beta_0)$, and thus the correlation parameter, hence the conditional copula $C_{x_0}$, does not depend on $x_0$ anymore: we call this *conditional invariance*, not to be mistaken with conditional independence. This last feature happens when $\beta_0 = \infty$, implying a correlation parameter $\beta(x_0) = 0$.

The three different forms of factor copulas are now formalized.

**Conditional independence.** Conditional independent one-factor copulas are those such that the inner copula is the independence copula. They correspond exactly to copulas of the form (3.1.1), described in Krupskii and Joe (2013), and their interpretation is such that, given the factor's value $X_0 = x_0$, the variables $U_1, \dots, U_d$ are independent. Let us note that, even in this simple case, the obtained models are quite reasonable and useful, as was demonstrated not only in Krupskii and Joe (2013) or Oh and Patton (2015), but also in view of the vast literature about conditional independent models, see Skrondal and Rabe-Hesketh (2007). In Section 3.6.2, we provide a novel procedure in order to test the assumption of conditional independence.

**Conditional invariance.** Conditional invariant one-factor copulas are those such that, in (3.2.8) for instance, $C_{x_0} = C_{x_0'}$ for any $x_0$ and $x_0'$. That is, there is conditional dependence, but this conditional dependence remains the same regardless of the factor's value.

**Conditional noninvariance.** Conditional noninvariant one-factor copulas are those which are not conditionally invariant. Note that, *a fortiori*, they are not conditionally independent either. Here, the conditional dependence structure is allowed to change with the factor's value. In Example 3.2.1, $\beta(x_0) \to 0$ as $x_0 \to \infty$ and therefore $C_{x_0} \to \Pi$, the independence copula. On the opposite, $\beta(x_0) \to 1$ as $x_0 \to 0$ and thus $C_{x_0} \to M$, the Fréchet-Hoeffding upper bound, characterizing complete positive dependence.

Natural parametric one-factor copulas can be built with the help of Kendall's tau and Spearman's rho. Recall that, given a bivariate copula $C$, Kendall's tau is a dependence coefficient in $[-1, 1]$ defined by

$$\tau = 4 \int_{[0,1]^2} C(u, v)\, dC(u, v) - 1. \tag{3.3.3}$$

A value of $\tau \approx 0$ hints at independence, and $\tau \approx -1$ (respectively $\tau \approx +1$) indicates negative (respectively positive) dependence. Example 3.3.2 illustrates the procedure.

*Example* 3.3.2. Let $X_0$ follow a standard uniform distribution and let $C_{x_0}$ be as

$$C_{x_0}(u_1, \ldots, u_d) = (u_1^{-\tau^{-1}(x_0)} + \cdots + u_d^{-\tau^{-1}(x_0)} - d + 1)^{-1/\tau^{-1}(x_0)},$$

where $\tau^{-1}$ is the inverse map of

$$\tau(\beta) = \frac{\beta}{\beta + 2}, \qquad \beta > 0. \tag{3.3.4}$$

In Example 3.3.2, for a fixed $x_0$, $C_{x_0}$ is recognized to be a Clayton copula with parameter $\tau^{-1}(x_0) = 2x_0/(1 - x_0)$ for $x_0 \in (0, 1)$. In general, the procedure works as follows. First, choose a parametric family of copulas, here the family of Clayton copulas

$$C_\beta(u_1, \ldots, u_d) = (u_1^{-\beta} + \cdots + u_d^{-\beta} - d + 1)^{-1/\beta}, \qquad \beta > 0. \tag{3.3.5}$$

Second, compute Kendall's tau (there is only one, since all pairs have the same distribution), which in this case is given by (3.3.4). Third, choose the distribution of $X_0$ so that its support corresponds to the range of the map induced by (3.3.4), here $(0, 1)$. Fourth and last, replace $\beta$ by $\tau^{-1}(x_0)$ in (3.3.5).

The conditional dependence structure in Example 3.3.2 goes from conditional independence to conditional complete dependence. Indeed, when $x_0 \to 0$, $\beta(x_0) \to 0$ and $C_{\beta(x_0)} \to \Pi$. If $x_0 \to 1$ instead, $\beta(x_0) \to \infty$ and $C_{\beta(x_0)} \to M$,

the Fréchet-Hoeffding upper bound for copulas. If one rather defines $\beta(x_0) = -\log(x_0)$, then $\beta(x_0) \to \infty$ when $x_0 \to 0$ and $C_{\beta(x_0)} \to M$. Hence, in one case the dependence increases with respect to the factor, while in the other case it decreases.

### 3.3.2 Densities of EOFCs and NEOFCs

The density of an EOFC, as defined by (3.2.10) is straighforward to get:

$$\begin{aligned} c(u_1, \ldots, u_d) &= \frac{\partial^d C(u_1, \ldots, u_d)}{\partial u_1 \ldots \partial u_d} \\ &= \int_0^1 c_{t_1}^{\diamond_1}(H_{11}(u_1, t_1), \ldots, H_{d1}(u_d, t_1)) \prod_{i=1}^{d} c_i(u_i, t_1) dt_1, \end{aligned} \tag{3.3.6}$$

which is the integral of the product of one $d$-variate density and $d$ bivariate densities, and where $c_{t_1}^{\diamond_1}$ is the density of $C_{t_1}^{\diamond_1}$ while $c_i$ is the density of $C_i$.

The density of a NEOFC, as defined by (3.2.11), is however less straighforward to obtain and require first to figure out the result of $\frac{\partial G_i}{\partial u_i}$, also refer to (3.2.12) and (3.2.13). Observe that (chain rule):

$$\frac{\partial H_{ij}^{t_j} \circ H_{il}^{t_l}(u_i)}{\partial u_i} = \frac{\partial H_{ij}(H_{il}^{t_l}(u_i), t_j)}{\partial H_{il}^{t_l}(u_i)} \frac{\partial H_{il}^{t_l}(u_i)}{\partial u_i}.$$

Using this last result, $\frac{\partial G_i}{\partial u_i}$ can now be written as

$$\begin{aligned} \frac{\partial G_i}{\partial u_i} =& \frac{\partial H_{iw}^{t_w} \circ \cdots \circ H_{i1}^{t_1}(u_i)}{\partial u_i} \\ =& \frac{\partial H_{iw}(H_{i,w-1}^{t_{w-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i), t_w)}{\partial H_{i,w-1}^{t_{w-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i)} \frac{\partial H_{i,w-1}^{t_{w-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i)}{\partial u_i} \\ =& \prod_{k=w}^{3} \left( \frac{\partial H_{ik}(H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i), t_k)}{\partial H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i)} \right) \frac{\partial H_{i2}(H_{i1}^{t_1}(u_i), t_2)}{\partial H_{i1}^{t_1}(u_i)} \frac{\partial H_{i1}^{t_1}(u_i)}{\partial u_i}. \end{aligned} \tag{3.3.7}$$

The last expression in (3.3.7) is the product of $(w-2)+1+1 = w$ fractions. The last one is a bivariate copula density and the others each involve a bivariate density. Starting with the fraction on the right of the last expression in (3.3.7), we indeed have that

$$\frac{\partial H_{i1}^{t_1}(u_i)}{\partial u_i} = \frac{\partial H_{i1}(u_i, t_1)}{\partial u_i} = \frac{\partial^2 C_{i1}(u_i, t_1)}{\partial u_i \partial t_1} = c_{i1}(u_i, t_1).$$

If we replace $H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i)$ by $\bigcirc$ so that

$$\frac{\partial H_{ik}(H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i), t_k)}{\partial H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i)} = \frac{\partial H_{ik}(\bigcirc, t_k)}{\partial \bigcirc},$$

we see that

$$\frac{\partial H_{ik}(\bigcirc, t_k)}{\partial \bigcirc} = \frac{\partial^2 C_{ik}(\bigcirc, t_k)}{\partial \bigcirc \partial t_k} = c_{ik}(\bigcirc, t_k). \tag{3.3.8}$$

Therefore, we can write that

$$\frac{\partial G_i}{\partial u_i} = \prod_{k=w}^{3} \left( c_{ik}\left( H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i), t_k \right) \right) \times c_{i2}(H_{i1}^{t_1}(u_i), t_2) \times c_{i1}(u_i, t_1). \tag{3.3.9}$$

As an example, we can expand (3.3.9) for $w = 3$:

$$\begin{aligned} \frac{\partial G_i}{\partial u_i} =& c_{i3}(H_{i2}^{t_2}(H_{i1}^{t_1}(u_i)), t_3) c_{i2}(H_{i1}^{t_1}(u_i), t_2) c_{i1}(u_i, t_1) \\ =& c_{i3}(\frac{\partial C_{i2}(\frac{\partial C_{i1}(u_i,t_1)}{\partial t_1}, t_2)}{\partial t_2}, t_3) c_{i2}(\frac{\partial C_{i1}(u_i, t_1)}{\partial t_1}, t_2) c_{i1}(u_i, t_1). \end{aligned} \tag{3.3.10}$$

The density of a NEOFC can now be written:

$$\begin{aligned} & c_w^{\odot}(u_1, \ldots, u_d) = \frac{\partial^d C_w^{\odot}(u_1, \ldots, u_d)}{\partial u_1 \ldots \partial u_d} \\ =& \int_{[0,1]^w} c_{t_w}^{\diamond_w}\left( G_1(u_1; t_w, \ldots, t_1), \ldots, G_d(u_d; t_w, \ldots, t_1) \right) \prod_{i=1}^{d} \frac{\partial G_i}{\partial u_i} dt_w \ldots dt_1 \\ =& \int_{[0,1]^w} c_{t_w}^{\diamond_w}\left( \{G_i\} \right) \\ \times & \prod_{i=1}^{d} \left[ \prod_{k=w}^{3} \left( c_{ik}\left( H_{i,k-1}^{t_{k-1}} \circ \cdots \circ H_{i1}^{t_1}(u_i), t_k \right) \right) \times c_{i2}(H_{i1}^{t_1}(u_i), t_2) \times c_{i1}(u_i, t_1) \right] dt_w \ldots dt_1, \end{aligned} \tag{3.3.11}$$

where $c_{t_w}^{\diamond_w}$ is the density of $C_{t_w}^{\diamond_w}$ and $\{G_i\} = \{G_1(u_1; t_w, \ldots, t_1), \ldots, G_d(u_d; t_w, \ldots, t_1)\}$.

The density of a NEOFC, as displayed in the last expression of (3.3.11), is thus made of two parts: a $d$-variate copula density on the left, $c_{t_w}^{\diamond_w}\left(\{G_i\}\right)$, and a product that involves $d \times w$ bivariate copula densities on the right of $c_{t_w}^{\diamond_w}\left(\{G_i\}\right)$.

### 3.3.3 On the margins of EOFCs and NEOFCs

The multivariate margins of an EOFC are still EOFCs.

*Proof.* This can be easily seen by integrating $u_1$ out of Equation (3.3.6):

$$\begin{aligned}
&\int_0^1 c(u_1,\dots,u_d)\,du_1 \\
&= \int_0^1 \int_0^1 c_{t_1}^{\diamond_1}(H_{11}(u_1,t_1),\dots,H_{d1}(u_d,t_1)) \prod_{i=1}^{d} c_i(u_i,t_1) dt_1\,du_1 \\
&= \int_0^1 \int_0^1 c_{t_1}^{\diamond_1}(H_{11}(u_1,t_1),\dots,H_{d1}(u_d,t_1)) \prod_{i=1}^{d} \frac{\partial H_i(u_i,t_1)}{\partial u_i} dt_1\,du_1.
\end{aligned}$$

The integral requires the following change of variable to be solved:

$$\begin{aligned}
u &= H_{11}(u_1,t_1), \\
du &= \frac{\partial H_{11}(u_1,t_1)}{\partial u_1} du_1,
\end{aligned}$$

leading to

$$\int_0^1 \int_0^1 c_{t_1}^{\diamond_1}(u, H_{21}(u_2,t_1),\dots,H_{d1}(u_d,t_1)) \prod_{i=2}^{d} \frac{\partial H_{i1}(u_i,t_1)}{\partial u_i} dt_1\,du.$$

After reorganizing this last expression, one gets:

$$\begin{aligned}
&\int_0^1 c(u_1,\dots,u_d)\,du_1 \\
&= \int_0^1 \prod_{i=2}^{d} \frac{\partial H_{i1}(u_i,t_1)}{\partial u_i} \int_0^1 c_{t_1}^{\diamond_1}(u, H_{21}(u_2,t_1),\dots,H_{d1}(u_d,t_1))\,du\,dt_1.
\end{aligned}$$

The integral related to $du$ is the margin of $c_{t_1}^{\diamond_1}$ in $H_{21},\dots,H_{d1}$ when its first argument is integrated out. The final result is

$$c^{[2:d]}(u_2,\dots,u_d) = \int_0^1 c_{t_1}^{[2:d],\diamond_1}(H_{21}(u_2,t_1),\dots,H_{d1}(u_d,t_1)) \prod_{i=2}^{d} \frac{\partial H_{i1}(u_i,t_1)}{\partial u_i} dt_1,$$

which is the density of an EOFC. □

In general, in order to get a given margin from an EOFC, one just needs to drop the variables that are no longer needed in (3.3.6).

Similarly to this last result, the multivariate margins of a NEOFC are also NEOFCs.

*Proof.* The proof is similar to what was done in the case of EOFCs. Start from (3.3.11) and integrate $u_1$ out:

$$\begin{aligned}&\int_0^1 c_w^{\odot}(u_1,\dots,u_d)du_1\\ &=\int_0^1\int_{[0,1]^w} c_{t_w}^{\diamond_w}\big(G_1(u_1;t_w,\dots,t_1),\dots,G_d(u_d;t_w,\dots,t_1)\big)\prod_{i=1}^d\frac{\partial G_i}{\partial u_i}dt_w\dots dt_1\,du_1.\end{aligned} \tag{3.3.12}$$

Perform a change of variable:

$$\begin{aligned}u&=G_1(u_1;t_w,\dots,t_1),\\ du&=\frac{\partial G_1(u_1;t_w,\dots,t_1)}{\partial u_1}du_1,\end{aligned}$$

leading to

$$\int_0^1\int_{[0,1]^w} c_{t_w}^{\diamond_w}\big(u,G_2,\dots,G_d\big)\prod_{i=2}^d\frac{\partial G_i}{\partial u_i}dt_w\dots dt_1\,du.$$

Reorganize this last expression:

$$\int_{[0,1]^w}\prod_{i=2}^d\frac{\partial G_i}{\partial u_i}\int_0^1 c_{t_w}^{\diamond_w}\big(u,G_2,\dots,G_d\big)du\,dt_w\dots dt_1.$$

The integral related to $du$ is just the $(d-1)$ margin of $c_{t_w}^{\diamond_w}$ in $G_2,\dots,G_d$. One can now write:

$$\begin{aligned}&\int_0^1 c_w^{\odot}(u_1,\dots,u_d)\,du_1=c_w^{[2:d],\odot}(u_2,\dots,u_d)\\ &=\int_{[0,1]^w} c_{t_w}^{[2:d],\diamond_w}\big(G_2(u_2;t_w,\dots,t_1),\dots,G_d(u_d;t_w,\dots,t_1)\big)\prod_{i=2}^d\frac{\partial G_i}{\partial u_i}dt_w\dots dt_1,\end{aligned} \tag{3.3.13}$$

which is the expression of a NEOFC density. □

In general, the margins of a NEOFC are easy to get: just drop in expression (3.3.11) the elements corresponding to the variables that are not of interest.

### 3.3.4 The upper and lower Fréchet-Hoeffding bounds

The upper Fréchet–Hoeffding bound and is defined as $M(u_1, \ldots, u_d) = \min(u_1, \ldots, u_d)$. It is always a copula, and corresponds to comonotone random variables.

PROPOSITION 3.3.3. *In* (3.2.9)*, let* $C^{\diamond}_{Q_0(u_0)} = \Pi$ *and let* $C_i(u_i, u_0) = \min(u_i, u_0)$, $\forall i \in \{1, \ldots, d\}$*, where* $\min(u_i, u_0) = M(u_i, u_0)$ *is the upper Fréchet-Hoeffding bound on* $(U_i, U_0)$*. Then,* $C(u_1, \ldots, u_d) = \min(u_1, \ldots, u_d)$*.*

*Proof.* The outer copula is

$$C(u_1, \ldots, u_d) = \int_0^1 \frac{\partial \min(u_1, u_0)}{\partial u_0} \ldots \frac{\partial \min(u_d, u_0)}{\partial u_0} du_0,$$

and since $\frac{\partial \min(u_i, u_0)}{\partial u_0}$ is 0 as soon as $u_0 > u_i$, one can write:

$$C(u_1, \ldots, u_d) = \int_0^{\min(u_1, \ldots, u_d)} \frac{\partial \min(u_1, u_0)}{\partial u_0} \ldots \frac{\partial \min(u_d, u_0)}{\partial u_0} du_0.$$

Moreover, as long as $u_0 < \min(u_1, \ldots, u_d)$, we have that $\frac{\partial \min(u_i, u_0)}{\partial u_0} = \frac{\partial u_0}{\partial u_0} = 1, \forall i \in \{1, \ldots d\}$. Therefore, we have

$$C(u_1, \ldots, u_d) = \int_0^{\min(u_1, \ldots, u_d)} 1 \ldots 1 \, du_0 = \min(u_1, \ldots, u_d).$$

□

This result will turn out very useful to show that OFCs are not suited to model hierarchical dependence in Section 3.5.

The 2-dimensional lower Fréchet-Hoeffding bound is usually written as $W(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$. Note that, in contrast to the upper Fréchet-Hoeffding bound, its generalization, $W(u_1, \ldots, u_d) = \max(d - 1 + \sum_{i=1}^{d} u_i, 0)$, is not a copula anymore.

PROPOSITION 3.3.4. *In* (3.2.9)*, let* $d = 2$*,* $C^{\diamond}_{Q_0(u_0)} = \Pi$*,* $C_1(u_1, u_0) = \max(u_1 + u_0 - 1, 0)$ *and* $C_2(u_2, u_0) = \min(u_2, u_0)$*. Then,* $C(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$*.*

*Proof.* The outer copula is

$$C(u_1, u_2) = \int_0^1 \frac{\partial \max(u_1 + u_0 - 1, 0)}{\partial u_0} \frac{\partial \min(u_2, u_0)}{\partial u_0} du_0,$$

and since the integrand is 0 as long as $u_2 < u_0$, one can write that

$$C(u_1, u_2) = \int_0^{u_2} \frac{\partial \max(u_1 + u_0 - 1, 0)}{\partial u_0} \frac{\partial \min(u_2, u_0)}{\partial u_0} du_0.$$

Since, $\forall u_0 \in [0, u_2]$, $\frac{\partial \min(u_2, u_0)}{\partial u_0} = 1$, one can write

$$\begin{aligned} C(u_1, u_2) &= \int_0^{u_2} \frac{\partial \max(u_1 + u_0 - 1, 0)}{\partial u_0} du_0 \\ &= \max(u_1 + u_0 - 1, 0)|_{u_0 = u_2} - \max(u_1 + u_0 - 1, 0)|_{u_0 = 0} \\ &= \max(u_1 + u_2 - 1, 0), \end{aligned}$$

and the proof is complete.

□

### 3.3.5 Two degenerate cases

From Equation (3.2.15) or Equation (3.2.16), one can end up with

$$C(u_1, \ldots, u_d) = E_{U_0}\left[C^{\diamond}_{U_0}(u_1, \ldots, u_d)\right]$$

or with

$$C^{\odot}_w(u_1, \ldots, u_d) = E_{T_w}\left[C^{\diamond_w}_{T_w}(u_1, \ldots, u_d)\right].$$

PROPOSITION 3.3.5. *In Equation* (3.2.15)*, let* $C_i$*,* $\forall i \in \{1, \ldots, d\}$*, be such that* $C_i(u_i, u_0) = u_i u_0$*. Then,* $C(u_1, \ldots, u_d) = E_{U_0}\left[C^{\diamond}_{U_0}(u_1, \ldots, u_d)\right]$*.*

*Proof.* If $C_i$, $\forall i \in \{1, \ldots, d\}$, is such that $C_i(u_i, u_0) = u_i u_0$, and since we have that $\frac{\partial(u_i u_0)}{\partial u_0} = u_i$, Equation (3.2.15) becomes

$$C(u_1, \ldots, u_d) = \int_0^1 C^{\diamond}_{u_0}(u_1, \ldots, u_d)\, du_0 = E_{U_0}\left[C^{\diamond}_{U_0}(u_1, \ldots, u_d)\right].$$

□

The reasoning is the same for a NEOFC: one just needs to let all linking copulas be the independence copula. Note that in case of conditional invariance, we have that $C(u_1, \ldots, u_d) = C^{\diamond}(u_1, \ldots, u_d)$ and $C^{\odot}_w(u_1, \ldots, u_d) = C^{\diamond_w}(u_1, \ldots, u_d)$, that is, the outer copula is directly equal to the inner one. This result is important as it means that any $d$-dimensional copula in the literature is an EOFC where the linking copulas are the independence copula and the inner copula is equal to the $d$-dimensional copula of interest.

A second degenerate case, regarding the linking copulas, is now presented.

PROPOSITION 3.3.6. *Let the inner copula be the independence copula and set all linking copulas to the upper Fréchet-Hoeffding bound, except for one, $C_l$. In $C(u_1, \dots, u_d)$, set now all arguments to 1, except for $u_l$ and $u_k$, where $k$ is an arbitrary element of $\{1, \dots d\}\backslash\{l\}$. The resulting bivariate margin of $C$, $C^{[l,k]}(u_l, u_k)$, is then equal to $C_l(u_l, u_k)$.*

*Proof.* Let us write the expression of $C^{[l,k]}(u_l, u_k)$ when the inner copula is the independence one and $C_k$ is the upper Fréchet-Hoeffding bound:

$$C^{[l,k]}(u_l, u_k) = \int_0^1 \frac{\partial C_l(u_l, u_0)}{\partial u_0} \frac{\partial \min(u_k, u_0)}{\partial u_0} du_0.$$

The function $\min(u_k, u_0)$ will return $u_0$ as long as $u_0 < u_k$. Otherwise, $u_k$ is returned. Therefore, one can write

$$\begin{aligned} C^{[l,k]}(u_l, u_k) &= \int_0^{u_k} \frac{\partial C_l(u_l, u_0)}{\partial u_0} \frac{\partial u_0}{\partial u_0} du_0 + \int_{u_k}^1 \frac{\partial C_l(u_l, u_0)}{\partial u_0} \frac{\partial u_k}{\partial u_0} du_0 \\ &= \int_0^{u_k} \frac{\partial C_l(u_l, u_0)}{\partial u_0} du_0 + 0 \\ &= C_l(u_l, u_k) - C_l(u_l, 0) \\ &= C_l(u_l, u_k) \end{aligned}$$

□

Proposition 3.3.6 can be generalized for the case of a NEOFC as follows: if $C_{lm}$ is the linking copula of interest, $l \in \{1, \dots d\}$, $m \in \{1, \dots w\}$, then the $[l, k]$-margin of $C_w^{\odot}(u_1, \dots, u_d)$ is $C_{lm}(u_l, u_k)$ if $C_{t_w}^{\diamond_t}$ is the independence copula while the linking copulas related to $T_m$ are set to the upper Fréchet-Hoeffding bound, except for $C_{lm}$, and the remaining linking copulas are set to the independence copula. The proof that the result of this setup is $C^{[l,k],\odot}(u_l, u_k) = C_{lm}(u_l, u_k)$ is trivial in view of the proof of Proposition 3.3.6 and is therefore left to the reader.

### 3.3.6 Dependence properties

To start, let us write representation (3.2.9) when $d = 2$:

$$C(u_1, u_2) = \int_0^1 C_{Q_0(u_0)}^{\diamond} \left[ \frac{\partial C_1(u_1, u_0)}{\partial u_0}, \frac{\partial C_2(u_2, u_0)}{\partial u_0} \right] du_0. \tag{3.3.14}$$

Let's also recall some dependence properties from Nelsen (2006).

A copula $C$ is positively quadrant dependent (PQD) if $C(\mathbf{u}) \geq \Pi(\mathbf{u})$ for any $\mathbf{u} \in [0,1]^d$, where $\Pi(\mathbf{u})$ is the independence copula. Similarly, a copula $C$ is negatively quadrant dependent (NQD) if $C(\mathbf{u}) \leq \Pi(\mathbf{u})$ for any $\mathbf{u} \in [0,1]^d$.

If $C_1$ is the copula of $(U_1, U_0)$, then $U_1$ is said to be stochastically decreasing in $U_0$ if $\frac{\partial C_1(u_1,u_0)}{\partial u_0}$ is an increasing function of $u_0$, $\frac{\partial^2 C_1(u_1,u_0)}{\partial u_0^2} \geq 0$. This can be written as $SD(U_1|U_0)$ and it will often be said that $C_1$ is SD in $U_0$ or that $C_1(u_1, u_0)$ is SD in its second argument.

Similarly, if $C_1$ is the copula of $(U_1, U_0)$, then $U_1$ is said to be stochastically increasing in $U_0$ if $\frac{\partial C_1(u_1,u_0)}{\partial u_0}$ is a decreasing function of $u_0$, $\frac{\partial^2 C_1(u_1,u_0)}{\partial u_0^2} \leq 0$. This can be written as $SI(U_1|U_0)$ and it will often be said that $C_1$ is SI in $U_0$ or that $C_1(u_1, u_0)$ is SI in its second argument.

Finally, note that if a copula $C_1$ is stochastically increasing in either $U_0$ or $U_1$, then it is PQD. The same applies if $C_1$ is stochastically decreasing in either $U_0$ or $U_1$: $C_1$ is then NQD. The reverse is however not true: being SI or SD for a copula is a stronger dependence property than being PQD or NQD.

PROPOSITION 3.3.7. *Let the inner copula in* (3.3.14)*, $C^{\diamond}_{Q_0(u_0)}$, be a PQD copula, $\forall u_0$. Then if $C_2$ is NQD and $C_1$ is $SD(U_1|U_0)$ or if $C_2$ is PQD and $C_1$ is $SI(U_1|U_0)$, the outer copula is PQD, too.*

*Proof.* If $C^{\diamond}_{Q_0(u_0)}(v_1, v_2)$ is PQD, then $C^{\diamond}_{Q_0(u_0)}(v_1, v_2) \geq v_1 v_2$. Therefore, it is clear that

$$C(u_1, u_2) = \int_0^1 C^{\diamond}_{Q_0(u_0)}\left[\frac{\partial C_1(u_1, u_0)}{\partial u_0}, \frac{\partial C_2(u_2, u_0)}{\partial u_0}\right] du_0 \geq \int_0^1 \frac{\partial C_1(u_1, u_0)}{\partial u_0}\frac{\partial C_2(u_2, u_0)}{\partial u_0}\, du_0.$$

Using integration by parts, one can then show that

$$C(u_1, u_2) \geq u_2 \frac{\partial C_1(u_1, u_0)}{\partial u_0}\bigg|_{u_0=1} - \int_0^1 C_2(u_2, u_0)\frac{\partial^2 C_1(u_1, u_0)}{\partial u_0^2} du_0.$$

Since $C_1$ is $SD(U_1|U_0)$ and $C_2$ is NQD, we have that

$$0 \leq \int_0^1 C_2(u_2, u_0)\frac{\partial^2 C_1(u_1, u_0)}{\partial u_0^2} du_0 \leq \int_0^1 u_2 u_0 \frac{\partial^2 C_1(u_1, u_0)}{\partial u_0^2} du_0,$$

and therefore

$$C(u_1, u_2) \geq u_2 \frac{\partial C_1(u_1, u_0)}{\partial u_0}\bigg|_{u_0=1} - \int_0^1 u_2 u_0 \frac{\partial^2 C_1(u_1, u_0)}{\partial u_0^2} du_0.$$

Using integration by parts again, we have that

$$\int_0^1 u_2 u_0 \frac{\partial^2 C_1(u_1, u_0)}{\partial u_0^2} du_0 = u_2 \frac{\partial C_1(u_1, u_0)}{\partial u_0}\bigg|_{u_0=1} - u_1 u_2,$$

leading to

$$C(u_1, u_2) \geq u_1 u_2.$$

□

A similar proposition as the one above can be made, ensuring that the outer copula is this time NQD. The proof is almost the same as the one from the previous proposition and is therefore left to the reader.

PROPOSITION 3.3.8. *Let the inner copula in* (3.3.14)*, $C^{\diamond}_{Q_0(u_0)}$, be a NQD copula, $\forall u_0$. Then if $C_2$ is NQD and $C_1$ is $SI(U_1|U_0)$ or if $C_2$ is PQD and $C_1$ is $SD(U_1|U_0)$, the outer copula is NQD, too.*

COROLLARY 3.3.9. *Let the inner copula in a 2-dimensional EOFC be the independence copula, that is, we fall back to Equation* (3.1.1)*, where $X_0 = U_0$. Then if $C_2$ is PQD, $C_1$ is $SI(U_1|U_0)$ and since the independence copula is PQD, by Proposition 3.3.7, the outer copula is PQD. Similarly, if $C_2$ is NQD, $C_1$ is $SI(U_1|U_0)$ and since the independence copula is NQD, by Proposition 3.3.8, the outer copula is NQD.*

Note that Proposition 1 in Krupskii and Joe (2013) offers a similar result as the one from the above corollary, where it is shown that if both $C_1$ and $C_2$ are SI in $U_0$, the outer copula is PQD. Proposition 3.3.7 shows that both $C_1$ and $C_2$ do not need to be SI in $U_0$ for the outer copula to be PQD: as long as one of the two linking copulas is SI and the remaining one is just PQD, then the outer one is PQD.

Let us now write the expression of a NEOFC when $w, d = 2$. Then:

$$C_2^{\odot}(u_1, u_2) = \int_{[0,1]^2} C^{\diamond_2}_{t_2}\left(\frac{\partial C_{12}(\frac{\partial C_{11}(u_1,t_1)}{\partial t_1}, t_2)}{\partial t_2}, \frac{\partial C_{22}(\frac{\partial C_{21}(u_2,t_1)}{\partial t_1}, t_2)}{\partial t_2}\right) dt_2\, dt_1. \tag{3.3.15}$$

The following proposition ensures that the outer copula is going to be PQD.

PROPOSITION 3.3.10. *Let the inner copula in* (3.3.15)*, $C^{\diamond_2}_{t_2}$, be a PQD copula, $\forall t_2$. Then if $C_{21}$ is NQD and $C_{11}$ is SD in $T_1$, while $C_{22}$ is NQD and $C_{12}$ is SD in $T_2$, the outer copula, $C_2^{\odot}(u_1, u_2)$, is PQD.*

*Proof.* Let us write $C_2^{\odot}(u_1, u_2)$ as

$$C_2^{\odot}(u_1, u_2) = \int_0^1 C^{\diamond_1}_{t_1}\left(\frac{\partial C_{11}(u_1, t_1)}{\partial t_1}, \frac{\partial C_{21}(u_2, t_1)}{\partial t_1}\right) dt_1, \tag{3.3.16}$$

where

$$C_{t_1}^{\diamond_1}(v_1, v_2) = \int_0^1 C_{t_2}^{\diamond_2}\left(\frac{\partial C_{12}(v_1, t_2)}{\partial t_2}, \frac{\partial C_{22}(v_2, t_2)}{\partial t_2}\right) dt_2. \tag{3.3.17}$$

By Proposition 3.3.7, if $C_{t_1}^{\diamond_1}$ is PQD while $C_{21}$ is NQD and $C_{11}$ is SD in $T_1$, then $C_2^{\odot}(u_1, u_2)$ is PQD, too. By Proposition 3.3.7 again, in order for $C_{t_1}^{\diamond_1}$ to be PQD, one needs $C_{t_2}^{\diamond_2}$ to be PQD, while $C_{22}$ is NQD and $C_{12}$ is SD in $T_2$. □

Proposition 3.3.10 can be generalized as follows:

PROPOSITION 3.3.11. *Let the inner copula in a 2-dimensional NEOFC be PQD. If for each $j \in \{1, \ldots, w\}$ we have that $C_{1j}$ is NQD and $C_{2j}$ is SD with respect to the factor $T_j$ or that $C_{2j}$ is NQD and $C_{1j}$ is SD with respect to the factor $T_j$, then the outer copula, $C_w^{\odot}$, is PQD.*

The proof of this last proposition is trivial in regard of the proof of Proposition 3.3.10. Proposition 3.3.11 can also be expressed so that the resulting outer copula is NQD:

PROPOSITION 3.3.12. *Let the inner copula in a 2-dimensional NEOFC be NQD. If for each $j \in \{1, \ldots, w\}$ we have that $C_{1j}$ is NQD and $C_{2j}$ is SI with respect to the factor $T_j$ or that $C_{2j}$ is NQD and $C_{1j}$ is SI with respect to the factor $T_j$, then the outer copula, $C_w^{\odot}$, is NQD.*

Hereafter is an example of NEOFC meeting the requirements of Proposition 3.3.11.

*Example* 3.3.13. Start from 3.3.15. Replace the inner copula $C_{t_2}^{\diamond_2}$ by the Frank copula, with parameter

$$\theta^{\diamond_2}(t_2) = \frac{\log(1 - t_2)}{-\lambda},$$

that is, we use the inverse CDF of an exponential distribution on $t_2$ in order to calculate $\theta^{\diamond_2}$, where $\lambda > 0$ is the rate, that is, the maximum value of the corresponding exponential probability density function. Since the support of the exponential is $[0, \infty)$, the parameter $\theta^{\diamond_2}(t_2)$ is always positive and therefore $C_{t_2}^{\diamond_2}$ is PQD $\forall t_2 \in [0, 1]$.

Next, set both $C_{11}$ and $C_{12}$ to be a Mardia copula with a negative parameter's value, ensuring the copula is NQD (Nelsen, 2006, p. 188), while both $C_{21}$ and $C_{22}$ are AMH copulas, each with a negative parameter's value, to ensure that $C_{21}$ is SD in $T_1$ and $C_{22}$ is SD in $T_2$. Indeed, for an AMH copula $C_{\theta_{\mathsf{AMH}}}(\bullet, t_j)$, one have that

$$\frac{\partial^2 C_{\theta_{\mathsf{AMH}}}(\bullet, t_j)}{\partial t_j^2} = -\frac{2\theta_{\mathsf{AMH}} \times (\bullet - 1) \bullet (\theta_{\mathsf{AMH}} \times (\bullet - 1) + 1)}{(\theta_{\mathsf{AMH}} \times (t_j - 1)(\bullet - 1) - 1)^3},$$

which is a positive quantity as long as $\theta_{\mathsf{AMH}} < 0$.

The outer copula resulting from this construction is, by Proposition 3.3.11, PQD.

Next we introduce an important conjecture, which describes the relationship between the inner copula and the outer copula of an EOFC.

CONJECTURE 3.3.14. *Let the inner copula in* (3.3.14), $C^{\diamond}_{Q_0(u_0)}$, *be stochastically decreasing in both its arguments and be conditionally invariant (refer to Subsection 3.3.1 for details on conditional invariance). If, moreover,* $C_2$ *is NQD while* $C_1$ *is SD in* $U_0$, *then* $C(u_1,u_2) > C^{\diamond}_{Q_0(u_0)}(u_1,u_2)$, $\forall (u_1,u_2) \in [0,1]^2$.

The motivation of this conjecture is now given.

Let $\Delta_{u_i}(u_0) = \frac{\partial C_i(u_i,u_0)}{\partial u_0} - u_i$. Note that $\int_0^1 \frac{\partial C_i(u_i,u_0)}{\partial u_0} du_0 = C_i(u_i,u_0)\Big|_0^1 = u_i$, and therefore that $\int_0^1 \Delta_{u_i}(u_0) du_0 = 0$. Also note that if $C_i$ is SI in $U_0$, then $\Delta_{u_i}(u_0)$ is a decreasing function of $u_0$. Similarly, if $C_i$ is SD in $U_0$, then $\Delta_{u_i}$ is an increasing function of $u_0$.

Using integration by parts, one can show that

$$\int_0^1 \Delta_{u_1}(u_0)\Delta_{u_2}(u_0)\, du_0 = -\int_0^1 \Big(C_2(u_2,u_0) - u_2 u_0\Big) \times \frac{\partial^2 C_1(u_1,u_0)}{\partial u_0^2} du_0, \tag{3.3.18}$$

the integral on the left remaining positive $\forall (u_1,u_2) \in [0,1]^2$ if $C_1$ is SI in $U_0$ while $C_2$ is PQD. Likewise, the integral on the left remains negative $\forall (u_1,u_2) \in [0,1]^2$ if $C_1$ is SI in $U_0$ while $C_2$ is NQD.

Let us now write the Taylor expansion for a bivariate copula $C^{\diamond}_{Q_0(u_0)}$:

$$\begin{aligned}
&C^{\diamond}_{Q_0(u_0)}\Big(u_1 + \Delta_{u_1}(u_0), u_2 + \Delta_{u_2}(u_0)\Big) \\
&= C^{\diamond}_{Q_0(u_0)}(u_1,u_2) + \frac{\partial C^{\diamond}_{Q_0(u_0)}(u_1,u_2)}{\partial u_1}\Delta_{u_1}(u_0) + \frac{\partial C^{\diamond}_{Q_0(u_0)}(u_1,u_2)}{\partial u_2}\Delta_{u_2}(u_0) \\
&\quad + \frac{\partial^2 C^{\diamond}_{Q_0(u_0)}(u_1,u_2)}{\partial u_1 \partial u_2}\Delta_{u_1}(u_0)\Delta_{u_2}(u_0) + \frac{\partial^2 C^{\diamond}_{Q_0(u_0)}(u_1,u_2)}{\partial u_1^2}\Delta^2_{u_1}(u_0)/2 \\
&+ \frac{\partial^2 C^{\diamond}_{Q_0(u_0)}(u_1,u_2)}{\partial u_2^2}\Delta^2_{u_2}(u_0)/2 + R\Big(u_1 + \Delta_{u_1}(u_0), u_2 + \Delta_{u_2}(u_0)\Big).
\end{aligned} \tag{3.3.19}$$

If one neglects the remainder term $R\Big(u_1 + \Delta_{u_1}(u_0), u_2 + \Delta_{u_2}(u_0)\Big)$, then one can write that the outer copula in Equation (3.3.14) is, taking into account that $C^{\diamond}_{Q_0(u_0)}(u_1,u_2) = C^{\diamond}(u_1,u_2)$ is conditionally invariant,

, which implicitly means that one would assume that higher order derivatives of $C^{\diamond}_{Q_0(u_0)}(u_1,u_2)$ can be approximated by 0, that is,

$$\begin{aligned}
C(u_1,u_2) \approx C^{\diamond}(u_1,u_2)\, du_0 &+ \frac{\partial^2 C^{\diamond}(u_1,u_2)}{\partial u_1 \partial u_2}\int_0^1 \Delta_{u_1}(u_0)\Delta_{u_2}(u_0)\, du_0 \\
&+ \frac{\partial^2 C^{\diamond}(u_1,u_2)}{\partial u_1^2}\int_0^1 \frac{\Delta^2_{u_1}(u_0)}{2}\, du_0 + \frac{\partial^2 C^{\diamond}(u_1,u_2)}{\partial u_2^2}\int_0^1 \frac{\Delta^2_{u_2}(u_0)}{2}\, du_0.
\end{aligned}$$

Note that both

$$\frac{\partial^2 C^{\diamond}(u_1,u_2)}{\partial u_1^2}\int_0^1 \frac{\Delta_{u_1}^2(u_0)}{2}\,du_0 \tag{3.3.20}$$

and

$$\frac{\partial^2 C^{\diamond}(u_1,u_2)}{\partial u_2^2}\int_0^1 \frac{\Delta_{u_2}^2(u_0)}{2}\,du_0 \tag{3.3.21}$$

are positive since the inner copula $C^{\diamond}$ is stochastically decreasing in both its arguments and $\int_0^1 \frac{\Delta_{u_i}^2(u_0)}{2}\,du_0$ is necessarily positive $\forall i$.

Moreover, by Equation (3.3.18),

$$\int_0^1 \Delta_{u_1}(u_0)\Delta_{u_2}(u_0)\,du_0 = -\int_0^1 \Big(C_2(u_2,u_0)-u_2u_0\Big)\times\frac{\partial^2 C_1(u_1,u_0)}{\partial u_0^2}\,du_0.$$

is a positive quantity, since $C_2$ is NQD and $C_1$ is SD in $U_0$. Therefore the outer copula can be approximated by $C^{\diamond}(u_1,u_2)$ plus some positive quantity, $\forall(u_1,u_2)\in[0,1]^2$, leading to the claim of the conjecture that $C(u_1,u_2) > C^{\diamond}(u_1,u_2)$.

It is also possible to express a similar conjecture for $C(u_1,u_2) < C^{\diamond}(u_1,u_2)$:

Conjecture 3.3.15. *Let the inner copula in* (3.3.14)*, $C^{\diamond}_{Q_0(u_0)} = C^{\diamond}$, be stochastically increasing in both its arguments and be conditionally invariant. If, moreover, $C_2$ is NQD while $C_1$ is SI in $U_0$, then $C(u_1,u_2) < C^{\diamond}(u_1,u_2)$, $\forall(u_1,u_2)\in[0,1]^2$.*

The motivation is almost exactly the same as that of Conjecture 3.3.14 and is therefore left to the reader.

Of course, one might wonder at this point how reasonable it is to neglect the remainder term in (3.3.19). In what follows, an attempt at detecting at least one example where Conjecture 3.3.15 fails is presented.

Let the inner copula be one of the 9 following copulas:

- a Farlie-Gumbel-Morgenstern (Nelsen, 2006; Balakrishnan and Lai, 2009) copula with a parameter's value of 0.25, 0.5 or 1,
- a Plackett copula (Kemp et al., 1992; Nelsen, 2006) with a parameter's value of 3, 12 or 64 (ranging from low to high level of dependence),
- or a Frank copula with a parameter's value of 2.5, 6 or 14 (again, in an effort to have low, moderate and high dependence).

All these inner copulas are conditionally invariant and SI in both arguments as required by Conjecture 3.3.15. Hereafter is the proof that the proposed FGM copulas are SI in both arguments.

The expression of a FGM copula is

$$C_{\theta_{\text{FGM}}}(u_1,u_2) = u_1u_2 + \theta_{\text{FGM}}u_1u_2(1-u_1)(1-u_2),$$

where $\theta_{\text{FGM}} \in [-1, 1]$.

One can calculate that

$$\frac{\partial^2 C_{\theta_{\text{FGM}}}}{\partial u_1^2} = 2\theta_{\text{FGM}} \times (u_2(u_2 - 1)),$$

and that

$$\frac{\partial^2 C_{\theta_{\text{FGM}}}}{\partial u_2^2} = 2\theta_{\text{FGM}} \times (u_1(u_1 - 1)).$$

In order for the copula to be SI in both its arguments, these two expressions have to be negative. Obviously, this will be the case as long as $\theta_{\text{FGM}}$ is positive.

Let now $C_1$ be one of the following copulas:

- a Plackett copula with a parameter's value of 3, 12 or 64,
- a Frank copula with a parameter's value of 2.5, 6 or 14,
- or a FGM copula with a parameter's value of 0.25, 0.5, 1.

These copulas are all SI in (at least) their second argument, as requested by Conjecture 3.3.15.

Finally, let $C_2$ be one of the following 9 copulas:

- a Mardia copula with a parameter's value of -0.25, -0.5 or -0.75,
- a Frank copula with a parameter's value of -2.5, -6 or -14,
- or a FGM copula with a parameter's value of -0.25, -0.5 or -1.

These copulas are all NQD, as required by Conjecture 3.3.15. A Mardia copula (Mardia, 1970; Nelsen, 2006) is defined as

$$\begin{aligned} C_{\theta_{\text{Mardia}}}(u_1, u_2) = {} & \frac{\theta^2_{\text{Mardia}}(1 + \theta_{\text{Mardia}})}{2} M(u_1, u_2) \\ & + (1 - \theta^2_{\text{Mardia}})\Pi(u_1, u_2) + \frac{\theta^2_{\text{Mardia}}(1 - \theta_{\text{Mardia}})}{2} W(u_1, u_2), \end{aligned}$$

where $M(u_1, u_2)$ is the upper Fréchet–Hoeffding bound and $W(u_1, u_2)$ the lower one, with $\theta_{\text{Mardia}} \in [-1, 1]$. A Mardia copula is NQD as long as $\theta_{\text{Mardia}} < 0$ (Nelsen, 2006, p. 188).

In total, there are $9^3 = 729$ combinations or setups of interest for the 27 presented copulas. For each of these 729 combinations, it is checked if $C(u_1, u_2) < C^{\diamond}(u_1, u_2)$, that is, if the outer copula is below the inner one, for a grid of 100 points in $[0, 1]^2$. Since $C(u_1, u_2)$ is calculated through numerical integration (see the appendix chapter), one must take into account the error on $C(u_1, u_2)$ before being able to conclude that $C(u_1, u_2) < C^{\diamond}(u_1, u_2)$. This is performed through the following hypothesis test:

$$H_0: \quad C(u_1, u_2) \geq C^{\diamond}(u_1, u_2),$$
$$H_1: \quad C(u_1, u_2) < C^{\diamond}(u_1, u_2),$$

| | $\theta_{\text{Mardia}} = -0.25$ | $\theta_{\text{Frank}} = -2.5$ | $\theta_{\text{FGM}} = -0.25$ | $\theta_{\text{Mardia}} = -0.5$ | $\theta_{\text{Frank}} = -6$ | $\theta_{\text{FGM}} = -0.5$ | $\theta_{\text{Mardia}} = -0.75$ | $\theta_{\text{Frank}} = -14$ | $\theta_{\text{FGM}} = -0.75$ |
|---|---|---|---|---|---|---|---|---|---|
| $\theta_{\text{Plackett}} = 3$ | 1.82e-3 | <1e-7 | <1e-7 | 1.52e-6 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{Frank}} = 2.5$ | 2.43e-3 | <1e-7 | <1e-7 | 2.19e-5 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{FGM}} = 0.25$ | 4.05e-2 | <1e-7 | <1e-7 | 3.63e-3 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{Plackett}} = 12$ | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{Frank}} = 6$ | <1e-6 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{FGM}} = 0.5$ | 1.96e-2 | <1e-7 | <1e-7 | 1.69e-3 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{Plackett}} = 64$ | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{Frank}} = 14$ | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |
| $\theta_{\text{FGM}} = 1$ | 6.46e-3 | <1e-7 | <1e-7 | 3.33e-4 | <1e-7 | <1e-7 | <1e-7 | <1e-7 | <1e-7 |

Table 3.1: Averaged p-values for various setups when the inner copula is a Frank copula with a parameter's value of 14. Rows are the various $C_1$ copulas, while columns describe the $C_2$ copulas.

with the following test statistic:

$$\frac{\hat{C}(u_1, u_2) - C^{\diamond}(u_1, u_2)}{\text{error}} \overset{H_0}{\sim} N(0, 1),$$

and where the p-value is the surface on the left of this test statistic.

Since for a given a setup, the test has to be run over a grid of 100 points, 100 p-values is obtained for each of the $9^3$ setups. To summarize all these p-values, the average p-value is calculated for each setup.

**Results.** For all setups except the one using as inner copula a Frank copula with a parameter's value of 14, the average p-value was less than 1e-7. Results for the inner Frank copula with a parameter's value of 14 are reported in Table 3.1, where the rows are the various $C_1$ presented earlier, while the columns are the various $C_2$ presented earlier.

### 3.3.7 Identifiability

A model is said to be identifiable if it is theoretically possible to learn the true values of this model's underlying parameters after obtaining an infinite number of observations from it. Mathematically, this is equivalent to saying that different values of the parameters must generate different probability distributions of the observable variables. That is, if $(P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \boldsymbol{\Theta})$ is a statistical model, with $P_{\boldsymbol{\theta}}$ a probability measure on a fixed space, the model is identifiable if $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ implies that $P_{\boldsymbol{\theta}_1} \neq P_{\boldsymbol{\theta}_2}$.

Models built using Equations (3.2.14), (3.2.15) or (3.2.16) are not necessarily identifiable.

*Example* 3.3.16. In Equation (3.2.14), let $d = 2$ and $C_1, C_2$ be Farlie-Gumbel-Morgenstern copulas, that is,

$$C_i(u_i, u_0; \theta_i) = u_i u_0 + \theta_i u_i u_0 (1 - u_0)(1 - u_i),$$

with $\theta_i \in [1, -1]$. The outer copula can be showed to be

$$C(u_1, u_2; \theta_1, \theta_2) = \frac{u_1 u_2}{3}(\theta_1 \theta_2 (u_1 - 1)(u_2 - 1) + 3) = u_1 u_2 + \frac{\theta_1 \theta_2}{3} u_1 u_2 (1 - u_1)(1 - u_2).$$

Thus, one can see that

$$C(u_1, u_2; \theta_1, \theta_2) = C(u_1, u_2; \theta_1', \theta_2')$$

whenever $\theta_1 \theta_2 = \theta_1' \theta_2'$, and the last equation can be satisfied even if $(\theta_1, \theta_2) \neq (\theta_1', \theta_2')$.

*Example* 3.3.17. In Equation (3.2.14), let $C_i$ be the independance copula for $i \in \{2, \dots, d\}$. Then, the outer copula is

$$C(\boldsymbol{u}; \theta_1) = \int_0^1 \frac{\partial C_1(u_1, u_0; \theta_1)}{\partial u_0} \, u_2 \dots u_d \, du_0,$$

which can be simplified to

$$C(\boldsymbol{u}; \theta_1) = u_2 \dots u_d \Big( C_1(u_1, 1; \theta_1) - C_1(u_1, 0; \theta_1) \Big) = u_1 \dots u_d.$$

The model is thus not identifiable: all values of $\theta_1$ lead to the same model, the independence copula.

*Example* 3.3.18. In Equation (3.2.15), let $d = 2$, $C_1$ be a Gaussian copula with correlation $\Delta_1$ and $C_2$ be a Gaussian copula with correlation $\Delta_2$. Moreover, let $C^{\diamond}_{Q_0(u_0)}$ be a Gaussian copula with correlation $\rho_A$. Then, it can be shown that the outer copula is a Gaussian copula with correlation

$$\rho_A \sqrt{1 - \Delta_1^2} \sqrt{1 - \Delta_2^2} + \Delta_1 \Delta_2.$$

Even in the case where $\Delta_1 = \Delta_2 = \Delta$, this becomes

$$\rho_A \big(1 - \Delta^2\big) + \Delta^2,$$

for which one can easily find $(\rho_A, \Delta)$ and $(\rho_A', \Delta')$, $(\rho_A, \Delta) \neq (\rho_A', \Delta')$ such that $\rho_A \big(1 - \Delta^2\big) + \Delta^2 = \rho_A' \big(1 - (\Delta')^2\big) + (\Delta')^2$. Refer to Section 3.5 for more details on Gaussian copula built using Equation (3.2.15).

In order to shed more light on identifiability issues, a few case studies based on the Fisher information matrix are presented. The Fisher information matrix can indeed be used to check if a model is locally identifiable, see Rothenberg (1971) or Iskrev et al. (2010). If the model is not locally identifiable all over its parameter space, then it is not, in general, identifiable in that parameter space. Let $c(\boldsymbol{u}; \boldsymbol{\theta})$ be the density of an EOFC. Then, the related Fisher information is defined as

$$\mathcal{I}(\boldsymbol{\theta}) = E\left(\frac{\partial \log(c(\boldsymbol{u};\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} \frac{\partial \log(c(\boldsymbol{u};\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^T}\right). \tag{3.3.22}$$

Checking if a model is locally identifiable everywhere in $\boldsymbol{\Theta}$ is the same as checking where in $\boldsymbol{\Theta}$ this matrix is singular (Iskrev et al., 2010).

Numerical computation of this matrix at a point $\boldsymbol{\theta}_r \in \boldsymbol{\Theta}$ can be done in various ways. First, one can generate $n$ observations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$ from the target model at point $\boldsymbol{\theta}_r$ using Algorithm 4 and then compute

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial \log(c(\boldsymbol{u}_i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} \frac{\partial \log(c(\boldsymbol{u}_i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^T}\right)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r},$$

where $\frac{\partial \log(c(\boldsymbol{u}_i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}}\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$ is numerically approached.

So, for instance, if $\boldsymbol{\theta}_r = (\theta_{1r}, \theta_{2r}, \ldots)^T$, we have

$$\frac{\partial \log(c(\boldsymbol{u}_i;\theta_1))}{\partial \theta_1}\Big|_{\theta_1=\theta_{1r}} \approx \frac{\log(c(\boldsymbol{u}_i;\theta_{1r}+h)) - \log(c(\boldsymbol{u}_i;\theta_{1r}-h))}{2h},$$

for $h$ set to an arbitrary small value.

Another approach to compute the Fisher information in (3.3.22) is through numerical integration. Indeed,

$$\mathcal{I}(\boldsymbol{\theta}_r) = \int_{[0,1]^d}\left(\frac{\partial \log(c(\boldsymbol{u};\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}} \frac{\partial \log(c(\boldsymbol{u};\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^T}\right)\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r} c(\boldsymbol{u};\boldsymbol{\theta}_r)d\boldsymbol{u},$$

where $\frac{\partial \log(c(\boldsymbol{u};\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}}\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_r}$ has to be numerically approached, too.

Note that in both approaches, the computation of $c(\boldsymbol{u};\boldsymbol{\theta}_r)$ itself requires numerical integration: see the appendix chapter for details on numerical integration.

**Case study 1 to 3.** Let $C_1 = C_2$ be a Frank copula with a parameter's value such that $\tau_{\text{Frank}} = 0.25, 0.5$ and $0.75$. The inner copula in all three cases is a normal copula with correlation $\theta$. The model turned out to be locally identifiable for each of the 3 cases. Table 3.2 displays the results for case 3.

| $\theta$ | 0.09 | 0.18 | 0.27 | 0.36 | 0.45 | 0.55 | 0.64 | 0.73 | 0.82 | 0.91 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{\mathcal{I}}(\theta)$ | 0.595 | 0.745 | 0.949 | 1.246 | 1.698 | 2.438 | 3.908 | 6.762 | 15.617 | 58.255 |

Table 3.2: The Fisher information as a function of $\theta$ for case study 3.

**Case study 4.** In this case study, the inner copula is the independence one, $C_2$ is a Gumbel copula with a fixed parameter value corresponding to a moderate dependence and $C_1$ is the copula with the parameter of interest,

$\theta_{\text{Gumbel}}$. Table 3.3 shows a rapid decrease of the Fisher information as the parameter of interest increases.

| $\tau(\theta_{\text{Gumbel}})$ | 0.09 | 0.18 | 0.27 | 0.36 | 0.45 | 0.55 | 0.64 | 0.73 | 0.82 | 0.91 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\hat{\mathcal{I}}(\theta_{\text{Gumbel}})$ | 1.30 | 0.73 | 0.43 | 0.24 | 0.13 | 0.06 | 0.02 | <1e-2 | <1e-3 | <1e-5 |

Table 3.3: The Fisher information for case study 4.

**Case study 5.** In this last case study, $C_1$ is the independence copula, $C_2$ is the Frank copula with parameter $\theta_{\text{Frank}}$ and the inner copula is the normal copula with parameter $\theta_{\text{normal}}$. The Fisher information is this times a $2 \times 2$ matrix. Figure 3.1 shows the determinant of the matrix as a function of $\theta_{\text{Frank}}$, expressed as a Kendall's tau for convenience, and $\theta_{\text{normal}}$.

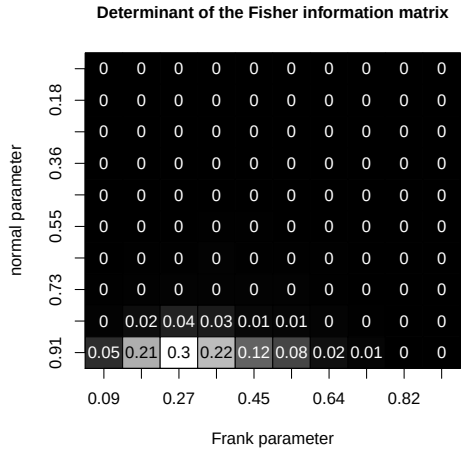



Figure 3.1: The determinant of the Fisher information matrix for case study 5.

## 3.4 Data generation

To generate one realization $(u_1, \ldots, u_d)$ of the random vector $(U_1, \ldots, U_d)$ with distribution $C$ given by (3.2.9), one takes the second part of $(u_0, u_1, \ldots, u_d)$, from $u_1$ to $u_d$, a realization of $(U_0, U_1, \ldots, U_d)$, where $U_0$ is the latent factor. Remembering that, given $U_0 = u_0$, the distribution of $(U_1, \ldots, U_d)$ can be split into the inner copula $C^{\diamond}_{u_0}$ and a set of univariate margins $\{C_{i|0}(u_i|u_0)\} = \{\frac{\partial C_i(u_i,u_0)}{\partial u_0}\}$, with $C^{-1}_{i|0}(\bullet|u_0)$ denoting the inverse function, $i \in \{1, \ldots, d\}$, the following algorithm produces the desired output.

**Algorithm 4** Generating one observation from (3.2.9).

1: Generate one observation $u_0$ from a standard uniform random variable.
2: Generate one observation $(u_1^\diamond, \ldots, u_d^\diamond)$ from $C_{u_0}^\diamond$.
3: Put $u_i = C_{i|0}^{-1}(u_i^\diamond | u_0)$ for $i = 1, \ldots, d$.

Note that, in the presence of conditional invariance, step 1 in the above algorithm is not required for step 2. Needless to say, in the first step, one could have sampled from $F_0$, the distribution of $X_0$, and in the second step, one would have sampled from $C_{x_0}$ instead of $C_{u_0}^\diamond$.

Data generation for a NEOFC can be performed using a generalized version of Algorithm 4. The main hurdle for this generalization is to define $G_i^{-1}$, such that

$$G_i^{-1}(G_i(u_i; t_w, \ldots t_1); t_w, \ldots, t_1) = u_i,$$

or, in short, such that $G_i^{-1}(G_i(u_i)) = u_i$.

Recall that

$$G_i(u_i; t_w, \ldots, t_1) = H_{iw}^{t_w} \circ \cdots \circ H_{i1}^{t_1}(u_i)$$

and that $H_{ij}^{t_j}(u_i) = \frac{\partial C_{ij}(u_i, t_j)}{\partial t_j}$.

Now define $H_{ij}^{-1,t_j}$ as the inverse function of $H_{ij}^{t_j}$, that is,

$$H_{ij}^{-1,t_j}(H_{ij}^{t_j}(u_i)) = u_i.$$

This is the same as writing

$$H_{ij}^{-1,t_j} \circ H_{ij}^{t_j}(u_i) = u_i.$$

Since $G_i$ is just a composition of the form $H_{iw}^{t_w} \circ \cdots \circ H_{i1}^{t_1}(u_i)$, we can extract back $u_i$ using the composition $H_{i1}^{-1,t_1} \circ \cdots \circ H_{iw}^{-1,t_w}(\bullet)$. The inverse of $G_i$ is therefore:

$$G_i^{-1}(\bullet) = H_{i1}^{-1,t_1} \circ \cdots \circ H_{iw}^{-1,t_w}(\bullet). \tag{3.4.1}$$

Now that $G_i^{-1}$ is defined, Algorithm 5, the generalized version of Algorithm 4, can be given.

**Input.** A NEOFC, that is, a $d$-variate copula $C_{t_w}^{\diamond_w}$, the related mappings, $d \times w$ bivariate copulas $\{C_{ij}\}$, and the related parameters.

**Output.** One observation from the input NEOFC, as a $d$-dimensional vector.

**Algorithm 5** Generating one observation from a NEOFC.

1: Generate $w$ observations from a uniform random variable on [0,1]. Denote these values as $t_1, \dots, t_w$.
2: Generate one observation from $C^{\diamond w}_{t_w}$. Denote the resulting vector as $u_1^{\diamond w} \dots u_d^{\diamond w}$.
3: On each $u_i^{\diamond w}$ in $\{u_1^{\diamond w}, \dots, u_d^{\diamond w}\}$, run the corresponding function $G_i^{-1}(\bullet; t_w, \dots, t_1)$. Denote the result as $u_i$.

The end result of the last step of Algorithm 5 is a vector $(u_1, \dots, u_d)$, one observation from the input NEOFC. Run Algorithm 5 multiple times to get multiple observations.

Critical to both Algorithms in this section is the inversion of $\frac{\partial C_i(u_i,u_0)}{\partial u_0}$ with respect to $u_i$ in the case of an EOFC or of $\frac{\partial C_{ij}(u_i,t_j)}{\partial t_j}$ with respect to $u_i$ in the case of a NEOFC. This is usually not hard to get symbolically. In the worst cases, $C^{-1}_{i|0}$ or $H^{-1,t_j}_{ij}$ can be get numerically. In the appendix chapter of the thesis, R code allowing to work with OFCs, EOFCs and NEOFCs is showcased, including a function allowing to generate observations according to the two algorithms presented in this section. At the moment, the inverse of $\frac{\partial C_i(u_i,u_0)}{\partial u_0}$ with respect to $u_i$ is implemented for more than 10 bivariate copulas. See the appendix chapter for a detailed list.

## 3.5 Models

In this section, several models built from Equation (3.2.9) or Equation (3.2.11) are presented, including many well-known copula models from the literature.

**Krupskii-Huser-Genton copulas.** In their paper, Krupskii et al. (2016) offer a new family of one-factor copulas that can be used to model replicated spatial data. Let $C_A$ be a $d$-variate Gaussian copula with correlation matrix $A = \{\rho_{ij}\}$. Let

$$F_\bullet(x_i) = \int_{-\infty}^{+\infty} \Phi(x_i - x_0) dF_0(x_0),$$

where $F_0$ is some univariate CDF, while $\Phi$ is the CDF of the univariate standard Gaussian distribution. Then, the KHG family of copulas corresponds to an EOFC where $C^\diamond_{Q_0(u_0)} = C^\diamond = C_A$ and $\frac{\partial C_i(u_i,u_0)}{\partial u_0} = \Phi(F_\bullet^{-1}(u_i) - F_0^{-1}(u_0))$, $\forall i \in \{1, \dots d\}$.

As shown by Krupskii et al. (2016), tail dependence and tail asymmetry for this model is possible, based on the choice of $F_0$ and $A$.

Let $C^{[i,j]}(u_i, u_j)$, $i \neq j$, be a bivariate margin of $C$, the outer copula. Recall

that the lower tail dependence of a bivariate copula, such as $C^{[i,j]}(u_i, u_j)$, is (Hartmann et al., 2004; McNeil et al., 2015):

$$\lambda_{ij}^L = \lim_{q \to 0} \frac{C^{[i,j]}(q,q)}{q},$$

while the upper one is defined as

$$\lambda_{ij}^U = \lim_{q \to 0} \frac{2q - 1 + C^{[i,j]}(1-q, 1-q)}{q}.$$

As an example, if

$$F_0(x_0) = 1 - K x_0^{\beta} e^{-\theta x_0^{\alpha}},$$

Krupskii et al. (2016) showed that for any bivariate margin $C^{[i,j]}(u_i, u_j)$ of $C$, as long as $\theta, K > 0$, $0 < \alpha < 1$ and $\beta \in \mathbb{R}$, or as long as $\theta, K > 0$, $\alpha = 0$ and $\beta < 0$,

$$\lambda_{ij}^U = 1.$$

A more interesting case however arises if $\theta, K > 0$, $\alpha = 1$ and $\beta \in \mathbb{R}$. Then,

$$\lambda_{ij}^U = 2\Phi\Big[ -\theta \sqrt{\frac{1-\rho_{ij}}{2}} \Big],$$

meaning that tail asymmetry becomes possible through the correlation matrix $A = \{\rho_{ij}\}$.

The upper tail dependence coefficient can also be made such that $\lambda_{ij}^U = 0$: one need to let $\theta, K > 0$, $\alpha > 1$ and $\beta \in \mathbb{R}$.

For more examples and details, refer to Krupskii et al. (2016).

**Farlie-Gumbel-Morgenstern copulas.** Let the inner copula of a bidimensional EOFC be such that $C^{\diamond}_{Q_0(u_0)} = \Pi$, while $C_1, C_2$ are Farlie-Gumbel-Morgenstern copulas with parameters $\theta_1$ and $\theta_2$. Then, the resulting outer copula is a FGM copula with parameter $\frac{\theta_1 \theta_2}{3}$. Although a bit tedious, the proof is trivial. Also refer to Example 3.3.16.

Note that in a bivariate FGM with parameter $\theta \in [-1, 1]$, the related Kendall's tau is $\tau = \frac{2\theta}{9}$ (Nelsen, 2006, p. 162). This means that even if the parameters $\theta_1$ and $\theta_2$ of $C_1$ and $C_2$ are set to their upper limit, the parameter of the resulting FGM will be 1/3, corresponding to a rather low Kendall's tau of 0.074. The model where $C^{\diamond}_{Q_0(u_0)} = \Pi$, while $C_1, C_2$ are FGM copulas is thus trapped between a correlation of -0.074 and 0.074 only.

In an EOFC, the inner copula can be viewed as the baseline level of dependence. Instead of letting $C^{\diamond}_{Q_0(u_0)} = \Pi$, one could let $C^{\diamond}_{Q_0(u_0)} = M$, where $M$ is the upper Fréchet-Hoeffding bound. In this scheme, the two FGM copulas

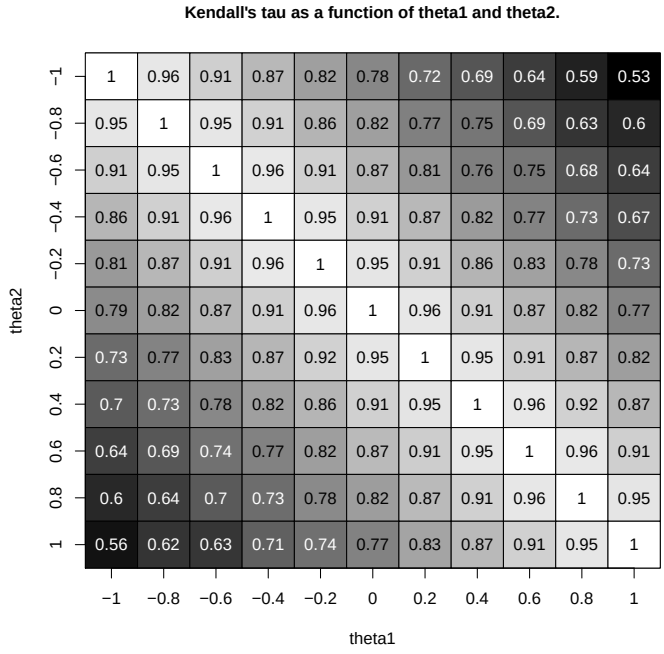



Figure 3.2: Empirical Kendall's tau for an EOFC where the inner copula is $M$ and $C_1, C_2$ are FGM copulas, described through $\theta_1 \in [-1, 1]$ and $\theta_2 \in [-1, 1]$.

$C_1, C_2$ can only be used to decrease the dependence between the two random variables of interest, $U_1$ and $U_2$. Figure 3.2 shows the empirical Kendall's tau calculated (sample size = 1000) for various combinations of $\theta_1$ and $\theta_2$ when the inner copula is $M$. As one can see, the improved model is now trapped between a correlation of 1 and $\sim 0.55$. This range can thus be tuned by changing the inner copula.

**Hierarchical NEOFCs.** OFCs are not suited to model hierarchical dependence. Assume that one wishes to build a one-factor copula on $(U_1, U_2, U_3, U_4)$ such that the resulting model is described by the following matrix of Kendall's taus:

$$\begin{array}{c|cccc} & U_1 & U_2 & U_3 & U_4 \\ U_1 & & 1 & 0.25 & 0.25 \\ U_2 & & & 0.25 & 0.25 \\ U_3 & & & & 1 \\ U_4 & & & & \end{array} \tag{3.5.1}$$

Since $U_1$ and $U_2$ are related through the maximum value of 1, it means that their bivariate margin must be

$$C^{[1,2]}(u_1, u_2) = \min(u_1, u_2) = \int_0^1 \frac{\partial \min(u_1, u_0)}{\partial u_0} \frac{\partial \min(u_2, u_0)}{\partial u_0} du_0.$$

Similarly, the bivariate margin of $U_3$ and $U_4$ is

$$C^{[3,4]}(u_3, u_4) = \min(u_3, u_4) = \int_0^1 \frac{\partial \min(u_3, u_0)}{\partial u_0} \frac{\partial \min(u_4, u_0)}{\partial u_0} du_0.$$

Therefore, the OFC on $(U_1, U_2, U_3, U_4)$ has to be

$$C(u_1, u_2, u_3, u_4) = \int_0^1 \frac{\partial \min(u_1, u_0)}{\partial u_0} \frac{\partial \min(u_2, u_0)}{\partial u_0} \frac{\partial \min(u_3, u_0)}{\partial u_0} \frac{\partial \min(u_4, u_0)}{\partial u_0} du_0.$$

By proposition 3.3.3 however, this corresponds to $C(u_1, u_2, u_3, u_4) = \min(u_1, u_2, u_3, u_4)$, for which the correlation matrix is not (3.5.1). A OFC will never be able to reproduce a matrix of Kendall's taus such as the one in (3.5.1). In general, OFCs do not seem to be suited to model matrices of the form

$$\begin{array}{c|cccc} & U_1 & U_2 & U_3 & U_4 \\ U_1 & & \tau_{12} & \tau_0 & \tau_0 \\ U_2 & & & \tau_0 & \tau_0 \\ U_3 & & & & \tau_{34} \\ U_4 & & & & \end{array} \tag{3.5.2}$$

in which $\tau_{12}, \tau_{34} > \tau_0$.

On the other hand, it is possible to build NEOFCs for which the matrix of Kendall's taus is as in (3.5.2).

Start from Equation (3.2.11), and let $w = 3$, that is, we have a triple layer of linking copulas. Let $C_{t_3}^{\diamond_3} = \Pi$. For $j = 3$, let all linking copulas $C_{ij}$ be a Frank copula with a same parameter $\theta_3$. This layer gives the baseline level of dependence between the four random variables of interest. Next, use the second layer, $j = 2$, to couple $U_1$ and $U_2$, that is, let $C_{12}$ and $C_{22}$ be a Frank copula with a common paramater $\theta_2$, while $C_{32}$ and $C_{42}$ are set to the independence copula. Finally use the first layer, $j = 1$, to couple $U_3$ and $U_4$, that is, let $C_{31}$ and $C_{41}$ be a Frank copula with a common paramater $\theta_1$, while $C_{11}$ and $C_{21}$ are set to the independence copula. Table 3.4 allows one to visualize this scheme.

| | $i=1$ | $i=2$ | $i=3$ | $i=4$ | |
|---|---|---|---|---|---|
| $j=1$ | $\Pi$ | $\Pi$ | Frank | Frank | $\theta_1$ |
| $j=2$ | Frank | Frank | $\Pi$ | $\Pi$ | $\theta_2$ |
| $j=3$ | Frank | Frank | Frank | Frank | $\theta_3$ |
| $C^\diamond$ | $\Pi$ | | | | |

Table 3.4: A hierarchical NEOFC with three layers and an inner copula set to independence.

*Example* 3.5.1. In Table 3.4, set $\theta_3$ to a value of 5.74, corresponding to a Kendall'tau of 0.5 for the related linking copulas and $\theta_2 = \theta_1$ to a value of 6.73, corresponding to a Kendall's tau of 0.55 for the related linking copulas. The empirical Kendall's taus based on 10000 observations generated from the model through Algorithm 5 are as shown hereafter (also refer to the appendix chapter).

```
> round(cor(generated.data, method="kendall"), 3)
      [,1]  [,2]  [,3]  [,4]
[1,] 1.000 0.559 0.119 0.123
[2,] 0.559 1.000 0.118 0.121
[3,] 0.119 0.118 1.000 0.563
[4,] 0.123 0.121 0.563 1.000
```

While the correlation between $U_1$, $U_2$ and $U_3$, $U_4$ is strong, one can note that even though $\theta_3$ corresponds to a correlation of 0.5, the correlation for the couples $(U_1, U_3)$, $(U_1, U_4)$, $(U_2, U_3)$ and $(U_2, U_4)$ appears to have dampen to $\sim 0.12$. Pushing $\theta_3$ towards higher values can help: if $\theta_3$ is set to a value of 14.14, corresponding to a Kendall's tau of 0.75, the empirical Kendall's taus become as shown hereafter.

```
> round(cor(generated.data, method="kendall"), 3)
      [,1]  [,2]  [,3]  [,4]
[1,] 1.000 0.743 0.220 0.222
[2,] 0.743 1.000 0.219 0.222
[3,] 0.220 0.219 1.000 0.738
[4,] 0.222 0.222 0.738 1.000
```

This increase remains however slow and even with a high value for $\theta_3$, such as 38.28, corresponding to a Kendall's tau of 0.9, the correlation between $U_1$ and $U_3$ will not exceed 0.27.

To overcome the issue of low dependence between the pairs $(U_1, U_3)$, $(U_1, U_4)$, $(U_2, U_3)$ and $(U_2, U_4)$, one can make use of the inner copula to set the baseline level of dependence, rather than using a layer of identical linking copulas. See Table 3.5.

| | $i=1$ | $i=2$ | $i=3$ | $i=4$ | |
|---|---|---|---|---|---|
| $j=1$ | $\Pi$ | $\Pi$ | Frank | Frank | $\theta_1$ |
| $j=2$ | Frank | Frank | $\Pi$ | $\Pi$ | $\theta_2$ |
| $C^\diamond$ | Frank | | | | $\theta^\diamond$ |

Table 3.5: A hierarchical NEOFC with 2 layers and an inner copula set to a Frank copula.

*Example* 3.5.2. In Table 3.5, set $\theta^{\diamond}$ to a value of 14.14, corresponding to a Kendall's $\tau$ of 0.75, $\theta_2$ to 1.38, corresponding to a Kendall's tau of 0.15, and $\theta_1$ to 6.73, corresponding to a Kendall's $\tau$ of 0.55. The empirical Kendall's taus based on 10000 observations generated from the related hierarchical NEOFC are shown hereafter.

```
> round(cor(generated.data, method="kendall"), 3)
      [,1]  [,2]  [,3]  [,4]
[1,] 1.000 0.755 0.388 0.385
[2,] 0.755 1.000 0.387 0.385
[3,] 0.388 0.387 1.000 0.809
[4,] 0.385 0.385 0.809 1.000
```

As last example of a hierarchical NEOFC, we revisit the tree on the right of Figure 2.23 from Chapter 2. This tree was built on daily log returns and corresponds to the hierarchical NEOFC from Table 3.6, where the symbols $\{\mathrm{D}_0, \mathrm{D}_{1234}, \mathrm{D}_{56}, \mathrm{D}_{12}, \mathrm{D}_{34}\}$ on the right of the table are there to help one compare the table with the tree of interest.

| | $i=1$ | $i=2$ | $i=3$ | $i=4$ | $i=5$ | $i=6$ | |
|---|---|---|---|---|---|---|---|
| $j=1$ | $\Pi$ | $\Pi$ | Frank | Frank | $\Pi$ | $\Pi$ | $\mathrm{D}_{34}$ |
| $j=2$ | Frank | Frank | $\Pi$ | $\Pi$ | $\Pi$ | $\Pi$ | $\mathrm{D}_{12}$ |
| $j=3$ | $\Pi$ | $\Pi$ | $\Pi$ | $\Pi$ | Frank | Frank | $\mathrm{D}_{56}$ |
| $j=4$ | Frank | Frank | Frank | Frank | $\Pi$ | $\Pi$ | $\mathrm{D}_{1234}$ |
| $C^{\diamond}$ | Frank | | | | | | $\mathrm{D}_0$ |

Table 3.6: The hierarchical NEOFC corresponding to the tree structure on the right of Figure 2.23.

Note that, given a tree structure, the corresponding NEOFC will have as many layers as there are internal nodes in the tree when the inner copula is set to independence, or, when the inner copula is used as root node, as many layers as there are internal nodes in the tree minus one.

*Example* 3.5.3. Set the various parameters of Table 3.6 to the arbitrary values shown in Table 3.7. The empirical Kendall's taus based on 10000 observations generated from the related hierarchical NEOFC are shown hereafter.

```
> round(cor(data, method="kendall"), 3)
      [,1]  [,2]  [,3]  [,4]  [,5]  [,6]
[1,] 1.000 0.776 0.400 0.406 0.476 0.478
[2,] 0.776 1.000 0.404 0.407 0.475 0.475
[3,] 0.400 0.404 1.000 0.775 0.485 0.483
[4,] 0.406 0.407 0.775 1.000 0.489 0.487
[5,] 0.476 0.475 0.485 0.489 1.000 0.755
[6,] 0.478 0.475 0.483 0.487 0.755 1.000
```

Note that, in contrast to what is usually required for nested Archimedean copulas, the dependence in hierarchical NEOFCs is not required to be a nondecreasing function as one goes farther away from the root node. The empirical

matrix from the last example indeed shows that the dependence at node $D_{12}$ is $\sim 0.776$, that of node $D_{1234} = D_{1:4}$ is lower, around 0.400, but that of the root is higher than 0.400, at around 0.475.

| | $i=1$ | $i=2$ | $i=3$ | $i=4$ | $i=5$ | $i=6$ | |
|---|---|---|---|---|---|---|---|
| $j=1$ | $\Pi$ | $\Pi$ | $\tau_{34}=0.4$ | $\tau_{34}=0.4$ | $\Pi$ | $\Pi$ | $D_{34}$ |
| $j=2$ | $\tau_{12}=0.4$ | $\tau_{12}=0.4$ | $\Pi$ | $\Pi$ | $\Pi$ | $\Pi$ | $D_{12}$ |
| $j=3$ | $\Pi$ | $\Pi$ | $\Pi$ | $\Pi$ | $\tau_{56}=0.2$ | $\tau_{56}=0.2$ | $D_{56}$ |
| $j=4$ | $\tau_{1234}=0.1$ | $\tau_{1234}=0.1$ | $\tau_{1234}=0.1$ | $\tau_{1234}=0.1$ | $\Pi$ | $\Pi$ | $D_{1:4}$ |
| $C^\diamond$ | $\tau_0=0.75$ | | | | | | $D_0$ |

Table 3.7: Parameters for Example 3.5.3, expressed as Kendall's taus for convenience.

**Archimedean copulas.** Let $\psi$ be a completely monotonic function on $[0,\infty]$, that is, $(-1)^k d^k/dt^k \psi(t) \geqslant 0$ for all integers $k$ and all $t > 0$, and such that $\psi(0) = 1$ while

$$\psi(\infty) = \lim_{t\to\infty} \psi(t) = 0.$$

If a copula $C$ can be written as

$$C(u_1,\dots,u_d) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d)),$$

then it is called an Archimedean copula with generator $\psi$ (McNeil, 2008). Let us note that, in order to make sure $C$ is a proper copula, the above-mentioned conditions on $\psi$ are sufficient, but not necessary. For sufficient and necessary conditions, see McNeil and Nešlehová (2009).

PROPOSITION 3.5.4. *In* (3.2.9)*, let* $C_{x_0} = \Pi$*, assume that the support of* $X_0$ *is* $[0,\infty]$*, and put*

$$C_i(u_i,u_0) = \int_0^{Q_0(u_0)} e^{-t\psi^{-1}(u_i)} f_0(t)\,dt,$$

*where*

$$\psi(x) = \int_0^\infty e^{-tx} f_0(t)\,dt,$$

$i \in \{1,\dots,d\}$*, with* $f_0$ *being the derivative of* $F_0$ *and* $Q_0$ *its inverse, with* $Q_0(u_0) = x_0$*. It can be checked that* $\psi$ *is completely monotonic, see for instance Joe (2001). Then* $C$*, the left-hand side of equation* (3.2.9)*, is an Archimedean copula with generator* $\psi$*.*

*Proof.* Checking that $C_i$ is a copula is straightforward. Moreover, since

$$\frac{\partial C_i(u_i,u_0)}{\partial u_0} = e^{-Q_0(u_0)\psi^{-1}(u_i)},$$

it holds that

$$C(u_1,\dots,u_d) = \int_0^\infty e^{-x_0 \sum_{i=1}^d \psi^{-1}(u_i)} f_0(x_0)\,dx_0 = \psi\Big(\sum_{i=1}^d \psi^{-1}(u_i)\Big).$$

□

Note that a reformulation of the above construction can be found in Joe (2001).

**Nested Archimedean copulas.** Archimedean copulas can be nested in order to get more flexible models. Nested Archimedean copulas (NACs) were introduced by Joe (2001) and have been the main topic of many research papers since, see for instance McNeil (2008), Hofert and Pham (2013), or Okhrin et al. (2013b). The simplest nested Archimedean copula consists of a bivariate Archimedean copula

$$C_{12}(u_1, u_2) = \psi_{12}(\psi_{12}^{-1}(u_1) + \psi_{12}^{-1}(u_2)),$$

which is nested into another bivariate Archimedean copula

$$C_{123}(\bullet, u_3) = \psi_{123}(\psi_{123}^{-1}(\bullet) + \psi_{123}^{-1}(u_3))$$

in order to get a copula of the form

$$\begin{aligned} C(u_1, u_2, u_3) &= C_{123}(C_{12}(u_1, u_2), u_3) \\ &= \psi_{123}\big(\psi_{123}^{-1}(\psi_{12}(\psi_{12}^{-1}(u_1) + \psi_{12}^{-1}(u_2))) + \psi_{123}^{-1}(u_3)\big). \end{aligned} \quad (3.5.3)$$

In general, an arbitrary pair of generators $(\psi_{123}, \psi_{12})$ does not ensure that the copula in Equation (3.5.3) is a proper copula. See Chapter 2 for more details on NACs. In this chapter, it is always assumed that the generators are such that the NAC is a proper copula.

PROPOSITION 3.5.5. *Define $\psi_{123}$ the same way $\psi$ was defined in Proposition 3.5.4. Also let $C_i$ as in Proposition 3.5.4. Further define*

$$C_{x_0}(u, v, w) = \exp\left( - x_0 \times \nu\left(\nu^{-1}\Big[\frac{1}{x_0}\log\Big(\frac{1}{u}\Big)\Big] + \nu^{-1}\Big[\frac{1}{x_0}\log\Big(\frac{1}{v}\Big)\Big]\right)\right) \times w;$$

*where $\nu(\bullet) = \psi_{123}^{-1}\big(\psi_{12}(\bullet)\big)$, $\nu(\bullet)^{-1} = \psi_{12}^{-1}\big(\psi_{123}(\bullet)\big)$ and $\psi_{12}(\bullet)$ is equal to the integral from 0 to $\infty$ of $\exp(-t\bullet)dF_{12}(t)$ with $F_{12}$ some distribution function. Then* (3.2.9) *is the copula given in* (3.5.3).

*Proof.* First, it is proven that $C_{x_0}$ is a copula. Define, for $0 \le u, v, w \le 1$,

$$\begin{aligned} G_{x_0}(u, v, w) = \exp\left( - x_0 \times \nu\left(\nu^{-1}\Big[\psi_{123}^{-1}\Big(u\Big)\Big] + \nu^{-1}\Big[\psi_{123}^{-1}\Big(v\Big)\Big]\right)\right) \\ \times \exp\left( - x_0 \times \psi_{123}^{-1}\Big(w\Big)\right). \end{aligned}$$

One can check that $C_{x_0}$ in Proposition 3.5.5 is the copula corresponding to the multivariate distribution $G_{x_0}$. And from Joe (2001), page 88, it can be easily deduced that $G_{x_0}$ is indeed a distribution function. Therefore $C_{x_0}$ is a proper copula.

The proof then proceeds by showing that $C$ in (3.2.9) is a nested Archimedean copula. $C$ in (3.2.9) becomes

$$C(u_1,u_2,u_3) = \int_0^1 \exp\Bigg( - Q_0(u_0) \times \nu\Big(\nu^{-1}\Big[\frac{1}{Q_0(u_0)}\log\Big(\frac{1}{C_{1|0}(u_1|u_0)}\Big)\Big] + \nu^{-1}\Big[\frac{1}{Q_0(u_0)}\log\Big(\frac{1}{C_{2|0}(u_2|u_0)}\Big)\Big]\Big)\Bigg) \times C_{3|0}(u_3|u_0)\,du_0. \tag{3.5.4}$$

In the proof of Proposition 3.5.4, it was shown that $\frac{\partial C_i(u_i,u_0)}{\partial u_0} = C_{i|0}(u_i|u_0) = \exp(-Q_0(u_0)\times\psi_{123}^{-1}(u_i))$. Replacing in (3.5.4) gives

$$\begin{aligned}
&\int_0^1 \exp\Bigg( - Q_0(u_0) \times \nu\Big(\nu^{-1}\Big[\psi_{123}^{-1}(u_1)\Big] + \nu^{-1}\Big[\psi_{123}^{-1}(u_2)\Big]\Big)\Bigg) \\
&\qquad\qquad \exp(-F_0^{-1}(u_0)\times\psi_{123}^{-1}(u_3))\,du_0 \\
&= \int_0^\infty \exp\Bigg( - x_0\Big[\psi_{123}^{-1}\Big(\psi_{12}\Big(\psi_{12}^{-1}(u_1)+\psi_{12}^{-1}(u_2)\Big)\Big) + \psi_{123}^{-1}(u_3)\Big]\Bigg) dF_0(x_0) \\
&= \psi_{123}\big(\psi_{123}^{-1}(\psi_{12}(\psi_{12}^{-1}(u_1)+\psi_{12}^{-1}(u_2))) + \psi_{123}^{-1}(u_3)\big),
\end{aligned}$$

which is the NAC from (3.5.3). □

**Gaussian copulas.** A Gaussian copula is a copula whose density $c$ satisfies

$$\log(c(u_1,\dots,u_d)) = -\frac{1}{2}\log(\det(R)) - \frac{1}{2}z^\top(R^{-1}-I)z, \tag{3.5.5}$$

where $R$ is a $d\times d$ invertible correlation matrix, $z=(z_1,\dots,z_d)^\top$ and $z_i$ is the quantile of order $u_i$ of the standard normal distribution. A Gaussian copula can be represented as in (3.2.9). Let $\Delta=(\Delta_1,\dots,\Delta_d)^\top$ be a real vector in $[0,1]^d$ and let $D$ be a diagonal matrix with elements given by $1-\Delta_i^2$, $i=1,\dots,d$. Finally let $C_A$ be a $d$-variate Gaussian copula with correlation matrix $A$.

PROPOSITION 3.5.6. *Let $C^\diamond_{Q_0(u_0)} = C^\diamond = C_A$ and let $C_i$ be a bivariate Gaussian copula with correlation $\Delta_i$, $i\in\{1,\dots,d\}$. Then the outer copula in* (3.2.9) *is a Gaussian copula with correlation matrix given by $R = D^{1/2}AD^{1/2} + \Delta\Delta^T$.*

*Proof.* Let $R$ be a symmetric nonnegative matrix whose diagonal elements are equal to 1 and whose element in the $i$-th row and $j$-th column is denoted

$\Delta_{ij}$. Let $(Z_1, \ldots, Z_d, Z_0)$ be distributed according to a $(d+1)$-variate centered Gaussian distribution with variance-covariance matrix given by

$$\begin{pmatrix} R & \Delta \\ {\scriptstyle d\times d} & {\scriptstyle d\times 1} \\ \Delta^T & 1 \\ {\scriptstyle 1\times d} & {\scriptstyle 1\times 1} \end{pmatrix},$$

so that $(Z_1, \ldots, Z_d | Z_0 = z_0) \sim N(\Delta z_0,\, R - \Delta\Delta^\top)$. The partial correlations are given by

$$\mathrm{Cov}(Z_i, Z_j | Z_0 = z_0) = \mathrm{Corr}(Z_i, Z_j | Z_0 = z_0) = \frac{\Delta_{ij} - \Delta_i \Delta_j}{\sqrt{(1-\Delta_i^2)(1-\Delta_j^2)}},$$

or, in other words, the partial correlation matrix is $A = D^{-1/2}(R - \Delta\Delta^\top)D^{-1/2}$. Given $Z_0 = z_0$, the margins $(Z_i | Z_0 = z_0)$ are $N(\Delta_i z_0, 1 - \Delta_i^2)$, hence

$$P(Z_i \le z | Z_0 = z_0) = \Phi((z - \Delta_i z_0)/\sqrt{1-\Delta_i^2}),$$

and, moreover, the corresponding copula is a Gaussian copula $C_A$, with correlation matrix $A$.

Calculate the copula corresponding to $(Z_1, \ldots, Z_d)$. Let $\Phi$ be the cumulative distribution function of the univariate standard Gaussian distribution.

$$\begin{aligned}
&C_{(Z_1,\ldots,Z_d)}(u_1, \ldots, u_d) \\
&= \int P(\Phi(Z_i) \le u_i,\, i = 1, \ldots, d | Z_0 = z_0) \Phi'(z_0) dz_0 \\
&= \int P(Z_i \le \Phi^{-1}(u_i),\, i = 1, \ldots, d | Z_0 = z_0) \Phi'(z_0) dz_0 \\
&= \int C_A \left\{ \Phi\left( \frac{\Phi^{-1}(u_i) - \Delta_i z_0}{\sqrt{1-\Delta_i^2}} \right), i = 1, \ldots, d \right\} \Phi'(z_0) dz_0 \\
&= \int_0^1 C_A \left\{ \Phi\left( \frac{\Phi^{-1}(u_i) - \Delta_i \Phi^{-1}(u_0)}{\sqrt{1-\Delta_i^2}} \right), i = 1, \ldots, d \right\} du_0.
\end{aligned}$$

This expression corresponds exactly to the copula given in (3.2.9), with

$$\frac{\partial C_i(u_i, u_0)}{\partial u_0} = \Phi\left( \frac{\Phi^{-1}(u_i) - \Delta_i \Phi^{-1}(u_0)}{\sqrt{1-\Delta_i^2}} \right),$$

see Meyer (2013) for details.

□

Note: in the case $d = 2$, the outer copula is a Gaussian copula with correlation given by

$$\rho_A \sqrt{1-\Delta_1^2}\sqrt{1-\Delta_2^2} + \Delta_1 \Delta_2.$$

Also see Example 3.3.18.

**C-Vine copulas.** Let $(U_0, U_1, \ldots, U_d)$ be a random vector following a C-Vine copula distribution truncated after the second level. The density of this truncated C-Vine is given by

$$c(u_0, \ldots, u_d) = \prod_{i=1}^{d-1} c^*_{1,1+i|0}(C^*_{1|0}(u_1|u_0), C^*_{1+i|0}(u_{1+i}|u_0)|u_0) \prod_{i=1}^{d} c^*_i(u_i, u_0) \tag{3.5.6}$$

where $c^*_i = \partial^2 C^*_i(u_i, u_0)/\partial u_i \partial u_0$, $C^*_{i|0}(u_i|u_0) = \partial C^*_i(u_i, u_0)/\partial u_0$, $\{C^*_i(u_i, u_0)\}$ is a set of $d$ arbitrary bivariate copulas, and $\{c^*_{1,1+i|0}\}$ is a set of $d-1$ abritrary copula densities for each $u_0$. Due to their extreme flexibility and ease of use (one only has to specify sets of bivariate copulas), Vine copulas have been used in an increasing number of applications and are still a hot topic of research, see for instance Aas et al. (2009), Kurowicka and Joe (2011) or Bedford and Cooke (2002).

Proposition 3.5.7. *If, in* (3.3.6)*, where* $T_1 = U_0$ *and* $t_1 = u_0$*, for each* $u_0$*,* $c_{u_0}$ *is defined as*

$$c_{u_0}(u_1, \ldots, u_d) = \prod_{i=1}^{d-1} c^*_{1,1+i|0}(u_1, u_{1+i}|u_0),$$

*and* $c_i(u_i, u_0) = c^*_i(u_i, u_0)$ *for all* $i$*, then the outer copula* $c$ *in* (3.3.6) *is the* $d$*-variate marginal distribution, with respect to* $u_0$*, of* (3.5.6)*, that is, its density is*

$$\begin{aligned} c(u_1, \ldots, u_d) &= \int_0^1 c_{u_0}(C^*_{1|0}(u_1|u_0), \ldots, C^*_{d|0}(u_d|u_0)) \prod_{i=1}^{d} c^*_i(u_i, u_0) du_0 \\ &= \int_0^1 \prod_{i=1}^{d-1} c^*_{1,1+i|0}(C^*_{1|0}(u_1|u_0), C^*_{1+i|0}(u_{1+i}|u_0)|u_0) \prod_{i=1}^{d} c^*_i(u_i, u_0)\, du_0. \end{aligned}$$

If one assumes that, in (3.5.6), none of the elements of $\{c^*_{1,1+i|0}\}$ actually depends on $u_0$, then the inner copula in Proposition 3.5.7 becomes

$$c_{u_0}(u_1, \ldots, u_d) = \prod_{i=1}^{d-1} c^*_{1,1+i}(u_1, u_{1+i}),$$

which is nothing more than a C-Vine on $(U_1, \ldots, U_d)$, truncated at the first level.

**$p$-factor models.** Define respectively $\Pi_1$-factor and $\Pi_2$-factor copulas as copulas of the form

$$C^{(\Pi_1)}(u_1,\dots,u_d) = \int_0^1 \prod_{i=1}^d C_{i|0}^{(2)}(u_i|v_2)\,dv_2, \text{ and} \tag{3.5.7}$$

$$C^{(\Pi_2)}(u_1,\dots,u_d) = \int_0^1 \int_0^1 \prod_{i=1}^d C_{i|0}^{(2)}(C_{i|0}^{(1)}(u_i|v_1)|v_2)\,dv_2\,dv_1, \tag{3.5.8}$$

where $C_{i|0}^{(k)}(u_i|v_k) = \partial C_i^{(k)}(u_i,v_k)/\partial v_k$ for $k=1,2$ and $i=1,\dots,d$, and where the $C_i^{(k)}$ are (arbitrary) bivariate copulas. $\Pi_1$-factor and $\Pi_2$-factor copulas have been studied in Krupskii and Joe (2013, 2015) as copula models for conditionally independent variables given respectively one and two latent factors.

The following (trivial) proposition aims at recovering $\Pi_1$-factor and $\Pi_2$-factor copulas as special cases of the model (3.2.9).

PROPOSITION 3.5.8. *Consider the copulas given in* (3.5.7) *and* (3.5.8). *In* (3.2.9), *put* $C_i = C_i^{(2)}$. *If, moreover,* $C^{\diamond}_{Q_0(u_0)} = \Pi$ *for each* $u_0$, *then the outer copula* $C$ *in* (3.2.9) *is the* $\Pi_1$*-factor copula given in* (3.5.7). *Likewise, if, in* (3.2.9), $C_i = C_i^{(1)}$ *and moreover,*

$$C^{\diamond}_{Q_0(u_0)}(u_1,\dots,u_d) = \int_0^1 \prod_{i=1}^d C_{i|0}^{(2)}(u_i|\tilde{u}_0)\,d\tilde{u}_0,$$

*then the outer copula* $C$ *in* (3.2.9) *is the* $\Pi_2$*-factor copula given in* (3.5.8).

Note that $C^{\diamond}_{Q_0(u_0)}$ in the above proposition actually does not depend on $u_0$ hence we have conditional invariance. This restriction can be easily removed as follows. Let, for each $u_0$, $\{\widetilde{C}_i(\bullet,\bullet;u_0)\}$ be a set of bivariate copulas and

$$C^{\diamond}_{Q_0(u_0)}(u_1,\dots,u_d) = \int_0^1 \prod_{i=1}^d \widetilde{C}_{i|0}(u_i|\tilde{u}_0;u_0)\,d\tilde{u}_0,$$

where $\widetilde{C}_{i|0}(u_i|\tilde{u}_0;u_0) = \partial \widetilde{C}_i(u_i,\tilde{u}_0;u_0)/\partial\tilde{u}_0$. The outer copula is then

$$C(u_1,\dots,u_d) = \int_0^1 \int_0^1 \prod_{i=1}^d \widetilde{C}_{i|0}(C_{i|0}(u_i|u_0)|\tilde{u}_0;u_0)\,d\tilde{u}_0\,du_0.$$

## 3.6 Inference

This section discusses estimation of models of the form (3.2.9) or (3.2.11), as well as how one can test for conditional independence for models of the form (3.2.9).

### 3.6.1 Estimation

Let, in (3.2.9), $C_i(u_i, u_0) = C_i(u_i, u_0; \alpha_i)$ and

$$C_{x_0}(u_1, \ldots, u_d) = C_{x_0}(u_1, \ldots, u_d; \beta(x_0)),$$

where $\beta$ is a mapping which, to each $x_0$ in the support of $X_0$, associates a parameter in the appropriate parameter space. If the mapping $\beta$ depends on a vector of parameters, as in (3.3.2), we also denote this vector by $\beta$. Likewise, we denote the parameter vector which contains the parameters of the quantile function $Q_0$ of $X_0$ by $\lambda$. Accordingly, the notation for the copula of $(U_1, \ldots, U_d | U_0 = u_0)$ becomes $C^{\diamond}_{Q_0(u_0)}(u_1, \ldots, u_d) = C^{\diamond}(u_1, \ldots, u_d; \beta, \lambda)$. Finally, let us denote by $(x_{h1}, \ldots, x_{hd})$, $h = 1, \ldots, n$, the sample of the distribution $F$ with margins $F_1, \ldots, F_d$ and copula $C$.

The pseudo log likelihood function to maximize is

$$L_n(\boldsymbol{\theta}) = \sum_{h=1}^{n} \log \int_0^1 c\left[C_{1|0}(\widehat{F}_1(x_{h1})|u_0; \alpha_1), \ldots, C_{d|0}(\widehat{F}_d(x_{hd})|u_0; \alpha_d); \beta, \lambda\right] \times \prod_{i=1}^{d} c_i(\widehat{F}_i(x_{hi}), u_0; \alpha_i)\, du_0, \quad (3.6.1)$$

where $\boldsymbol{\theta}$ stands for the complete parameter vector, that is, $\boldsymbol{\theta} = (\alpha, \beta, \lambda)$, $\alpha = (\alpha_1, \ldots, \alpha_d)$ and $\widehat{F}_i$ denotes an estimate of $F_i$, $i \in \{1, \ldots, d\}$. There are many ways to estimate $F_i$. For instance, $\widehat{F}_i$ may be the empirical distribution function, as in Genest et al. (1995), or may be a parametric estimate, as in Joe and Xu (1996).

Regarding the computational aspects, especially in higher dimensions and for large datasets, the likelihood (3.6.1) may be costly to compute due to the repeated use of integrals (as many as the sample size). A brief discussion on these computational aspects are given in Section 3.7.1, also see the appendix chapter.

### 3.6.2 Testing for conditional independence

This section provides procedures to test for conditional independence in models based on the representation (3.2.9). Indeed, being able to assess if the variables of interest are dependent or independent conditioned on the latent factor seems a crucial issue. Conditional independence would mean that the factor captures all the dependence in the data whereas no conditional independence would mean that there is a remaining, intrinsic dependence in the variables even though the factor has been accounted for.

Throughout this subsection, the bivariate copulas $C_i$, $i \in \{1, \ldots, d\}$, are assumed to belong to some known parametric families. The inner copula $C_{u_0} =$

$C^{\diamond}_{Q_0(u_0)}$, however, can be left fully unspecified: either nothing is known about it, either a parametric form is assumed. The possibility to carry out a hypothesis test in this setting is new to the literature.

The hypothesis test for conditional independence is of the form

$$\begin{aligned} H_0 : C^{\diamond}_{Q_0(u_0)} &= \Pi \text{ for all } u_0 \\ \text{versus } H_1 : \text{ there exists some } u_0 &\text{ such that } C^{\diamond}_{Q_0(u_0)} \neq \Pi \end{aligned}$$

(recall that $\Pi$ stands for the independence copula or the product function), where for two functions $f$ and $g$, $f = g$ means that $f(t) = g(t)$ for all $t$ in their domain.

If a certain parametric form is assumed for $C^{\diamond}_{Q_0(u_0)}$, such as in Section 3.2, then most likely the test will reduce to testing for a parameter to equate a certain value, and no conceptual difficulties refrain the task. For instance, in (3.3.2), testing for conditional independence amounts to testing for $\beta_0 = \infty$ or $\beta_1 = \infty$ (conceptually). Let us remark that testing for conditional invariance is also feasible: one need to test $\beta_1 = 0$.

If $C^{\diamond}_{Q_0(u_0)}$ is left fully unspecified, the alternative hypothesis needs to be slightly restricted in order for a test to exist. Consider

$$\begin{aligned} H_0 : C^{\diamond}_{Q_0(u_0)} &= \Pi \text{ for all } u_0 \\ \text{versus } H_1 : C^{\diamond}_{Q_0(u_0)} &> \Pi \text{ for all } u_0. \end{aligned} \tag{3.6.2}$$

Then, the alternative hypothesis is: "conditioned on the factor, the variables of interest are positively dependent".

Let $\pi$ be the risk of type I error. One rejects $H_0$ if $T_n \leq c_\pi$, where $c_\pi$ is chosen so that $P_{H_0}(T_n \leq c_\pi) = \pi$ and where

$$T_n = \sup_{t \in [0,1]^d} (M(t) - \widehat{C}(t)), \tag{3.6.3}$$

where $M(t) = M(t_1, \ldots, t_d) = \min(t_1, \ldots, t_d)$ is the Fréchet-Hoeffding upper bound for copula and

$$\widehat{C}(t) = \frac{1}{n} \sum_{h=1}^{n} \mathbf{1}(\{\widehat{F}_i(X_{hi}) \leq t_i\},\, i = 1, \ldots, d) \tag{3.6.4}$$

is the empirical estimator of $C$ ; $(X_{h1}, \ldots, X_{hd})$, $h \in \{1, \ldots, n\}$, being the data ; and $\widehat{F}_i$ being the empirical distribution function of $X_{hi}$.

The heuristic underlying the expression of $T_n$, the test statistic, is as follows. Denote by $C^{(H_0)}$ the outer copula under $H_0$, that is, one substitutes $\Pi$ for the inner copula $C^{\diamond}_{Q_0(u_0)}$ in (3.2.9) and gets

$$C^{(H_0)}(u_1, \ldots, u_d) = \int_0^1 \prod_{i=1}^{d} C_{i|0}(u_i|u_0)\, du_0. \tag{3.6.5}$$

If $H_0$ is true, then the true copula $C$ verifies $C = C^{(H_0)}$. But if $H_0$ is false, $C_{u_0} > \Pi$ implies $C > C^{(H_0)}$. In order to reject the null, one could therefore compare the empirical copula $\widehat{C}$ to $\widehat{C}^{(H_0)}$, the latter being an estimate of $C^{(H_0)}$, and reject the null if $\widehat{C} - \widehat{C}^{(H_0)}$ is too large. However, as the distribution of this difference under $H_0$ is unknown, one would need to use bootstrap and fit $C^{(H_0)}$ to simulated data a large number of times, a process far too expensive for most hardware.

To overcome this issue, the comparison of $\widehat{C}$ to $\widehat{C}^{(H_0)}$ is done indirectly using the Fréchet-Hoeffding upper bound as a baseline (note that the Fréchet-Hoeffding lower bound for copulas could also be considered as a baseline for comparison, the upper one was however picked since it has the nice property to be a copula $\forall d$, although this property is by no means required for the test). The result of this suggestion is (3.6.3), which describe a test statistic always positive and expected to take small values when $H_0$ is false.

The bootstrap of $T_n$ is now described. Note that under $H_0$, the outer copula in (3.6.5) is fully parametric: one can obtain an estimate $\widehat{C}^{(H_0)}$ by maximum pseudo-likelihood (Krupskii and Joe, 2013). New bootstrap samples (say $N$ of them) are then drawn using Algorithm 4 in order to get $N$ test statistics $T_n^{(1)}, \dots, T_n^{(N)}$. These can be used, for instance, to compute a p-value as $P_{H_0}\left(T_n \leq T_n^{(obs)}\right) \approx N^{-1} \sum_{k=1}^{N} \mathbf{1}\left(T_n^{(k)} \leq T_n^{(obs)}\right)$. In other words, the p-value is estimated by counting how many test statistics in $\{T_n^{(t)}\}, t \in \{1, \dots, N\}$, are lower or equal than the test statistic $T_n^{(obs)}$, built using the data from the original sample.

Finally, let us note that the test $H_0 : C^{\diamond}_{Q_0(u_0)} = \Pi$ against $H_1 : C^{\diamond}_{Q_0(u_0)} < \Pi$ can be carried out by considering (3.6.3) again, but this time with a rejection region on the right, that is, we reject $H_0$ if $T_n \geqslant c_\pi$, where $c_\pi$ is chosen so that $P_{H_0}(T_n \geq c_\pi) = \pi$.

## 3.7 Illustrations

The purpose of this section is to illustrate how one can take advantage of the framework presented in Section 3.2 in practice. We first provide a few technical details on how some numerical operations were performed.

### 3.7.1 Computational aspects

In this chapter, log-likelihoods are maximized using either gradient descent algorithms, which can be found in the `optim` function of the statistical software `R`, or differential evolution (Storn and Price, 1997; Qin and Suganthan, 2005; Price et al., 2006), for which a `R` package also exists: `DEoptim` (Mullen et al.,

2011).

Note that gradient descent algorithms usually require to provide a starting parameter vector. It is advised to try several such points and retain the best result for the likelihood.

For the numerical evaluation of the integrals in (3.6.1), refer to the appendix chapter.

### 3.7.2 Revisiting the daily log returns from Section 2.6, Chapter 2

In Section 2.6, Chapter 2, daily log returns from January 2010 to December 2012 for 6 indices were gathered with the help of Yahoo! Finance and analyzed using nested Archimedean copulas. The result was the estimation of the two tree structures in Figure 2.23.

In this subsection, we fit the hierarchical NEOFC described by Table 3.6 on the same data, in an effort to reproduce the results from Section 2.6. This is performed by maximization of the corresponding likelihood using the `DEoptim` function. The estimated parameters for this model are, expressed as Kendall's taus for convenience,

- $\tau_0 = 0.176813$,
- $\tau_{1234} = 0.072114$,
- $\tau_{12} = 0.251929$,
- $\tau_{34} = 0.354454$,
- and $\tau_{56} = 0.383513$.

Also refer to Table 3.7.

The Kendall's taus between the random variables of interest induced by the fitted model are however unknown but can be calculated through simulation. Hereafter is the matrix of Kendall's taus between the random variables of interest, calculated using 10000 observations from the fitted hierarchical NEOFC.

```
      ANF AMZN  ChM  PCh GBLB  KBC
 ANF 1.00 0.28 0.13 0.14 0.12 0.12
AMZN .... 1.00 0.12 0.12 0.13 0.13
 ChM .... .... 1.00 0.34 0.10 0.11
 PCh .... .... .... 1.00 0.10 0.11
GBLB .... .... .... .... 1.00 0.36
 KBC .... .... .... .... .... 1.00
```

This can be compared to the matrix of Kendall's taus induced by the tree structure on the right of Figure 2.23:

```
      ANF AMZN  ChM  PCh GBLB  KBC
 ANF 1.00 0.31 0.08 0.08 0.18 0.18
AMZN .... 1.00 0.08 0.08 0.18 0.18
 ChM .... .... 1.00 0.35 0.18 0.18
 PCh .... .... .... 1.00 0.18 0.18
GBLB .... .... .... .... 1.00 0.44
 KBC .... .... .... .... .... 1.00
```

The Kendall's taus from the fitted hierarchical NEOFC however match the ones induced by the model on the left of Figure 2.23 more:

```
      ANF AMZN  ChM  PCh GBLB  KBC
 ANF 1.00 0.31 0.14 0.14 0.14 0.14
AMZN .... 1.00 0.14 0.14 0.14 0.14
 ChM .... .... 1.00 0.35 0.14 0.14
 PCh .... .... .... 1.00 0.14 0.14
GBLB .... .... .... .... 1.00 0.44
 KBC .... .... .... .... .... 1.00
```

In conclusion, the hierarchical NEOFC described by Table 3.6, fitted on the data from Section 2.6, Chapter 2, is able to capture a similar hierarchical dependence structure as the one captured by the tree on the left of Figure 2.23.

### 3.7.3 Power of the test for conditional independence

In this section, we study the power of the test statistic $T_n$ in (3.6.3) by means of a simulation experiment. We considered the test (3.6.2) and set the type I error risk to $\pi = 0.1$. We drew $N = 500$ datasets of size $n = 50$ and 500 from the model (3.2.9), with $d = 3$ and $C_i$ being Clayton copulas as in (3.3.1) with parameters $\alpha_i$, $i = 1, 2, 3$, such that Kendall's $\tau$ coefficients are equal to 0.4 for $i = 1$, 0.5 for $i = 2$ and 0.6 for $i = 3$. The inner copula $C_{x_0}$ was a normal copula as in (3.5.5) with correlation matrix

$$R = \begin{pmatrix} 1 & & \\ & \ddots & \beta \\ & \beta & \ddots \\ & & & 1 \end{pmatrix}, \tag{3.7.1}$$

for $\beta = 0.0, 0.1, 0.2, 0.3, 0.4$, and, finally, 0.5. (There are $N = 500$ samples of size $n = 50$ and 500 for each $\beta$). Note that $\beta = 0$ corresponds to the null hypothesis $H_0$.

For each sample, $k = 1, \ldots, N$, we calculated a $p$-value $p^{(k)}$ based on 200 boostrap replications. That is, we calculated the proportion of 200 simulated test statistics that where lower or equal than the observed one. As rejection occurs whenever the $p$-value is lower or equal to the type I error risk $\pi$, we approximated the power by the proportion of the $p^{(k)}$ falling below $\pi$. See Section 3.6.2 for details.

Figure 3.3 shows the estimated power of $T_n$ in (3.6.3). As it was expected, the power of the test increases as $n$ and $\beta$ grow. Furthermore the power is equal to the type I error risk $\pi$ under the null, that is when $\beta = 0$.

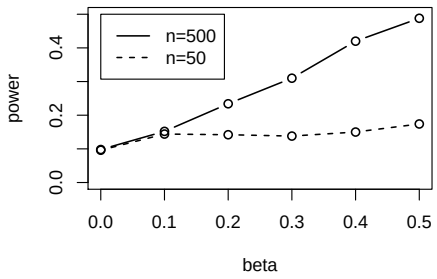



Figure 3.3: Power of (3.6.3) as a function of $\beta$, for the setting described above.

### 3.7.4 Estimating the distribution of a financial market through the dependence of its individual assets

It is commonly assumed that dependence within financial markets is higher in "crisis times" than in "stable times" (see e.g. Tajvidi et al. (2015) for a statistical analysis supporting this view). If one wishes to turn this assumption into a statistical model, then certainly the approach developed in this paper would be useful. Indeed, one would let $X_0$ be the crisis indicator and $C_{x_0}$ account for the dependence in the market as a function of its state — state which would range from "no crisis", coded by the number 0, to "extreme crisis", coded by the number 1, with values usch as 0.5, 0.2 or 0.7 describing an in-between state.

Once a particular model would have been chosen, many things may be of interest. One may be interested in estimating the latent crisis indicator distribution and study its evolution through time. Or would assess the goodness-of-fit of the model in order to infirm or confirm it, in particular the functional dependence related to the latent factor, or, in other words, how the individual assets respond to the market state.

#### Data

Our market consists of $d = 10$ arbitrary components[1] of the NASDAQ index. We gathered weekly data from Yahoo! Finance at `http://www.yahoo.com/` between 2005 and 2013. The log-return at the $h$-th week and $k$-th year for the $i$-th component is denoted

$$X_{hi}^{(k)} = \log(V_{hi}^{(k)}/V_{h-1,i}^{(k)}),$$

[1] Alexion, Apple, Biogen, Rob, Citrix, Costco, EA, Fast, Garmin and Henry

where $V_{hi}^{(k)}$ stands for the raw prices. The log-returns are uniformized as

$$R_{hi}^{(k)}/(n_k+1),$$

where $n_k$ is the number of observations for the $k$-th year (usually 52) and $R_{hi}^{(k)}$ is the rank of $X_{hi}^{(k)}$ among $X_{1i}^{(k)},\dots,X_{ni}^{(k)}$. The latent crisis indicator at the $h$-th week and $k$-th year is denoted $X_{h0}^{(k)}$. For the sake of simplification, we assume that the vectors $(X_{h0}^{(k)},X_{h1}^{(k)},\dots,X_{hd}^{(k)})$, $h=1,\dots,n$, are independent and identically distributed for each fixed $k$.

**Models and methods**

Our choice for the 3 components required (remember Section 3.2) to build a parametric model are given here. For convenience, the latent crisis indicator is assumed to be beta distributed, so that it has a flexible distribution over the interval $[0,1]$. So, for each year $k$,

$$f_0^{(k)}(x_0;\lambda_1^{(k)},\lambda_2^{(k)})=\frac{\Gamma(\lambda_1^{(k)}+\lambda_2^{(k)})}{\Gamma(\lambda_1^{(k)})\Gamma(\lambda_2^{(k)})}x_0^{\lambda_1^{(k)}-1}(1-x_0)^{\lambda_2^{(k)}-1},$$

where $0\le x_0\le 1$, $\lambda_1^{(k)},\lambda_2^{(k)}>0$ and $\Gamma$ is the Gamma function defined in (3.2.4). Consequently, the factor's median may also be interpreted as a deterministic crisis indicator: the closer the median is to 1, the more likely we are in a crisis. The copula of $(X_{h0}^{(k)},X_{hi}^{(k)})$ is assumed to be a Frank copula for all $i=1,\dots,d$ and all $k$, that is,

$$C_i^{(k)}(u_i,u_0;\alpha_i^{(k)})=-\frac{1}{\alpha_i^{(k)}}\log\left(1+\frac{(e^{-\alpha_i^{(k)}u_i}-1)(e^{-\alpha_i^{(k)}u_0}-1)}{e^{-\alpha_i^{(k)}}-1}\right),$$

where $\alpha_i^{(k)}\neq 0$ and $-\infty<\alpha_i^{(k)}<\infty$ (see e.g. Nelsen (2006) p. 116). The inner copula is a Gaussian copula with an exchangeable correlation matrix, so that $c_{x_0}$ has formula (3.2.5) with

$$R(x_0)=\begin{pmatrix}1 & & & \\ & \ddots & x_0 & \\ & x_0 & \ddots & \\ & & & 1\end{pmatrix}.$$

In other words, the dependence between the individual assets increases linearly as the crisis becomes more severe.

For each year, there were 12 parameters to estimate: 10 for the 10 bivariate Frank copulas $(\alpha_1^{(k)},\dots,\alpha_d^{(k)})$ and 2 for the latent factor beta distribution

$(\lambda_1^{(k)}, \lambda_2^{(k)})$. Estimation was performed by pseudo maximum likelihood, as described in Section 3.6.1.

In order to assess the goodness of fit of our model, we compared the average of the pairwise Kendall's tau coefficient model-based estimates to its empirical counterpart. Let us focus first on the empirical coefficient. Denote by $\tau_{ii'}^{(k)}$ (respectively $\hat{\tau}_{ii'}^{(k)}$) the true (respectively empirical) Kendall's tau coefficient between the $i$-th and $i'$-th individual assets for the $k$-th year. Let $\tau^{(k)} = \sum_{i<i'} \tau_{ii'}^{(k)}/(d(d-1)/2)$ and $\overline{\tau}^{(k)} = \sum_{i<i'} \hat{\tau}_{ii'}^{(k)}/(d(d-1)/2)$ be the respective pairwise averages. Recall that the empirical estimate of Kendall's tau coefficient between the $i$-th and $i'$-th individual assets for the $k$-th year is given by

$$\hat{\tau}_{ii'}^{(k)} = \binom{n}{2}^{-1} \sum_{h<h'} \operatorname{sign}\left((X_{hi}^{(k)} - X_{h'i}^{(k)})(X_{hi'}^{(k)} - X_{h'i'}^{(k)})\right),$$

where $\operatorname{sign}(x) = 1$ if $x > 0$, $-1$ if $x < 0$ and 0 if $x = 0$.

Confidence intervals around $\tau^{(k)}$ can be built as follows. Let $\boldsymbol{\tau}^{(k)}$ be the vector whose coordinates are the $\tau_{ii'}^{(k)}$ and let $\hat{\boldsymbol{\tau}}^{(k)}$ be its empirical counterpart. The vector $\sqrt{n}(\hat{\boldsymbol{\tau}}^{(k)} - \boldsymbol{\tau}^{(k)})$ tends to a centered normal distribution of dimension $d(d-1)/2$. The asymptotic variance-covariance matrix can be estimated by bootstrap from formula (12) in Mazo et al. (2015b). The convergence in distribution of $\sqrt{n}(\overline{\tau}^{(k)} - \tau^{(k)})$ to a centered normal distribution and standard deviation, say $\sigma^{(k)}$, comes after applying the Delta method. Again, on can compute an estimate, say $\hat{\sigma}^{(k)}$, of $\sigma^{(k)}$ by bootstrap. As a result, one can easily compute a confidence interval of level 99% as $\overline{\tau}^{(k)} \pm z_{.995}\hat{\sigma}^{(k)}/\sqrt{n}$, where $z_{.995}$ is a number such that the probability of a standard normal variable to lie between $-z_{.995}$ and $+z_{.995}$ is 99%.

### Results

Figure 3.4 pictures the average of the pairwise Kendall's tau coefficient model-based estimates. The shaded area represents the 99% confidence intervals. The results presented in Figure 3.4 support the plausibility of our model as the curve lies inside the confidence intervals. In particular, these results also demonstrate the model flexibility, as the curve seems to "follow" the empirical confidence intervals.

Figure 3.5 pictures three indices normalized so that their shape through time could be drawn and compared on the same picture. The normalizations are of the form

$$f_{\text{normalized}}(t) = (f(t) - F_-)/(F_+ - F_-)$$

where $F_-$ and $F_+$ are the minimum and maximum values of $f$ respectively. Thus, up to normalization, the dashed line represents the NASDAQ loss rate, that is, $(X_{hi}^{(k)} - X_{hi}^{(k+1)})/X_{hi}^{(k)}$ for $k \in \{2007, \ldots, 2013\}$; the dotted line[2] repre-

[2]The data were downloaded from `http://www.cboe.com/data/putcallratio.aspx`

sents the put-call ratio on the CBOE total exchange volume (see below for an explanation) and the plain line represents the latent factor estimated median.

The put-call ratio[3] on a certain market is, as its name tells, the ratio between the put options and the call options on that market. Put options are contracts on assets which give one the right, but not the obligation, to sell that asset in the future at a price fixed today, so that a profit can be made if the price of the asset goes down. Conversely, a call option is a contract on a asset which allows one to buy in the future at a price fixed today. Thus, arguably, the ratio of the put to the call can be seen as the overall attitude of investors toward a financial market. As such, we might see it as a crisis indicator. Likewise, the index of a market, such as the NASDAQ, is commonly regarded as mirroring the state of some part of the economy, and, therefore, its loss rate can be seen again as a crisis indicator.

Therefore, we have at our disposal, on the one hand, two crisis indicators computed independently from our model, and our latent factor estimated median, which we chose to interpret as a crisis indicator. Arguably, if these three indices — the NASDAQ loss rate, the put-call ratio and the latent factor estimated median — exhibit a similar behavior, this would support, first, our choice to interpret $X_0$ as a crisis indicator, second, the functional form of our copula $C_{x_0}$ and third, losely speaking, our model.

In Figure 3.5, all the indices have a similar shape: that of the letter "M": it goes up, down, up and down again. Moreover, they all pick down at around the same location, corresponding to 2010. Also, they all pick up at around 2008, corresponding to the world financial crisis that took its root into the subprimes crisis in the summer of 2007. After 2010, they start to go up again, perhaps corresponding to the European sovereign debt crisis. While we have, admittedly, no expertise to discuss the relevancy or confidence one can have in these indices, it is still noticeable how they agree with each other, how they tell the same story. In particular, the behavior of our latent factor is consistent with the behaviors of the other crisis indicators. Therefore, we believe that our model passed an important empirical test and we hope to have convinced the reader of its usefulness in this situation.

[3]see e.g. `http://www.investopedia.com/terms/p/putcallratio.asp`

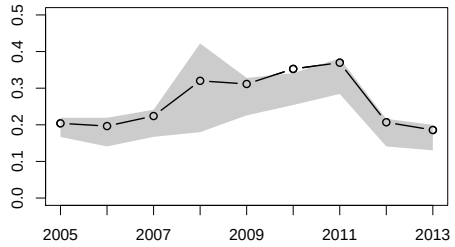



Figure 3.4: Pairwise Kendall's tau estimated coefficients average under the considered model along with empirical confidence intervals through time.

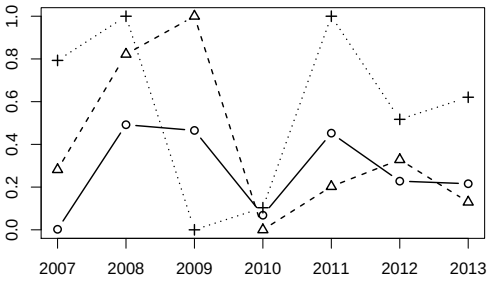



Figure 3.5: Three normalized financial markets trackers through time: the plain curve represents the latent factor estimated median, the dashed curve represents the NASDAQ loss rate and the dotted curve represents the put-call ratio.

## 3.8 Discussion

In this chapter, we extended the scope of one-factor copulas thanks to Equations (3.2.8), (3.2.9) and (3.2.11). Models built using these equations can now feature a varying conditional dependence structure and a factor's distribution not restricted to be the standard uniform. General dependence properties for these models were given and explicit examples were discussed. The usefulness of these models was illustrated by considering the estimation of the behavior of a financial market through the dependence of its individual components. Furthermore, a novel hypothesis test was constructed in order to assess whether conditional independence holds or not.

Note that although not part of this chapter, the appendix chapter of this thesis was built on top of it and showcases how one can in practice, using R, build models using Equations (3.2.8), (3.2.9) or(3.2.11). Explicit examples of code is given, as well as details on how numerical integration can be performed,

including a simulation study to compare five different numerical integration methods.

# General Conclusion

This thesis began with nested Archimedean copulas and their trees. Being able to nest Archimedean copulas is not only elegant, but also useful. It allows not only to break the bounds of regular Archimedean copulas, offering more flexibility, but also to really understand the link between the various variables at play, thanks to the tree structure arising from the nesting scheme.

Nonetheless, nested Archimedean copulas still fall short when it comes to flexibility. They might look flexible, and they certainly are more flexible than Archimedean copulas, but they remain too limited.

This fact was painfully visible while trying to estimate the tree structure from a NAC using the approach from Subsection 2.3.2 in Chapter 2. In this approach, pieces of the target tree are marginally estimated and then assembled to build a "global" estimated tree. When the data actually come from a nested Archimedean copula, the approach yields very good results. But when the data are not from a nested Archimedean copula, one always ends up with pieces that conflict with one another in such a way that a global estimated tree is impossible to retrieve. Yet, each marginal piece is strongly supported by the data. Even if there was a procedure allowing to correct some of the pieces so that they play well with the others, this will not change the fact that the fit of the resulting tree will be bad on the level where the correction was applied. How is it possible to have conflicting pieces that are all strongly supported by the data at the same time? Simple: the data are not coming from a nested Archimedean copula and fitting one will not accurately model the dependencies between the random variables of interest.

The approach developed in Subsection 2.4 is a response to this problem of conflicting pieces. Even if this problem is actually a sign that fitting a nested Archimedean copula to the data is not a good idea, one should still be able to do it. Subsection 2.4 thus gives the key on how to force a tree on the data, regardless of whether or not the underlying target copula is actually a NAC.

The last chapter of this thesis was devoted to factor copulas. One-factor copulas (OFCs) form a flexible class of copulas that's easy to understand and interpret. They however are unable to model certain hierarchical dependencies, as shown in Section 3.5. In Chapter 3, they are extended, allowing for more

flexibility, and NACs are shown to be a particular case of this extension. EOFCs and NEOFCs, developed in this thesis, still face many challenges, but they nonetheless offer flexibility on demand, interpretability and could find their place alongside vine copulas.

**Possible future work. Speaking at the first person in this part of the thesis.** While my desire to build new high-dimensional copulas remains strong, I feel the urge to start using the ones already developed or studied on real data. What are the applications of copulas? How are they used by researchers around the world? A small review of the literature outside of the realm of statistics lead me to some surprising results, such as a paper from Jiang et al. (2009) where copulas were used for the study of extra solar planets, one from Scherrer et al. (2009) on the study of large scale structures in the universe, one from Salinas-Gutiérrez et al. (2009) where copulas are used for mathematical optimization, one from Onken et al. (2009) on the study of short-term noise dependencies of spike-counts in macaque prefrontal cortex, or one from De Waal and van Gelder (2005), using copulas for the modeling of extreme wave heights and periods. These are only a few of the papers found. Most papers using copulas use bivariate copulas only, hinting at a huge potential for applications using $d$-dimensional copulas, with $d > 2$. As a former chemist with a strong background in physics and a general background in most natural sciences, it seems I'm in an excellent position to serve as bridge between the world of copulas and researchers from other fields of science.

Regarding new copula models, nested extended one-factor copulas are one of the most intriguing constructions of this thesis and are expected to fuel many future researches. Things of interest regarding them are, among other things, numerical challenges, identifiability issues and a better understanding of the properties of hierarchical NEOFCs.

Finally, coming back to hierarchy, a new copula model recently emerged: the hierarchical Kendall copula, as seen for instance in Brechmann (2013) or Brechmann (2014). This copula seems to retain the main quality of nested Archimedean copulas, which is its ability to be represented as a tree structure, while offering more flexibility. Estimation is, as far as I know, still an issue and could be the subject of future researches.

# Appendix: the OFC package

In this part of the thesis, related to Chapter 3, the three main functions of the OFC package are showcased, these functions allowing one to work with one-factor copulas, extended one-factor copulas and nested extended one-factor copulas, using R. Numerical integration is also discussed.

The three functions are `dOFC`, `pOFC` and `rOFC`. The outer copula one wishes to work with must be specified to these functions as a list, the first element of which being a data frame specifying the inner and bivariate (or linking) copulas. Example 1 shows how to use `dOFC` with a simple OFC, Example 2 shows how to use `pOFC` with an EOFC and Example 3 shows how to use `rOFC` with a NEOFC.

**Example 1: dOFC with a simple OFC.** One must here define the points at which the density has to be calculated, as well as the OFC of interest.

```
> myOFC
$innerandbiv
      i     j  family param1 param2
1 inner inner     ind     NA     NA
2     1     1   Frank    2.0     NA
3     2     1   Frank    2.0     NA
4     3     1   Frank    2.0     NA

> u
           [,1]      [,2]      [,3]
[1,] 0.21034650 0.3257784 0.7907567
[2,] 0.05175068 0.7129468 0.8002865

> dOFC(u=u,OFC=myOFC)
$values
[1] 0.9067347 0.8080970

$errors
[1] 0.006275317 0.003494095
```

In this example, the inner copula is the independence copula. The max value of `j` being 1, no nesting is involved, and since the max value of `i` is 3,

this is a copula that spans on 3 random variables, that is, $d = 3$. Hereafter, the code used to create the object `myOFC` from Example 1:

```
myOFC=list(innerandbiv=data.frame(
i=c("inner", 1, 2, 3),
j=c("inner", 1, 1, 1),
family=c("ind", "Frank", "Frank", "Frank"),
param1=c(NA, 2, 2, 2),
param2=c(NA, NA, NA, NA)
))
```

**Note 1**. Legal families for the inner copula and bivariate copulas are, at the time of writing: `"Independence"` (or `"ind"`), `"Normal"`, `"Student"` (or `"t"`), `"Frank"`, `"Joe"`, `"Gumbel"`, `"Clayton"`, `"FGM"`, `Plackett"`, `"min"` and `"Amh"`. Often, the first letter can be capitalized or not. The FGM copula should not be used as inner copula when $d > 2$ (a warning will be issued if the user nonetheless tries, although this won't stop the code from running), while using the Plackett copula in such cases will actually cause the `dOFC` function to stop, issuing an error message, as one can see in the hereafter example.

```
> myOFC
$innerandbiv
      i     j   family param1 param2
1 inner inner Plackett    3.0     NA
2     1     1    Frank    2.0     NA
3     2     1    Frank    2.0     NA
4     3     1    Frank    2.0     NA

> u
           [,1]      [,2]      [,3]
[1,] 0.21034650 0.3257784 0.7907567
[2,] 0.05175068 0.7129468 0.8002865
> dOFC(u=u, OFC=myOFC)
Error in dOFC(u = u, OFC = myOFC) :
  The Plackett copula is a bivariate copula only.
  Your d is >2, thus you cannot use it as inner copula in this case.
```

**Note 2**. When used as shown in Example 1, `dOFC` is a fully vectorized function. The memory usage can therefore be high, although few users are expected to run into trouble. In any case, should one run out of memory, it is suggested to loop through the lines of `u` rather than giving the full `u` matrix to `dOFC`.

**Note 3**. Using `dOFC` to work with a NEOFC rather than with an EOFC does not make any difference regarding memory usage. It does make a difference regarding calculation times: `dOFC` indeed runs slower with large values of $w$. See **Example 3** for how to build a NEOFC.

**Example 2. pOFC with an EOFC.**

```
$innerandbiv
      i     j  family param1 param2
1 inner inner student     NA     20
2     1     1 clayton    1.5     NA
3     2     1 clayton    1.5     NA

$map1
function (tw, param)
{
    qbeta(tw, param[1], param[2])
}

$map1.param
[1] 5 5

> u
          [,1]      [,2]
[1,] 0.8468635 0.1547030
[2,] 0.7673620 0.1715961
> pOFC(u=u, OFC=myOFC)
$values
[1] 0.1598121 0.1755701

$errors
[1] 0.007517979 0.007742175
```

In this example, the inner copula is a $d$-variate exchangeable student copula with 20 degrees of freedom. The max value of `j` is 1, therefore, no nesting is involved. Since the max value of `i` is 2, the outer copula spans on 2 random variables, that is, $d = 2$. The correlation for the inner copula is related to `map1`. `map1` is the quantile function of a beta distribution, the two shape parameters of which are 5 (see `map1.param`). Therefore, the correlation for the inner copula comes from a symmetric beta distribution, the expected value of which being 0.5.

Hereafter is the code allowing one to build the `myOFC` object from Example 2.

```
myOFC=list(innerandbiv=data.frame(
i=c("inner", 1, 2),
j=c("inner", 1, 1),
family=c("student", "clayton", "clayton"),
param1=c(NA, 1.5, 1.5),
param2=c(20, NA, NA)),

map1=function(tw, param) {

qbeta(tw, param[1], param[2])

},

map1.param=c(5, 5))
```

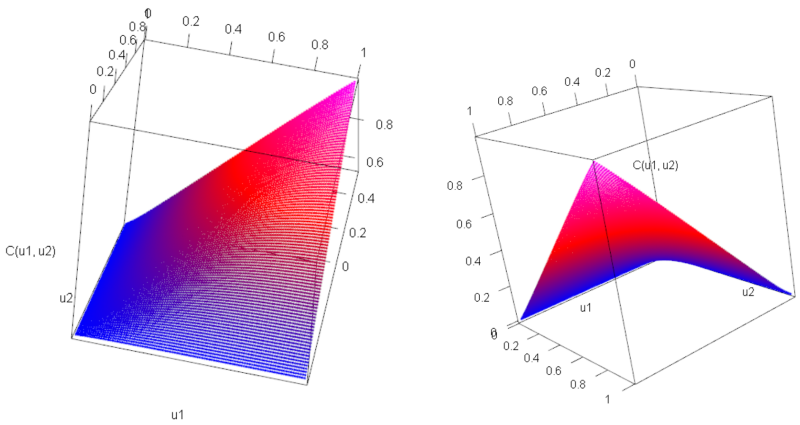



Figure 5.6: The copula from Example 2, as calculated by pOFC.

Figure 5.6 displays the copula from this example, as calculated by `pOFC`.

**Note.** When working with a map for the inner copula, make sure the map is able to handle `tw` as a vector and outputs a vector of the same size.

**Example 3. rOFC with a NEOFC.**

`rOFC` is a function with only two arguments: `n` and `OFC`. The function generates `n` observations from the input factor copula.

```
> myOFC
$innerandbiv
       i     j  family   param1 param2
1  inner inner student       NA     NA
2      1     1  gumbel 1.386588     NA
3      2     1  gumbel 1.220615     NA
4      3     1  gumbel 1.293084     NA
5      4     1  gumbel 1.236429     NA
6      5     1  gumbel 1.897415     NA
7      1     2  gumbel 1.943960     NA
8      2     2  gumbel 1.397582     NA
9      3     2  gumbel 1.441990     NA
10     4     2  gumbel 1.891171     NA
11     5     2  gumbel 1.692865     NA
$map1
function (tw, param)
{
    qbeta(tw, param[1], param[2])
}

$map1.param
[1] 5 5

$map2
function (tw, param)
```

```
{
    apply(cbind(round(tw * param), 1), 1, max)
}

$map2.param
[1] 100

> rOFC(2, myOFC)
          [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.3260713 0.2661169 0.1693356 0.2536162 0.4851917
[2,] 0.3460410 0.5187450 0.1131389 0.2921746 0.1783821
```

In the above example, the inner copula is again a $d$-variate exchangeable student copula, but in this case both the correlation and the degrees of freedom are related to $t_w$ through `map1` and `map2`. As it was the case before, the correlation of the inner copula comes from the quantile function related to a beta distribution, with both shape parameters set to 5 (`map1.param`). The degrees of freedom of the copula are equal to the rounded value of $t_w$ multiplied by `map2.param`. In case the result is 0, the value of 1 is used instead. Finally, since the max value of `i` is 5, $d = 5$, and since the max value of `j` is 2, $w = 2$. `myOFC` is thus a NEOFC in Example 3.

The code allowing to build `myOFC` for this example is given hereafter (the function `copGumbel@iTau` requires the `copula` package):

```
myOFC=list(innerandbiv=data.frame(
i=c("inner", 1:5, 1:5),
j=c("inner", rep(1, 5), rep(2, 5)),
family=c("student", rep("gumbel", 10)),
param1=c(NA, copGumbel@iTau(runif(10, 0.1, 0.5))),
param2=c(rep(NA, 11))),

map1=function(tw, param) {
qbeta(tw, param[1], param[2])
},

map1.param=c(5, 5),

map2=function(tw, param) {
apply(cbind(round(tw*param), 1), 1, max)
},

map2.param=100
)
```

Figure 5.7 shows cloud of points one gets from this NEOFC while keeping only the generated values for $U_1, U_2$ and $U_3$.

**Numerical integration.** The functions `dOFC` and `pOFC` require to solve a complicated integral. By default, both functions perform naive Monte-Carlo integration, where the integration space is sampled a thousand times according to a uniform distribution that stretches over the integration space.

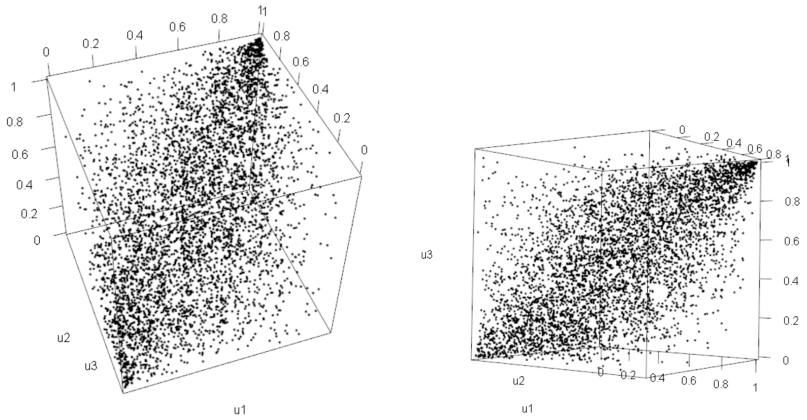



Figure 5.7: Cloud of points of the first three random variables in Example 3, related to a NEOFC.

In case of a one-factor copula or of an extended one-factor copula, the integration space is the segment $[0,1]$. However, with a nested extended one-factor copula, the integration space becomes $[0,1]^w$ and is therefore multi-dimensional.

In any case, the integrand is calculated at the 1000 sampled points and `dOFC` (or `pOFC`) outputs the average result. An error is also given and can be decreased by increasing how many times the integration space is sampled.

With `pOFC`, in case of a NEOFC, the integral one needs to solve is

$$I = \int_{[0,1]^w} C_{t_w}^{\diamond_w}\left(G_1(u_1; t_w, \dots, t_1), \dots, G_d(u_d; t_w, \dots, t_1)\right) dt_w \dots dt_1,$$

which will be rewritten as

$$I = \int_{[0,1]^w} f(\bar{t}) d\bar{t}$$

for convenience, and where $\bar{t} = (t_w, \dots, t_1)'$. Let $\bar{t}_1, \dots, \bar{t}_N$ be $N$ points sampled uniformly on $[0,1]^w$. Then, what `pOFC` outputs is

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^{N} f(\bar{t}_i),$$

while the error is

$$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(f(\bar{t}_i)-\hat{I}_N)^2}.$$

The default value of $N = 1000$ in `pOFC` and `dOFC` can be increased or decreased through the argument `input.integration`, as seen in the following example:

```
> myOFC
$innerandbiv
      i     j  family param1 param2
1 inner inner     ind     NA     NA
2     1     1   Frank    2.0     NA
3     2     1   Frank    2.0     NA
4     3     1 student    0.4     10

> u
            u1        u2        u3
[1,] 0.1137034 0.6092747 0.8609154
[2,] 0.6222994 0.6233794 0.6403106
> dOFC(u=u,OFC=myOFC)
$values
[1] 0.8252743 1.0727231

$errors
[1] 0.004866552 0.013615694

> dOFC(u=u,OFC=myOFC,input.integration=10000)
$values
[1] 0.8282513 1.0797866

$errors
[1] 0.001523137 0.004332387
```

The decrease of the error is however very slow, in $\sqrt{N}$.

Both `pOFC` and `dOFC` however allow for more sophisticated numerical integration approaches, offering full compatibility with the R packages cubature (Steven and Balasubramanian, 2013) and R2Cuba (Thomas et al., 2015). In the following example, the naive Monte Carlo integration is overriden and replaced by adaptative multidimensional integration:

```
> adaptIntegrate(f=dOFC, lowerLimit=0, upperLimit=1,
+ naive.integration=FALSE, u=u[1,], OFC=myOFC)
$integral
[1] 0.8255346

$error
[1] 7.004233e-06

$functionEvaluations
[1] 735
```

```
$returnCode
[1] 0
```

The key to override the default naive Monte Carlo integration is to set the argument `naive.integration` to `FALSE` in `pOFC` or `dOFC`. Notice that, in the above example using `adaptIntegrate`, the error on the result is rather small, while the integrand was only evaluated 735 times. This however doesn't mean that one will get a faster result using the function `adaptIntegrate` from the `cubature` package: the naive Monte Carlo integration used by default by `pOFC` or `dOFC` is significantly faster at evaluating the integrand over and over again.

The following examples show how one can make use of the integration methods from the package R2Cuba:

```
> cuhre(ndim=1, ncomp=1, integrand=dOFC,
+ naive.integration=FALSE, u=u[1,], OFC=myOFC)
Iteration 1:  11 integrand evaluations so far
[1] 0.82709 +- 0.0524425        chisq 0 (0 df)
.
.
.
Iteration 13:  275 integrand evaluations so far
[1] 0.825543 +- 0.000734674     chisq 0.00243551 (12 df)
integral: 0.8255427 (+-0.00073)
nregions: 13; number of evaluations:  275; probability:  1.110223e-16
```

```
> vegas(ndim=1, ncomp=1, integrand=dOFC,
+ naive.integration=FALSE, u=u[1,], OFC=myOFC)
Iteration 1:  1000 integrand evaluations so far
[1] 0.826074 +- 0.00484348      chisq 0 (0 df)
Iteration 2:  2500 integrand evaluations so far
[1] 0.82546 +- 0.00178692       chisq 0.0141172 (1 df)
Iteration 3:  4500 integrand evaluations so far
[1] 0.825522 +- 0.000719784     chisq 0.0141605 (2 df)
integral: 0.8255218 (+-0.00072)
number of evaluations:  4500; probability:  0.007055242
```

```
> suave(ndim=1, ncomp=1, integrand=dOFC,
+ naive.integration=FALSE, u=u[1,], OFC=myOFC)
Iteration 1:  1000 integrand evaluations so far
[1] 0.826074 +- 0.00484348      chisq 0 (0 df)
Iteration 2:  2000 integrand evaluations so far
[1] 0.826075 +- 0.00159638      chisq 0.000569991 (2 df)
Iteration 3:  3000 integrand evaluations so far
[1] 0.82565 +- 0.000763983      chisq 6.91287 (5 df)
integral: 0.8256503 (+-0.00076)
nregions: 3; number of evaluations:  3000; probability:  0.7727983
```

To conclude this appendix chapter, the five available numerical integration methods (naive, adaptIntegrate, cuhre, vegas and suave) are compared for the three models used in Example 1, 2 and 3. Given model $k$, $k \in \{1, 2, 3\}$, the

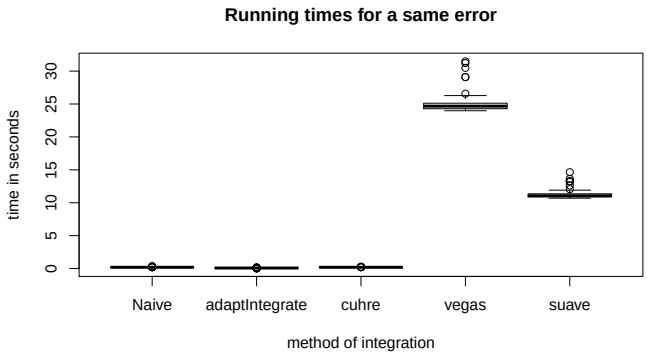



Figure 5.8: (1/2) Running times for the model in Example 1.

function `pOFC` is run 100 times on $u_i = 0.5, \forall i \in \{1, \dots d\}$ using each of the five integration methods until each of these methods reaches a prespecified error $e_k$. The running times are displayed in Figures 5.8 to 5.13. As one can see, `adaptIntegrate` appears to reach a prespecified error faster than any of the four other methods.

Some suggested references for the reader eager to learn more about numerical integration are Lepage (1978), Press and Farrar (1990), Berntsen et al. (1991), Caflisch (1998) or Robert and Casella (2013).

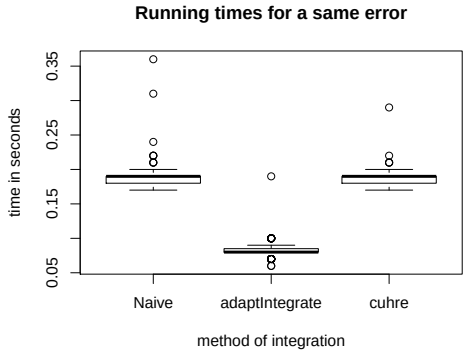



Figure 5.9: (2/2) Running times for the model in Example 1, where vegas and suave have been removed.

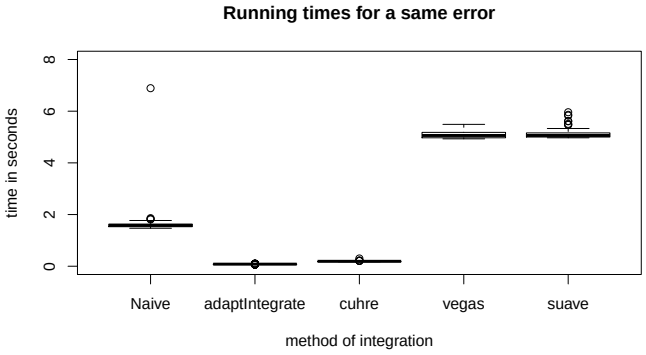



Figure 5.10: (1/2) Running times for the model in Example 2.

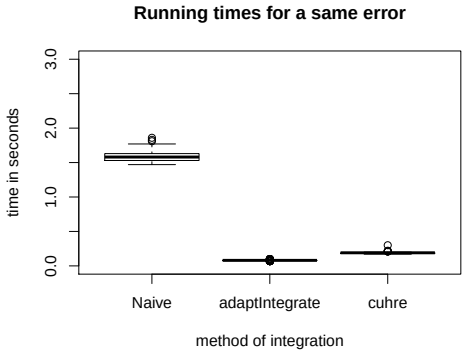



Figure 5.11: (2/2) Running times for the model in Example 2, where vegas and suave have been removed.

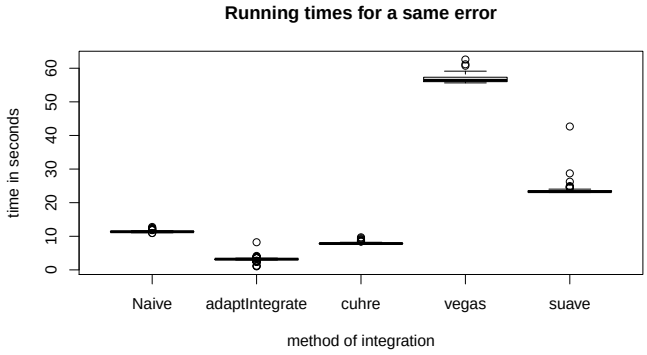



Figure 5.12: (1/2) Running times for the model in Example 3.

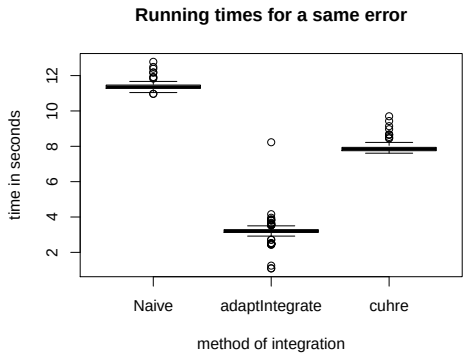



Figure 5.13: (2/2) Running times for the model in Example 3, where vegas and suave have been removed.

# Bibliography


Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44(2):182–198, 2009.

Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

Narayanaswamy Balakrishnan and Chin-Diew Lai. *Continuous bivariate distributions*. Springer Science & Business Media, 2009.

Philippe Barbe, Christian Genest, Kilani Ghoudi, and Bruno Rémillard. On Kendall's process. *Journal of Multivariate Analysis*, 58(2):197–229, 1996.

Tim Bedford and Roger M Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence*, 32(1-4):245–268, 2001.

Tim Bedford and Roger M. Cooke. Vines - a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.

Jarle Berntsen, Terje O. Espelid, and Alan Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software (TOMS)*, 17(4):437–451, 1991.

Norman Biggs. *Algebraic graph theory*. Cambridge University Press, 1993.

Norman Biggs, E. Keith Lloyd, and Robin J. Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1976.

Olaf Bininda-Emonds. The evolution of supertrees. *Trends in Ecology & Evolution*, 19(6):315–322, 2004.

Eike Christian Brechmann. Sampling from hierarchical kendall copulas. *Journal de la Société Française de Statistique*, 154(1):192–209, 2013.

Eike Christian Brechmann. Hierarchical kendall copulas: properties and inference. *Canadian Journal of Statistics*, 42(1):78–108, 2014.

Eike Christian Brechmann and Ulf Schepsmeier. Modeling dependence with C- and D-vine copulas: The R-package CDVine. *Journal of Statistical Software*, 52(3):1–27, 2013.

Eike Christian Brechmann, Claudia Czado, and Kjersti Aas. Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85, 2012.

Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

Gary Chartrand. *Introduction to graph theory*. Tata McGraw-Hill Education, 2006.

Gary Chartrand, Linda Lesniak, and Ping Zhang. *Graphs & digraphs*, volume 39. CRC Press, 2010.

Claudia Czado. Pair-copula constructions of multivariate copulas. In *Copula theory and its applications*, pages 93–109. Springer, 2010.

Claudia Czado, Ulf Schepsmeier, and Aleksey Min. Maximum likelihood estimation of mixed C-vines with application to exchange rates. *Statistical Modelling*, 12(3):229–255, 2012.

D.J. De Waal and P. van Gelder. Modelling of extreme wave heights and periods through copulas. *Extremes*, 8(4):345–356, 2005.

Andrew Ehrenberg. On sampling from a population of rankers. *Biometrika*, 39 (1/2):82–87, 1952.

Paul Embrechts. Copulas: A personal view. *Journal of Risk and Insurance*, 76 (3):639–650, 2009.

Paul Embrechts, Rdiger Frey, and Alexander McNeil. Quantitative risk management. *Princeton Series in Finance, Princeton*, 10, 2005.

Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Incorporated, 2004.

Christian Genest and Louis-Paul Rivest. Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034–1043, 1993.

Christian Genest and Louis-Paul Rivest. On the multivariate probability integral transformation. *Statistics & Probability Letters*, 53(4):391–399, 2001.

Christian Genest, Kilani Ghoudi, and Louis-Paul Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3):543–552, 1995.

Christian Genest, Johanna Nešlehová, and Noomen Ben Ghorbal. Estimators based on Kendall's tau in multivariate copula models. *Australian & New Zealand Journal of Statistics*, 53(2):157–177, 2011a.

Christian Genest, Johanna Nešlehová, and Johanna Ziegel. Inference in multivariate Archimedean copula models. *Test*, 20(2):223–256, 2011b.

Jan Górecki, Marius Hofert, and Martin Holeňa. An approach to structure determination and estimation of hierarchical archimedean copulas and its application to bayesian classification. *Journal of Intelligent Information Systems*, pages 1–39, 2015.

Philipp Hartmann, Stefan Straetmans, and Casper G. De Vries. Asset market linkages in crisis periods. *Review of Economics and Statistics*, 86(1):313–326, 2004.

Wassily Hoeffding. A non-parametric test of independence. *The Annals of Mathematical Statistics*, pages 546–557, 1948.

Marius Hofert. Efficiently sampling nested Archimedean copulas. *Computational Statistics & Data Analysis*, 55(1):57–70, 2011.

Marius Hofert and Martin Maechler. Nested Archimedean copulas meet R: the nacopula package. *Journal of Statistical Software*, 39(9), 2011.

Marius Hofert and David Pham. Densities of nested Archimedean copulas. *Journal of Multivariate Analysis*, 118:37–52, 2013.

Martin Holeňa, Lukáš Bajer, and Martin Ščavnickỳ. Using copulas in data mining based on the observational calculus. *IEEE Transactions on Knowledge and Data Engineering*, 27(10):2851–2864, 2015.

Nikolay Iskrev et al. Evaluating the strength of identification in DSGE models. An a priori approach. In *Society for Economic Dynamics, 2010 Meeting Papers*, 2010.

Kiyosi Itô. *Encyclopedic dictionary of mathematics*, volume 1. MIT press, 1993.

Guey Jiang, Li-Chin Yeh, Yen-Chang Chang, and Wen-Liang Hung. On the fundamental mass-period functions of extrasolar planets. *The Astrophysical Journal Supplement Series*, 186(1):48, 2009.

Harry Joe. *Multivariate models and dependence concepts*. Chapman & Hall/CRC, 2001.

Harry Joe. *Dependence Modeling with Copulas*. Chapman & Hall, 2014.

Harry Joe and James Jianmeng Xu. The estimation method of inference functions for margins for multivariate models. Technical report, Department of Statistics, University of British Columbia, Vancouver, 1996.

Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

A.W. Kemp, T.P. Hutchinson, and Chin Diew Lai. Continuous bivariate distributions, emphasising applications, 1992.

Pavel Krupskii and Harry Joe. Factor copula models for multivariate data. *Journal of Multivariate Analysis*, 120:85–101, 2013.

Pavel Krupskii and Harry Joe. Structured factor copula models: Theory, inference and computation. *Journal of Multivariate Analysis*, 138:53–73, 2015.

Pavel Krupskii, Raphael Huser, and Marc G. Genton. Factor copula models for replicated spatial data. *arXiv:1511.03000*, 2016.

D. Kurowicka and H. Joe. *Dependence Modeling: Vine Copula Handbook*. World Scientific, 2011.

Peter Lepage. A new algorithm for adaptive multidimensional integration. *Journal of Computational Physics*, 27(2):192–203, 1978.

Kantilal Varichand Mardia. *Families of bivariate distributions*, volume 27. Griffin London, 1970.

Dmytro Matsypura, Emily Neo, and Artem Prokhorov. Estimation of hierarchical archimedean copulas as a shortest path problem. 2016.

Gildas Mazo and Nathan Uyttendaele. Building conditionally dependent parametric one-factor copulas. 2016. In preparation.

Gildas Mazo, Stephane Girard, and Florence Forbes. A flexible and tractable class of one-factor copulas. *Statistics and Computing*, pages 1–15, 2015a.

Gildas Mazo, Stephane Girard, and Florence Forbes. Weighted least-squares inference based on dependence coefficients for multivariate copulas. *ESAIM: Probability and Statistics*, 19:746–765, 2015b.

Alexander J. McNeil. Sampling nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, 78(6):567–581, 2008.

Alexander J. McNeil and Johanna Nešlehová. Multivariate Archimedean copulas, $d$-monotone functions and $l_1$-norm symmetric distributions. *The Annals of Statistics*, 37:3059–3097, 2009.

Alexander J. McNeil, Rüdiger Frey, and Paul Embrechts. *Quantitative risk management: Concepts, techniques and tools*. Princeton university press, 2015.

Christian Meyer. The bivariate normal copula. *Communications in Statistics-Theory and Methods*, 42(13):2402–2422, 2013.

Katharine Mullen, David Ardia, David Gil, Donald Windover, and James Cline. DEoptim: An R package for global optimization by differential evolution. *Journal of Statistical Software*, 40(6):1–26, 2011.

Roger B. Nelsen. *An introduction to copulas*. Springer, New York, 2006.

Roger B. Nelsen, José Juan Quesada-Molina, José Antonio Rodríguez-Lallena, and Manuel Úbeda-Flores. Kendall distribution functions. *Statistics & Probability Letters*, 65(3):263–268, 2003.

Meei Pyng Ng and Nicholas C Wormald. Reconstruction of rooted trees from subtrees. *Discrete Applied Mathematics*, 69(1):19–31, 1996.

Kevin C. Nixon. The parsimony ratchet, a new method for rapid parsimony analysis. *Cladistics*, 15(4):407–414, 1999.

Dong Hwan Oh and Andrew J. Patton. Modelling dependence in high dimensions with factor copulas. *Journal of Business & Economic Statistics*, 2015.

Ostap Okhrin and Alexander Ristig. Hierarchical Archimedean Copulae: The HAC Package. *Journal of Statistical Software*, 58:1–20, 2014.

Ostap Okhrin, Yarema Okhrin, and Wolfgang Schmid. On the structure and estimation of hierarchical Archimedean copulas. *Journal of Econometrics*, 173(2):189–204, 2013a.

Ostap Okhrin, Yarema Okhrin, and Wolfgang Schmid. Properties of hierarchical Archimedean copulas. *Statistics & Risk Modeling*, 30(1):21–54, 2013b.

Arno Onken, Steffen Grünewälder, Matthias H.J. Munk, and Klaus Obermayer. Analyzing short-term noise dependencies of spike-counts in macaque prefrontal cortex using copulas and the flashlight transformation. *PLOS Computational Biology*, 5(11), 2009.

Andrew J. Patton. Modelling asymmetric exchange rate dependence. *International economic review*, 47(2):527–556, 2006.

William H. Press and Glennys R. Farrar. Recursive stratified sampling for multidimensional monte carlo integration. *Computers in Physics*, 4(2):190–195, 1990.

Kenneth Price, Rainer Storn, and Jouni Lampinen. *Differential Evolution - A Practical Approach to Global Optimization*. Natural Computing. Springer-Verlag, January 2006.

Kai Qin and Ponnuthurai Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *2005 IEEE congress on evolutionary computation*, volume 2, pages 1785–1791. IEEE, 2005.

Liam J. Revell. Phytools: An R package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution*, 3:217–223, 2012.

Mohsen Rezapour. On the construction of nested archimedean copulas for d-monotone generators. *Statistics & Probability Letters*, 101:21–32, 2015.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Thomas J. Rothenberg. Identification in parametric models. *Econometrica: Journal of the Econometric Society*, pages 577–591, 1971.

František Rublík. Estimates of the covariance matrix of vectors of u-statistics and confidence regions for vectors of kendall's tau. *Kybernetika*, 52(2):280–293, 2016.

Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4): 406–425, 1987.

Rogelio Salinas-Gutiérrez, Arturo Hernández-Aguirre, and Enrique R Villa-Diharce. Using copulas in estimation of distribution algorithms. In *MICAI 2009: Advances in Artificial Intelligence*, pages 658–668. Springer, 2009.

Robert J. Scherrer, Andreas A. Berlind, Qingqing Mao, and Cameron K. McBride. From finance to cosmology: The copula of large-scale structure. *The Astrophysical Journal Letters*, 708(1), 2009.

Berthold Schweizer. Thirty years of copulas. In *Advances in probability distributions with given marginals*, pages 13–50. Springer, 1991.

Johan Segers and Nathan Uyttendaele. Nonparametric estimation of the tree structure of a nested Archimedean copula. *Computational Statistics & Data Analysis*, 72:190–204, 2014.

Abe Sklar. Fonction de répartition dont les marges sont données. *Institut de statistique de l'Université de Paris*, 8:229–231, 1959.

Anders Skrondal and Sophia Rabe-Hesketh. Latent variable modelling: A survey. *Scandinavian Journal of Statistics*, 34(4):712–745, 2007.

Johnson Steven and Narasimhan Balasubramanian. *Cubature: Adaptive multivariate integration over hypercubes*, 2013. R package version 1.1-2.

Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

M. Shel Swenson, Rahul Suri, C. Randal Linder, and Tandy Warnow. Superfine: fast and accurate supertree estimation. *Systematic biology*, 61(2):214–227, 2012.

Nader Tajvidi, Seksan Kiatsupaibul, Sunti Tirapat, and Chonnart Panyangam. Behavior of extreme dependence between stock markets when the regime shifts. *Sri Lankan Journal of Applied Statistics*, 16(1):21–40, 2015.

Hahn Thomas, Bouvier Annie, and Kiêu Kiên. *R2Cuba: Multidimensional Numerical Integration*, 2015. R package version 1.1-0.

Nathan Uyttendaele. On the estimation of nested archimedean copulas: A theoretical and an experimental comparison. *Computational Statistics*, 2016.

Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

Mark Wilkinson, James A Cotton, Chris Creevey, Oliver Eulenstein, Simon R Harris, Francois-Joseph Lapointe, Claudine Levasseur, James O Mcinerney, Davide Pisani, and Joseph L Thorley. The shape of supertrees to come: tree shape related properties of fourteen supertree methods. *Systematic biology*, 54(3):419–431, 2005.

Robin J. Wilson. *An introduction to graph theory*. Pearson Education India, 1970.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.