Dependence: From classical copula modeling to neural networks

by

Avinash Srikanta Prasad

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2020

© Avinash Srikanta Prasad 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Debbie Dupuis Professor, Department of Decision Sciences, HEC Montréal
Supervisors:	Marius Hofert Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo Mu Zhu Professor, Department of Statistics and Actuarial Science, University of Waterloo
Internal Members:	Ali Ghodsi Professor, Department of Statistics and Actuarial Science, University of Waterloo Christiane Lemieux Professor, Department of Statistics and Actuarial Science, University of Waterloo
Internal-External Member:	Chris Bauch Professor, Department of Applied Mathematics, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The material presented in Chapter 2 was co-authored with Dr. Marius Hofert, Dr. Mu Zhu and Dr. Wayne Oldford. This work has been published in

> Hofert, M., Oldford, W., Prasad, A., & Zhu, M. (2019), A framework for measuring association of random vectors via collapsed random variables, *Journal of Multivariate Analysis, 172*, 5-27.

The material presented in Chapter 3 was co-authored with Dr. Marius Hofert and Dr. Raphaël Huser. This work has been published in

> Hofert, M., Huser, R., & Prasad, A. (2018), Hierarchical Archimax copulas, *Journal of Multivariate Analysis, 167*, 195-211.

The material presented in Chapter 4 was co-authored with Dr. Marius Hofert and Dr. Mu Zhu and is currently under revision. The contents of this chapter can be found in

> Hofert, M., Prasad, A. and Zhu, M. (2018) Quasi-random sampling for multivariate distributions via generative neural networks. arXiv preprint arXiv:1811.00683.

The material presented in Chapter 5 was co-authored with Dr. Marius Hofert and Dr. Mu Zhu and has been submitted for review. The submitted manuscript can be found in

> Hofert, M., Prasad, A. and Zhu, M. (2020) Multivariate time-series modeling with generative neural networks. arXiv preprint arXiv:2002.10645.

Abstract

The development of tools to measure and to model dependence in high-dimensional data is of great interest in a wide range of applications including finance, risk management, bioinformatics and environmental sciences. The copula framework, which allows us to extricate the underlying dependence structure of any multivariate distribution from its univariate marginals, has garnered growing popularity over the past few decades. Within the broader context of this framework, we develop several novel statistical methods and tools for analyzing, interpreting and modeling dependence.

In the first half of this thesis, we advance classical copula modeling by introducing new dependence measures and parametric dependence models. To that end, we propose a framework for quantifying dependence between random vectors. Using the notion of a collapsing function, we summarize random vectors by single random variables, referred to as collapsed random variables. In the context of this collapsing function framework, we develop various tools to characterize the dependence between random vectors including new measures of association computed from the collapsed random variables, asymptotic results required to construct confidence intervals for these measures, collapsed copulas to analytically summarize the dependence for certain collapsing functions and a graphical assessment of independence between groups of random variables. We explore several suitable collapsing functions in theoretical and empirical settings. To showcase tools derived from our framework, we present data applications in bioinformatics and finance.

Furthermore, we contribute to the growing literature on parametric copula modeling by generalizing the class of Archimax copulas (AXCs) to hierarchical Archimax copulas (HAXCs). AXCs are typically used to model the dependence at non-extreme levels while accounting for any asymptotic dependence between extremes. HAXCs then enhance the flexibility of AXCs by their ability to model partial asymmetries. We explore two ways of inducing hierarchies. Furthermore, we present various examples of HAXCs along with their stochastic representations, which are used to establish corresponding sampling algorithms.

While the burgeoning research on the construction of parametric copulas has yielded some powerful tools for modeling dependence, the flexibility of these models is already limited in moderately high dimensions and they can often fail to adequately characterize complex dependence structures that arise in real datasets. In the second half of this thesis, we explore utilizing generative neural networks instead of parametric dependence models. In particular, we investigate the use of a type of generative neural network known as the generative moment matching network (GMMN) for two critical dependence modeling tasks. First, we demonstrate how GMMNs can be utilized to generate quasi-random samples from

a large variety of multivariate distributions. These GMMN quasi-random samples can then be used to obtain low-variance estimates of quantities of interest. Compared to classical parametric copula methods for multivariate quasi-random sampling, GMMNs provide a more flexible and universal approach. Moreover, we theoretically and numerically corroborate the variance reduction capabilities of GMMN randomized quasi-Monte Carlo estimators. Second, we propose a GMMN–GARCH approach for modeling dependent multivariate time series, where ARMA–GARCH models are utilized to capture the temporal dependence within each univariate marginal time series and GMMNs are used to model the underlying cross-sectional dependence. If the number of marginal time series is large, we embed an intermediate dimension reduction step within our framework. The primary objective of our proposed approach is to produce empirical predictive distributions (EPDs), also known as probabilistic forecasts. In turn, these EPDs are also used to forecast certain risk measures, such as value-at-risk. Furthermore, in the context of modeling yield curves and foreign exchange rate returns, we show that the flexibility of our GMMN–GARCH models leads to better EPDs and risk-measure forecasts, compared to classical copula–GARCH models.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Dr. Marius Hofert and Dr. Mu Zhu, for their patient guidance, encouragement and support. I greatly appreciate the constructive and constant feedback I received when discussing new ideas, working on R code, confronting failed experiments, and completing manuscripts.

I would like to thank Dr. Wayne Oldford and Dr. Raphaël Huser for their insightful comments and guidance which helped to develop and improve the research projects presented in Chapters 2 and 3 respectively.

I would also like to thank my other thesis committee members, Dr. Debbie Dupuis, Dr. Chris Bauch, Dr. Christiane Lemieux and Dr. Ali Ghodsi for taking the time to read my thesis and providing me with suggestions.

I am very grateful to the support staff of the Department of Statistics and Actuarial Science. Particularly, I would like to thank Mary Lou Dufton for assisting me in many different ways throughout the years.

I would like to acknowledge the funding support provided by the Natural Sciences and Engineering Research Council of Canada and the University of Waterloo.

Lastly, I would like to thank my parents for their unconditional love and support throughout my many years of graduate study.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of high-dimensional dependent data in a variety of applications including finance, risk management, bioinformatics, hydrology and environmental sciences motivates the need for statistical tools to measure and to model dependence. An important framework prevalent in multivariate analysis for characterizing dependence is that of copulas. Copulas are multivariate distribution functions with standard uniform univariate margins. For a d-dimensional random vector $\boldsymbol{X}$, Sklar's Theorem (Sklar, 1959) allows us to tailor multivariate distribution functions $F_{\boldsymbol{X}}$ with specific margins $F_{X_1}, \ldots, F_{X_d}$ through copulas $C : [0,1]^d \to [0,1]$ via

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = C(F_{X_1}(x_1), \ldots, F_{X_d}(x_d)), \quad \boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d.$$

The key attractive feature of this decomposition is the ability to separately model the margins and the dependence structure of $\boldsymbol{X}$. Leveraging this key feature, one can flexibly model multivariate data. Also arising from this copula framework is the notion of dependence measures which solely depend on C, more formally known as *measures of concordance* (Scarsini, 1984). Such measures provide us with a consistent method for ranking dependencies.

In this thesis, we develop novel statistical methods for measuring and modeling dependence using the copula framework. These methods are utilized in various key tasks including detecting dependence, ranking of dependencies, modeling dependence structures of extreme events, quasi-random sampling and multivariate time series modeling. In what follows, we describe the primary contributions of this thesis in detail along with an outline of the subsequent chapters.

1.1 Measuring association of random vectors

The problem of measuring association of random variables X and Y has been extensively studied with axiomatic frameworks well established to characterize the desirable properties that bivariate measures of association should exhibit. *Measures of dependence* were defined using the list of properties first proposed in Rényi (1959) and later revised by Schweizer and Wolff (1981). Scarsini (1984) then additionally proposed the *concordance property* which ensured that bivariate measures of concordance satisfied a partial ordering on the set of copulas. Moreover, concordance measures, by virtue of solely depending on the copula of (X, Y), are invariant to strictly increasing transforms of X and Y. Prominent examples of such concordance measures popular in the copula literature include Spearman's rho and Kendall's tau. More recently, the notion of an *equitable dependence measure*, which extends the invariance property established by concordance measures to include invariance under non-monotone transforms of X and Y, was introduced by Reshef et al. (2011) and more formally established by Kinney and Atwal (2014).

While there exists thorough research on bivariate measures of association, the extension to random vectors remains somewhat of an open problem. There are some proposed measures of association between random vectors in the literature, the more notable of which include the kernel canonical correlation coefficient (Bach and Jordan, 2002), the distance covariance coefficient (Székely et al., 2007), the Hilbert Schmidt independence criterion (Gretton et al., 2008) and multivariate extensions of Spearman's rho and Kendall's tau as defined by Grothe et al. (2014).

In Chapter 2 of this thesis, we propose a general framework for measuring association of random vectors which not only subsumes some of the existing measures discussed above but also allows us to formulate various new measures. Using the notion of a *collapsing function*, the random vectors are summarized by single random variables, referred to as *collapsed random variables*. Measures of association computed from the collapsed random variables are then used to measure dependence between random vectors. To this end, we explore suitable collapsing functions and it is through the choice of this collapsing function that certain existing and new measures of association between random vectors are obtained. Additionally, we derive non-parametric estimators for the collapsed measures of association along with their corresponding asymptotic properties.

Furthermore, we introduce the notion of a collapsed distribution function and a collapsed copula. Collapsed copulas in particular provide us with the dependence structure between collapsed random variables. Moreover for certain collapsing functions, we can directly link the collapsed to the higher-dimensional copula. This investigation yields interesting

analytical results including a multivariate extension of the well-known *Kendall distribution function* (Barbe et al., 1996).

Naturally each collapsing function considered has its own unique set of features, advantages and disadvantages, which we explore in detail via discussions, theoretical analyses and data applications. One such application is motivated by a problem in bioinformatics which involves the ranking of a protein's side chain pairs by dependence. In addition, we considered an application in finance involving the dependence between S&P 500 business sectors.

The contents of Chapter 2 along with the additional theoretical results, proofs and examples presented in Appendix A have been published in:

> Hofert, M., Oldford, W., Prasad, A., & Zhu, M. (2019), A framework for measuring association of random vectors via collapsed random variables, *Journal of Multivariate Analysis, 172*, 5-27 (Hofert et al., 2019).

1.2 Dependence modeling with hierarchical Archimax copulas

Over the past few decades, there has been a burgeoning research interest in the construction of parametric copula classes for dependence modeling. Elliptical and Archimedean copulas are among the most well-known classes of copulas. Elliptical copulas typically arise from extracting the implicit copula of elliptical distributions via Sklar's Theorem. Some prominent members of this class of copulas include the Gaussian and t copulas. However, elliptical copulas are radially symmetric which is a limitation in modeling dependence structures with strong dependence in one of the two joint tails.

On the other hand, Archimedean copulas, which are explicit copulas constructed via a *generator* function, can model radial asymmetries; see Genest and MacKay (1986) and Nelsen (2006, Chapter 4). One drawback, however, is that members of this class of copulas are exchangeable, that is, permutation symmetric in their arguments, which can be restrictive, especially for higher dimensional dependence modeling, since all multivariate margins of the same dimension are equal. To relax this restrictive assumption, the class of *nested Archimedean copulas (NACs)*, which are constructed by plugging Archimedean copulas into each other at different levels, was introduced to model partial asymmetries in the dependence structure. For further details regarding NACs see Joe (1997, p. 87) and McNeil (2008).

Motivated by applications in environmental sciences and risk management, there has been growing interest in modeling the dependence structure of extreme events. For example, characterizing the joint risk of flooding at multiple locations is of great significance for disaster planning and development of suitable insurance products. A key statistical tool for modeling such joint risks is that of multivariate extreme value distributions, which arise as limiting distributions of properly scaled componentwise maxima of independent and identically distributed random vectors, the underlying copulas of which are referred to as *extreme value copulas (EVCs)*. For further details regarding EVCs see Jaworski et al. (2010, p. 128) and Segers (2012).

Capéraà et al. (2000) and Charpentier et al. (2014a) then generalized both the Archimedean and extreme value copula classes by proposing the class of *Archimax copulas (AXCs)*. Hence AXCs can simultaneously model the dependence at non-extreme levels while accounting for any asymptotic dependence between extremes. In Chapter 3 of this thesis, we introduce a more flexible class of AXC copulas, known as *hierarchical Archimax copulas (HAXCs)*, to account for partial asymmetries. HAXCs are *hierarchical* in the sense that they possess an underlying tree structure to characterize the dependence and thus have different pairwise margins depending on whether two variables belong to the same group, cluster or business sector. We propose two ways of inducing such hierarchies. Various examples of HAXCs are then discussed along with a general sampling algorithm which is obtained by working with the stochastic representation.

We also provide some derivations involving densities of Archimax copulas and briefly address the construction of nested Archimax copulas, which form a sub-class of HAXCs due to certain sufficient nesting conditions, in Appendix B. This work has been published in:

> Hofert, M., Huser, R., & Prasad, A. (2018), Hierarchical Archimax copulas, *Journal of Multivariate Analysis, 167*, 195-211 (Hofert et al., 2018a).

1.3 Dependence modeling with generative neural networks

While the substantial research focus on the construction of parametric copula classes has yielded some powerful tools for dependence modeling, the flexibility of these models is already limited in moderately large dimensions and they can fail to provide an adequate fit for complex dependence structures that arise in many data applications; see for example Hofert and Oldford (2018). Moreover, using parametric copulas in applications requires

model-specific algorithms for parameter estimation, goodness-of-fit assessment and finally model selection from a large collection of copula classes. To address these problems, we introduce *generative neural networks (GNNs)* for dependence modeling.

GNNs have enjoyed a meteoric rise to popularity in the past five years riding on the coattails of the broader deep learning revolution. The primary objective of a GNN is to *learn* the underlying distribution of a high-dimensional data set using neural networks and consequently provide a fast sampling mechanism to *generate* samples from this distribution. That is, given data points $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ from $F_{\boldsymbol{X}}$, a GNN aims to learn a deep neural network mapping $f_{\hat{\boldsymbol{\theta}}}$, such that, provided with a sample $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_m$ from an input distribution $F_{\boldsymbol{Z}}$, we can generate a *new sample* $f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_1), \dots, f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_m)$ from $F_{\boldsymbol{X}}$. Typical choices for $F_{\boldsymbol{Z}}$ are $\mathrm{U}(0,1)^d$ or $\mathrm{N}(\boldsymbol{0}, I_d)$. For uniform input distribution, the resulting sampling procedure is reminiscent of, at least in principle, the classical inverse transform method, where the mapping $f_{\hat{\boldsymbol{\theta}}}$ we learn is, in a sense, the inverse Rosenblatt transform.

The explosion of research into GNNs has yielded a plethora of approaches for learning the map $f_{\hat{\boldsymbol{\theta}}}$. Among the most popular types of GNNs are *generative adversarial networks (GANs)* (Goodfellow et al., 2014), *variational autoencoders (VAEs)* (Kingma and Welling, 2014), and *generative moment matching networks (GMMNs)* (Li et al., 2015). In this thesis, we focus on utilizing GMMNs where the *maximum mean discrepancy (MMD)* loss function (Gretton et al., 2007) used to learn $f_{\hat{\boldsymbol{\theta}}}$ is specifically designed for learning the entire distribution $F_{\boldsymbol{X}}$. In contrast, GANs and VAEs, in their standard formulation, are more geared towards ensuring that any single generated observation, typically an image, is very realistic. While certain versions of GANs and VAEs have incorporated the MMD loss function (see Li et al. (2017) and Zhao et al. (2017)), they typically require more sophisticated and computationally expensive learning mechanisms in comparison to GMMNs.

Although our objective is to model $F_{\boldsymbol{X}}$, in practice it is often useful to leverage Sklar's Theorem to separate the modeling of $F_{X_1}, \dots, F_{X_d}$ from the modeling of C. Hence, we typically instead focus on using GMMNs to learn the underlying dependence structure C. In applications, this essentially translates to normalizing the dataset to $[0,1]^d$ by removing the marginal information, which also allows us to more efficiently learn the deep neural network map $f_{\hat{\boldsymbol{\theta}}}$. Dependence modeling remains the key challenging aspect of modeling multivariate data. As mentioned earlier, there exists certain challenges in obtaining good parametric copula fits for complex dependence structures already in moderately high dimensions. Hence using parametric copulas could result in misspecified dependence models, which in certain applications, would lead to severe underestimation of joint risks. GMMNs offer a more *universal* and highly *flexible* approach for modeling dependence. Once we learn to generate samples from C, it is not difficult to fit appropriate marginal distributions and use them to

generate samples from $F_{\boldsymbol{X}}$.

In Chapters 4 and 5 of this thesis, we showcase the benefits of utilizing GMMNs for dependence modeling in the contexts of quasi-random sampling and multivariate time series modeling.

1.3.1 Quasi-random sampling for multivariate distributions

The basic idea behind quasi-random numbers is to replace pseudo-$\mathrm{U}(0,1)^d$ random numbers with a low-discrepancy point set $P_m = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ to produce a more homogeneous coverage of $[0,1]^d$; in applications a randomized point set $\tilde{P}_m = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_m\}$ is used instead to obtain unbiased estimators of quantities of interest. This idea is visualized for the $d = 2$ case in Figure 1.1 which displays scatter plots of pseudo-random (left) and quasi-random (right) samples from $\mathrm{U}(0,1)^2$. As can be observed from this figure, the generated quasi-random sample has fewer *gaps* and *clusters* compared to the pseudo-random sample. Consequently, with respect to a certain discrepancy measure, the empirical distribution of the quasi-random sample is typically closer to the uniform distribution $\mathrm{U}(0,1)^2$ than that of the pseudo-random sample.

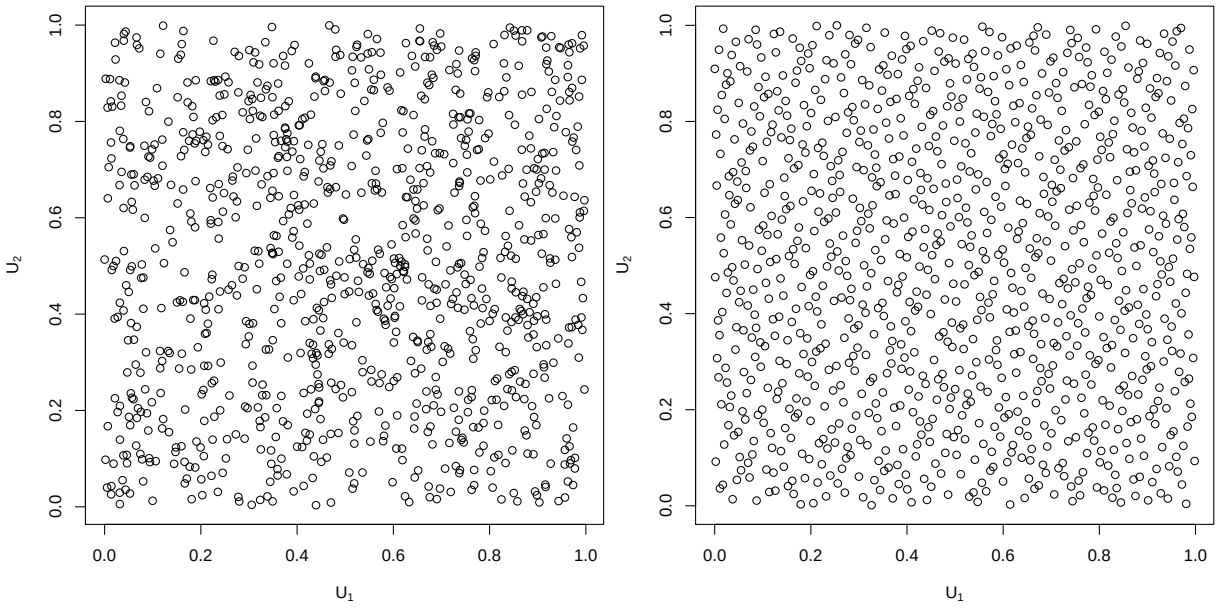

Figure 1.1: Pseudo-random (left) and quasi-random samples (right) from $\mathrm{U}(0,1)^2$.

We are interested in the natural extension of this concept to non-uniform multivariate distributions $F_{\boldsymbol{X}}$. Currently in the literature, the most comprehensive approach for generating quasi-random samples from multivariate distributions is via parametric copula models; see Cambou et al. (2017). However, this approach is feasible for only a limited number of parametric copulas where either a numerically tractable transformation of $\tilde{P}_m$ is attainable via the *conditional distribution method* (based on inverse Rosenblatt transform) or a simple stochastic representation exists from which one can construct a tractable transformation. Additionally, in certain cases it can be tricky to ensure that these transformations preserve the low-discrepancy of $\tilde{P}_m$ and a careful manipulation of the model-specific sampling procedure is required. Moreover, there exists complex dependence structures in data which cannot be adequately captured by available parametric copulas.

In Chapter 4 of this thesis, we propose a new approach for quasi-random sampling from $F_{\boldsymbol{X}}$ with any underlying copula C, using GMMNs. This approach greatly extends in two ways, the range of multivariate distributions for which we can readily construct quasi-random samples. Firstly, we can generate quasi-random samples from $F_{\boldsymbol{X}}$ with *any* underlying parametric copula C by transforming a randomized point set $\tilde{P}_m$ using the neural network map $f_{\hat{\boldsymbol{\theta}}}$ obtained by training the GMMN on a pseudo-random sample from C. Hence, in comparison to the current quasi-random sampling strategy for parametric copulas, GMMNs provide a more *universal* approach. Secondly, in a similar manner, we can generate quasi-random samples from empirical distributions $\hat{F}_{\boldsymbol{X}}$ of real data sets with underlying dependence structures $\hat{C}$ (empirical copulas) not adequately captured by parametric copulas, thanks to the *greater flexibility* of GMMNs. We demonstrate that GMMNs are capable of learning the dependence structures of numerous sophisticated parametric copulas including nested Archimedean, Marshall-Olkin and mixture copulas along with the complex dependence structures that arise when modeling multivariate financial returns data. Moreover, we show that all corresponding GMMN quasi-random samples essentially *preserve* the low discrepancy of $\tilde{P}_m$ upon transformation.

The main application of quasi-random sampling is to approximate expectations of the form $\mu = \mathbb{E}(\Psi(\boldsymbol{X}))$, where $\Psi : \mathbb{R}^d \to \mathbb{R}$ is an integrable function, with variance reduction. To this end, we utilize GMMN *randomized quasi-Monte Carlo (RQMC)* estimators that are constructed based on quasi-random samples from $F_{\boldsymbol{X}}$. In various settings, we theoretically establish convergence rates for RQMC estimators under smoothness conditions on $f_{\hat{\boldsymbol{\theta}}}$ and Ψ. Furthermore, we numerically demonstrate that GMMN RQMC estimators achieve variance reduction and improved convergence rates compared to MC estimators constructed based on pseudo-random samples from $F_{\boldsymbol{X}}$. We also present a real-data example inspired by applications in finance and risk management that showcases the flexibility of GMMNs and the variance reduction capabilities of our GMMN RQMC estimators for estimating various

quantities of interest μ.

To keep the main body of Chapter 4 concise, we relegated certain technical details, proofs and numerical results to Appendix C. The research presented in Chapter 4 and Appendix C is currently under revision.

> Hofert, M., Prasad, A. and Zhu, M. (2018) Quasi-random sampling for multivariate distributions via generative neural networks. arXiv preprint arXiv:1811.00683 (Hofert et al., 2018c).

1.3.2 Multivariate time series modeling

Conceptually, multivariate time series (MTS) aim at capturing two types of dependence structures — the serial dependence within each univariate marginal time series and the cross-sectional dependence between the individual time series. A popular approach for modeling MTS data is to separate these two dependence modeling tasks.

There exists a wide variety of univariate time series models which capture different types of temporal patterns. In finance and econometrics, *generalized auto-regressive conditional heteroscedasticity (GARCH)* models (Bollerslev, 1986) are particularly popular due to their ability to capture the volatility clustering effect that is often present in financial time series data. Within the GARCH framework, the extension to MTS modeling is achieved by modeling the corresponding joint innovations using a suitable multivariate distribution. To this end, Jondeau and Rockinger (2006) and Patton (2006) introduced the *copula–GARCH* approach where parametric copulas are used to characterize the cross-sectional dependence. Over the past decade, the research on calibration of copula–GARCH models has grown in tandem with the burgeoning literature on parametric copula classes.

In Chapter 5 of this thesis, we introduce the *GMMN–GARCH* framework, where GMMNs are used in place of parametric copulas to model the cross-sectional dependence of MTS data. For higher dimensional time series which are amenable to good approximations by lower dimensional representations, we also incorporate a dimension reduction step in our framework.

The primary objective of MTS modeling via GMMN–GARCH (or copula–GARCH) models is to construct empirical predictive distributions, also known as probabilistic forecasts. Probabilistic forecasting is of great interest in a variety of applications including hydrology (Krzysztofowicz, 2001), energy forecasting (Wan et al., 2013; Hong et al., 2016), finance and quantitative risk management where the empirical predictive distribution is often

used to forecast risk measures such as *Value-at-Risk (VaR)* or *expected shortfall (ES)* via simulation. We showcase the applicability and flexibility of our GMMN–GARCH framework in forecasting yield curves and foreign exchange rates.

Note that since we borrow much of the same GMMN setup detailed in Chapter 4, the relevant background materials is not repeated in Chapter 5. The work presented in this chapter has been submitted for review.

> Hofert, M., Prasad, A. and Zhu, M. (2020) Multivariate time-series modeling with generative neural networks. arXiv preprint arXiv:2002.10645 (Hofert et al., 2020).

Chapter 2

A framework for measuring association of random vectors via collapsed random variables

2.1 Introduction

While there are numerous well established methods to measure dependence between random variables, the extension to random vectors poses a significant challenge. This challenge arises from the lack of a unique axiomatic framework that states desirable properties a measure of association between random vectors should exhibit. Moreover, there is no unique extension of bivariate measures of association to arbitrary dimensions and the available multivariate measures of association do not naturally capture dependence between more than one random vector as is of interest for applications in areas such as bioinformatics, finance, insurance or risk management.

Proposed solutions to this problem are rather difficult to find in the literature. A classical methodology for summarizing linear dependence between random vectors is the well known canonical correlation coefficient; see Hotelling (1936). A non-linear extension of canonical correlation has been suggested through the use of kernel functions in Bach and Jordan (2002) and Ghoraie et al. (2015a). A faster version of the kernel canonical correlation method has been developed by adopting the idea of randomized kernels; see Lopez-Paz et al. (2013). Székely et al. (2007) proposed a novel distance covariance coefficient, defined as a weighted $\mathcal{L}^2$-norm between the joint characteristic function and the product of marginal characteristic functions of the random vectors under consideration. In the context of copula

modeling, Grothe et al. (2014) recently derived versions of Spearman's rho and Kendall's tau between random vectors including corresponding estimation procedures. Our framework will generalize their approach and allow us to derive a couple of interesting results as by-products.

Note that there is neither an inherently correct nor a canonical way of measuring dependence between random vectors. As a result, one can think of multiple ways of quantifying such dependence. Approaches are primarily motivated by the purpose, for example, detection or ranking of dependencies, or the dataset under investigation. In this chapter, we subsume several such approaches under a general framework which allows us to detect, quantify, visualize and check dependence between random vectors.

The chapter is organized as follows. In Section 2.2 we present a framework for measuring dependence between random vectors. Furthermore, we discuss non-parametric estimators for the measures of association arising from the framework and their corresponding asymptotic properties. Section 2.3 develops the notion of a collapsed distribution function and a collapsed copula. Moreover, analytical formulas of these collapsed distributions are presented for a number of collapsing functions. Empirical examples from the areas of bioinformatics and finance are covered in Section 2.4. In addition, a visual assessment of independence between groups of random variables is introduced. Section 2.5 provides concluding remarks for this chapter.

2.2 The framework

For introducing a framework for measuring dependence between random vectors, it suffices to consider the case of two random vectors, a p-dimensional $\boldsymbol{X} = (X_1, \ldots, X_p)$ with continuous marginal distribution functions $F_{X_1}, \ldots, F_{X_p}$ and a q-dimensional $\boldsymbol{Y} = (Y_1, \ldots, Y_q)$ with continuous marginal distribution functions $F_{Y_1}, \ldots, F_{Y_q}$, defined on some probability space with probability measure $\mathbb{P}$. Our target is to measure dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$ with a measure of association

$$\chi = \chi(\boldsymbol{X}, \boldsymbol{Y})$$

mapping to either $[-1, 1]$ or $[0, 1]$; note that, depending on the context, various notions of dependence are possible.

A natural first step is to establish the properties that χ should satisfy. For bivariate measures of association, that is, measures of association between two random variables X and Y, one set of such properties is listed in Rényi (1959), with minor revisions later in

Schweizer and Wolff (1981), and slightly modified in Embrechts et al. (2002); the resulting measures of association were termed "measures of dependence". Scarsini (1984) introduced another set of properties, including a pointwise partial ordering on the set of copulas known as concordance ordering; the resulting measures of association are thus called concordance measures or also rank-correlation measures; see Embrechts et al. (2002) for their advantages over the classical linear correlation coefficient. Prominent examples are Kendall's tau and Spearman's rho. Another type of bivariate measure of association, focusing on the extremal dependence in the joint tails of bivariate distributions, are the coefficients of tail dependence.

More recently, Reshef et al. (2011) described an ideal measure of association in the bivariate case as the so-called "equitable dependence measure", which extends the invariance property of concordance measures to include invariance under non-monotone marginal transforms. However, the maximal information coefficient (MIC) introduced in Reshef et al. (2011), which empirically satisfies the equitability condition under various non-monotone transformations, is purely data-driven and heuristic. As a result, the MIC measure does not naturally fit into our probabilistic framework. Various versions of this equitability condition have since been proposed including more mathematically formal definitions; see, for example, Kinney and Atwal (2014). Hence, there is some consensus concerning an "ideal" bivariate measure of association but our problem demands generalizations of these properties to vector-based measures of association, which is non-trivial.

Grothe et al. (2014) recently approached this problem and listed properties of a concordance measure that carry over from random variables X, Y to random vectors $\boldsymbol{X}, \boldsymbol{Y}$. These include:

(P1) $\chi(\boldsymbol{X}, \boldsymbol{Y}) \in [-1, 1]$;

(P2) $\chi(\boldsymbol{X}, \boldsymbol{Y})$ is invariant to permutations of the components of the random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$;

(P3) independence of $\boldsymbol{X}$ and $\boldsymbol{Y}$ implies $\chi(\boldsymbol{X}, \boldsymbol{Y}) = 0$;

(P4) (*Invariance Property*) $\chi(\boldsymbol{X}, \boldsymbol{Y})$ is invariant to strictly increasing transformations of the components of the random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ (that is, χ is copula-based);

(P5) (*Concordance Ordering Property*) if two $(p+q)$-dimensional copulas C_1 and C_2 with the same marginal copulas corresponding to the first p and the second q dimensions satisfy $C_1 \preceq C_2$ (that is, $C_1(\boldsymbol{u}) \leq C_2(\boldsymbol{u})$ for all $\boldsymbol{u} \in [0,1]^{p+q}$) and if, $\boldsymbol{U}_1 \sim C_1$ and $\boldsymbol{U}_2 \sim C_2$ then $\chi(\boldsymbol{U}_1) \leq \chi(\boldsymbol{U}_2)$. .

Extending the invariance and concordance ordering properties to the vector case can be done in many ways with (P4) and (P5) being just one set of possible generalizations. In particular, (P5) focuses only on the dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$ to establish the concordance ordering with equal p- and q-dimensional marginal dependence requirements. The difficulty lies in hypothesizing invariance and concordance properties when the marginal distributions $F_{X_1}, \ldots, F_{X_p}$ and $F_{Y_1}, \ldots, F_{Y_q}$ and the copulas $C_{\boldsymbol{X}}$ and $C_{\boldsymbol{Y}}$ of $\boldsymbol{X}$ and $\boldsymbol{Y}$ can all vary; generalizing the concept of equitable dependence faces similar difficulties.

Additionally, our framework subsumes measures of association χ which map to $[0, 1]$. Thus we consider the following Rényi axiom Rényi (1959) as an alternative to property (P1):

(P1)$'$ $\chi(\boldsymbol{X}, \boldsymbol{Y}) \in [0, 1]$.

While Property (P1)$'$ is just a special case of Property (P1), it is useful to particularly identify measures of association χ which map to $[0, 1]$ (see later).

2.2.1 Collapsed random variables

The framework we suggest consists of collapsing the two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ to single random variables $S_{\boldsymbol{X}}(\boldsymbol{X})$ and $S_{\boldsymbol{Y}}(\boldsymbol{Y})$, referred to as *collapsed random variables*. The functions $S_{\boldsymbol{X}}$ and $S_{\boldsymbol{Y}}$ map random vectors to random variables and are referred to as *collapsing functions*; see also Grabisch et al. (2009) for a related notion known as aggregation functions. For the sake of simplicity, we will restrict ourselves to using the same collapsing function to collapse $\boldsymbol{X}$ and $\boldsymbol{Y}$ and will simply denote this function by S (even if $p \neq q$; for example, S could simply be the sum over p components for $\boldsymbol{X}$ and the sum over q components for $\boldsymbol{Y}$). We also assume that $\boldsymbol{X}$ and $\boldsymbol{Y}$ have continuous margins, to facilitate development of theoretical results. The bivariate distribution function of $(S(\boldsymbol{X}), S(\boldsymbol{Y}))$ is called the *collapsed distribution function* in our framework and its copula, if unique, is termed the *collapsed copula*. In later discussions of the collapsed distribution function and copula, we will also assume that $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$ are continuously distributed random variables.

We interpret the notion of a collapsing function quite generally, the only requirement being that a random vector is mapped to a single random variable. As we will see later, a collapsing function for $\boldsymbol{X}$ does not necessarily have to be a p-variate function, it can also be a $2p$-variate function, in which case the collapsed random variable will denoted by $S(\boldsymbol{X}, \boldsymbol{X}')$, where $\boldsymbol{X}'$ is an independent copy of $\boldsymbol{X}$.

Finally, we note that while our framework is restricted to one level of collapsing with two groups of random variables, it can be extended for the purposes of modeling to several groups of random variables and multiple hierarchical levels. Such hierarchical constructions have been studied in the literature using certain collapsing functions. Examples include the hierarchical Kendall copula in Brechmann (2014) and the hierarchical aggregation models in Arbenz et al. (2012) and Côté and Genest (2015).

2.2.2 Collapsed measures of association

The two collapsed random variables, $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$, can be used to detect, quantify and check dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$ using classical and well understood bivariate measures of association. Any such measure will be referred to as *collapsed measure of association* in our framework.

Remark 2.2.1
At this point, some remarks are in order. First, our approach of measuring dependence between random vectors by measuring dependence between their collapsed random variables is different from multivariate extensions of measures of association: The latter aim to summarize dependence within a single random vector; see Schmid et al. (2010) and the references therein for a comprehensive treatment. Second, it should be clear that measuring association between collapsed random variables can only be a summary of the dependence between the components of $(\boldsymbol{X}, \boldsymbol{Y})$, and that the measures of association χ we consider will not generally uniquely determine the dependence of $(\boldsymbol{X}, \boldsymbol{Y})$; the same is well-known for random variables ($p = q = 1$). Even so, we will present results under which one can explicitly determine the copula of $(S(\boldsymbol{X}), S(\boldsymbol{Y}))$ given that of $(\boldsymbol{X}, \boldsymbol{Y})$. □

Although there are various choices of collapsed measures of association, for ease of illustration we will mainly focus on Pearson's correlation coefficient ρ and consider

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}. \tag{2.1}$$

This choice is less restrictive than it might appear. Spearman's rho, Kendall's tau and Blomqvist's beta all appear as special cases of (2.1) for appropriate collapsing functions S. Spearman's rho ρ_S is simply Pearson's correlation coefficient ρ of the probability-integral-transformed random variables. That is, if $F_{S(\boldsymbol{X})}$ denotes the distribution function of the collapsed random variable $S(\boldsymbol{X})$, the collapsing functions $\tilde{S}(\boldsymbol{x}) = F_{S(\boldsymbol{X})}\{S(\boldsymbol{x})\}$ and $\tilde{S}(\boldsymbol{y}) = F_{S(\boldsymbol{Y})}\{S(\boldsymbol{y})\}$ give

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{\tilde{S}(\boldsymbol{X}), \tilde{S}(\boldsymbol{Y})\} = \rho_S\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}.$$

For Kendall's tau τ, the collapsing function is an example of a $2p$-variate function. In particular, one can show that if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are continuously distributed random vectors and $(\boldsymbol{X}', \boldsymbol{Y}')$ is an independent copy of $(\boldsymbol{X}, \boldsymbol{Y})$, then the collapsing function $\tilde{S}(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{1}\{S(\boldsymbol{x}) \leq S(\boldsymbol{x}')\}$ leads to

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{\tilde{S}(\boldsymbol{X}, \boldsymbol{X}'), \tilde{S}(\boldsymbol{Y}, \boldsymbol{Y}')\} = \tau\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}. \tag{2.2}$$

For Blomqvist's beta β, let $\tilde{S}(\boldsymbol{x}) = \mathbb{1}\{S(\boldsymbol{x}) \leq F^{-}_{S(\boldsymbol{X})}(1/2)\}$, where $F^{-}_{S(\boldsymbol{X})}$ denotes the quantile function of $F_{S(\boldsymbol{X})}$, and note that

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{\tilde{S}(\boldsymbol{X}), \tilde{S}(\boldsymbol{Y})\} = \beta\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}.$$

Tail dependence is often expressed by λ, the lower (or upper) coefficient of tail dependence implied by the corresponding copula. Applied to the collapsed copula would give

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \lambda\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}, \tag{2.3}$$

which provides a measure of tail dependence between random vectors $\boldsymbol{X}$, $\boldsymbol{Y}$, within our framework. Although there are multivariate notions of tail dependence (Jaworski et al., 2010, Chapter 10), to the best of our knowledge no measure of tail dependence exists between two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$. The simple and intuitive formulation above gives many such measures, depending on the choice of collapsing functions. Moreover, this approach can be straightforwardly extended to more than two random vectors by considering matrices; see Embrechts et al. (2016).

In what follows, we will focus on cases where the collapsed measure of association is Pearson's correlation coefficient.

2.2.3 Choosing the collapsing function

There is no universal way to choose the collapsing function S for measuring dependence between random vectors; the choice will largely depend on context, as we shall illustrate later in the applications section. We start by introducing various options for S, summarized in Table 2.1. While collapsing functions can be as sophisticated as deep neural networks (Andrew et al., 2013), the focus in this chapter will be on the elementary summary functions listed in Table 2.1. Note that the $2p$-variate collapsing functions listed in Table 2.1 require us to invoke an independent copy $\boldsymbol{X}'$ of the random vector $\boldsymbol{X}$. Thus, unlike p-variate collapsing functions, we need a pair of realizations, $\boldsymbol{x}$ and $\boldsymbol{x}'$, from $\boldsymbol{X}$ to evaluate S once.

	Type of S	Collapsing function S
p-variate	Weighted sum	$S(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$
	Maximum (or minimum)	$S(\boldsymbol{x}) = \max_{1 \leq j \leq p} \{x_j\}$ (or $S(\boldsymbol{x}) = \min_{1 \leq j \leq p} \{x_j\}$)
	Probability integral transform	$S(\boldsymbol{x}) = F_{\boldsymbol{X}}(\boldsymbol{x})$
$2p$-variate	Distance	$S(\boldsymbol{x}, \boldsymbol{x}') = D(\boldsymbol{x}, \boldsymbol{x}')$
	Kernel similarity	$S(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}, \boldsymbol{x}')$
	Multivariate rank	$S(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{1}\{\boldsymbol{x} \leq \boldsymbol{x}'\}$

Table 2.1: Examples of collapsing functions S of a random vector $\boldsymbol{X}$ (with realizations $\boldsymbol{x}$ and $\boldsymbol{x}'$); note that the inequality $\boldsymbol{x} \leq \boldsymbol{x}'$ in the multivariate rank collapsing function is understood componentwise.

Consequentially, to estimate χ based on a sample of size $n \geq 2$, S is computed for all $\binom{n}{2}$ pairs of multivariate observations.

The following sections consider each collapsing function listed in Table 2.1 in more detail.

The weighted sum collapsing function

The weighted sum function is a classical choice of collapsing function. Its key feature is the flexibility presented in the choice of weights. These weights can be chosen arbitrarily, e.g., taken to be equal or optimally chosen with respect to some objective function. The canonical correlation coefficient of Hotelling (1936) is a classical approach involving optimal weights, where

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \sup_{\boldsymbol{w}_{\boldsymbol{X}} \in \mathbb{R}^p, \boldsymbol{w}_{\boldsymbol{Y}} \in \mathbb{R}^q} \rho(\boldsymbol{w}_{\boldsymbol{X}}^\top \boldsymbol{X}, \boldsymbol{w}_{\boldsymbol{Y}}^\top \boldsymbol{Y}). \tag{2.4}$$

Note that by replacing $\boldsymbol{X} = (X_1, \ldots, X_p)$ with $(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))$, $(\mathbb{1}\{X_1 \leq X_1'\}, \ldots, \mathbb{1}\{X_p \leq X_p'\})$, or $(\mathbb{1}\{X_1 \leq F_{X_1}^-(1/2)\}, \ldots, \mathbb{1}\{X_p \leq F_{X_p}^-(1/2)\})$ and doing the same with $\boldsymbol{Y} = (Y_1, \ldots, Y_q)$ in Equation (2.4), we can construct Spearman's rho, Kendall's tau, or Blomqvist's beta variants of canonical correlation, respectively. Another approach to constructing rank-based versions of canonical correlation is given in Alfons et al. (2017). In addition to the canonical correlation approach, hierarchical aggregation modeling techniques such as in Arbenz et al. (2012) utilize the sum collapsing function.

Restricting the weights to sum to one would yield the weighted average collapsing function. With equal weights we can ensure no random variable in the group is fully ignored.

The application of interest can inform the choice of weights given the interpretation of each random variable within a group. Alternatively, one could consider the *m-largest* (or *m-smallest*) *weighted average*, that is, the average over the m largest (or m smallest) order statistics per group of random variables. This can be of interest in the context of financial risk management, where one needs to keep track of the m largest (or m smallest) losses in two or more portfolios or asset classes.

Measures of association arising from the weighted sum collapsing function satisfy the basic properties (P1) and (P3) listed in the introduction of Section 2.2. However, property (P2) is only satisfied when we use equal weights. An exception is the canonical correlation coefficient which satisfies property (P1)′ instead of (P1). Property (P4) is in general only satisfied if we replace $(X_1, \ldots, X_p)$ by $(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))$ in S. The key feature of this collapsing function is the set of weight parameters which provides some flexibility, can be readily adapted to certain optimization problems, and can be easily interpreted for applications arising, for example, in finance and bioinformatics. While there is an easy way to obtain invariance to marginal distributions as noted above, this comes at the expense of interpretability as it pertains to the weights involved. Additionally, note that if a subset of the weights is chosen to be zero without proper justification, the resulting measure χ could potentially be a misleading summary of the dependence between random vectors; see also Remark 2.2.1 in this regard.

The maximum collapsing function

The componentwise maximum (or minimum) is a special case of the aforementioned extreme weighted case, with 1-largest (or 1-smallest) weighted average as collapsing function, that is,

$$S(\boldsymbol{x}) = \max\{x_1, \ldots, x_p\} \quad (\text{or } S(\boldsymbol{x}) = \min\{x_1, \ldots, x_p\}).$$

This requires all dimensions of the random vector $\boldsymbol{X}$ to have a comparable interpretation. It may be useful, for example, when quantifying dependence between market return data grouped into sectors where dependence between different market sectors would be measured through the best (or worst) performer in each sector.

Measures of association arising from the maximum or minimum collapsing functions also satisfy the basic properties (P1)–(P3) listed in the introduction of Section 2.2. Property (P4) is in general only satisfied if we replace $(X_1, \ldots, X_p)$ by $(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))$ in S. A key feature of the maximum collapsing function is its interpretability especially in certain applications such as finance. Mathematically, we can directly link the collapsed and

original distribution function or copula as described in Section 2.3.2 below. This allows one to use the maximum collapsing function in hierarchical models. A drawback of the maximum collapsing function arises when $X_i \leq X_j$ almost surely for some $i, j \in \{1, \ldots, p\}$. In this case, the nature of dependence between random vectors captured by the maximum collapsing function can be misleading since a subset of the random variables within the group will be fully ignored.

The probability integral transform collapsing function

The probability integral transform (PIT) collapsing function bears some resemblance to the multivariate extension of Spearman's rho discussed in Grothe et al. (2014). However, the definition of χ within our framework and its estimation procedure differ. The PIT-transformed collapsed random variable $F_{\boldsymbol{X}}(\boldsymbol{X})$ has distribution function $K_{\boldsymbol{X}}(t) = \mathbb{P}\{F_{\boldsymbol{X}}(\boldsymbol{X}) \leq t\}$, $t \in [0, 1]$, known as the *Kendall distribution*. Since $F_{\boldsymbol{X}}(\boldsymbol{X}) = C_{\boldsymbol{X}}\{F_{X_1}(X_1), \ldots, F_{X_p}(X_p)\} = C_{\boldsymbol{X}}(U_1, \ldots, U_p)$ for $\boldsymbol{U} = (U_1, \ldots, U_p) \sim C_{\boldsymbol{X}}$, $K_{\boldsymbol{X}}$ only depends on the copula $C_{\boldsymbol{X}}$ of $\boldsymbol{X}$ and can thus be viewed as a summary of the dependence of the components of $\boldsymbol{X}$ in the form of a p-variate function. Unfortunately, $K_{\boldsymbol{X}}$ itself is rarely analytically tractable for dimensions of $\boldsymbol{X}$ larger than two. Notable exceptions are Archimedean copulas $C_{\boldsymbol{X}}$ with generators ψ, for which a calculation based on the stochastic representation and a connection with the Poisson distribution function can be used to show that

$$K_{\boldsymbol{X}}(t) = \sum_{k=0}^{p-1} \frac{\psi^{(k)}\{\psi^{-1}(t)\}}{k!} \{-\psi^{-1}(t)\}^k, \quad t \in [0, 1]; \tag{2.5}$$

see the proof of Proposition 2.3.5 for this approach or Barbe et al. (1996) for the first appearance of this result.

Measures of association arising from the PIT collapsing function satisfy all five properties (P1)–(P5). For property (P4), note that by Sklar's Theorem, see Sklar (1959),

$$\begin{aligned} S(X_1, \ldots, X_p) &= F_{\boldsymbol{X}}(\boldsymbol{X}) = C_{\boldsymbol{X}}\{F_{X_1}(X_1), \ldots, F_{X_p}(X_p)\} \\ &= F_{(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))}\{F_{X_1}(X_1), \ldots, F_{X_p}(X_p)\} = F_{(U_1, \ldots, U_p)}(U_1, \ldots, U_p) \\ &= S(U_1, \ldots, U_p), \end{aligned}$$

where $(U_1, \ldots, U_p) = (F_{X_1}(X_1), \ldots, F_{X_p}(X_p)) \sim C_{\boldsymbol{X}}$. Therefore, the PIT as collapsing function also does not depend on the univariate marginal distributions. Property (P5) follows as a consequence of the resulting measure χ being copula-based together with the fact that this collapsing function naturally summarizes within-vector dependence. The

satisfaction of (P5) is a notable advantage of this collapsing function. Additionally, the measure χ here is a multivariate extension of Spearman's rho. Furthermore, using the PIT collapsing function, we can link the collapsed and original distribution functions and copulas more directly (see Section 2.3.3), thus allowing it to be used in hierarchical models; this was investigated in the context of Kendall copulas in Brechmann (2014). A downside is that the measures of association arising from the PIT collapsing function are not immediately amenable to the general asymptotic analysis using the tools we present in Section 2.2.4.

The pairwise distance collapsing function

One can choose virtually any type of distance D, for example, Euclidean, Manhattan, Canberra, and Minkowski as collapsing function S. Nested within our framework for this choice of collapsing function is a partial connection with the distance correlation of Székely et al. (2007). In particular, for the choice of Euclidean distance, we obtain a measure χ similar to that of the sample version of distance correlation. By default one can choose the Euclidean distance collapsing function given its link to distance correlation. However, numerical experiments have shown that it can sometimes be advantageous to choose the Canberra distance to avoid issues related to large distances resulting from outliers in the data. Beyond this, experimentation within the context of the data application objective is required.

Measures of association arising from the distance collapsing function satisfy properties (P1)′, (P2) and (P3). As with all previous measures, property (P4) is in general only satisfied if we replace $(X_1, \ldots, X_p)$ and $(X_1', \ldots, X_p')$ by $(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))$ and $(F_{X_1}(X_1'), \ldots, F_{X_p}(X_p'))$ in S. Measures χ resulting from the distance collapsing functions have the potential to detect various non-linear associations; see Székely et al. (2007) and the numerical experiments in Lopez-Paz et al. (2013) and Simon and Tibshirani (2014) for some corroboration in the context of distance correlations. In our framework, the broader choice of distance collapsing functions can potentially be utilized to capture a wider variety of non-linearities as needed. In particular, the differentiating advantage over copula-based measures lies in the ability to detect non-monotone associations. From an empirical perspective, the increased sample size in the collapsed space can be helpful for smaller datasets. However, with this comes the added computational burden and memory when dealing with larger datasets. The lack of interpretability, especially in the context of collapsed distribution functions and copulas is a disadvantage of all distance collapsing functions. Additionally, the inability to differentiate between positive and negative associations between groups of random variables is a notable drawback of distance collapsing functions.

The pairwise kernel collapsing function

One can choose any kernel function K, some of which are listed in Table 2.2.

Type of K	Kernel function $K(\cdot\,;\cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$
Linear	$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = \boldsymbol{x}_i^\top \boldsymbol{x}_k$
Polynomial (of order d)	$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = (1 + \boldsymbol{x}_i^\top \boldsymbol{x}_k)^d$
Gaussian	$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = \exp\left[-\{\|\boldsymbol{x}_i - \boldsymbol{x}_k\|_2^2/(2\sigma^2)\}\right]$
von Mises	$K(\boldsymbol{x}_i, \boldsymbol{x}_k) = \prod_{t=1}^p \exp\{\kappa_t \cos(x_{it} - x_{kt})\}$

Table 2.2: Examples of kernel functions.

By default one can choose the Gaussian kernel which is widely used as a sort of universal approximator; see Micchelli et al. (2006). For angular data, the von-Mises kernel based on its corresponding multivariate distribution as given in Mardia et al. (2008) would be a natural choice.

There are a few measures of association in the literature constructed with kernel similarity functions but with different formulations. These include the Hilbert Schmidt independence criterion of Gretton et al. (2008) and kernel canonical correlation coefficient of Bach and Jordan (2002).

Measures of association arising from the kernel collapsing function satisfy properties (P1)′, (P2) and (P3). Again, property (P4) is in general only satisfied if we replace $(X_1, \ldots, X_p)$ and $(X_1', \ldots, X_p')$ by $(F_{X_1}(X_1), \ldots, F_{X_p}(X_p))$ and $(F_{X_1}(X_1'), \ldots, F_{X_p}(X_p'))$ in S. Like the distance functions, the kernel collapsing functions can potentially detect various non-linear and non-monotone associations. Furthermore, the augmented sample size in the collapsed space is a benefit for smaller datasets and a computational burden for larger datasets. A notable drawback of using the kernel collapsing function is its inability to differentiate between positive and negative associations between groups of random variables. Additionally, the collapsed measure of association, the collapsed copula and the collapsed distribution function all lack interpretability.

The multivariate rank collapsing function

Using the multivariate rank collapsing function $S(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{1}\{\boldsymbol{x} \leq \boldsymbol{x}'\}$ to reduce multi-dimensional random vectors to a single dimension yields a rank-based measure of association χ. As usual, the inequality $\boldsymbol{x} \leq \boldsymbol{x}'$ is understood componentwise. The resulting

measure of association was first introduced in Grothe et al. (2014) as one possible multivariate extension of Kendall's tau. Analogously, a multivariate extension of Blomqvist's beta between random vectors can be obtained using $S(\boldsymbol{x}) = \mathbb{1}\{\boldsymbol{x} \leq F^{-}_{\boldsymbol{X}}(\mathbf{1/2})\}$, where $F^{-}_{\boldsymbol{X}}(\mathbf{1/2}) = (F^{-}_{X_1}(1/2), \ldots, F^{-}_{X_p}(1/2))$.

Measures of association arising from the rank collapsing function satisfy all five properties (P1)–(P5) listed in the introduction of Section 2.2. As to property (P4), note that if $F_{X_1}, \ldots, F_{X_p}$ are strictly increasing and continuous, the multivariate rank transform satisfies

$$\begin{aligned} S\{(X_1, \ldots, X_p),\ (X'_1, \ldots, X'_p)\} &= \mathbb{1}\{X_1 \leq X'_1, \ldots, X_p \leq X'_p\} \\ &= \mathbb{1}\{F_{X_1}(X_1) \leq F_{X_1}(X'_1), \ldots, F_{X_p}(X_p) \leq F_{X_p}(X'_p)\} \\ &= S\Big\{(F_{X_1}(X_1), \ldots, F_{X_p}(X_p)),\ (F_{X_1}(X'_1), \ldots, F_{X_p}(X'_p))\Big\} \end{aligned}$$

and thus does not depend on the marginal distributions $F_{X_1}, \ldots, F_{X_p}$. Property (P5) follows as a consequence of the resulting measure χ being copula-based, coupled with the fact that this collapsing function summarizes within-vector dependence; see Grothe et al. (2014) for further details. The satisfaction of (P5) is a notable feature of this collapsing function. Moreover, this measure χ represents a multivariate extension of Kendall's tau. One drawback in the context of our framework is the difficulty in interpreting the collapsed distribution function and copula. It is also worth noting that the computational burden for computing this measure χ can be high; this is a $2p$-variate collapsing function so computations are of the same order as the distance and kernel collapsing functions, because χ must be estimated from $\binom{n}{2}$ pairs of samples.

2.2.4 Estimation and asymptotic properties

In this section, we study estimators of χ in Equation (2.1), and derive asymptotic results which can be used to compute their standard errors. Also, see Appendix A.1.3, where we study estimators of τ in Equation (2.2).

Assume we have a random sample $(\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n)$ from $F_{\boldsymbol{X},\boldsymbol{Y}}$. Furthermore let $(\boldsymbol{X}', \boldsymbol{Y}')$ be an independent copy of $(\boldsymbol{X}, \boldsymbol{Y})$. An estimator χ_n of $\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}$ can be constructed by replacing ρ by the sample correlation coefficient. The following section investigates some properties of this estimator for collapsing functions S with known analytical forms. This excludes S that are stochastic and data-dependent such as the PIT collapsing function; the estimation of the PIT collapsing function is treated separately in Section 2.3.3.

We follow Grothe et al. (2014) and view χ_n through the lens of U-statistics to derive its asymptotic distribution.

Proposition 2.2.2 (Asymptotic distribution of χ_n)
Suppose $\chi_n(\boldsymbol{X}, \boldsymbol{Y})$ is defined as the sample correlation coefficient between $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$. Then, as $n \to \infty$,

$$\sqrt{n}(\chi_n - \chi) \xrightarrow{d} \mathrm{N}(0, \sigma_\chi^2),$$

where

$$\sigma_\chi^2 = \begin{cases} (\nabla f_{5\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_1 (\nabla f_{5\times 1}|_{\boldsymbol{\mu}}), & \text{if } S \text{ is a } p\text{-variate function,} \\ 4(\nabla f_{5\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_2 (\nabla f_{5\times 1}|_{\boldsymbol{\mu}}), & \text{if } S \text{ is a } 2p\text{-variate function.} \end{cases}$$

Here, $\nabla f_{5\times 1}|_{\boldsymbol{\mu}}$ denotes the gradient vector of the function

$$f(a, b, c, d, e) = \frac{e - ab}{\sqrt{c - a^2}\sqrt{d - b^2}},$$

evaluated at the population mean $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_{xx}, \mu_{yy}, \mu_{xy})$, where $\mu_x = \mathbb{E}\{S(\boldsymbol{X})\}$, $\mu_y = \mathbb{E}\{S(\boldsymbol{Y})\}$, $\mu_{xx} = \mathbb{E}\{S(\boldsymbol{X})^2\}$, $\mu_{yy} = \mathbb{E}\{S(\boldsymbol{Y})^2\}$, $\mu_{xy} = \mathbb{E}\{S(\boldsymbol{X})S(\boldsymbol{Y})\}$. Furthermore, Σ_1 denotes the covariance matrix of $(S(\boldsymbol{X}), S(\boldsymbol{Y}), S(\boldsymbol{X})^2, S(\boldsymbol{Y})^2, S(\boldsymbol{X})S(\boldsymbol{Y}))$ and Σ_2 denotes the covariance matrix of $(\mathbb{E}_{\boldsymbol{X}'}\{S(\boldsymbol{X}, \boldsymbol{X}')\}, \mathbb{E}_{\boldsymbol{Y}'}\{S(\boldsymbol{Y}, \boldsymbol{Y}')\}, \mathbb{E}_{\boldsymbol{X}'}\{S(\boldsymbol{X}, \boldsymbol{X}')^2\}, \mathbb{E}_{\boldsymbol{Y}'}\{S(\boldsymbol{Y}, \boldsymbol{Y}')^2\}, \mathbb{E}_{(\boldsymbol{X}', \boldsymbol{Y}')}\{S(\boldsymbol{X}, \boldsymbol{X}')S(\boldsymbol{Y}, \boldsymbol{Y}')\})$.

Proof. See Appendix A.1.1. □

Remark 2.2.3 (Estimation of σ_χ^2)
To estimate the asymptotic variance σ_χ^2 we adopt a plug-in approach as suggested by Grothe et al. (2014). This procedure has two key ingredients as summarized below and it will slightly differ between the two cases given in the proof of Proposition 2.2.2. Note moreover that the notation below is also explained in the proof of Proposition 2.2.2 in Appendix A.1.1. The two cases are those where S is a p-variate (Case 1) or a $2p$-variate (Case 2) function.

1. In a first step, evaluate the gradient vector $\nabla f_{5\times 1}$ at $\boldsymbol{m}^{(k)} = (m_x^{(k)}, m_y^{(k)}, m_{xx}^{(k)}, m_{yy}^{(k)}, m_{xy}^{(k)})$, $k \in \{1, 2\}$ corresponding to the sample quantities in Case k. The analytical form of the gradient vector evaluated at the appropriate values is given in Appendix A.1.2.

2. Now distinguish the two cases: In Case 1, estimate Σ_1 by the sample covariance matrix $\Sigma_{n,1}$ of $\Big(S(\boldsymbol{X}_i), S(\boldsymbol{Y}_i), S(\boldsymbol{X}_i)^2, S(\boldsymbol{Y}_i)^2, S(\boldsymbol{X}_i)S(\boldsymbol{Y}_i)\Big)$, $i \in \{1, \ldots, n\}$. In Case 2,

estimate Σ_2 by the sample covariance matrix $\Sigma_{n,2}$ of $\Big(g_x(\boldsymbol{X}_i), g_y(\boldsymbol{Y}_i), g_{xx}(\boldsymbol{X}_i), g_{yy}(\boldsymbol{Y}_i), g_{xy}(\boldsymbol{X}_i, \boldsymbol{Y}_i)\Big)$, $i \in \{1, \ldots, n\}$, where

$$g_x(\boldsymbol{X}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j), \quad g_y(\boldsymbol{Y}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} S(\boldsymbol{Y}_i, \boldsymbol{Y}_j),$$

$$g_{xx}(\boldsymbol{X}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j)^2, \quad g_{yy}(\boldsymbol{Y}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} S(\boldsymbol{Y}_i, \boldsymbol{Y}_j)^2,$$

$$g_{xy}(\boldsymbol{X}_i, \boldsymbol{Y}_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j) S(\boldsymbol{Y}_i, \boldsymbol{Y}_j).$$

The quantities g_x, g_y, g_{xx}, g_{yy}, g_{xy} estimate the conditional expectations $\mathbb{E}_{\boldsymbol{X}'}\{S(\boldsymbol{X}, \boldsymbol{X}')\}$, $\mathbb{E}_{\boldsymbol{Y}'}\{S(\boldsymbol{Y}, \boldsymbol{Y}')\}$, $\mathbb{E}_{\boldsymbol{X}'}\{S(\boldsymbol{X}, \boldsymbol{X}')^2\}$, $\mathbb{E}_{\boldsymbol{Y}'}\{S(\boldsymbol{Y}, \boldsymbol{Y}')^2\}$, $\mathbb{E}_{(\boldsymbol{X}', \boldsymbol{Y}')}\{S(\boldsymbol{X}, \boldsymbol{X}')S(\boldsymbol{Y}, \boldsymbol{Y}')\}$, respectively, and can be motivated using the jackknife methodology as Grothe et al. (2014) identified.

3. Then, $\sigma^2_{n,\chi} = (\nabla f|_{\boldsymbol{m}^{(1)}})^\top \Sigma_{n,1} (\nabla f|_{\boldsymbol{m}^{(1)}})$ in Case 1 and $\sigma^2_{n,\chi} = 4(\nabla f|_{\boldsymbol{m}^{(2)}})^\top \Sigma_{n,2} (\nabla f|_{\boldsymbol{m}^{(2)}})$ in Case 2.

2.3 Collapsed distribution functions and their copulas

While we can always compute and visualize realizations from the empirical collapsed copula – for $2p$-variate functions this requires using all $\binom{n}{2}$ pairs of observations – deriving an analytical form of the collapsed distribution function or collapsed copula in terms of the joint distribution of $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$ may be challenging. As an example, consider S to be the sum, in which case the marginal distributions of $(S(\boldsymbol{X}), S(\boldsymbol{Y}))$ are generally not analytically tractable even if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are vectors of independent components so that $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$ are convolutions.

There are two scenarios under which one may be able to derive explicit results. First, when $\boldsymbol{X}$ and $\boldsymbol{Y}$ have a specific dependence structure, as in the following subsection, and second, when the collapsing functions are suitable, as in the subsections thereafter. Most notably, an example of the latter scenario will yield a multivariate extension of the well-known Kendall distribution.

2.3.1 General collapsing functions

The following result uses the concept of strong-comonotonicity of Puccetti and Scarsini (2010), according to which $(\boldsymbol{X}, \boldsymbol{Y})$ is called *s-comonotone* if $\boldsymbol{X} = (F^-_{X_1}(U), \ldots, F^-_{X_p}(U))$ and $\boldsymbol{Y} = (F^-_{Y_1}(U), \ldots, F^-_{Y_q}(U))$ for $U \sim \mathrm{U}(0,1)$. An immediate extension of this result leads to the notion of strong-countermonotonicity: We call $(\boldsymbol{X}, \boldsymbol{Y})$ *s-countermonotone* if $\boldsymbol{X} = (F^-_{X_1}(U), \ldots, F^-_{X_p}(U))$ and $\boldsymbol{Y} = (F^-_{Y_1}(1-U), \ldots, F^-_{Y_q}(1-U))$ for $U \sim \mathrm{U}(0,1)$.

Proposition 2.3.1 (Independence, s-comonotonicity, s-countermonotonicity)
Let $\boldsymbol{X} \sim F_{\boldsymbol{X}}$ be a p-dimensional and $\boldsymbol{Y} \sim F_{\boldsymbol{Y}}$ be a q-dimensional random vector, both with continuously distributed margins.

1. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent, then $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) = uv$ for $u, v \in [0,1]$.
2. If $(\boldsymbol{X}, \boldsymbol{Y})$ is s-comonotone and $g(u) = S\Big\{(F^-_{X_1}(u), \ldots, F^-_{X_p}(u))\Big\}$ and $h(u) = S\Big\{(F^-_{Y_1}(u), \ldots, F^-_{Y_q}(u))\Big\}$ are strictly increasing functions, then $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) = \min(u,v)$ and thus the collapsed copula is the upper Fréchet–Hoeffding bound.
3. If $(\boldsymbol{X}, \boldsymbol{Y})$ is s-countermonotone and $g(u) = S\Big\{(F^-_{X_1}(u), \ldots, F^-_{X_p}(u))\Big\}$ and $h(u) = S\Big\{(F^-_{Y_1}(u), \ldots, F^-_{Y_q}(u))\Big\}$ are strictly increasing functions, then $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) = \max(u+v-1, 0)$ and thus the collapsed copula is the lower Fréchet–Hoeffding bound.

Proof.

1. By the Grouping Lemma (see (Resnick, 2014, Lemma 4.4.1) and (Durrett, 2004, Theorem 2.1.6)), which states that measurable functions of independent random variables are independent, $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$ are independent and so $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ is the independence copula.
2. For $U \sim \mathrm{U}(0,1)$, we have that $S(\boldsymbol{X}) = S\Big\{(F^-_{X_1}(U), \ldots, F^-_{X_p}(U))\Big\} = g(U)$ and $S(\boldsymbol{Y}) = S\Big\{(F^-_{Y_1}(U), \ldots, F^-_{Y_q}(U))\Big\} = h(U)$, so both collapsed random variables are increasing functions of the same $U \sim \mathrm{U}(0,1)$. Therefore, they are comonotone and thus their copula equals the upper Fréchet–Hoeffding bound.
3. For $U \sim \mathrm{U}(0,1)$, we have that $S(\boldsymbol{X}) = S\Big\{(F^-_{X_1}(U), \ldots, F^-_{X_p}(U))\Big\} = g(U)$ and $S(\boldsymbol{Y}) = S\Big\{(F^-_{Y_1}(1-U), \ldots, F^-_{Y_q}(1-U))\Big\} = h(1-U)$, so $S(\boldsymbol{X})$ is a strictly increasing and $S(\boldsymbol{Y})$ is a strictly decreasing function of U. Therefore, they are countermonotone and thus their copula equals the lower Fréchet–Hoeffding bound.

□

Note that the upper Fréchet–Hoeffding bound may also appear if $(\boldsymbol{X}, \boldsymbol{Y})$ is not s-comonotone. For a random vector $\boldsymbol{U}$ with $\mathrm{U}(0,1)$ margins and $\boldsymbol{X} = (F^-_{X_1}(U_1), \dots, F^-_{X_p}(U_p)) = \boldsymbol{Y}$, one has that $(S(\boldsymbol{X}), S(\boldsymbol{Y})) = (S(\boldsymbol{X}), S(\boldsymbol{X}))$ and thus that the collapsed copula is the upper Fréchet–Hoeffding bound. Any concordance measure of the collapsed random variables would therefore be 1.

The following examples show that many collapsing functions fall under the setup of Proposition 2.3.1 Parts 2 and 3 and thus that s-comontonicity and s-countermonotonicity imply that the collapsed copulas are the upper and lower Fréchet–Hoeffding bound, respectively.

Example 2.3.2 (Strict monotonicity of g, h for various collapsing functions)

1. Let S be the maximum collapsing function and let $F_{X_1}, \dots, F_{X_p}, F_{Y_1}, \dots, F_{Y_q}$ be continuous, so that $F^-_{X_1}, \dots, F^-_{X_p}, F^-_{Y_1}, \dots, F^-_{Y_q}$ are strictly increasing. Then $g(u) = S\Big\{(F^-_{X_1}(u), \dots, F^-_{X_p}(u))\Big\} = \max_{1\le j\le p}\{F^-_{X_j}(u)\}$ and $h(u) = S\Big\{(F^-_{Y_1}(u), \dots, F^-_{Y_q}(u))\Big\} = \max_{1\le k\le q}\{F^-_{Y_k}(u)\}$, which are strictly increasing in u.

2. Let S be the PIT collapsing function and let the diagonals of $C_{\boldsymbol{X}}$ and $C_{\boldsymbol{Y}}$ be strictly increasing. Then, by Sklar's Theorem, $g(u) = S\Big\{(F^-_{X_1}(u), \dots, F^-_{X_p}(u))\Big\} = F_{\boldsymbol{X}}\{F^-_{X_1}(u), \dots, F^-_{X_p}(u)\} = C_{\boldsymbol{X}}(u, \dots, u)$ and $h(u) = S\Big\{(F^-_{Y_1}(u), \dots, F^-_{Y_q}(u))\Big\} = F_{\boldsymbol{Y}}\{F^-_{Y_1}(u), \dots, F^-_{Y_q}(u)\} = C_{\boldsymbol{Y}}(u, \dots, u)$, which are strictly increasing in u.

3. Let S be the weighted sum collapsing function S with non-negative weights $\boldsymbol{w}$ and let $F_{X_1}, \dots, F_{X_p}, F_{Y_1}, \dots, F_{Y_q}$ be continuous, so that $F^-_{X_1}, \dots, F^-_{X_p}, F^-_{Y_1}, \dots, F^-_{Y_q}$ are strictly increasing. Then $g(u) = S\Big\{(F^-_{X_1}(u), \dots, F^-_{X_p}(u))\Big\} = \sum_{j=1}^p w_j F^-_{X_j}(u)$ and $h(u) = S\Big\{(F^-_{Y_1}(u), \dots, F^-_{Y_q}(u))\Big\} = \sum_{k=1}^q w_k F^-_{Y_k}(u)$, which are strictly increasing in u.

2.3.2 Maximum collapsing function

We now focus on the maximum collapsing function. An appealing property allows us to derive the collapsed distribution function and collapsed copula explicitly given that we know the joint distribution function of $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$.

Proposition 2.3.3 (The collapsed distribution and its copula for the maximum collapsing function)

Let $X_1, \dots, X_p, Y_1, \dots, Y_q$ be continuously distributed random variables with distribution functions $F_{X_1}, \dots, F_{X_p}, F_{Y_1}, \dots, F_{Y_q}$, respectively. Furthermore, let $F_{\boldsymbol{X},\boldsymbol{Y}}$ denote the distribution function of $(\boldsymbol{X}, \boldsymbol{Y})$ and consider the maximum collapsing function S. Then the collapsed distribution function $F_{(S(\boldsymbol{X}),S(\boldsymbol{Y}))}$ is $F_{(S(\boldsymbol{X}),S(\boldsymbol{Y}))}(x, y) = F_{\boldsymbol{X},\boldsymbol{Y}}(x, \dots, x, y, \dots, y)$ with corresponding collapsed copula

$$C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u, v) = F_{\boldsymbol{X},\boldsymbol{Y}}\{F^-_{S(\boldsymbol{X})}(u), \dots, F^-_{S(\boldsymbol{X})}(u), F^-_{S(\boldsymbol{Y})}(v), \dots, F^-_{S(\boldsymbol{Y})}(v)\}, \quad u, v \in [0, 1],$$

where $F^-_{S(\boldsymbol{X})}(u)$ and $F^-_{S(\boldsymbol{Y})}(v)$ denote the quantile functions of the distribution functions $F_{S(\boldsymbol{X})}(x) = F_{\boldsymbol{X},\boldsymbol{Y}}(x, \dots, x, \infty, \dots, \infty)$ and $F_{S(\boldsymbol{Y})}(y) = F_{\boldsymbol{X},\boldsymbol{Y}}(\infty, \dots, \infty, y, \dots, y)$, respectively.

Proof. Since $F_{(S(\boldsymbol{X}),S(\boldsymbol{Y}))}(x, y) = \mathbb{P}\{\max(X_1, \dots, X_p) \leq x, \max(Y_1, \dots, Y_q) \leq y\} = \mathbb{P}(X_1 \leq x, \dots, X_p \leq x, Y_1 \leq y, \dots, Y_q \leq y) = F_{\boldsymbol{X},\boldsymbol{Y}}(x, \dots, x, y, \dots, y)$ with margins $F_{S(\boldsymbol{X})}(x) = F_{\boldsymbol{X},\boldsymbol{Y}}(x, \dots, x, \infty, \dots, \infty)$ and $F_{S(\boldsymbol{Y})}(y) = F_{\boldsymbol{X},\boldsymbol{Y}}(\infty, \dots, \infty, y, \dots, y)$, Sklar's Theorem implies that the collapsed copula $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ is given as stated. □

Deriving the collapsed copula in special cases can provide a concrete understanding of the way in which the maximum collapsing function summarizes dependence between $\boldsymbol{X}$ and $\boldsymbol{Y}$. We present one example below; another one is given in Appendix A.2.

Example 2.3.4 (Meta nested Archimedean copula model and the maximum collapsing function)
Let $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y}) \sim F_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) = C_0\Big[C_1\{F_{X_1}(x_1), \dots, F_{X_p}(x_p)\}, C_2\{F_{Y_1}(y_1), \dots, F_{Y_q}(y_q)\}\Big]$, where $X_j \sim F_{X_j}$, $j \in \{1, \dots, p\}$, and $Y_k \sim F_{Y_k}$, $k \in \{1, \dots, q\}$, are continuously distributed. Let C_0, C_1, C_2 be Archimedean copulas with generators ψ_0, ψ_1, ψ_2 satisfying the sufficient nesting condition; see Hofert (2012) or McNeil (2008) for more details. Furthermore, consider the maximum collapsing function S.

1. If F_{X_j} is equal to F_X for all $j \in \{1, \dots, p\}$ and F_{Y_j} is equal to F_Y for all $j \in \{1, \dots, q\}$, then

$$S(\boldsymbol{X}) \sim F_{S(\boldsymbol{X})}(x) = C_1\{F_X(x), \dots, F_X(x)\} = \psi_1\Big[p\psi_1^{-1}\{F_X(x)\}\Big],$$
$$S(\boldsymbol{Y}) \sim F_{S(\boldsymbol{Y})}(y) = C_2\{F_Y(y), \dots, F_Y(y)\} = \psi_2\Big[q\psi_2^{-1}\{F_Y(y)\}\Big],$$

   with corresponding quantile functions

$$F^-_{S(\boldsymbol{X})}(u) = F^-_X\Big[\psi_1\{\psi_1^{-1}(u)/p\}\Big], \quad F^-_{S(\boldsymbol{Y})}(v) = F^-_Y\Big[\psi_2\{\psi_2^{-1}(v)/q\}\Big].$$

Proposition 2.3.3 implies that the collapsed copula equals

$$\begin{aligned}
C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) &= F_{\boldsymbol{X},\boldsymbol{Y}}\{F^-_{S(\boldsymbol{X})}(u),\dots,F^-_{S(\boldsymbol{X})}(u),F^-_{S(\boldsymbol{Y})}(v),\dots,F^-_{S(\boldsymbol{Y})}(v)\}\\
&= C_0\Big[C_1\Big\{F_X\Big(F_X^-\Big[\psi_1\{\psi_1^{-1}(u)/p\}\Big]\Big),\dots,F_X\Big(F_X^-\Big[\psi_1\{\psi_1^{-1}(u)/p\}\Big]\Big)\Big\},\\
&\qquad C_2\Big\{F_Y\Big(F_Y^-\Big[\psi_2\{\psi_2^{-1}(v)/q\}\Big]\Big),\dots,F_Y\Big(F_Y^-\Big[\psi_2\{\psi_2^{-1}(v)/q\}\Big]\Big)\Big\}\Big]\\
&= C_0\Big(C_1\Big[\psi_1\{\psi_1^{-1}(u)/p\},\dots,\psi_1\{\psi_1^{-1}(u)/p\}\Big],\\
&\qquad C_2\Big[\psi_2\{\psi_2^{-1}(v)/q\},\dots,\psi_2\{\psi_2^{-1}(v)/q\}\Big]\Big)\\
&= C_0(u,v), \quad u,v\in[0,1].
\end{aligned}$$

This is an intuitive result, as any two random variables (X_j, Y_k) have marginal copula C_0 under this model as do the group maxima as long as the marginal distributions are equal per group. This implies that any collapsed measure of concordance is precisely the one corresponding to the copula C_0 in this case.

2. In the general case where the margins of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are not all equal, we know that

$$\begin{aligned}
S(\boldsymbol{X}) \sim F_{S(\boldsymbol{X})}(x) &= \mathbb{P}\{\max_{1\le j\le p}(X_j)\le x\} \le \mathbb{P}(X_j\le x) = F_{X_j}(x), \quad j\in\{1,\dots,p\},\\
S(\boldsymbol{Y}) \sim F_{S(\boldsymbol{Y})}(y) &= \mathbb{P}\{\max_{1\le k\le q}(Y_k)\le y\} \le \mathbb{P}(Y_k\le y) = F_{Y_k}(y), \quad k\in\{1,\dots,q\},
\end{aligned}$$

and thus that

$$F^-_{S(\boldsymbol{X})}(u)\ge F^-_{X_j}(u), \quad j\in\{1,\dots,p\}, \qquad F^-_{S(\boldsymbol{Y})}(v)\ge F^-_{Y_k}(v), \quad k\in\{1,\dots,q\}.$$

We thus obtain the following lower bound for $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$

$$\begin{aligned}
C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) &= F_{\boldsymbol{X},\boldsymbol{Y}}\{F^-_{S(\boldsymbol{X})}(u),\dots,F^-_{S(\boldsymbol{X})}(u),F^-_{S(\boldsymbol{Y})}(v),\dots,F^-_{S(\boldsymbol{Y})}(v)\}\\
&= C_0\Big(C_1\Big[F_{X_1}\{F^-_{S(\boldsymbol{X})}(u)\},\dots,F_{X_p}\{F^-_{S(\boldsymbol{X})}(u)\}\Big],\\
&\qquad C_2\Big[F_{Y_1}\{F^-_{S(\boldsymbol{Y})}(v)\},\dots,F_{Y_q}\{F^-_{S(\boldsymbol{Y})}(v)\}\Big]\Big)\\
&\ge C_0\Big(C_1\Big[F_{X_1}\{F^-_{X_1}(u)\},\dots,F_{X_p}\{F^-_{X_p}(u)\}\Big],\\
&\qquad C_2\Big[F_{Y_1}\{F^-_{Y_1}(v)\},\dots,F_{Y_q}\{F^-_{Y_q}(v)\}\Big]\Big)\\
&= C_0\{C_1(u,\dots,u),\ C_2(v,\dots,v)\}, \quad u,v\in[0,1].
\end{aligned}$$

2.3.3 PIT collapsing function

For the PIT collapsing function, the collapsed distribution function and copula have notable terminology and notation following from the copula literature. In that spirit, we will present them as extensions of the Kendall distribution and adopt the same notation.

Definition

Let $\boldsymbol{U} \sim C_{\boldsymbol{X}}$ and $\boldsymbol{V} \sim C_{\boldsymbol{Y}}$ for the copulas $C_{\boldsymbol{X}}$ and $C_{\boldsymbol{Y}}$ of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. For $t_1, t_2 \in [0,1]$, define the *multivariate* (or *joint*) *Kendall distribution* by

$$K_{\boldsymbol{X},\boldsymbol{Y}}(t_1, t_2) = \mathbb{P}\{F_{\boldsymbol{X}}(\boldsymbol{X}) \leq t_1,\ F_{\boldsymbol{Y}}(\boldsymbol{Y}) \leq t_2\} = \mathbb{P}\{C_{\boldsymbol{X}}(\boldsymbol{U}) \leq t_1,\ C_{\boldsymbol{Y}}(\boldsymbol{V}) \leq t_2\}.$$

It is straightforward to define higher-dimensional Kendall distributions having univariate Kendall distributions as margins. The copula of $K_{\boldsymbol{X},\boldsymbol{Y}}(t_1, t_2)$, if uniquely determined, follows from Sklar's Theorem via

$$C_K(u_1, u_2) = K_{\boldsymbol{X},\boldsymbol{Y}}\{K_{\boldsymbol{X}}^-(u_1), K_{\boldsymbol{Y}}^-(u_2)\}, \quad u_1, u_2 \in [0,1], \tag{2.6}$$

where $K_{\boldsymbol{X}}^-$ and $K_{\boldsymbol{Y}}^-$ denote the quantile functions of the marginal Kendall distributions $K_{\boldsymbol{X}}$ and $K_{\boldsymbol{Y}}$, respectively. We call C_K the *Kendall copula*. Kendall copulas have previously appeared in Brechmann (2014) as hierarchical Kendall copulas without explicitly investigating the notion of joint Kendall distributions; the latter naturally appear in our framework for measuring dependence between random vectors.

Properties

We now briefly discuss some basic properties of multivariate Kendall distributions and Kendall copulas (as before, we focus on the bivariate case); see also Appendix A.3 where several measures of association $\chi(\boldsymbol{X}, \boldsymbol{Y})$ are expressed in terms of the multivariate Kendall distribution.

As in (2.5), where an analytical formula for univariate Kendall distributions for Archimedean copulas is given, an explicit form can be given for multivariate Kendall distributions.

Proposition 2.3.5 (Multivariate Kendall distribution in the Archimedean case)
Let $(\boldsymbol{X}, \boldsymbol{Y})$ be a $(p+q)$-dimensional random vector with Archimedean copula C with

completely monotone generator ψ. Then, for all $t_1, t_2 \in [0,1]$,

$$K_{\boldsymbol{X},\boldsymbol{Y}}(t_1,t_2) = \sum_{m=0}^{(p-1)(q-1)} \Big[\sum_{n=0}^{m} \frac{\{\psi^{-1}(t_1)\}^n}{n!} \frac{\{\psi^{-1}(t_2)\}^{m-n}}{(m-n)!} \Big] (-1)^m \psi^{(m)}\{\psi^{-1}(t_1) + \psi^{-1}(t_2)\}. \quad (2.7)$$

Proof. Let $V \sim F_V$, where F_V is the Laplace–Stieltjes inverse of ψ and let $E_{11}, \dots, E_{1p}, E_{21}, \dots, E_{2q} \overset{\text{ind.}}{\sim} \text{Exp}(1)$. Furthermore, let

$$\boldsymbol{U} = \Big(\psi\Big(\frac{E_{11}}{V}\Big), \dots, \psi\Big(\frac{E_{1p}}{V}\Big)\Big) \quad \text{and} \quad \boldsymbol{V} = \Big(\psi\Big(\frac{E_{21}}{V}\Big), \dots, \psi\Big(\frac{E_{2q}}{V}\Big)\Big).$$

Note that $(\boldsymbol{U}, \boldsymbol{V}) \sim C$ and that $\boldsymbol{U} \sim C_{\boldsymbol{X}}$ and $\boldsymbol{V} \sim C_{\boldsymbol{Y}}$, where $C_{\boldsymbol{X}}, C_{\boldsymbol{Y}}$ are (also) Archimedean copulas with generator ψ. As a result, $(\boldsymbol{X}, \boldsymbol{Y})$ allows for the stochastic representation

$$(\boldsymbol{X}, \boldsymbol{Y}) = (F^-_{X_1}(U_{11}), \dots, F^-_{X_p}(U_{1p}),\ F^-_{Y_1}(U_{21}), \dots, F^-_{Y_q}(U_{2q})).$$

Thus

$$\begin{aligned}
&K_{\boldsymbol{X},\boldsymbol{Y}}(t_1,t_2) = \mathbb{P}\{F_{\boldsymbol{X}}(\boldsymbol{X}) \le t_1,\ F_{\boldsymbol{Y}}(\boldsymbol{Y}) \le t_2\} = \mathbb{P}\{C_{\boldsymbol{X}}(\boldsymbol{U}) \le t_1,\ C_{\boldsymbol{Y}}(\boldsymbol{V}) \le t_2\} \\
&= \mathbb{P}\{E_{11} + \dots + E_{1p} > V\psi^{-1}(t_1),\ E_{21} + \dots + E_{2q} > V\psi^{-1}(t_2)\} \\
&= \int_0^\infty \mathbb{P}\{E_{11} + \dots + E_{1p} > v\psi^{-1}(t_1),\ E_{21} + \dots + E_{2q} > v\psi^{-1}(t_2)\}\, \mathrm{d}F_V(v) \\
&= \int_0^\infty \mathbb{P}\{E_{11} + \dots + E_{1p} > v\psi^{-1}(t_1)\}\, \mathbb{P}\{E_{21} + \dots + E_{2q} > v\psi^{-1}(t_2)\}\, \mathrm{d}F_V(v) \\
&= \int_0^\infty F_{\text{Poi}\{v\psi^{-1}(t_1)\}}(p-1) F_{\text{Poi}\{v\psi^{-1}(t_2)\}}(q-1)\, \mathrm{d}F_V(v) \\
&= \int_0^\infty \exp\{-v\psi^{-1}(t_1)\} \sum_{k=0}^{p-1} \frac{\{v\psi^{-1}(t_1)\}^k}{k!} \exp\{-v\psi^{-1}(t_2)\} \sum_{l=0}^{q-1} \frac{\{v\psi^{-1}(t_2)\}^l}{l!}\, \mathrm{d}F_V(v) \\
&= \int_0^\infty \exp\Big[-v\{\psi^{-1}(t_1) + \psi^{-1}(t_2)\}\Big] \sum_{m=0}^{(p-1)(q-1)} \Big[\sum_{n=0}^{m} \frac{\{\psi^{-1}(t_1)\}^n}{n!} \frac{\{\psi^{-1}(t_2)\}^{m-n}}{(m-n)!} \Big] v^m\, \mathrm{d}F_V(v) \\
&= \sum_{m=0}^{(p-1)(q-1)} \Big[\sum_{n=0}^{m} \frac{\{\psi^{-1}(t_1)\}^n}{n!} \frac{\{\psi^{-1}(t_2)\}^{m-n}}{(m-n)!} \Big] \psi^{(m)}\{\psi^{-1}(t_1) + \psi^{-1}(t_2)\}(-1)^m,
\end{aligned}$$

where we used the fact that the survival function of an Erlang distribution can be expressed as the distribution function F_{Poi} of a Poisson distribution. □

Note that (2.5) follows from (2.7) as a special case. Moreover, it is straightforward to extend (2.7) to higher dimensions. In this case, each random vector in the construction

corresponds to a single dimension of the multivariate Kendall distribution. As a special case, when each such random vector consists of only a single random variable, the multivariate Kendall distribution equals the copula of these random variables.

Figures 2.1 and 2.2 display scatter plots of $n = 1000$ independent observations of the bivariate Gumbel and Clayton Kendall copulas. The parameter of the underlying Gumbel and Clayton generator are chosen such that Kendall's tau equals 0.5. The different plots depict how varying dimensions p, q impact the dependence structure between the two random vectors. This difference manifests itself in the form of asymmetry (lower vs upper tails) and the strength of dependence (comparing the case $(p, q) = (2, 2)$ to $(p, q) = (50, 50)$). Concerning the latter, one can give a heuristic argument. With the notation as before, note that $C_{\boldsymbol{X}}(\boldsymbol{U}) = \psi(p\bar{E}_{1\cdot}/V)$ and $C_{\boldsymbol{Y}}(\boldsymbol{V}) = \psi(q\bar{E}_{2\cdot}/V)$, where, almost surely by the Strong Law of Large Numbers, $\bar{E}_{1\cdot} = \sum_{j=1}^{p} E_{1j}/p \to 1$ and $\bar{E}_{2\cdot} = \sum_{k=1}^{q} E_{2k}/q \to 1$. Therefore, for large p and q, $C_{\boldsymbol{X}}(\boldsymbol{U}) \approx \psi(p/V)$ and $C_{\boldsymbol{Y}}(\boldsymbol{V}) \approx \psi(q/V)$ which are increasing functions of the same random variable V and thus their copula approximates the upper Fréchet–Hoeffding bound. Finally, we note the asymmetry in Figures 2.1 and 2.2 in the pull of the realizations towards the diagonal representing perfect dependence, which is stronger below the diagonal if $p > q$ and above the diagonal if $p < q$.

Note that besides Example 2.3.2 Part 2, the PIT collapsing function S also leads to the collapsed copula being the upper Fréchet–Hoeffding bound under the notion of π-comonotonicity of Puccetti and Scarsini (2010). If $p = q$, $\boldsymbol{X} = (F^-_{X_1}(U_1), \dots, F^-_{X_p}(U_p))$ and $\boldsymbol{Y} = (F^-_{Y_1}(U_1), \dots, F^-_{Y_p}(U_p))$ for $\boldsymbol{U} \sim C$ for some copula C, then, by Sklar's Theorem, $S(\boldsymbol{X}) = F_{\boldsymbol{X}}(\boldsymbol{X}) = C(U_1, \dots, U_p)$ and $S(\boldsymbol{Y}) = F_{\boldsymbol{Y}}(\boldsymbol{Y}) = C(U_1, \dots, U_p)$, so the collapsed random variables are comonotone and thus the collapsed copula is the upper Fréchet–Hoeffding bound.

Nonparametric estimators of univariate Kendall distributions based on a random sample $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i \in \{1, \dots, n\}$, can be constructed as follows. Let

$$\boldsymbol{W}_i = (W_{i1}, W_{i2}) = \Big(\frac{1}{n-1} \sum_{\substack{k=1 \\ k \neq i}}^{n} \mathbb{1}\{\boldsymbol{X}_k \leq \boldsymbol{X}_i\}, \ \frac{1}{n-1} \sum_{\substack{k=1 \\ k \neq i}}^{n} \mathbb{1}\{\boldsymbol{Y}_k \leq \boldsymbol{Y}_i\} \Big),$$

where, as usual, the inequalities are understood componentwise. Similar to Barbe et al. (1996) and Genest and Rivest (1993) in the univariate case, one can use the empirical distribution function

$$K_n(\boldsymbol{t}) = K_n(t_1, t_2) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\boldsymbol{W}_i \leq \boldsymbol{t}\} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{W_{i1} \leq t_1, W_{i2} \leq t_2\}, \quad \boldsymbol{t} = (t_1, t_2) \in [0, 1]^2,$$

in the multivariate case as a nonparametric estimator of $K_{\boldsymbol{X},\boldsymbol{Y}}(t_1, t_2)$.

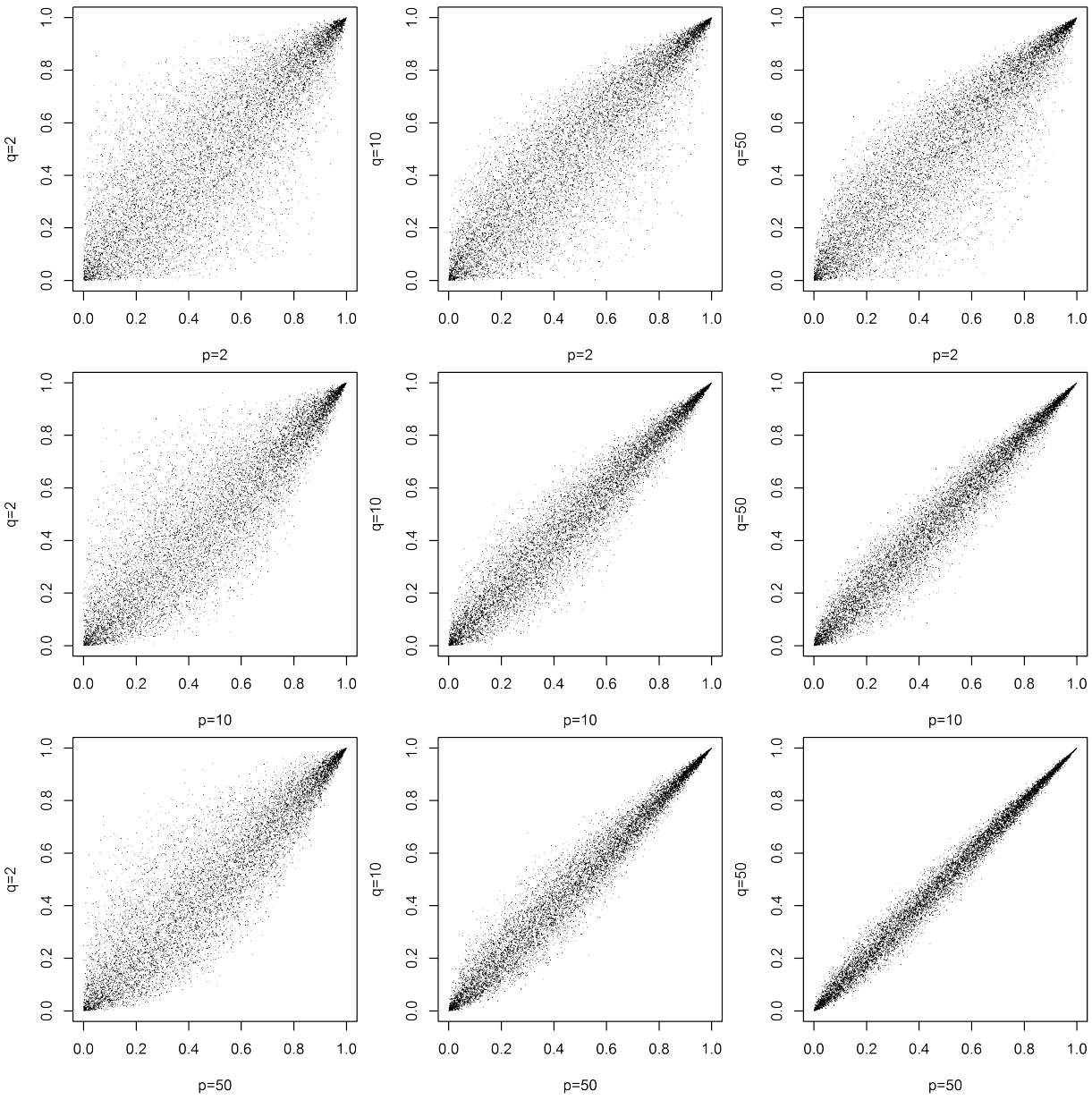

Figure 2.1: $n = 1000$ independent observations from different Gumbel Kendall copulas (with Gumbel parameter chosen such that Kendall's tau of the underlying generator equals 0.5) corresponding to the joint Kendall distribution function as specified in (2.7). Note the dimensions of the two sectors are varied with $p \in \{2, 10, 50\}$ and $q \in \{2, 10, 50\}$, thus leading to nine different variations.

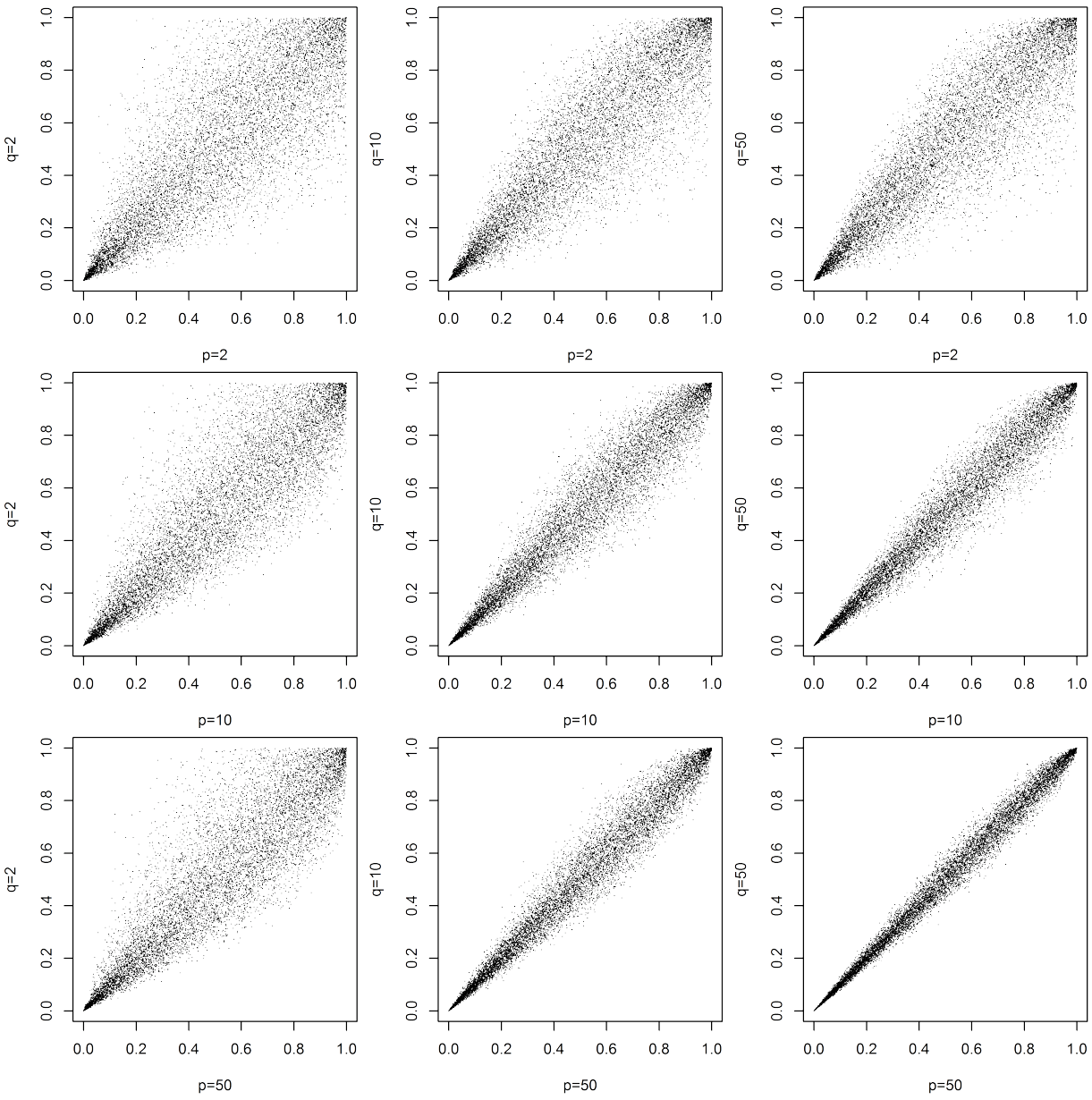

Figure 2.2: $n = 1000$ independent observations from different Clayton Kendall copulas (with Clayton parameter chosen such that Kendall's tau of the underlying generator equals 0.5) corresponding to the joint Kendall distribution function as specified in (2.7). Note the dimensions of the two sectors are varied with $p \in \{2, 10, 50\}$ and $q \in \{2, 10, 50\}$, thus leading to nine different variations.

Estimation for the PIT collapsing function

We now discuss the construction of an estimator for $\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\}$. To begin with, let $W_1 = F_{\boldsymbol{X}}(\boldsymbol{X})$ and $W_2 = F_{\boldsymbol{Y}}(\boldsymbol{Y})$. As in Barbe et al. (1996), we consider the pseudo-observations

$$W_{i1} = \frac{1}{n-1} \sum_{\substack{k=1 \\ k \neq i}}^{n} \mathbb{1}\{\boldsymbol{X}_k \leq \boldsymbol{X}_i\}, \quad W_{i2} = \frac{1}{n-1} \sum_{\substack{k=1 \\ k \neq i}}^{n} \mathbb{1}\{\boldsymbol{Y}_k \leq \boldsymbol{Y}_i\}, \quad i \in \{1, \ldots, n\},$$

where the inequalities are understood componentwise. As before, an estimator for the measure of association $\chi(\boldsymbol{X}, \boldsymbol{Y})$ can simply be constructed via the sample correlation coefficient, that is, $\chi_n(\boldsymbol{X}, \boldsymbol{Y}) = \rho_n(W_{i1}, W_{i2})$. As this particular estimator does not fit in the U-statistic framework, it is harder to derive asymptotic normality with an expression for the asymptotic variance for this collapsing function. One can construct bootstrap confidence intervals provided that convergence in distribution is established for χ_n. However, this asymptotic result remains to be found.

Based on the pseudo-observations defined above, one can also estimate $\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho_S\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\}$ by $\chi_n(\boldsymbol{X}, \boldsymbol{Y}) = \rho_n\{K_{n,\boldsymbol{X}}(W_{i1}), K_{n,\boldsymbol{Y}}(W_{i2})\}$, where

$$K_{n,\boldsymbol{X}}(t_1) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{W_{i1} \leq t_1\}, \quad K_{n,\boldsymbol{Y}}(t_2) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{W_{i2} \leq t_2\}, \quad t_1, t_2 \in [0, 1].$$

2.4 Applications

In practice, the choice of collapsing functions will depend on the context. The weighted sum/average collapsing function is often the most obvious and interpretable choice. Another strong contender is the distance collapsing function, which is related to the distance correlation of Székely et al. (2007) for which there exist well established theoretical results. In this section we present two applications to illustrate how our framework can be applied in practice.

2.4.1 Protein data: An application from bioinformatics

Proteins are complex molecules composed of sequences of amino acid residues of which there are 20 different types. All share a generic structure, R-CH(NH_2)-COOH, where the component labelled "R", also known as a side chain, identifies the specific type of amino acid.

In bioinformatics, scientists are interested in understanding how conformational changes at different side chains may be coupled together (Ghoraie et al., 2015a). For example, if two residues are far apart in the sequence but their side chains tend to change conformation together, it may be an indication that they are close in 3D. In turn, this may shed light on the all-important underlying protein folding process.

The conformation of a side chain can be characterized by a set of dihedral angles. To understand this, picture a side chain as a sequence of atoms spanning off the backbone of the protein. The angle between planes formed by atoms 1–3 and atoms 2–4 in the sequence is referred to as the *first* dihedral angle, and so on. Typically, there are zero to four such dihedral angles depending on the size of the underlying amino acid.

Thus, let $\boldsymbol{X} = (X_1, \ldots, X_p)$, $0 \leq p \leq 4$, and $\boldsymbol{Y} = (Y_1, \ldots, Y_q)$, $0 \leq q \leq 4$, represent the dihedral angles of two side chains, respectively. We need a measure of association between the two random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$. To quantify their dependence, Ghoraie et al. (2015b) applied the Graphical LASSO (GLASSO) developed by Friedman et al. (2008), while Ghoraie et al. (2015a) used "kernelized partial canonical correlation analysis" (KPCCA). Here, we apply our framework of collapsing functions.

Analysis

We report results using various collapsing functions – in particular, the weighted average, the pairwise distance, the pairwise kernel, and the PIT.

For the weighted average, we consider putting more weight on the first few dihedral angles. This is because dihedral angles closer to the backbone of the protein are more restricted in their motion, so changes in their conformations contain much more biological information than those further away. In particular, we consider the extreme case $\boldsymbol{w} = (1, 0, ..., 0)$, that is, full weight on the first dimension, which results in the bivariate measure of association $\rho(X_1, Y_1)$. For the pairwise distance, we include only the Euclidean distance because, after experimenting with other distance functions, there was little to no difference for this application.

For the pairwise kernel, we follow Ghoraie et al. (2015a) and use a multivariate von-Mises kernel,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{t=1}^{p} \exp\{\kappa_t \cos(x_{it} - x_{jt})\},$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^p$ are two different conformations of a given side chain. We simply use the same concentration parameters as adopted and justified by Ghoraie et al. (2015a), so

$\kappa_1 = 8$, $\kappa_2 = 8$, $\kappa_3 = 4$ and $\kappa_4 = 2$. These choices were made because atoms farther away from the backbone have more freedom of motion.

Finally, the PIT is a general-purpose choice of collapsing function that can capture both positive and negative association. However, for the purpose of ranking dependencies we are only interested in the strength of dependence, so we use $|\chi(\boldsymbol{X}, \boldsymbol{Y})|$ as the ranking criteria.

We use the same dataset as Ghoraie et al. (2015a) which allows for a direct comparison of the results. Altogether, Ghoraie et al. (2015a) studied eight different types of proteins from three different families (Ras, Rho and Rab). Each protein has a varying number of residues approximately in the range of 160–190. Through a specific procedure explained in Ghoraie et al. (2015a), roughly 16,000–18,000 sample conformations for these proteins were generated.

Note that working with the pairwise distance and kernel collapsing functions is computationally prohibitive. For each protein we have up to 18,000 sample conformations which would have resulted in $\binom{18{,}000}{2}$ samples in the collapsed space. We thus consider ten random subsets of size 5000 without replacement from the original dataset and compute the relevant evaluation criteria as an average across the subsets.

The objective is to rank all pairs of residues in a protein according to various measures of association, and to verify whether "known couplings" appear in the top-ranked pairs. Following Ghoraie et al. (2015a), "known couplings" were based on the Contact Rearrangement Network (CRN) method from Daily et al. (2008). The receiver-operating characteristic (ROC) curve – in particular, the area under the ROC curve (AUC) – is used as a summarizing evaluation criterion to determine how well the rankings produced by different measures agree with the CRN method's results; the AUC is well-known to have the interpretation of being the probability that an algorithm ranks a true signal ahead of a false one (Hanley and McNeil, 1982).

Results and discussion

We compare the resulting AUC values from the chosen collapsing functions with results from KPCCA Ghoraie et al. (2015a) and GLASSO Ghoraie et al. (2015b).

From Table 2.3, we see that measures of association resulting from all collapsing functions have AUC values much greater than 50%, so they are all significantly better than detecting allosteric couplings at random. Particularly, the distance collapsing function yielded the best allosteric coupling detection amongst measures of association arising from our framework. For three proteins 1G16, 1KAO, and 1XTQ, measures of association resulting from the

Protein PBD ID	H-Ras 4Q21	RhoA 1FTN	Rap2A 1KAO	Rheb 1XTQ	Sec4 1G16	Cdc42 1ANO	Rac1 1HH4(A)	Ypt7p 1KY3
KPCCA	80	75	69	70	68	68	67	72
Distance	77	73	72	71	69	66	64	64
Weighted Average	78	74	72	65	71	66	67	59
GLASSO	78	72	68	71	68	68	59	67
Kernel	73	71	69	71	68	69	65	64
PIT	74	68	70	71	68	61	59	57

Table 2.3: AUC with respect to CRN, where the AUC values are in percent. The rows and columns are organized in decreasing order of row and column means. Note that the "PDB ID" is a unique identifier of the inactive state of the protein; see Berman et al. (2006).

distance collapsing function yielded better results compared to KPCCA and GLASSO. Furthermore, simple yet meaningful collapsing functions, such as the weighted average with $\boldsymbol{w} = (1, 0, ..., 0)$, often yielded comparable and for 1G16 and 1KAO superior results to KPCCA and GLASSO. This is an interesting observation, given that this particular collapsing function is considerably faster and easier to understand than the mathematically sophisticated KPCCA or GLASSO methods and the computationally cumbersome distance collapsing function.

2.4.2 S&P 500: An application from finance

Numerous problems in finance and risk management require the study of dependence between random vectors or groups of random variables. In this section, we explore such a problem by investigating dependence between S&P 500 business sectors. As we are dealing with time series data, this problem can be viewed both through the lens of static and dynamic dependence. Fixing a time period, we can assess whether the business sectors are independent by visualizing the dependence between them. Additionally, we can compute time-varying measures of association to dynamically capture dependence between business sectors.

S&P 500 constituent data

For the static case, we consider the 465 available constituent time series from the S&P 500 in the time period from 2007-01-01 to 2009-12-31 (756 trading days); see the R package

`qrmdata` of Hofert and Hornik (2016). For the dynamic case, we consider 461 of these constituent time series after accounting for missing data. We use the ten Global Industry Classification Standard (GICS) sectors as business sectors. Nine of the ten GICS sectors have Exchange Traded Funds (ETFs) which track the performance of each business sector; they are also known as sector Standard and Poor's Depository Receipt (SPDR) ETFs. We use a bivariate measure of association between any two sector ETFs as a market-determined benchmark for comparison.

To pre-process the dataset, we work with negative log-returns for each constituent and fit ARMA(1,1)-GARCH(1,1) models to each time series. We then extract the corresponding standardized residuals to investigate dependence between the component series; see Patton (2006) for this procedure. The same pre-processing is applied to each of the nine ETF time series.

A snapshot of S&P 500 sector dependence

Before thinking about modeling dependence, one should test the hypothesis $\mathcal{H}_0$ that all random variables are independent. Note that the hypothesis $\mathcal{H}_{0,c}$ that the collapsed random variables are independent is a subset of $\mathcal{H}_0$. If $\mathcal{H}_{0,c}$ is rejected, so is $\mathcal{H}_0$. The following algorithm, which easily extends to more than two groups of random variables, provides a simple graphical assessment of $\mathcal{H}_{0,c}$.

Algorithm 2.4.1 (Graphical assessment of independence for two groups of random variables)
Let $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i \in \{1, \dots, n\}$, be a random sample from $(\boldsymbol{X}, \boldsymbol{Y})$ and assume $S(\boldsymbol{X})$ and $S(\boldsymbol{Y})$ are continuously distributed. To visually check independence of $\boldsymbol{X}$ and $\boldsymbol{Y}$ based on $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$, $i \in \{1, \dots, n\}$, do:

1. Compute the collapsed variables $S_{i1} = S(\boldsymbol{X}_i)$ and $S_{i2} = S(\boldsymbol{Y}_i)$, $i \in \{1, \dots, k\}$, where $k = n$ for p-variate functions and $k = \binom{n}{2}$ for $2p$-variate functions.

2. Compute the pseudo-observations $U_{k,ij} = R_{ij}/(k+1)$, $i \in \{1, \dots, k\}$, $j \in \{1, 2\}$, where, for each $j \in \{1, 2\}$, R_{ij} denotes the rank of S_{ig} among $S_{1g}, \dots, S_{kj}$.

3. Plot the pseudo-observations $(U_{k,i1}, U_{k,i2})$, $i \in \{1, \dots, k\}$. The less the visualized samples resemble realizations from $\mathrm{U}(0,1)^2$, the greater the evidence against $\mathcal{H}_{0,c}$ and thus $\mathcal{H}_0$.

An interesting question is whether our visual assessment of independence is independent of the marginal distributions of the $p+q$ components of $(\boldsymbol{X}, \boldsymbol{Y})$. This certainly depends on

the collapsing function. In general, it does not matter for an assessment of independence, but for better interpretability one could of course build pseudo-observations of the given data from $(\boldsymbol{X}, \boldsymbol{Y})$ before applying Algorithm 2.4.1; note that in this case, one would apply pseudo-observations at two levels, to the original variables and to the collapsed variables.

Following Algorithm 2.4.1, we can perform an assessment of independence between business sectors. In particular, we use Euclidean distance, equally-weighted average, maximum, and PIT collapsing functions. We also visualize the dependence between all 36 ETF sector pairs for comparison. There are only 36 ETF sector pairs since the Telecommunications sector does not have an ETF.

Figure 2.3 illustrates this graphical assessment of independence with four zenplots, one for each choice of collapsing function. Zenplots are zigzag expanded navigation plots where adjacent bivariate plots share the same variable. This leads to more flexible plot layouts which are less wasteful concerning space than scatter-plot matrices. If necessary, the bivariate plots can also be ordered according to some measure from "most" to "least" important; see Hofert and Oldford (2017) for more details. As can be clearly detected from the chosen collapsing functions, the business sectors cannot be assumed to be independent. To facilitate the comparison of the collapsed variables with the benchmark, Figure 2.4 also shows the pairwise dependence structures between the nine sector ETFs.

The four zenplots in Figure 2.3 can be interpreted as realizations from the underlying and unknown collapsed copula. Realizations from the Euclidean distance collapsed copula are denser in comparison to realizations from the other three collapsing functions because there are $\binom{756}{2}$ realizations as opposed to just 756. In particular, due to the nature of the distance function, it is difficult to interpret features of the dependence structure between business sectors, such as tail dependence, asymmetry and shape in the context of the original variables portrayed in the corresponding zenplot. As a result, for applications in finance, the distance collapsing function should only be used for a quick graphical assessments of independence.

Since the weighted average collapsing function is most natural for return data, the interpretations of tail dependence and asymmetry translate well from the bivariate case. We naturally see the similarity in the dependence structures between the weighted average collapsing function and the benchmark ETFs in Figure 2.4. Furthermore, since the PIT collapsing function leads to realizations from the Kendall copula, it also yields an attractive interpretation of the dependence structure depicted in its corresponding zenplot. For instance, as noted in Example A.3.4, the tail dependence coefficients in this case can be interpreted as natural multivariate extensions of bivariate tail dependence. Owing to the justification of these two collapsing functions and interpretability, one could potentially fit

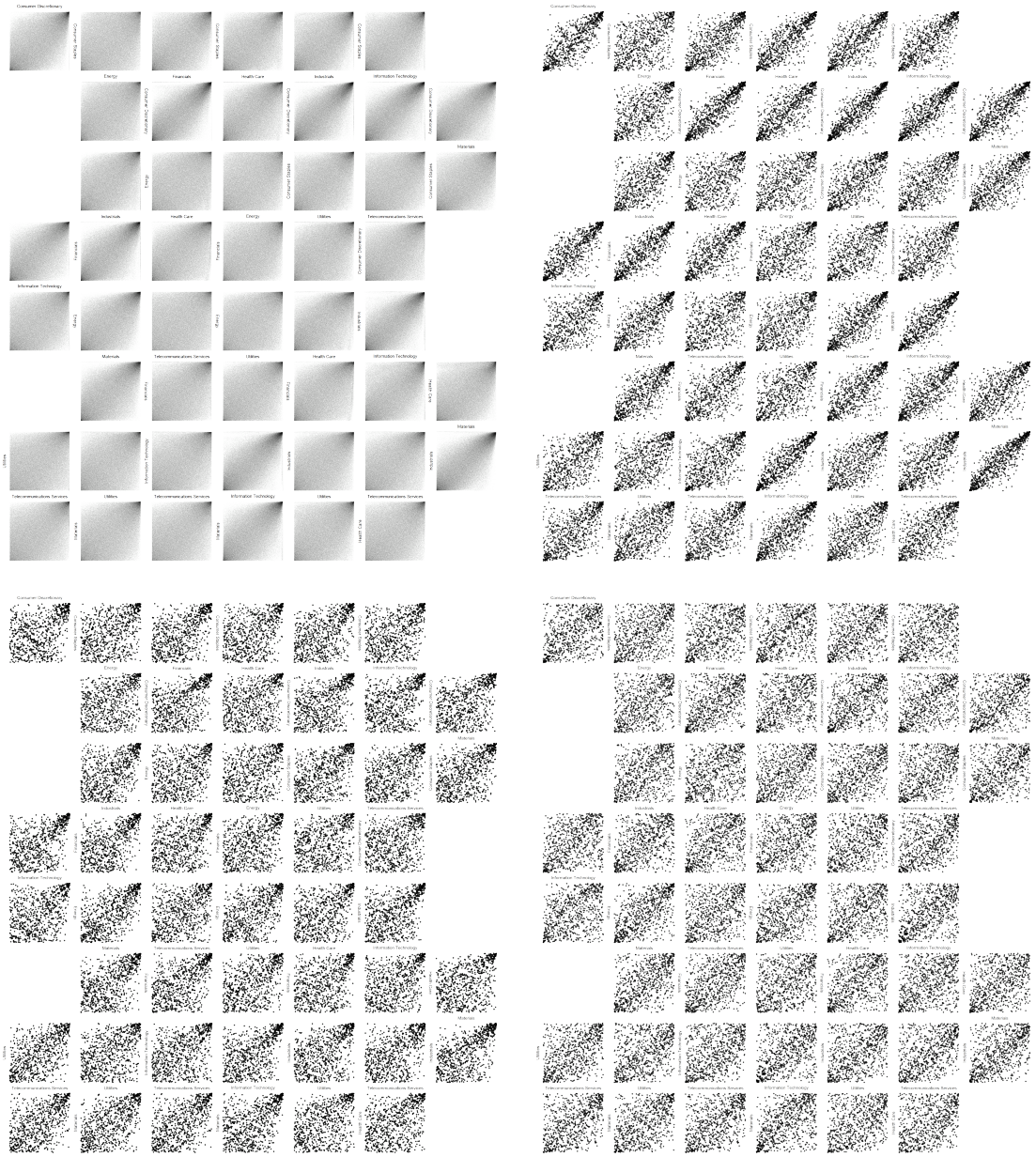

Figure 2.3: Zenplots displaying all pairs of pseudo-observations for the 10 GICS sectors of the 465-dimensional S&P 500 data based on the Euclidean distance (top left), weighted average (top right), PIT (bottom left), and maximum (bottom right) collapsing functions.

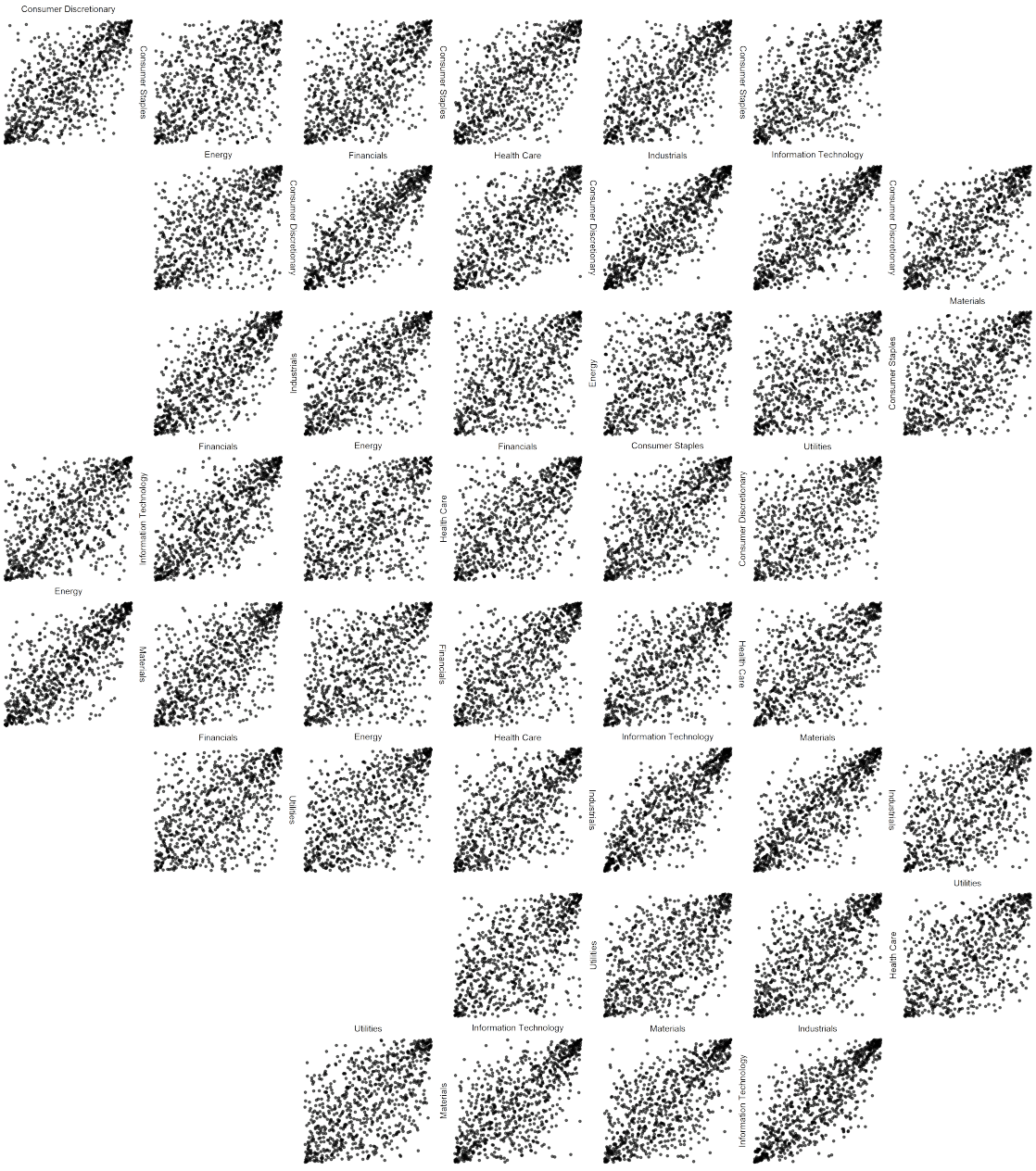

Figure 2.4: Zenplot displaying all pairs of pseudo-observations for the nine GICS Sector ETFs.

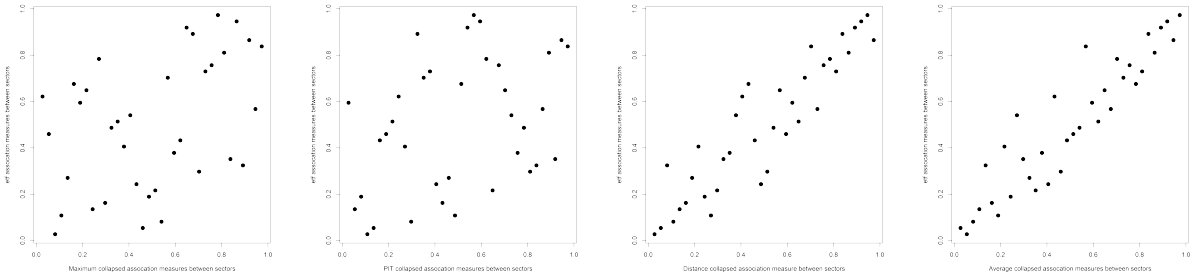

Figure 2.5: Scatter plots displaying pseudo-observations of maximum, PIT, Euclidean distance, and average collapsed measures of association between the nine GICS sectors versus measures of association between the corresponding nine GICS Sector ETFs.

a copula model directly to the collapsed variables to model a notion of dependence between groups of random variables, but this framework will in general not offer an analytically tractable link back to the original random variables.

The maximum collapsing function appears to capture a weaker form of dependence compared to the other collapsing functions and the benchmark. This is to be expected as this collapsing function describes a notion of dependence between the worst performers only. In particular, it describes this notation in a plural sense in that the constituent chosen as the maximum can change daily in each business sector over the time period considered.

To check which collapsing functions best capture dependence with respect to the benchmark, Figure 2.5 shows scatter plots of pseudo-observations of collapsed measures of association between nine GICS sectors versus measures of association between the corresponding GICS sector ETFs. In particular, in increasing order of concordance between the collapsed and ETF measures of association, we have the maximum, PIT, distance, and average collapsing functions. We can clearly infer that the measures of association arising from the distance and average collapsing functions match the dependencies between the sector ETFs more closely. Since ETFs which are tradeable securities are marketed as weighted averages of sector constituents, the measures of association arising from the average collapsing function match the sector ETF dependencies most naturally. The notable observation from this check is the ability of the Euclidean distance collapsing function to match the ETF dependencies closely.

Dynamic S&P 500 sector dependence

We will now capture the dynamic dependence between these sectors using a moving window. In particular, we investigate how between-sector dependencies changed over time from 2006-01-01 to 2015-12-31. Using a 150-day moving window, Figure 2.6 depicts the time-varying dependence as captured by the distance, average, maximum, and PIT collapsing functions for each of four randomly chosen pairs of business sectors. Also included for comparison with the benchmark is the measure of association between ETFs for each pair. While the measures of association resulting from different collapsing functions lie on different scales, they all capture the same shifts in dependence not only with respect to each other but also with respect to the market-determined ETF dependence series. This indicates the suitability of any of these collapsing functions to the task of detecting dependence and the shifts in the strength of dependence over time. Furthermore, ETFs are marketed as weighted averages of sector constituents, but are tradeable securities in their own right and thus exposed to market forces. Such a construction of ETFs explains why the average collapsing function would most closely track the dependence between ETFs despite the use of equal weights in our collapsing function and despite the influence that market forces might have on the dependence between sector ETFs.

Figure 2.7 shows the time-varying dependence as captured by the distance, average, and maximum collapsing functions with their corresponding confidence intervals constructed using Proposition 2.2.2 and Remark 2.2.3. Shown in the background are all pairwise bivariate time-varying measures of association between individual constituents of the two sectors. This juxtaposition highlights that the measures of association between collapsed random variables capture fairly similar shifts in strength of dependence over time compared with all the pairwise measures of association between the sectors. Furthermore, one can see that the width of confidence intervals for the various collapsed measures of association is well within the width of the background band representing all the bivariate dependence series between individual constituents from each sector. This provides further intuitive corroboration that the collapsing functions capture time-varying dependence between groups of random variables at least when compared to a series of matrices of pairwise measures of association.

2.5 Discussion

There is no universal notion of a "best" collapsing function. All reasonable collapsing functions we investigated tend to capture dependence between random vectors in a similar

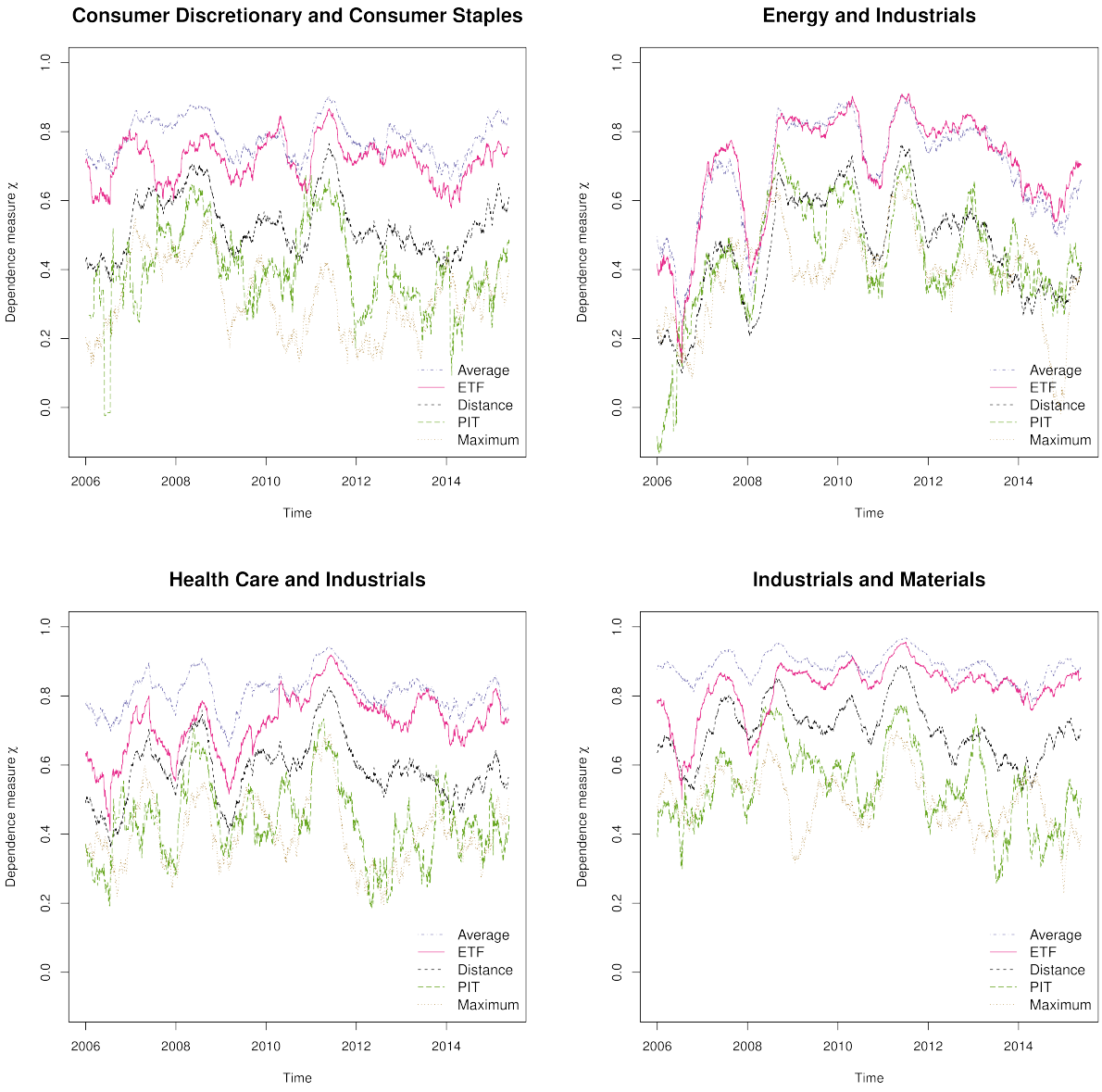

Figure 2.6: Time-varying measure of association for various collapsing functions and the ETFs between a few selected pairs of business sectors. The four pairs of sectors arbitrarily selected are as follows: Consumer Discretionary vs. Consumer Staples (top left), Energy vs. Industrials (top right), Health Care vs. Industrials (bottom left), and Industrials vs Materials (bottom right).

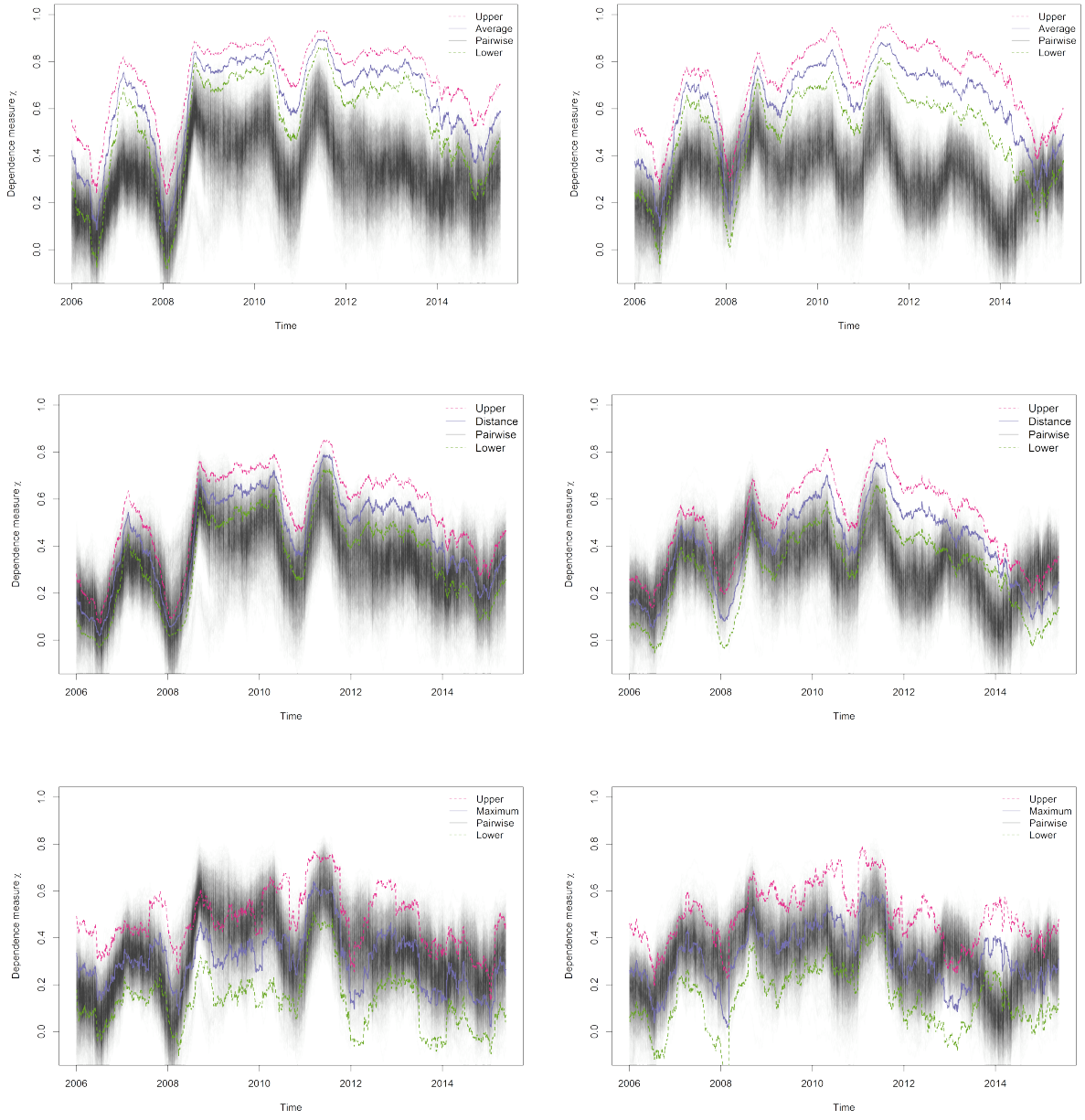

Figure 2.7: Time-varying measure of association for average (top plots), distance (middle plots), and maximum (bottom plots) collapsing functions with 95% confidence intervals against a backdrop of all pairwise time-varying measures between assets in the two business sectors considered. On the left panel we present the plots for Consumer Discretionary vs. Energy sectors and on the right panel we present the plots for Energy vs. Health Care sectors.

fashion. As we outlined in detail in Section 2.2.3, there are notable properties, advantages, and disadvantages for each collapsing function. From our protein data application, we found that the Euclidean distance and the weighted-average collapsing functions yielded competitive results in the ranking task. Moreover, the special case of the weighted average function was linked with a particular biological meaning thus offering some interpretability and insight for scientists in the field. From our finance example and particularly in the static dependence context, we saw that the equally-weighted average and Euclidean distance collapsing functions most closely matched the ETF dependence between S&P 500 sectors. The weighted average function is a natural choice in the context of finance and its tracking of the benchmark dependence was further evident in the dynamic context of our finance example. The other salient observation extracted from the two data applications is the usefulness of the general purpose Euclidean distance collapsing function in measuring dependence between random vectors. While it lacks interpretability and adaptability in terms of the collapsed distribution functions and copulas, it is closely related to the distance correlation of Székely et al. (2007) and appears to be a competitive metric for measuring and ranking dependencies.

Chapter 3

Hierarchical Archimax copulas

3.1 Introduction

The class of Archimax copulas, see Capéraà et al. (2000) and Charpentier et al. (2014a), generalizes Archimedean copulas to incorporate a stable tail dependence function as known from extreme-value copulas. As special cases, Archimax copulas can be Archimedean or extreme-value copulas and thus extend both of these classes of copulas. They provide a link between dependence structures arising in multivariate extremes and Archimedean copulas, which have intuitive and computationally appealing properties. One feature of Archimedean copulas is that they can be *nested* in the sense that one can (under assumptions detailed later) plug Archimedean copulas into each other and still obtain proper copulas. Such a construction is *hierarchical* in the sense that certain multivariate margins are exchangeable, yet the copula overall is not; this additional flexibility to allow for (partial) asymmetry over an exchangeable model is typically used to model components belonging to different groups, clusters or business sectors. In this work, we raise the following natural question (see Sections 3.2 and 3.3):

> How can hierarchical Archimax copulas be constructed?

Since we work with stochastic representations, sampling is also covered. Constructing nested Archimax copulas is largely an open problem which we discuss in Appendix B.2. Moreover, to fill a gap in the literature, we present a general formula for the density and its evaluation of Archimax copulas; see Appendix B.1.

In what follows, we assume the reader to be familiar with the basics of Archimedean copulas (ACs) and extreme-value copulas (EVCs); see, for example, McNeil and Nešlehová

(2009) for the former (from which we also adopt the notation) and (Jaworski et al., 2010, Chapter 6) for the latter.

3.2 Hierarchical extreme-value copulas via hierarchical stable tail dependence functions

3.2.1 Connection between d-norms and stable tail dependence functions

A copula C is an extreme-value copula if and only if it is *max-stable*, that is, if

$$C(\boldsymbol{u}) = C(u_1^{1/m}, \dots, u_d^{1/m})^m, \quad m \in \mathbb{N},\ \boldsymbol{u} = (u_1, \dots, u_d) \in [0,1]^d;$$

see, for example, Jaworski et al. (2010, Theorem 6.2.1). An extreme-value copula C can be characterized in terms of its stable tail dependence function $\ell : [0,\infty)^d \to [0,\infty)$ via

$$C(\boldsymbol{u}) = \exp\{-\ell(-\ln u_1, \dots, -\ln u_d)\}, \quad \boldsymbol{u} \in [0,1]^d; \tag{3.1}$$

see, for example, Beirlant et al. (2004, Section 8.2) and Jaworski et al. (2010, Chapter 6). A characterization of stable tail dependence functions ℓ is given in Charpentier et al. (2014a) and Ressel (2013). (being homogeneous of order 1, being 1 when evaluated at the unit vectors in $\mathbb{R}^d$ and being fully d-max decreasing).

Sampling from EVCs is usually quite challenging and time-consuming for the most popular models. Examples which are comparably easy to sample are Gumbel and nested Gumbel copulas, the only Archimedean and nested Archimedean EVCs, respectively, where a stochastic representation is available; see Nelsen (2006, Theorem 4.5.2).

- The *Gumbel* (or *logistic*) *copula* C with parameter $\alpha \in (0,1]$ and stable tail dependence function $\ell(\boldsymbol{x}) = (x_1^{1/\alpha} + \cdots + x_d^{1/\alpha})^\alpha$, $\boldsymbol{x} \in [0,\infty)^d$, can be sampled using the algorithm of Marshall and Olkin (1988). It utilizes the stochastic representation

$$\boldsymbol{U} = \left(\psi\left(\frac{E_1}{V}\right), \dots, \psi\left(\frac{E_d}{V}\right)\right) \sim C, \tag{3.2}$$

  where $\psi(t) = \exp(-t^\alpha)$ is a Gumbel generator, $E_1, \dots, E_d \overset{\text{ind.}}{\sim} \text{Exp}(1)$, independently of the *frailty* $V \sim \mathcal{PS}(\alpha) = \text{S}(\alpha, 1, \cos^{1/\alpha}(\alpha\pi/2), \mathbb{1}_{\{\alpha=1\}}; 1)$; see (Nolan, 2017, p. 8) for the parameterization of this α-stable distribution.

- Nested Gumbel copulas, see Tawn (1990), can also be sampled based on a stochastic representation corresponding to the nesting structure; see McNeil (2008). The main idea is to replace the single frailty V by a sequence of dependent frailties (all α-stable for different α), nested in a specific way; see Section 3.3.

For more complicated EVCs, Schlather (2002), Dieker and Mikosch (2015) and Dombry et al. (2016) have proposed approximate or exact simulation schemes based on the following stochastic representation of max-stable processes; see de Haan (1984), Penrose (1992) and Schlather (2002).

Theorem 3.2.1 (Spectral representation of max-stable processes)
Let $\{W_i(\boldsymbol{s})\}_{i=1}^{\infty}$ be independent copies of the random process $W(\boldsymbol{s})$, $\boldsymbol{s} \in \mathcal{S} \subseteq \mathbb{R}^q$, such that $W(\boldsymbol{s}) \geq 0$ and $\mathbb{E}\{W(\boldsymbol{s})\} = 1$, $\boldsymbol{s} \in \mathcal{S}$. Furthermore, let $\{P_i\}_{i=1}^{\infty}$ be points of a Poisson point process on $[0, \infty)$ with intensity $x^{-2}\,\mathrm{d}x$. Then

$$Z(\boldsymbol{s}) = \sup_{i \geq 1}\{P_i W_i(\boldsymbol{s})\} \tag{3.3}$$

is a max-stable random process with unit Fréchet margins and

$$\ell(x_1, \ldots, x_d) = \mathbb{E}(\max_{1 \leq j \leq d}\{x_j W(\boldsymbol{s}_j)\}), \quad x_1, \ldots, x_d > 0, \tag{3.4}$$

is the associated stable tail dependence function of the random vector $(Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_d))$ for fixed $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d$. Therefore, if a process $Z(\boldsymbol{s})$ can be expressed as in (3.3), the distribution function of the random vector $(Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_d))$ is $\mathbb{P}\{Z(\boldsymbol{s}_1) \leq x_1, \ldots, Z(\boldsymbol{s}_d) \leq x_d\} = \exp\{-\ell(1/x_1, \ldots, 1/x_d)\}$, that is, $(Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_d))$ has EVC C with stable tail dependence function ℓ and unit Fréchet margins $\exp(-1/x_j)$, $j \in \{1, \ldots, d\}$.

For completeness, Algorithm 3.2.2 below describes the traditional approach for simulating max-stable processes constructed using (3.3). This algorithm goes back to Schlather (2002) and provides approximate simulations by truncating the supremum to a finite number of processes in (3.3). When the random process $W(\boldsymbol{s})$ is bounded almost surely, a stopping criterion may be designed to optimally select the number of Poisson points N to perform exact simulation. For more general exact sampling schemes, we refer to Dieker and Mikosch (2015) and Dombry et al. (2016).

Algorithm 3.2.2 (Approximate sampling of max-stable processes based on (3.3))

1. Simulate N Poisson points $\{P_i\}_{i=1}^{N}$ in decreasing order as $P_i = 1/\sum_{k=1}^{i} E_k$, $i \in \{1, \ldots, N\}$, where $E_k \overset{\text{ind.}}{\sim} \text{Exp}(1)$, $k \in \{1, \ldots, N\}$.

2. Simulate N independent copies $\{W_i(\boldsymbol{s})\}_{i=1}^N$ of the process $W(\boldsymbol{s})$ at a finite set of locations $\boldsymbol{s} \in \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d\}$.

3. For each location $\boldsymbol{s} \in \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d\}$, set $Z(\boldsymbol{s}) = \max_{1 \leq i \leq N}\{P_i W_i(\boldsymbol{s})\}$.

By choosing the spatial domain $\mathcal{S}$ in (3.3) to be finite and replacing $W(\boldsymbol{s}_1), \ldots, W(\boldsymbol{s}_d)$ by non-negative random variables $W_1, \ldots, W_d$ with $\mathbb{E}(W_j) = 1$, $j \in \{1, \ldots, d\}$, thus replacing the random process $W(\boldsymbol{s})$ by the non-negative random vector $\boldsymbol{W} = (W_1, \ldots, W_d)$, this representation also provides a characterization of, and sampling algorithms for, (finite-dimensional) EVCs; from here on we will adopt this "vector case" for W and accordingly for Z.

We now turn to the link between max-stable random vectors $(Z_1, \ldots, Z_d)$ and d-norms as recently described in Aulbach et al. (2015). A norm $\|\cdot\|_d$ on $\mathbb{R}^d$ is called a *d-norm* if there exists a random vector $\boldsymbol{W} = (W_1, \ldots, W_d)$ with $W_j \geq 0$ and $\mathbb{E}(W_j) = 1$, $j \in \{1, \ldots, d\}$, such that

$$\|\boldsymbol{x}\|_d = \mathbb{E}(\max_{1 \leq j \leq d}\{|x_j| W_j\}) = \mathbb{E}(\|\boldsymbol{x}\boldsymbol{W}\|_\infty), \quad \boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d, \tag{3.5}$$

where $\|\cdot\|_\infty$ denotes the supremum norm and $\boldsymbol{x}\boldsymbol{W}$ is understood componentwise. In this case, $\boldsymbol{W}$ is called *generator* of $\|\cdot\|_d$. One can compare (3.4) and (3.5) to identify the correspondence

$$\ell(\boldsymbol{x}) = \|\boldsymbol{x}\|_d = \mathbb{E}(\|\boldsymbol{x}\boldsymbol{W}\|_\infty), \quad \boldsymbol{x} \in [0, \infty)^d, \tag{3.6}$$

between d-norms and stable tail dependence functions on $[0, \infty)^d$. Specifying a generator $\boldsymbol{W}$ thus defines a stable tail dependence function which in turn characterizes an EVC. The link (3.6) with d-norms provides us with a useful method for constructing and sampling EVCs which can also be exploited for constructing hierarchical EVCs (HEVCs).

We now provide a few examples of d-norm generators for well known copulas which can serve as building blocks for HEVCs (and, see Section 3.3, hierarchical Archimax copulas).

Example 3.2.3

1. If $\boldsymbol{W} = (1, \ldots, 1)$ with probability one, then $\|\boldsymbol{x}\|_d = \max_{1 \leq j \leq d} |x_j|$. This characterizes comonotonicity, that is, the upper Fréchet–Hoeffding bound with stable tail dependence function $\ell(\boldsymbol{x}) = \max\{x_1, \ldots, x_d\}$.

2. If $\boldsymbol{W}$ is a random permutation of $(d, 0, \ldots, 0) \in \mathbb{R}^d$, then $\|\boldsymbol{x}\|_d = d \sum_{j=1}^d |x_j|/d = \sum_{j=1}^d |x_j|$. This characterizes independence with the stable tail dependence function $\ell(\boldsymbol{x}) = x_1 + \cdots + x_d$.

3. If $\boldsymbol{W} = (W_1, \dots, W_d)$ is such that for some $0 < \alpha < 1$, $\Gamma(1-\alpha)W_j \overset{\text{ind.}}{\sim} \exp(-x^{-1/\alpha})$, $x \in [0, \infty)$, where Γ denotes the gamma function, a straightforward computation shows that $\|\boldsymbol{x}\|_d = (\sum_{j=1}^d |x_j|^{1/\alpha})^\alpha$. This implies that $\ell(\boldsymbol{x}) = (\sum_{j=1}^d x_j^{1/\alpha})^\alpha$ and thus that the max-stable dependence structure is the Gumbel (logistic) copula with parameter $\alpha \in (0, 1)$.

4. If $\boldsymbol{W}$ is such that for some $\theta > 0$, $W_j = \Gamma(1 + 1/\theta)W_j^*$ with $W_j^* \overset{\text{ind.}}{\sim} \exp(-x^\theta)$, $x \in [0, \infty)$, then the stable tail dependence function can be calculated to be

$$\ell(\boldsymbol{x}) = \sum_{\emptyset \neq J \subseteq \{1,\dots,d\}} (-1)^{|J|+1} \left(\sum_{j \in J} x_j^{-\theta} \right)^{-1/\theta},$$

and thus the max-stable dependence structure is the negative logistic copula with parameter $\theta > 0$; see, for example, Dombry et al. (2016).

5. If $\boldsymbol{W} = (W_1, \dots, W_d) \sim (\sqrt{2\pi}\max\{0, \varepsilon_1\}, \dots, \sqrt{2\pi}\max\{0, \varepsilon_d\})$, where $(\varepsilon_1, \dots, \varepsilon_d) \sim \mathrm{N}_d(\boldsymbol{0}, P)$ with correlation matrix P, a *Schlather model* results; see Schlather (2002).

6. If $\boldsymbol{W} = (W_1, \dots, W_d) \sim (\max\{0, \varepsilon_1\}^\nu / c_\nu, \dots, \max\{0, \varepsilon_d\}^\nu / c_\nu)$, where $(\varepsilon_1, \dots, \varepsilon_d) \sim \mathrm{N}_d(\boldsymbol{0}, P)$ with correlation matrix P, $\nu > 0$, and $c_\nu = 2^{\nu/2-1}\Gamma\{(\nu+1)/2\}/\sqrt{\pi}$, then the *extremal t model* of Opitz (2013) results; for $\nu = 1$, the Schlather model is obtained as a special case. The stable tail dependence function $\ell(\boldsymbol{x})$ of the extremal t model in dimension d is given by

$$\ell(\boldsymbol{x}) = \sum_{j=1}^d x_j t_{d-1}\left(\nu + 1,\ P_{-j,j},\ \frac{P_{-j,-j} - P_{-j,j}P_{j,-j}}{\nu + 1}\right)\{(\boldsymbol{x}_{-j}/x_j)^{-1/\nu}\}, \tag{3.7}$$

where $t_d(\nu, \boldsymbol{\mu}, \Sigma)(\boldsymbol{x})$ denotes the d-variate Student t distribution function with ν degrees of freedom, location vector $\boldsymbol{\mu}$ and dispersion matrix Σ evaluated at $\boldsymbol{x}$ as in (McNeil et al., 2015, Example 6.7), $P_{-j,-j}$ (respectively, $P_{-j,j}$, $P_{j,-j}$) denotes the submatrix obtained by removing the jth row and the jth column (respectively, jth row, jth column) from P and $\boldsymbol{x}_{-j} = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_d)$.

7. If $\boldsymbol{W} = (W_1, \dots, W_d) \sim (\exp(\varepsilon_1 - \sigma_1^2/2), \dots, \exp(\varepsilon_d - \sigma_d^2/2))$, where $(\varepsilon_1, \dots, \varepsilon_d) \sim \mathrm{N}_d(\boldsymbol{0}, \Sigma)$ for a covariance matrix Σ with diagonal entries $\Sigma_{jj} = \sigma_j^2$, $j \in \{1, \dots, d\}$, and corresponding correlation matrix P (such that $\Sigma_{ij} = \sigma_i\sigma_j P_{ij}$, $i, j \in \{1, \dots, d\}$), a *Brown–Resnick model* results; see Kabluchko et al. (2009). This model can also be obtained as a certain limit of the extremal t model when the degrees of freedom $\nu \to \infty$; see Nikoloulopoulos et al. (2009). The Brown–Resnick model is characterized

by the *Hüsler–Reiss copula*; see Hüsler and Reiss (1989). Its stable tail dependence function $\ell(\boldsymbol{x})$ is available in any dimension d, see Huser and Davison (2013) and Nikoloulopoulos et al. (2009), and given by

$$\ell(\boldsymbol{x}) = \sum_{j=1}^{d} x_j \Phi_{d-1}(\boldsymbol{0}, \Sigma_j)(\boldsymbol{\eta}_j), \tag{3.8}$$

where $\Phi_d(\boldsymbol{\mu}, \Sigma)(\boldsymbol{x})$ denotes the d-variate normal distribution function with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ evaluated at $\boldsymbol{x}$, Σ_j is the $(d-1)\times(d-1)$ covariance matrix with entries

$$\Sigma_{j,ik} = \begin{cases} 2\gamma_{ij}, & \text{if } k = i \in \{1,\ldots,d\}\backslash\{j\}, \\ \gamma_{ij} + \gamma_{jk} - \gamma_{ik}, & \text{if } k \neq i, \end{cases}$$

where $\gamma_{ij} = \sigma_i^2 + \sigma_j^2 - \sigma_i\sigma_j P_{ij}$, and $\boldsymbol{\eta}_j$ is the $(d-1)$-dimensional vector with ith entry $\gamma_{ij} - \ln(x_i/x_j)$.

8. If $\boldsymbol{W} = (W_1, \ldots, W_d) \sim H$ for a distribution function H with margins $F_1, \ldots, F_d$ on $[0, \infty)$ such that $\mathbb{E}(W_j) = 1$, $j \in \{1, \ldots, d\}$, then, by Sklar's Theorem, if C denotes the copula of H, one can derive the general form of ℓ via (3.6). If $\boldsymbol{U} \sim C$, then the stochastic representation $\boldsymbol{W} = (F_1^-(U_1), \ldots, F_d^-(U_d))$ can be used to see that, for all $\boldsymbol{x} > \boldsymbol{0}$,

$$\begin{aligned} G_{\boldsymbol{x}}(y) &= \mathbb{P}(\max_{1\le j\le d}\{|x_j|W_j\} \le y) = \mathbb{P}(W_1 \le y/x_1, \ldots, W_d \le y/x_d) \\ &= \mathbb{P}\{U_1 \le F_1(y/x_1), \ldots, U_d \le F_d(y/x_d)\} = C\{F_1(y/x_1), \ldots, F_d(y/x_d)\}. \end{aligned}$$

Applying the chain rule for differentiating this expression with respect to y leads to the density

$$g_{\boldsymbol{x}}(y) = \sum_{j=1}^{d} \mathrm{D}_j\, C\{F_1(y/x_1), \ldots, F_d(y/x_d)\} f_j(y/x_j)/x_j,$$

where $\mathrm{D}_j\, C(\boldsymbol{u})$ denotes the partial derivatives of C with respect to the jth argument evaluated at $\boldsymbol{u}$. By (3.6) and the substitution $z_j = y/x_j$, we thus have that, for all

$\boldsymbol{x} > \mathbf{0}$,

$$\begin{aligned}\ell(\boldsymbol{x}) &= \int_0^\infty y g_{\boldsymbol{x}}(y)\,\mathrm{d}y = \sum_{j=1}^d \frac{1}{x_j}\int_0^\infty y\,\mathrm{D}_j\,C\{F_1(y/x_1),\ldots,F_d(y/x_d)\}f_j(y/x_j)\,\mathrm{d}y \\ &= \sum_{j=1}^d x_j \int_0^\infty z_j\,\mathrm{D}_j\,C\{F_1(z_j x_j/x_1),\ldots,F_d(z_j x_j/x_d)\}f_j(z_j)\,\mathrm{d}z_j \\ &= \sum_{j=1}^d x_j \mathbb{E}[Z_j\,\mathrm{D}_j\,C\{F_1(Z_j x_j/x_1),\ldots,F_d(Z_j x_j/x_d)\}],\end{aligned}$$

where $Z_1 \sim F_1, \ldots, Z_d \sim F_d$ are independent. This formula resembles (3.7) and (3.8). If required, it can be evaluated by Monte Carlo, for example. Note that it only poses a restriction on the marginal distributions (being non-negative and scalable to have mean 1), not the dependence of the components of $\boldsymbol{W}$.

3.2.2 Hierarchical stable tail dependence functions

Let us now turn to a construction method for HEVCs by exploiting the link between d-norm generators and stable tail dependence functions established in Section 3.2.1. The idea is to build stable tail dependence functions with a hierarchical structure at the level of the associated d-norm generator. Although our approach is similar in spirit to Lee and Joe (2017) who recently proposed factor extreme-value copula models, the two constructions differ.

By analogy with the construction of nested Archimedean copulas (outlined in Section 3.3) we define hierarchical d-norm generators $\boldsymbol{W} = (W_1, \ldots, W_d)$ in terms of a tree structure with d leaves. Under this framework, each component W_j, $j \in \{1, \ldots, d\}$, is obtained as a measurable, non-negative function g_j of intermediate variables $\{W_k^*\}_{k\in \mathrm{Anc}(j)}$, lying on the tree nodes along the path from the seed W_0^* at the root of the tree to the jth leaf represented by the variable W_j itself. In other words, the variable W_j may be expressed in terms of its ancestor variables identified by the index set $\mathrm{Anc}(j)$, some of which may be shared with other variables W_k, $k \neq j$, thus inducing dependence between the components of the vector $\boldsymbol{W}$. To fix ideas, consider the tree represented in Figure 3.1. In this case, one has, for example, $W_2 = g_2(W_0^*, W_2^*, W_{21}^*)$ and $W_7 = g_7(W_0^*, W_3^*, W_{32}^*, W_{323}^*)$. To define a valid d-norm generator, we need to assume that this system of variables and the corresponding functions g_j are such that $\mathbb{E}(W_j) = 1$ for each $j \in \{1, \ldots, d\}$. However, there is no further restriction on the dependence structure of these latent variables, which yields a very general framework.

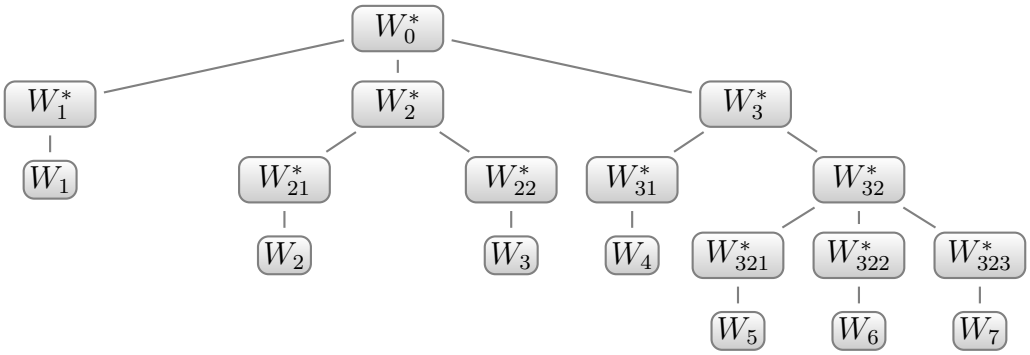

Figure 3.1: Tree representation of a hierarchical d-norm generator with $d = 7$ for the construction of a HEVC.

The inherent hierarchical structure of the d-norm generator defined in this way carries over to the EVC derived from (3.4). Such hierarchical d-norm generators yield HEVCs.

We now describe several example models of HEVCs constructed using this general framework. We first consider the well known nested Gumbel copula and show that it arises as HEVCs in our framework; see McFadden (1978), Stephenson (2003) and Tawn (1990) for early references. Nested Gumbel (or logistic) copulas have been applied in a variety of applications, such as Hofert and Scherer (2011) in the realm of pricing collateralized debt obligations or Vettori et al. (2017) where they are used to group various air pollutants into clusters with homogeneous extremal dependence strength.

Example 3.2.4 (Nested Gumbel copulas with two nesting levels)
For $0 < \alpha_1, \dots, \alpha_S \le \alpha_0 \le 1$, consider independent random variables organized in S groups:

$$\begin{aligned}
\text{Root:} \quad & W_0^* = 1, \\
\text{Level 1:} \quad & W_s^* \overset{\text{ind.}}{\sim} \mathcal{PS}(\alpha_s/\alpha_0), \quad s \in \{1, \dots, S\}, \\
\text{Level 2:} \quad & W_{sj}^* \overset{\text{ind.}}{\sim} \exp(-x^{-1/\alpha_s}),\ x > 0, \quad s \in \{1, \dots, S\},\ j \in \{1, \dots, d_s\}.
\end{aligned}$$

As outlined above, the leaves of the tree correspond to the d-norm generator $\boldsymbol{W} = (\boldsymbol{W}_1, \dots, \boldsymbol{W}_S)$, with $\boldsymbol{W}_s = (W_{s1}, \dots, W_{sd_s})$, $s \in \{1, \dots, S\}$, with $d = \sum_{s=1}^S d_s$, where

$$W_{sj} = g_{sj}(W_0^*, W_s^*, W_{sj}^*) = \frac{W_s^{*\alpha_s} W_{sj}^*}{\Gamma(1-\alpha_0)}, \quad s \in \{1, \dots, S\},\ j \in \{1, \dots, d_s\}. \tag{3.9}$$

It can be verified that, indeed, $W_{sj} \ge 0$ and $\mathbb{E}(W_{sj}) = 1$ for all s and j. Then, the stable tail dependence function corresponding to the d-norm generator $\boldsymbol{W}$ is given by

$$\ell(\boldsymbol{x}) = \ell_{\alpha_0}\{\ell_{\alpha_1}(\boldsymbol{x}_1), \dots, \ell_{\alpha_S}(\boldsymbol{x}_S)\}, \tag{3.10}$$

where $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_S)$, $\boldsymbol{x}_s = (x_{s1}, \ldots, x_{sd_s})$, $s \in \{1, \ldots, S\}$, and $\ell_\alpha(x_1, \ldots, x_d) = (\sum_{j=1}^d x_j^{1/\alpha})^\alpha$ is the stable tail dependence function of a Gumbel copula with parameter α.

Proof. It directly follows from (3.6) that

$$\ell(\boldsymbol{x}) = \mathbb{E}(\max_{1\le s\le S}\{\max_{1\le j\le d_s}\{x_{sj}W_{sj}\}\}), \quad \boldsymbol{x} \in [0,\infty)^d.$$

By (3.9) and with $Y_{\boldsymbol{x}} = \max_{1\le s\le S}\{\max_{1\le j\le d_s}\{x_{sj}W_s^{*\alpha_s}W_{sj}^*\}\}$, one obtains that

$$\begin{aligned}\ell(\boldsymbol{x}) &= \frac{1}{\Gamma(1-\alpha_0)}\mathbb{E}\Big(\max_{1\le s\le S}\{\max_{1\le j\le d_s}\{x_{sj}W_s^{*\alpha_s}W_{sj}^*\}\}\Big) = \frac{1}{\Gamma(1-\alpha_0)}\mathbb{E}(Y_{\boldsymbol{x}}) \\ &= \frac{1}{\Gamma(1-\alpha_0)}\int_0^\infty \mathbb{P}(Y_{\boldsymbol{x}} > y)\,\mathrm{d}y, \quad \boldsymbol{x} \in [0,\infty)^d.\end{aligned}$$

Conditioning on W_s^*, $s \in \{1, \ldots, S\}$, we obtain that

$$\begin{aligned}\mathbb{P}(Y_{\boldsymbol{x}} \le y) &= \mathbb{P}\{W_{sj}^* \le y/(x_{sj}W_s^{*\alpha_s}),\ s \in \{1,\ldots,S\},\ j \in \{1,\ldots,d_s\}\} \\ &= \mathbb{E}\Big[\prod_{s=1}^S\prod_{j=1}^{d_s}\exp\Big\{\Big(-\frac{y}{x_{sj}W_s^{*\alpha_s}}\Big)^{-\frac{1}{\alpha_s}}\Big\}\Big] = \prod_{s=1}^S\mathbb{E}\Big[\exp\Big\{-W_s^*\sum_{j=1}^{d_s}\Big(\frac{y}{x_{sj}}\Big)^{-\frac{1}{\alpha_s}}\Big\}\Big],\end{aligned}$$

where the last equality holding since $W_1^*, \ldots, W_S^*$ are independent. Since $W_s^* \sim \mathcal{PS}(\alpha_s/\alpha_0)$, this leads to

$$\mathbb{P}(Y_{\boldsymbol{x}} \le y) = \prod_{s=1}^S\exp\Big[-\Big\{\sum_{j=1}^{d_s}\Big(\frac{y}{x_{sj}}\Big)^{-\frac{1}{\alpha_s}}\Big\}^{\frac{\alpha_s}{\alpha_0}}\Big] = \exp\Big(-y^{-\frac{1}{\alpha_0}}\Big[\sum_{s=1}^S\Big\{\sum_{j=1}^{d_s}(x_{sj}^{\frac{1}{\alpha_s}})\Big\}^{\frac{\alpha_s}{\alpha_0}}\Big]\Big).$$

With $t = \sum_{s=1}^S\{\sum_{j=1}^{d_s}(x_{sj}^{1/\alpha_s})\}^{\alpha_s/\alpha_0}$, the substitution $z = y^{-1/\alpha_0}t$, and integration by parts, the stable tail dependence function is thus

$$\begin{aligned}\ell(\boldsymbol{x}) &= \frac{1}{\Gamma(1-\alpha_0)}\int_0^\infty\{1-\exp(-y^{-\frac{1}{\alpha_0}}t)\}\,\mathrm{d}y = \frac{t^{\alpha_0}}{\Gamma(1-\alpha_0)}\int_0^\infty\{1-\exp(-z)\}\alpha_0 z^{-\alpha_0-1}\,\mathrm{d}z \\ &= \frac{t^{\alpha_0}}{\Gamma(1-\alpha_0)}\int_0^\infty z^{-\alpha_0}\exp(-z)\,\mathrm{d}z = \frac{t^{\alpha_0}}{\Gamma(1-\alpha_0)}\Gamma(1-\alpha_0) = t^{\alpha_0} = \Big\{\sum_{s=1}^S\Big(\sum_{j=1}^{d_s}x_{sj}^{\frac{1}{\alpha_s}}\Big)^{\frac{\alpha_s}{\alpha_0}}\Big\}^{\alpha_0} \\ &= \ell_{\alpha_0}\{\ell_{\alpha_1}(\boldsymbol{x}_1), \ldots, \ell_{\alpha_S}(\boldsymbol{x}_S)\},\end{aligned}$$

which is the stable tail dependence function of a nested Gumbel copula constructed by nesting on the level of the d-norms. □

The construction underlying Example 3.2.4 may easily be generalized to trees with arbitrary nesting levels using the same line of proof. The construction, extending Stephenson (2003), is outlined in the following example.

Example 3.2.5 (Nested Gumbel copulas with arbitrary nesting levels)
To construct a nested Gumbel copula with arbitrary nesting levels, we mimic the construction with two nesting levels in Example 3.2.4. Let p_j be the path starting from the root of the tree and leading to the jth leaf representing the d-norm generator component W_j. We can write the corresponding node variables along this path as $W_0^*, W_{p_j(1)}^*, W_{p_j(2)}^*, \dots, W_{p_j(L_j)}^*, W_j$, where L_j denotes the number of intermediate variables (or levels) between W_0^* and W_j. Assume that all latent variables $W_{p_j(k)}^*$, $j \in \{1, \dots, d\}$, $k \in \{1, \dots, L_j\}$, are mutually independent within and across paths, and that

$$\begin{aligned}
\text{Root:} \quad & W_0^* = 1, \\
\text{Level 1:} \quad & W_{p_j(1)}^* \sim \mathcal{PS}(\alpha_{p_j(1)}/\alpha_0), \\
\text{Level } k\text{:} \quad & W_{p_j(k)}^* \sim \mathcal{PS}(\alpha_{p_j(k)}/\alpha_{p_j(k-1)}), \quad k \in \{2, \dots, L_j - 1\}, \\
\text{Level } L_j\text{:} \quad & W_{p_j(L_j)}^* \sim \exp(-x^{-1/\alpha_{p_j(L_j-1)}}), \quad x > 0,
\end{aligned}$$

where, for each path p_j, the parameters of the positive α-stable variables on this path are ordered as $0 < \alpha_{p_j(L_j-1)} \leq \cdots \leq \alpha_{p_j(1)} \leq \alpha_0 < 1$. We can then construct the component W_j of the d-norm generator via

$$W_j = g_j(W_0^*, W_{p_j(1)}^*, \dots, W_{p_j(L_j)}^*) = \frac{W_{p_j(1)}^{*\alpha_{p_j(1)}} \cdots W_{p_j(L_j-1)}^{*\alpha_{p_j(L_j-1)}} W_{p_j(L_j)}^*}{\Gamma(1-\alpha_0)}, \quad j \in \{1, \dots, d\}.$$

By recursively conditioning on the variables along each path, one can show that the resulting d-norm generator corresponds to the nested Gumbel copula based on the same tree structure and that its stable tail dependence function can be obtained by applying (3.10) recursively at each nesting level of the tree.

The construction principle for hierarchical d-norm generators also allows us to construct the following two HEVCs.

Example 3.2.6 (Hierarchical Hüsler–Reiss copula)
For simplicity, consider the two-level case

$$\begin{aligned}
\text{Root:} \quad & W_0^* = 1, \\
\text{Level 1:} \quad & (W_1^*, \dots, W_S^*) \sim \mathrm{N}_S(\mathbf{0}, \Sigma_0), \\
\text{Level 2:} \quad & (W_{s1}^*, \dots, W_{sd_s}^*) \sim \mathrm{N}_{d_s}(\mathbf{0}, \Sigma_s), \quad s \in \{1, \dots, S\},
\end{aligned}$$

where the vectors $(W_1^*, \ldots, W_S^*)$ and $(W_{s1}^*, \ldots, W_{sd_s}^*)$, $s \in \{1, \ldots, S\}$, are independent. Furthermore, assume that the covariance matrix Σ_0 may be expressed in terms of the variances $\sigma_1^{*2}, \ldots, \sigma_S^{*2}$ and the correlation matrix P_0 via $\Sigma_{0,ik} = \mathrm{Cov}(W_i^*, W_k^*) = \sigma_i^* \sigma_k^* P_{0,ik}$. Similarly, denote by $\sigma_{s1}^{*2}, \ldots, \sigma_{sd_s}^{*2}$ and P_s the respective quantities for the vector $(W_{s1}^*, \ldots, W_{sd_s}^*)$, $s \in \{1, \ldots, S\}$. Writing the d-norm generator as $\boldsymbol{W} = (\boldsymbol{W}_1, \ldots, \boldsymbol{W}_S)$, with $\boldsymbol{W}_s = (W_{s1}, \ldots, W_{sd_s})$, $s \in \{1, \ldots, S\}$, as in Example 3.2.4, we define the components by

$$W_{sj} = \exp\{(W_s^* + W_{sj}^*) - (\sigma_s^{*2} + \sigma_{sj}^{*2})/2\}, \quad s \in \{1, \ldots, S\},\ j \in \{1, \ldots, d_s\}. \tag{3.11}$$

It is immediate from Part 7 of Example 3.2.3 and by writing $\varepsilon_{sj} = W_s^* + W_{sj}^*$ that the resulting extreme-value distribution has the Hüsler–Reiss copula as underlying dependence structure. It is characterized by an overall dispersion matrix Σ whose entries are given by

$$\mathrm{Cov}(\varepsilon_{s_1 j_1}, \varepsilon_{s_2 j_2}) = \begin{cases} \Sigma_{0,s_1 s_1} + \Sigma_{s_1,j_1 j_2} = \sigma_{s_1}^{*2} + \sigma_{s_1 j_1}^* \sigma_{s_1 j_2}^* P_{s_1,j_1 j_2}, & s_1 = s_2 \text{ (same groups)}, \\ \Sigma_{0,s_1 s_2} = \sigma_{s_1}^* \sigma_{s_2}^* P_{0,s_1 s_2}, & s_1 \neq s_2 \text{ (different groups)}. \end{cases}$$

Hence, in this case, the underlying hierarchical d-norm generator results in a hierarchical structure of the covariance matrix Σ and the corresponding stable tail dependence function is of the same form. It is straightforward to verify that this hierarchical structure allows to model stronger dependence within groups than between groups. This simple two-level example can easily be generalized to trees with arbitrary nesting levels, and it could be interesting for spatial modeling, where different homogeneous regions exhibit different extreme-value behaviors.

Example 3.2.7 (Hierarchical extremal t and Schlather copula)
Example 3.2.6 can be adapted to a hierarchical extremal t model by replacing (3.11) by

$$W_{sj} = \max\{0, (W_s^* + W_{sj}^*)/(\sigma_s^{*2} + \sigma_{sj}^{*2})^{1/2}\}^\nu / c_\nu, \quad s \in \{1, \ldots, S\},\ j \in \{1, \ldots, d_s\},$$

where $\nu > 0$ is the degree of freedom and c_ν is the same constant appearing in Part 6 of Example 3.2.3. For $\nu = 1$, we obtain a hierarchical Schlather model.

3.3 Hierarchical Archimax copulas

3.3.1 Archimax copulas

Let Ψ be the set of all *(Archimedean) generators*, that is, all $\psi : [0, \infty) \to [0, 1]$ which are continuous, decreasing, strictly decreasing on $[0, \inf\{t : \psi(t) = 0\}]$ and satisfying $\psi(0) = 1$

and $\psi(\infty) = \lim_{t\to\infty} \psi(t) = 0$. According to Capéraà et al. (2000) and Charpentier et al. (2014a), a copula is an *Archimax copula (AXC)* if it admits the form

$$C(\boldsymbol{u}) = \psi[\ell\{\psi^{-1}(u_1), \dots, \psi^{-1}(u_d)\}], \quad \boldsymbol{u} \in [0,1]^d, \tag{3.12}$$

for an Archimedean generator $\psi \in \Psi$ and a stable tail dependence function ℓ; note that the form (3.12) in d dimensions was originally conjectured in Mesiar and Jágr (2013). In what follows, we focus on the case where ψ is completely monotone. Since $\psi(0) = 1$, Bernstein's Theorem, see Bernstein (1928) or (Feller, 1971, p. 439), implies that ψ is the Laplace–Stieltjes transform of a distribution function F on the positive real line, that is, $\psi(t) = \mathcal{LS}[F](t) = \int_0^\infty \exp(-tx)\,\mathrm{d}F(x)$, $t \in [0,\infty)$, in this case. A stochastic representation for $\boldsymbol{U} \sim C$ is given by

$$\boldsymbol{U} = \left(\psi\left(\frac{E_1}{V}\right), \dots, \psi\left(\frac{E_d}{V}\right)\right) = \left(\psi\left(\frac{-\ln Y_1}{V}\right), \dots, \psi\left(\frac{-\ln Y_d}{V}\right)\right) \sim C, \tag{3.13}$$

where $(E_1, \dots, E_d) = (-\ln Y_1, \dots, -\ln Y_d)$ (which has Exp(1) margins) for $\boldsymbol{Y} = (Y_1, \dots, Y_d) \sim D$ for a d-dimensional EVC D with stable tail dependence function ℓ and $V \sim F = \mathcal{LS}^{-1}[\psi]$ is the frailty in the construction (which is independent of $\boldsymbol{Y}$). Note that, as a special case, if D is the independence copula, in other words $\ell(\boldsymbol{x}) = \sum_{j=1}^d x_j$, then C in (3.12) is Archimedean. Moreover, if $\psi(t) = \exp(-t)$, $t \geq 0$, then C in (3.12) is an EVC with stable tail dependence function ℓ (compare with (3.1)) and $\boldsymbol{U} = \boldsymbol{Y}$, so $C = D$. Although not relevant for the remainder of this chapter, but important for statistical applications, let us mention that, if it exists, the density of an AXC allows for a rather explicit form (derived in Proposition B.1.1) which makes computing the logarithmic density numerically feasible (see Proposition B.1.5).

3.3.2 Two ways of inducing hierarchies

There are two immediate ways to introduce a hierarchical structure on Archimax copulas following from (3.13), thus leading to *hierarchical Archimax copulas (HAXCs)*: At the level of the EVC D through its stable tail dependence function (via d-norms) and at the level of the frailty V by using a sequence of dependent frailties instead of a single V. Since the former was addressed in Section 3.2, we now focus on the latter.

Let D be a d-dimensional EVC with stable tail dependence function ℓ as before. The stochastic representation (3.13) can be generalized by replacing the single frailty V by a sequence of dependent frailties. Their hierarchical structure and dependence is best described in terms of a concrete example. To this end, consider Figure 3.2. The hierarchical

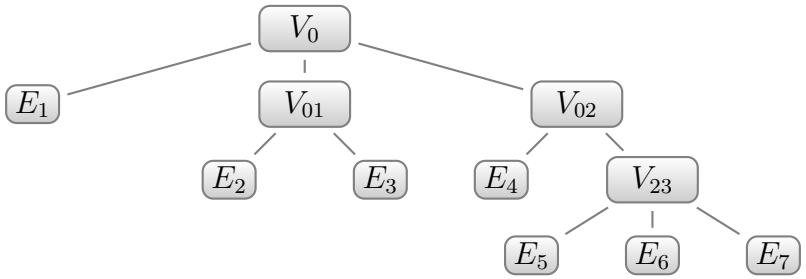

Figure 3.2: Tree representation of hierarchical frailties for the construction of a HAXC.

frailties are shown as nodes and the corresponding (dependent) Exp(1) random variables as leaves. The frailty at each level is drawn from a distribution on the positive real line which depends on the frailty from one level before: First $V_0 \sim F_0$ is drawn; then, independently of each other, $V_{01} \sim F_{01}(\cdot; V_0)$ and $V_{02} \sim F_{02}(\cdot; V_0)$ are drawn (note that V_0 thus acts as a parameter on the distributions F_{01} of V_{01} and F_{02} of V_{02}); finally, $V_{23} \sim F_{23}(\cdot; V_{02})$ is drawn. This procedure can easily be generalized (level by level) to more hierarchical levels if so desired. Similar to the Archimax case, if $(E_1, \dots, E_7)$ has EVC D and Exp(1) margins, one considers

$$\left(\frac{E_1}{V_0}, \frac{E_2}{V_{01}}, \frac{E_3}{V_{01}}, \frac{E_4}{V_{02}}, \frac{E_5}{V_{23}}, \frac{E_6}{V_{23}}, \frac{E_7}{V_{23}}\right) \tag{3.14}$$

and the survival copula of this random vector is then the HAXC C. For the latter step one needs the marginal survival functions of this random vector which are typically not known explicitly. However, they are known under the so-called *sufficient nesting condition* which is based on certain Laplace–Stieltjes transforms involved and which is also utilized in the construction of nested Archimedean copulas (NACs); see, for example, Hofert (2011), (Joe, 1997, pp. 87) or McNeil (2008). To introduce these Laplace–Stieltjes transforms, it is convenient to have the construction principle of NACs in mind. The NAC corresponding to Figure 3.2 is given by $C_0[u_1, C_1\{u_2, u_3\}, C_2\{u_4, C_3(u_5, u_6, u_7)\}]$, where C_k is generated by the completely monotone generator ψ_k, $k \in \{0, 1, 2, 3\}$. For this case, the sufficient nesting condition requires the appearing *nodes* $\psi_0^{-1} \circ \psi_1$, $\psi_0^{-1} \circ \psi_2$ and $\psi_2^{-1} \circ \psi_3$ in NAC to have completely monotone derivatives; see Hofert (2010) for examples and general results when this holds. This implies that the functions $\psi_{kl}(t; v) = \exp[-v\psi_k^{-1}\{\psi_l(t)\}]$, $t \in [0, \infty)$, $v \in (0, \infty)$, for $(k, l) = (0, 1)$, $(k, l) = (0, 2)$ and $(k, l) = (2, 3)$ are completely monotone generators for every v; see (Feller, 1971, p. 441). As such, by Bernstein's Theorem, they correspond to distribution functions on the positive real line. The important part now is that if the frailties V_0, V_{01}, V_{02} and V_{23} are chosen level-by-level such that

1. $V_0 \sim F_0 = \mathcal{LS}^{-1}[\psi_0]$;
2. $V_{01} \,|\, V_0 \sim F_{01} = \mathcal{LS}^{-1}[\psi_{01}(\cdot\,; V_0)]$ and $V_{02} \,|\, V_0 \sim F_{02} = \mathcal{LS}^{-1}[\psi_{02}(\cdot\,; V_0)]$; and
3. $V_{23} \,|\, V_{02} \sim F_{23} = \mathcal{LS}^{-1}[\psi_{23}(\cdot\,; V_{02})]$.

Then, by following along the lines as described in Hofert (2012), one can show that the corresponding HAXC has the stochastic representation

$$\boldsymbol{U} = \left(\psi_0\left(\frac{E_1}{V_0}\right), \psi_1\left(\frac{E_2}{V_{01}}\right), \psi_1\left(\frac{E_3}{V_{01}}\right), \psi_2\left(\frac{E_4}{V_{02}}\right), \psi_3\left(\frac{E_5}{V_{23}}\right), \psi_3\left(\frac{E_6}{V_{23}}\right), \psi_3\left(\frac{E_7}{V_{23}}\right)\right). \tag{3.15}$$

By comparison with (3.14), we see that if the distribution functions F_0, F_{01}, F_{02}, F_{23} of $V_0 \sim F_0$, $V_{01} \sim F_{01}(\cdot; V_0)$, $V_{02} \sim F_{02}(\cdot; V_0)$, $V_{23} \sim F_{23}(\cdot; V_{02})$ are chosen such that the Laplace–Stieltjes transforms ψ_0, ψ_1, ψ_2, ψ_3 (associated to $V_0, V_{01}, V_{02}, V_{23}$ via the structure of a NAC) satisfy the sufficient nesting condition, then the marginal survival functions of (3.14) are not only known, but they are equal to $\psi_0, \psi_1, \psi_2, \psi_3$ such that the resulting HAXC has a stochastic representation (see (3.15)) similar to that of a HAXC with single frailty (see (3.13)), just with different frailties.

Remark 3.3.1

1. Clearly, the stochastic representation of a HAXC based on hierarchical frailties as in (3.15) immediately allows for a sampling algorithm. The hierarchical frailties involved can easily be sampled in many cases, see Hofert (2010) or the R package `copula` of Hofert et al. (2005) for details.
2. Note that the stochastic representation of a HAXC constructed with hierarchical frailties equals that of a NAC, except for the fact that for the latter, the EVC D of $(E_1, \dots, E_7)$ is the independence copula.
3. The two types of constructing HAXCs presented here can also be mixed, one can use a HEVC and hierarchical frailties. Interestingly, the two types of hierarchies introduced this way do not have to coincide; see the following section for such an example.

All the figures shown in the following examples can be reproduced with the vignette `HAXC` of the R package `copula` (version $\geq 0.999.19$).

Example 3.3.2 (ACs vs AXCs vs NACs vs (different) HAXCs)
Figure 3.3 shows scatter-plot matrices of five-dimensional copula samples of size 1000 from the following models for $\boldsymbol{U} = (U_1, \dots, U_5) \sim C$.

1. Top left: (Archimedean) Clayton copula with stochastic representation
$$\boldsymbol{U} = \left(\psi\left(\frac{E_1}{V}\right), \dots, \psi\left(\frac{E_5}{V}\right)\right), \tag{3.16}$$
where $V \sim \Gamma(1/\theta, 1)$ for $\theta = 4/3$ (the frailty is gamma distributed) and $E_1, \dots, E_5 \overset{\text{ind.}}{\sim} \text{Exp}(1)$; see also (3.2). The copula parameter is chosen such that Kendall's tau equals 0.4.

2. Top right: AXC based on Clayton's family with gamma frailties recycled from the top left plot and stochastic representation as in (3.16) where $(E_1, \dots, E_5) = (-\ln Y_1, \dots, -\ln Y_5)$ for $(Y_1, \dots, Y_5)$ having a Gumbel EVC (with parameter such that Kendall's tau equals 0.5); note that the margins of $(E_1, \dots, E_5)$ are again Exp(1) (but its components are dependent in this case).

3. Middle left: NAC based on Clayton's family with hierarchical frailties such that two sectors of sizes 2 and 3 result, respectively, with parameters $(\theta_0, \theta_1, \theta_2)$ chosen such that Kendall's tau equals 0.2 between the two sectors, 0.4 within the first sector and 0.6 within the second sector. A stochastic representation for this copula is given by
$$\boldsymbol{U} = \left(\psi_1\left(\frac{E_1}{V_{01}}\right), \psi_1\left(\frac{E_2}{V_{01}}\right), \psi_2\left(\frac{E_3}{V_{02}}\right), \psi_2\left(\frac{E_4}{V_{02}}\right), \psi_2\left(\frac{E_5}{V_{02}}\right)\right), \tag{3.17}$$
where $V_0 \sim \Gamma(2)$ and
$$\begin{aligned} V_{01} \,|\, V_0 \sim F_{01} &= \mathcal{LS}^{-1}\left[\exp[-V_0\{(1+t)^{\theta_0/\theta_1} - 1\}]\right], \\ V_{02} \,|\, V_0 \sim F_{02} &= \mathcal{LS}^{-1}\left[\exp[-V_0\{(1+t)^{\theta_0/\theta_2} - 1\}]\right] \end{aligned}$$
are independent (see (Hofert, 2011, Theorem 3.6) for more details) and $E_1, \dots, E_5 \overset{\text{ind.}}{\sim} \text{Exp}(1)$.

4. Middle right: HAXC based on Clayton's family with hierarchical frailties recycled from the middle left plot and stochastic representation as in (3.17) where $(E_1, \dots, E_5) = (-\ln Y_1, \dots, -\ln Y_5)$ for $(Y_1, \dots, Y_5)$ having a Gumbel EVC (realizations recycled from the top right plot). Note that the hierarchical structure is only induced by the frailties in this case.

5. Bottom left: HAXC based on Clayton's family with hierarchical frailties recycled from the middle left plot and stochastic representation
$$\boldsymbol{U} = \left(\psi_1\left(\frac{E_{11}}{V_{01}}\right), \psi_1\left(\frac{E_{12}}{V_{01}}\right), \psi_2\left(\frac{E_{21}}{V_{02}}\right), \psi_2\left(\frac{E_{22}}{V_{02}}\right), \psi_2\left(\frac{E_{23}}{V_{02}}\right)\right),$$

where $(E_{11}, E_{12}, E_{21}, E_{22}, E_{23}) = (-\ln Y_{11}, -\ln Y_{12}, -\ln Y_{21}, -\ln Y_{22}, -\ln Y_{23})$ for $(Y_{11}, Y_{12}, Y_{21}, Y_{22}, Y_{23})$ having a nested Gumbel EVC (with sector sizes 2 and 3 and parameters such that Kendall's tau equals 0.2 between the two sectors, 0.5 within the first sector and 0.7 within the second sector). Note that the hierarchical structure is induced both at the level of the frailties and at the level of the EVC in this case, and that the hierarchical structure (sectors, sector dimensions) is the same.

6. Bottom right: HAXC as in the bottom left plot (realizations recycled) with stochastic representation

$$\boldsymbol{U} = \left(\psi_1\left(\frac{E_{11}}{V_{01}}\right), \psi_1\left(\frac{E_{12}}{V_{01}}\right), \psi_1\left(\frac{E_{21}}{V_{01}}\right), \psi_2\left(\frac{E_{22}}{V_{02}}\right), \psi_2\left(\frac{E_{23}}{V_{02}}\right)\right).$$

Note that the hierarchical structure for the frailties (sector sizes 3 and 2, respectively) and for the nested Gumbel EVC (sector sizes 2 and 3, respectively) differ in this case.

Example 3.3.3 (EVCs vs HEVCs vs (different) HAXCs)
Similar to Figure 3.3, Figure 3.4 shows scatter-plot matrices of five-dimensional copula samples of size 1000 from the following models for $\boldsymbol{U} = (U_1, \dots, U_5) \sim C$; for simulating from the extremal t EVC, we use the R package `mev` of Belzile et al. (2017).

1. Top left: Extremal t EVC with $\nu = 3.5$ degrees of freedom and homogeneous correlation matrix P with off-diagonal entries 0.7.

2. Top right: Extremal t HEVC with two sectors of sizes 2 and 3, respectively, such that the correlation matrix P has entries 0.2 for pairs belonging to different sectors, 0.5 for pairs belonging to the first sector and 0.7 for pairs belonging to the second sector.

3. Middle left: HAXC with single Clayton frailty (as in Example 3.3.2 Part 1) and extremal t HEVC recycled from the top right plot.

4. Middle right: HAXC with hierarchical Clayton frailties (as in Example 3.3.2 Part 3) and extremal t EVC recycled from the top left plot.

5. Bottom left: HAXC with hierarchical Clayton frailties (as in Example 3.3.2 Part 3) and extremal t HEVC recycled from the top right plot. Note that there are two types of hierarchies involved, at the level of the (hierarchical) frailties and at the level of the (hierarchical) extremal t EVC. Furthermore, the two hierarchical structures match.

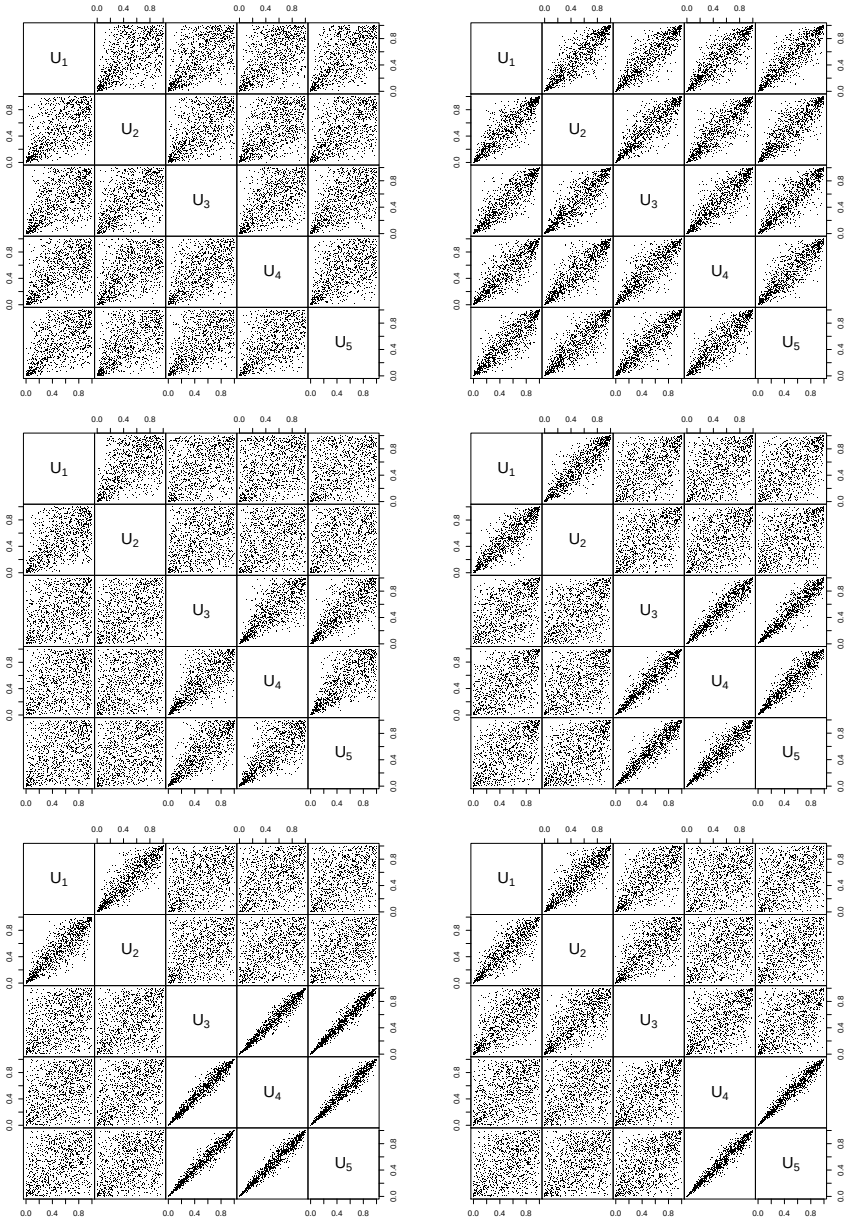

Figure 3.3: Scatter-plot matrices of five-dimensional copula samples of size 1000 of a Clayton copula (top left), an AXC with Clayton frailties and Gumbel EVC (top right), a nested Clayton copula (middle left), a HAXC with hierarchical Clayton frailties and Gumbel EVC (middle right), a HAXC with hierarchical Clayton frailties and nested Gumbel EVC of the same hierarchical structure (bottom left) and a HAXC with hierarchical Clayton frailties and nested Gumbel EVC of different hierarchical structure (bottom right).

6. Bottom right: HAXC as in the bottom left plot, but the hierarchical structures of the frailties (sector sizes 3 and 2, respectively) and of the HEVC (sector sizes 2 and 3, respectively) differ in this case.

Note that we can sample from a hierarchical Schlather model (special case of extremal t for $\nu = 1$), a hierarchical Brown–Resnick model, and their corresponding HAXCs in a similar fashion.

3.4 Conclusion

We extended the class of AXCs to HAXCs. Hierarchies can take place in two forms, either separately or simultaneously. First, the EVC involved in the construction of AXCs can have a hierarchical structure. To this end we presented a new approach for constructing hierarchical stable tail dependence functions based on a connection between stable tail dependence functions and d-norms. Second, a hierarchical structure can be imposed at the level of frailties similarly as NACs arise from ACs. Even more flexible constructions can be obtained by choosing a different hierarchical structure for the HEVC and the hierarchical frailties in the construction. Since all presented constructions are based on stochastic representations, sampling is immediate; see also the presented examples and vignette.

As a contribution to the literature on AXCs, we also derived a general formula for the density of AXCs and the computation of the corresponding logarithmic density. Furthermore, we briefly addressed the question when nested AXCs (NAXCs) can be constructed (either through nested stable tail dependence functions alone or, additionally, through hierarchical frailties). This is, in principle, possible, but as discussed in Appendix B.2, there is currently only one family of examples known when all the assumptions involved are fulfilled. Further research is thus required to find out whether this is the only possible case for which NAXCs result.

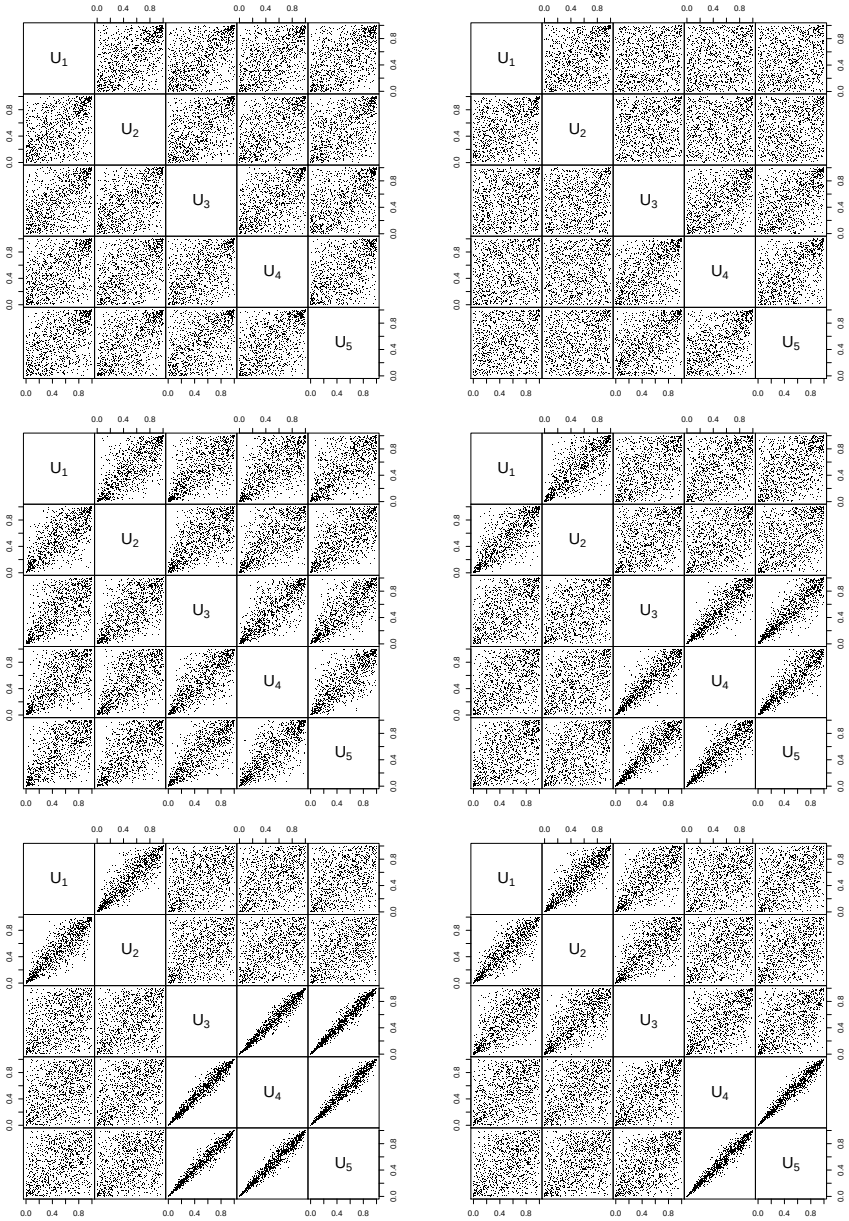

Figure 3.4: Scatter-plot matrices of five-dimensional copula samples of size 1000 of an extremal t EVC (top left), a hierarchical extremal t copula (a HEVC; top right), a HAXC with single Clayton frailty and extremal t HEVC (middle left), a HAXC with hierarchical Clayton frailties and extremal t EVC (middle right), a HAXC with hierarchical Clayton frailties and extremal t HEVC of the same hierarchical structure (bottom left) and a HAXC with hierarchical Clayton frailties and extremal t HEVC of different hierarchical structure (bottom right).

Chapter 4

Quasi-random sampling for multivariate distributions via generative neural networks

4.1 Introduction

Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a d-dimensional random vector with distribution function $F_{\boldsymbol{X}}$ and continuous margins $F_{X_1}, \ldots, F_{X_d}$. It is not a trivial task in general to generate quasi-random samples $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ from $F_{\boldsymbol{X}}$, i.e., samples that mimic realizations from $F_{\boldsymbol{X}}$ but preserve low-discrepancy in the sense of being locally more homogeneous with fewer "gaps" or "clusters" (Cambou et al., 2017, Section 4.2).

By Sklar's Theorem, we always have the decomposition

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = C(F_{X_1}(x_1), \ldots, F_{X_d}(x_d)), \quad \boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d, \tag{4.1}$$

where $C : [0,1]^d \to [0,1]$ is the unique underlying copula Nelsen (2006); Joe (2014). Since, in distribution, $\boldsymbol{X} = F_{\boldsymbol{X}}^{-1}(\boldsymbol{U})$ for $\boldsymbol{U} \sim C$ and $F_{\boldsymbol{X}}^{-1}(\boldsymbol{u}) = (F_{X_1}^{-1}(u_1), \ldots, F_{X_d}^{-1}(u_d))$, we shall mostly focus on the problem of generating quasi-random samples $\boldsymbol{U}_1, ..., \boldsymbol{U}_n$ from C rather than $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ from $F_{\boldsymbol{X}}$, as the latter are easily obtained from the former.

4.1.1 Existing difficulties

For the independence copula, $C(\boldsymbol{u}) = u_1 \cdot \ldots \cdot u_d$, quasi-random samples can be obtained simply by using randomized quasi-Monte Carlo (RQMC) point sets such as randomized

Sobol' or generalized Halton sequences; see, for example, Lemieux (2009, Chapter 5).

Recently, Cambou et al. (2017) demonstrated that for a limited number of copulas C (normal, t or Clayton copulas), one can obtain quasi-random samples by transforming RQMC point sets with the inverse Rosenblatt transform of C (Rosenblatt, 1952); for pseudo-random numbers this sampling method is known as the *conditional distribution method (CDM)* — see, e.g., Embrechts et al. (2003) or (Hofert, 2010, p. 45). Cambou et al. (2017) also showed that transformations to quasi-random copula samples may exist for copulas with a sufficiently simple stochastic representation. For most copulas, the latter is not the case and the CDM is numerically intractable. In other words, there exists no universal and numerically tractable transformation from RQMC point sets to quasi-random samples from copulas. For the majority of copula models, including grouped normal variance mixture copulas, Archimax copulas, nested Archimedean copulas or extreme-value copulas, we simply do not know how to generate quasi-random samples from them.

4.1.2 Our contribution

The main contribution of this chapter is to introduce a new approach for quasi-random sampling from $F_{\boldsymbol{X}}$ with *any* underlying copula C, using generative neural networks. Even when we do *not* know the distribution $F_{\boldsymbol{X}}$, our approach can still provide quasi-random samples from the corresponding empirical distribution $\hat{F}_{\boldsymbol{X}}$ as long as we have a dataset from $F_{\boldsymbol{X}}$. This is especially useful when the dependence structure in the data cannot be adequately captured by a readily available parametric copula; see Section 4.5 where we present a real-data example to show how useful our approach can be in this case where no adequate copula model is known in the first place.

Specifically, let $f_{\boldsymbol{\theta}}$ denote a neural network (NN) parameterized by $\boldsymbol{\theta}$. We train $f_{\boldsymbol{\theta}}$ so that, given a p-dimensional input $\boldsymbol{Z} \sim F_{\boldsymbol{Z}}$ with independent components $Z_1, \dots, Z_p$ from known distributions $F_{Z_1}, \dots, F_{Z_p}$, the trained NN can generate d-dimensional output from the desired distribution, $f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}) \sim F_{\boldsymbol{X}}$, where $\hat{\boldsymbol{\theta}}$ denotes the parameter vector of the trained NN. We can thus turn a uniform RQMC point set, $\{\tilde{\boldsymbol{v}}_1, \dots, \tilde{\boldsymbol{v}}_n\}$, into a quasi-random sample from $F_{\boldsymbol{X}}$ by letting

$$\boldsymbol{Y}_i = f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}(\tilde{\boldsymbol{v}}_i), \quad i = 1, \dots, n, \tag{4.2}$$

where $F_{\boldsymbol{Z}}^{-1}(\boldsymbol{u}) = (F_{Z_1}^{-1}(u_1), \dots, F_{Z_p}^{-1}(u_p))$.

4.1.3 Assessment

The theoretical properties of quasi-randomness (or low-discrepancy) under dependence are hard to assess; see Cambou et al. (2017) and Appendix C.1. In low-dimensional cases (see Section 4.3.2), we use data visualization tools to assess the quality of the generated quasi-random samples, such as contour plots (or level curves) showing that the empirical copula of our GMMN quasi-random samples is closer to the true target copula than that of GMMN pseudo-random samples. In higher-dimensional cases (see Section 4.3.3), we use a Cramér-von-Mises goodness-of-fit statistic to make the same point.

Since the main application of quasi-random sampling is to obtain low-variance Monte-Carlo estimates of

$$\mu = \mathbb{E}(\Psi(\boldsymbol{X})) = \mathbb{E}\Big(\Psi(F_{\boldsymbol{X}}^{-1}(\boldsymbol{U}))\Big) \quad \text{for} \quad \boldsymbol{X} \sim F_{\boldsymbol{X}},\ \boldsymbol{U} \sim C \tag{4.3}$$

for an integrable $\Psi : \mathbb{R}^d \to \mathbb{R}$, we also assess our method in such a specific context. The *Monte Carlo estimator* approximates this expectation by

$$\hat{\mu}_n^{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} \Psi(F_{\boldsymbol{X}}^{-1}(\boldsymbol{U}_i)), \tag{4.4}$$

where $\boldsymbol{U}_1, \dots, \boldsymbol{U}_n \overset{\text{ind.}}{\sim} C$. Using NN-generated quasi-random samples $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n$ from (4.2), we can approximate $\mathbb{E}\Big(\Psi(F_{\boldsymbol{X}}^{-1}(\boldsymbol{U}))\Big)$ by

$$\hat{\mu}_n^{\mathrm{NN}} = \frac{1}{n}\sum_{i=1}^{n} \Psi(\boldsymbol{Y}_i) = \frac{1}{n}\sum_{i=1}^{n} \Psi(f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}(\tilde{\boldsymbol{v}}_i)). \tag{4.5}$$

Theoretically (Section 4.2.3 and Appendix C.1), we establish various guarantees that the estimation error of (4.3) by (4.5) will be small as long as both $f_{\hat{\boldsymbol{\theta}}}$ and Ψ are sufficiently smooth; we also establish the corresponding convergence rates. Empirically (Section 4.4), we verify that (4.5) indeed has lower variance and converges faster than (4.4).

Although being the main focus in this chapter, let us stress that estimating expectations such as (4.3) is not the only application of quasi-random sampling. For example, quasi-random sampling is also useful for estimating quantiles of the distribution of a sum of dependent random variables.

All results presented in this chapter (and more) are reproducible with the demos `GMMN_QMC_paper`, `GMMN_QMC_data` and `GMMN_QMC_timings` as part of the new developed R package `gnn`.

4.2 Quasi-random GMMN samples

4.2.1 Generative moment matching networks

In this chapter, we work with the *multi-layer perceptron (MLP)*, which is regarded as the quintessential *neural network (NN)*. Let L be the number of (hidden) layers in the NN and, for each $l = 0, \ldots, L+1$, let d_l be the dimension of layer l, that is, the number of neurons in layer l. In this notation, layer $l = 0$ refers to the *input layer* which consists of the *input* $\boldsymbol{z} \in \mathbb{R}^p$ for $d_0 = p$, and layer $l = L+1$ refers to the *output layer* which consists of the *output* $\boldsymbol{y} \in \mathbb{R}^d$ for $d_{L+1} = d$. Layers $l = 1, \ldots, L+1$ can be described in terms of the output $\boldsymbol{a}_{l-1} \in \mathbb{R}^{d_{l-1}}$ of layer $l-1$ via

$$\begin{aligned}
\boldsymbol{a}_0 &= \boldsymbol{z} \in \mathbb{R}^{d_0}, \\
\boldsymbol{a}_l &= f_l(\boldsymbol{a}_{l-1}) = \phi_l(W_l \boldsymbol{a}_{l-1} + \boldsymbol{b}_l) \in \mathbb{R}^{d_l}, \quad l = 1, \ldots, L+1, \\
\boldsymbol{y} &= \boldsymbol{a}_{L+1} \in \mathbb{R}^{d_{L+1}},
\end{aligned}$$

with *weight matrices* $W_l \in \mathbb{R}^{d_l \times d_{l-1}}$, *bias vectors* $\boldsymbol{b}_l \in \mathbb{R}^{d_l}$ and *activation functions* ϕ_l; note that for vector inputs the activation function ϕ_l is understood to be applied componentwise.

Figure 4.1 visualizes this construction and the notation we use.

The NN $f_{\boldsymbol{\theta}} : \mathbb{R}^p \to \mathbb{R}^d$ can then be written as the composition

$$f_{\boldsymbol{\theta}} = f_{L+1} \circ f_L \circ \cdots \circ f_2 \circ f_1,$$

with its (flattened) parameter vector given by $\boldsymbol{\theta} = (W_1, \ldots, W_{L+1}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{L+1})$. To fit $\boldsymbol{\theta}$, we use the backpropagation algorithm (a stochastic gradient descent) based on a *loss function* $\mathcal{L}$. Conceptually, $\mathcal{L}$ computes a distance between the *target output* $\boldsymbol{x} \in \mathbb{R}^d$ and the *actual output* $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{z}) \in \mathbb{R}^d$ predicted by the NN; what is actually computed is a sample version of $\mathcal{L}$ based on a subsample (the so-called *mini-batch*), see Section 4.2.2.

The expressive power of NNs is primarily characterized by the *universal approximation theorem*; see Goodfellow et al. (2016, Chapter 6). In particular, given suitable activation functions, a single hidden layer NN with a finite number of neurons can approximate any continuous function on a compact subset of the multidimensional Euclidean space; see Nielsen (2015, Chapter 4) for a visual account of the validity of the universal approximation theorem. Cybenko (1989) first proposed such universal approximation results for the sigmoid activation function $\phi_l(x) = 1/(1 + \mathrm{e}^{-x})$ and Hornik (1991, Theorem 1) then generalized the results to include arbitrary bounded and non-constant activation functions. In recent years, the *rectified linear unit (ReLU)* $\phi_l(x) = \max\{0, x\}$ has become the most popular activation

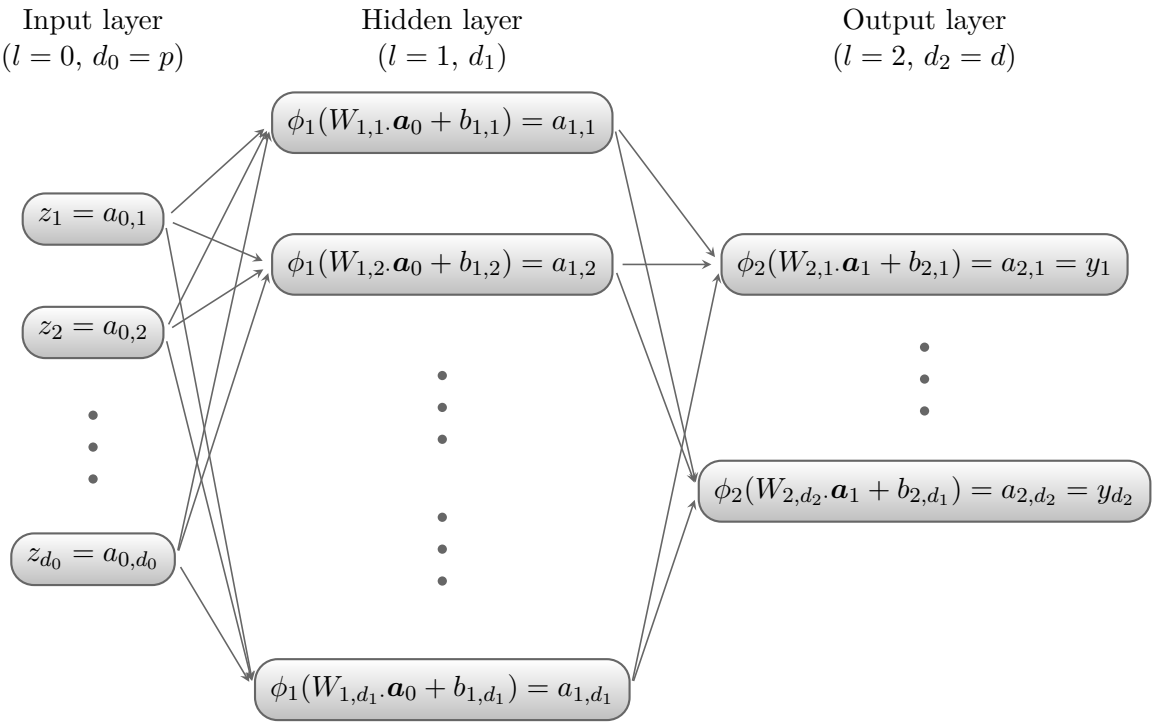

Figure 4.1: Structure of a NN with input $\boldsymbol{z} = (z_1, \ldots, z_{d_0})$, $L = 1$ hidden layer with output $\boldsymbol{a}_1 = f_1(\boldsymbol{a}_0) = \phi_1(W_1\boldsymbol{a}_0 + \boldsymbol{b}_1)$ and output layer with output $\boldsymbol{y} = \boldsymbol{a}_2 = f_2(\boldsymbol{a}_1) = \phi_2(W_2\boldsymbol{a}_1 + \boldsymbol{b}_2)$; note that in the figure, $W_{l,j\cdot}$ denotes the jth row of W_l and $b_{l,j}$ the jth row of $\boldsymbol{b}_l$.

function for efficiently training NNs. This unbounded activation function does not satisfy the assumptions of the universal approximation theorem in Hornik (1991). However, there have since been numerous theoretical investigations into the expressive power of NNs with ReLU activation functions; see, for example, Pascanu et al. (2013), Montufar et al. (2014) or Arora et al. (2016). In particular, for certain conditions on the number of layers and neurons in the NN, Arora et al. (2016) provide a similar universal approximation theorem for NNs with ReLU activation functions.

Li et al. (2015) and Dziugaite et al. (2015) simultaneously introduced a type of generative neural network known as the *generative moment matching network (GMMN)* or the Maximum Mean Discrepancy (MMD) net. GMMNs are NNs $f_{\boldsymbol{\theta}}$ of the above form which utilize a (kernel) maximum mean discrepancy statistic as the loss function (see later). Conceptually, they can be thought of as parametric maps of a given sample $\boldsymbol{Z} = (Z_1, \ldots, Z_p)$ from an *input distribution* $F_{\boldsymbol{Z}}$ to a sample $\boldsymbol{X} = (X_1, \ldots, X_d)$ from the *target distribution* $F_{\boldsymbol{X}}$. As is standard in the literature, we assume independence among the components of $\boldsymbol{Z} = (Z_1, \ldots, Z_p)$. Typical choices for the distribution of the Z_j's are $\mathrm{U}(0,1)$ or $\mathrm{N}(0,1)$. The objective is then to generate samples from the target distribution via the trained GMMN $f_{\hat{\boldsymbol{\theta}}}$. The MMD nets introduced in Dziugaite et al. (2015) are almost identical to GMMNs but with a slight difference in the training procedure; additionally, Dziugaite et al. (2015) provided a theoretical framework for analyzing optimization algorithms with (kernel) MMD loss functions.

4.2.2 Loss function and training of GMMNs

To learn $f_{\boldsymbol{\theta}}$ (or, statistically speaking, to estimate the parameter vector $\boldsymbol{\theta}$) we assume that we have n_{trn} training data points $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n_{\text{trn}}}$ from $\boldsymbol{X}$, either in the form of a pseudo-random sample from $F_{\boldsymbol{X}}$ or as real data. Based on a sample $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{n_{\text{gen}}}$ from the input distribution, the GMMN generates the output sample $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{n_{\text{gen}}}$, where $\boldsymbol{Y}_i = f_{\boldsymbol{\theta}}(\boldsymbol{Z}_i)$, $i = 1, \ldots, n_{\text{gen}}$. Stacking $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n_{\text{trn}}}$ into an $n_{\text{trn}} \times d$ matrix X and likewise $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_{n_{\text{gen}}}$ into an $n_{\text{gen}} \times d$ matrix Y, we are thus interested in whether the two samples X and Y come from the same distribution.

To this end, GMMNs utilize as loss function $\mathcal{L}$ the *maximum mean discrepancy (MMD)* statistic from the kernel two-sample test introduced by Gretton et al. (2007). For a given embedding function $\varphi : \mathbb{R}^d \mapsto \mathbb{R}^{d'}$, the MMD measures the distance between two sample

statistics, $(1/n_{\text{trn}})\sum_{t_1=1}^{n_{\text{trn}}}\varphi(\boldsymbol{X}_{t_1})$ and $(1/n_{\text{gen}})\sum_{t_2=1}^{n_{\text{gen}}}\varphi(\boldsymbol{Y}_{t_2})$, in the embedded space $\mathbb{R}^{d'}$ via

$$\begin{aligned}&\text{MMD}(X,Y)\\ &=\left\|\frac{1}{n_{\text{trn}}}\sum_{t_1=1}^{n_{\text{trn}}}\varphi(\boldsymbol{X}_{t_1})-\frac{1}{n_{\text{gen}}}\sum_{t_2=1}^{n_{\text{gen}}}\varphi(\boldsymbol{Y}_{t_2})\right\|_2\\ &=\sqrt{\frac{1}{n_{\text{trn}}^2}\sum_{t_1=1}^{n_{\text{trn}}}\sum_{t_2=1}^{n_{\text{trn}}}\varphi(\boldsymbol{X}_{t_1})^\top\varphi(\boldsymbol{X}_{t_2})-\frac{2}{n_{\text{trn}}n_{\text{gen}}}\sum_{t_1=1}^{n_{\text{trn}}}\sum_{t_2=1}^{n_{\text{gen}}}\varphi(\boldsymbol{X}_{t_1})^\top\varphi(\boldsymbol{Y}_{t_2})+\frac{1}{n_{\text{gen}}^2}\sum_{t_1=1}^{n_{\text{gen}}}\sum_{t_2=1}^{n_{\text{gen}}}\varphi(\boldsymbol{Y}_{t_1})^\top\varphi(\boldsymbol{Y}_{t_2})}.\end{aligned}$$

If we can choose $\varphi(\cdot)$ to be a kind of "distributional embedding", for example, in the sense that the two statistics — $(1/n_{\text{trn}})\sum_{t_1=1}^{n_{\text{trn}}}\varphi(\boldsymbol{X}_{t_1})$ and $(1/n_{\text{gen}})\sum_{t_2=1}^{n_{\text{gen}}}\varphi(\boldsymbol{Y}_{t_2})$ — contain all empirical moments of X and Y, respectively, then the MMD criterion will have achieved our desired purpose (of measuring whether the two samples have the same distribution). Amazingly, such embedding does exist.

By the so-called "kernel trick", known as early as Mercer (1909) but not widely until support vector machines became popular almost a century later, the inner product $\varphi(\boldsymbol{x}_t)^\top\varphi(\boldsymbol{y}_t)$ can be computed in a reproducing kernel Hilbert space by $K(\boldsymbol{x}_t,\boldsymbol{y}_t)$, where $K(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\mapsto\mathbb{R}$ denotes a kernel similarity function. Hence, for a given kernel function $K(\cdot,\cdot)$, the MMD statistic above is equivalent to

$$\begin{aligned}&\text{MMD}(X,Y)\\ &=\sqrt{\frac{1}{n_{\text{trn}}^2}\sum_{t_1=1}^{n_{\text{trn}}}\sum_{t_2=1}^{n_{\text{trn}}}K(\boldsymbol{X}_{t_1},\boldsymbol{X}_{t_2})-\frac{2}{n_{\text{trn}}n_{\text{gen}}}\sum_{t_1=1}^{n_{\text{trn}}}\sum_{t_2=1}^{n_{\text{gen}}}K(\boldsymbol{X}_{t_1},\boldsymbol{Y}_{t_2})+\frac{1}{n_{\text{gen}}^2}\sum_{t_1=1}^{n_{\text{gen}}}\sum_{t_2=1}^{n_{\text{gen}}}K(\boldsymbol{Y}_{t_1},\boldsymbol{Y}_{t_2})}.\end{aligned}\tag{4.6}$$

If $K(\cdot,\cdot)$ is chosen to be a so-called universal kernel function, such as a Gaussian or Laplace kernel, then the associated implicit embedding $\varphi:\mathbb{R}^d\mapsto\mathbb{R}^\infty$ is indeed a "distributional embedding" in the sense described above, and one can show that the MMD converges in probability to 0 for $n_{\text{trn}},n_{\text{gen}}\to\infty$ if and only if $\boldsymbol{Y}=\boldsymbol{X}$ in distribution; see Gretton et al. (2007, 2012a).

Thus, to train the GMMN $f_{\boldsymbol{\theta}}$, we perform the optimization

$$\min_{\boldsymbol{\theta}}\ \text{MMD}(X,(f_{\boldsymbol{\theta}}(Z))),$$

where the $n_{\text{gen}}\times p$ matrix Z is obtained by stacking $\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_{n_{\text{gen}}}$, and the NN transform $f_{\boldsymbol{\theta}}$ is understood to be applied row-wise. For the sake of convenience, we always simply set $n_{\text{gen}}=n_{\text{trn}}$ while training the GMMN. However, note that after training we can still generate an arbitrary number of samples from $f_{\hat{\boldsymbol{\theta}}}$.

Computing $\mathrm{MMD}(X, Y)$ in (4.6) requires one to evaluate the kernel for all $\binom{n_{\mathrm{trn}}}{2}$ pairs of observations, which is memory-prohibitive for even moderately large n_{trn}. As suggested by Li et al. (2015), we thus adopt a mini-batch optimization procedure. Instead of directly optimizing the MMD for the entire training dataset, we partition the data into *batches* of size n_{bat} and use the batches sequentially to update the parameters $\boldsymbol{\theta}$ of the GMMN with the Adam optimizer of Kingma and Ba (2014). Rather than following the gradient at each iterative step, the Adam optimizer essentially uses a "memory-sticking gradient" — a weighted combination of the current gradient and past gradients from earlier iterations. After all the training data are exhausted, i.e., roughly after $(n_{\mathrm{trn}}/n_{\mathrm{bat}})$-many batches or gradient steps, one *epoch* of the training of the GMMN is completed. The overall training procedure is considered completed after n_{epo} epochs. The training of the GMMN can thus be summarized as follows:

Algorithm 4.2.1 (Training GMMNs)

1. Fix the number n_{epo} of epochs and the batch size $1 \leq n_{\mathrm{bat}} \leq n_{\mathrm{trn}}$ per epoch, where n_{bat} is assumed to divide n_{trn}. Initialize the epoch counter $k = 0$ and the GMMN's parameter vector $\boldsymbol{\theta}$; we follow Glorot and Bengio (2010) and initialize the components of $\boldsymbol{\theta}$ as $W_l \sim \mathrm{U}(-\sqrt{6/(d_l + d_{l-1})}, \sqrt{6/(d_l + d_{l-1})})^{d_l \times d_{l-1}}$ and $\boldsymbol{b}_l = \boldsymbol{0}$ for $l = 1, \ldots, L+1$.
2. For epoch $k = 1, \ldots, n_{\mathrm{epo}}$, do:
   (a) Randomly partition the input distribution sample $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{n_{\mathrm{trn}}}$ and training sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n_{\mathrm{trn}}}$ into corresponding $n_{\mathrm{trn}}/n_{\mathrm{bat}}$ non-overlapping batches $\boldsymbol{Z}_1^{(b)}, \ldots, \boldsymbol{Z}_{n_{\mathrm{bat}}}^{(b)}$ and $\boldsymbol{X}_1^{(b)}, \ldots, \boldsymbol{X}_{n_{\mathrm{bat}}}^{(b)}$, $b = 1, \ldots, n_{\mathrm{trn}}/n_{\mathrm{bat}}$, of size n_{bat} each.
   (b) For batch $b = 1, \ldots, n_{\mathrm{trn}}/n_{\mathrm{bat}}$, do:
      i. Compute the GMMN output $\boldsymbol{Y}_i^{(b)} = f_{\boldsymbol{\theta}}(\boldsymbol{Z}_i^{(b)})$, $i = 1, \ldots, n_{\mathrm{bat}}$.
      ii. Compute the gradient $\frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{MMD}(X^{(b)}, Y^{(b)})$ from the samples $X^{(b)}$ (stacking $\boldsymbol{X}_1^{(b)}, \ldots, \boldsymbol{X}_{n_{\mathrm{bat}}}^{(b)}$) and $Y^{(b)}$ (stacking $\boldsymbol{Y}_1^{(b)}, \ldots, \boldsymbol{Y}_{n_{\mathrm{bat}}}^{(b)}$) via automatic differentiation.
      iii. Take a gradient step to update $\boldsymbol{\theta}$ with the Adam optimizer popularized by Kingma and Ba (2014, Algorithm 1).
3. Return $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$; the fitted GMMN is then $f_{\hat{\boldsymbol{\theta}}}$.

4.2.3 Pseudo- and quasi-random sampling by GMMNs

The following algorithm describes how to obtain a pseudo-random sample of $\boldsymbol{Y}$ via the trained GMMN $f_{\hat{\boldsymbol{\theta}}}$ from a pseudo-random sample $\boldsymbol{Z} \sim F_{\boldsymbol{Z}}$.

Algorithm 4.2.2 (Pseudo-random sampling by GMMN)

1. Fix the number n_{gen} of samples to generate from $\boldsymbol{Y}$.
2. Draw $\boldsymbol{Z}_i \overset{\text{ind.}}{\sim} F_{\boldsymbol{Z}}$, $i = 1, \dots, n_{\text{gen}}$, for example, via $\boldsymbol{Z}_i = F_{\boldsymbol{Z}}^{-1}(\boldsymbol{U}_i')$, $i = 1, \dots, n_{\text{gen}}$, where $\boldsymbol{U}_1', \dots, \boldsymbol{U}_{n_{\text{gen}}}' \overset{\text{ind.}}{\sim} \mathrm{U}(0,1)^p$.
3. Return $\boldsymbol{Y}_i = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_i)$, $i = 1, \dots, n_{\text{gen}}$; to obtain a sample from C, return the pseudo-observations of $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{n_{\text{gen}}}$ (Genest et al., 1995).

To obtain quasi-random samples from $F_{\boldsymbol{X}}$ with underlying copula C, we replace $\boldsymbol{U}_1', \dots, \boldsymbol{U}_{n_{\text{gen}}}' \overset{\text{ind.}}{\sim} \mathrm{U}(0,1)^p$ in Algorithm 4.2.2 by an RQMC point set $\tilde{P}_{n_{\text{gen}}} = \{\tilde{\boldsymbol{v}}_1, \dots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$, where $\tilde{\boldsymbol{v}}_i \sim \mathrm{U}(0,1)^p$, $i = 1, \dots, n_{\text{gen}}$, to obtain the following algorithm; the randomization is done to obtain unbiased QMC estimators and estimates of their variances. Note that while individual RQMC points $\tilde{\boldsymbol{v}}_i$ mimic $\boldsymbol{U}_i'$, $\tilde{\boldsymbol{v}}_1, \dots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}$ are dependent.

Algorithm 4.2.3 (Quasi-random sampling by GMMN)

1. Fix the number n_{gen} of samples to generate from $\boldsymbol{Y}$.
2. Compute an RQMC point set $\tilde{P}_{n_{\text{gen}}} = \{\tilde{\boldsymbol{v}}_1, \dots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$ (for example, a randomized Sobol' or a generalized Halton sequence) and $\boldsymbol{Z}_i = F_{\boldsymbol{Z}}^{-1}(\tilde{\boldsymbol{v}}_i)$, $i = 1, \dots, n_{\text{gen}}$.
3. Return $\boldsymbol{Y}_i = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_i)$, $i = 1, \dots, n_{\text{gen}}$; to obtain a sample from C, return the pseudo-observations of $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{n_{\text{gen}}}$.

As mentioned in the introduction, Cambou et al. (2017) presented transformations to convert $\tilde{P}_{n_{\text{gen}}}$ to samples which mimic samples from C but locally provide a more homogeneous coverage. Unfortunately, these transformations are only available for a few specific cases of C and their numerical evaluation in a fast and robust way is even more challenging. We can avoid these problems by first training a GMMN on pseudo-random samples from $F_{\boldsymbol{X}}$ with any copula C. Then, the trained GMMN $f_{\hat{\boldsymbol{\theta}}}$ can be used to generate quasi-random samples from $F_{\boldsymbol{X}}$ as in Algorithm 4.2.3. Alternatively, quasi-random samples which follow the same empirical distribution as any given dataset can be obtained by training a GMMN on the given dataset itself. An additional advantage is that GMMNs provide a sufficiently smooth map from the RQMC point set to the target distribution which helps preserve the low-discrepancy of the point set upon transformation and hence guarantees the improved performance of RQMC estimators compared to the MC estimator (see Section 4.4 and Appendix C.1).

With the mapping $F_{\boldsymbol{Z}}^{-1}(\boldsymbol{u}) = (F_{Z_1}^{-1}(u_1), \ldots, F_{Z_p}^{-1}(u_p))$ to the input distribution and the trained GMMN $f_{\hat{\boldsymbol{\theta}}}$ at hand, define a transform

$$q(\boldsymbol{u}) = f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}(\boldsymbol{u}), \quad \boldsymbol{u} \in (0,1)^p.$$

Based on the RQMC point set $\tilde{P}_{n_{\text{gen}}} = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$ of size n_{gen}, we can then obtain quasi-random samples by

$$\boldsymbol{Y}_i = q(\tilde{\boldsymbol{v}}_i), \quad i = 1, \ldots, n_{\text{gen}},$$

(compare with (4.2)) and define a *GMMN RQMC estimator* of (4.3) by

$$\hat{\mu}_{n_{\text{gen}}}^{\text{NN}} = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(\boldsymbol{Y}_i) = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(q(\tilde{\boldsymbol{v}}_i)) = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi\Big(f_{\hat{\boldsymbol{\theta}}}(F_{\boldsymbol{Z}}^{-1}(\tilde{\boldsymbol{v}}_i))\Big). \tag{4.7}$$

We thus have the approximations

$$\mathbb{E}(\Psi(\boldsymbol{X})) \approx \mathbb{E}(\Psi(\boldsymbol{Y})) \approx \hat{\mu}_{n_{\text{gen}}}^{\text{NN}}. \tag{4.8}$$

The error in the first approximation is small if the GMMN is trained well and the error in the second approximation is small if the unbiased estimator $\hat{\mu}_{n_{\text{gen}}}^{\text{NN}}$ has a small variance. The primary *bottleneck* in this setup is the error in the first approximation which is determined by the size n_{trn} of the training dataset and, in particular, by the batch size n_{bat} which is the major factor determining training efficiency of the GMMN we found in all our numerical studies. Given a sufficiently large n_{bat} and, by extension, n_{trn}, the GMMN is trained well, which renders the first approximation error in (4.8) negligible. However, in practice the batch size n_{bat} is constrained by the quadratically increasing memory demands to compute the MMD loss function of the GMMN. For a theoretical result regarding this approximation error, see Dziugaite et al. (2015) where a bound on the error between optimizing a sample version and a population version of $\text{MMD}(X, Y)$ was investigated. Finally, let us note that the task of GMMN training and generation are separate steps which ensures that, once trained, generating quasi-random GMMN samples is comparably fast; see Appendix C.2.

The error in the second approximation in (4.8) is small if the composite function $\Psi \circ q$ is sufficiently smooth. The transform q is sufficiently smooth for GMMNs $f_{\hat{\boldsymbol{\theta}}}$ constructed using standard activation functions and commonly used input distributions; see the discussion following Corollary C.1.4. Given a sufficiently smooth Ψ, we can establish a rate of convergence $O(n_{\text{gen}}^{-3}(\log n_{\text{gen}})^{p-1})$ for the variance (and $O(n_{\text{gen}}^{-3/2}(\log n_{\text{gen}})^{(p-1)/2})$ for the approximation error) of the GMMN RQMC estimator $\hat{\mu}_{n_{\text{gen}}}^{\text{NN}}$ constructed by scrambling a digital net to obtain $\{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$; see Appendix C.1.4. With a stronger assumption on

the behavior of the composite function $\Psi \circ q$, we can show that the Koksma–Hlawka bound on the error between the (non-randomized) GMMN QMC estimator $\frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(q(\boldsymbol{v}_i))$ and $\mathbb{E}(\Psi(\boldsymbol{Y}))$ is satisfied which in turn implies a rate of convergence $O(n_{\text{gen}}^{-1}(\log n_{\text{gen}})^p)$ for the (non-randomized) GMMN QMC estimator; see Appendix C.1.2. If the Koksma–Hlawka bound holds, we can also establish a rate of convergence $O(n_{\text{gen}}^{-2}(\log n_{\text{gen}})^{2p})$ for the variance of GMMN RQMC estimators constructed using the digital shift method as randomization technique; see Appendix C.1.4.

4.3 GMMN pseudo- and quasi-random samples for copula models

In this section we assess the quality of pseudo-random samples and quasi-random samples generated from GMMNs. In both cases we train GMMNs on pseudo-random samples $\boldsymbol{U}_1, \dots, \boldsymbol{U}_{n_{\text{trn}}} \sim C$ from the respective copula C. We start by addressing key implementation details and hyperparameters of Algorithm 4.2.1 that we used in all examples thereafter. By utilizing this algorithm to train $f_{\boldsymbol{\theta}}$ for a wide variety of copula families, we then investigate the quality of the samples $\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_{n_{\text{gen}}}$, once generated by Algorithm 4.2.2 and once by Algorithm 4.2.3.

4.3.1 GMMN architecture, choice of kernel and training setup

We find a single hidden layer architecture ($L = 1$) to be sufficient for all the examples we considered. This is because, in this chapter, we largely consider the cases of $d \in \{2, ..., 10\}$. Learning an entire distribution nonparametrically for $d > 10$ would most likely require $L > 1$, but it would also require a much larger sample size n_{trn} and become much more challenging computationally for GMMNs — recall from Section 4.2.2 that the loss function requires $\binom{n_{\text{trn}}}{2}$ evaluations. After experimentation, we fix $d_1 = 300$, ϕ_1 to be ReLU (it offers computational efficiency via non-expensive and non-vanishing gradients) and ϕ_2 to be sigmoid (to obtain outputs in $[0, 1]^d$).

To avoid the need of fine-tuning the bandwidth parameter, we follow Li et al. (2015) and use a mixture of Gaussian kernels with different bandwidth parameters as our kernel function for the MMD statistic in (4.6); specifically,

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n_{\text{krn}}} K(\boldsymbol{x}, \boldsymbol{y}; \sigma_i), \tag{4.9}$$

where n_{krn} denotes the number of mixture components and $K(\boldsymbol{x}, \boldsymbol{y}; \sigma) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|_2^2/(2\sigma^2))$ is the Gaussian kernel with bandwidth parameter $\sigma > 0$. After experimentation, we fix $n_{\text{krn}} = 6$ and choose $(\sigma_1, \dots, \sigma_6) = (0.001, 0.01, 0.15, 0.25, 0.50, 0.75)$; note that copula samples are in $[0,1]^d$.

Unless otherwise specified, we use the following setup across all examples. We use $n_{\text{trn}} = 60\,000$ training data points and find this to be sufficiently large to obtain reliable $f_{\hat{\boldsymbol{\theta}}}$. As dimension of the input distribution $F_{\boldsymbol{Z}}$, we choose $p = d$, that is, the GMMN $f_{\boldsymbol{\theta}}$ is set to be a d-to-d transformation. For $F_{\boldsymbol{Z}}$, we choose $\boldsymbol{Z} \sim \mathrm{N}(\boldsymbol{0}, I_d)$, where I_d denotes the identity matrix in $\mathbb{R}^{d \times d}$, so $\boldsymbol{Z}$ consists of independent standard normal random variables; this choice worked better than $\mathrm{U}(0,1)^d$ in practice despite the fact that $\mathrm{N}(\boldsymbol{0}, I_d)$ does not satisfy the assumptions of Proposition C.1.1. We choose a batch size of $n_{\text{bat}} = 5000$ in Algorithm 4.2.1; this decision is motivated from a practical trade-off that a small n_{bat} will lead to poor estimates of the population MMD loss function but a large n_{bat} will incur quadratically growing memory requirements due to (4.6). As the number of epochs we choose $n_{\text{epo}} = 300$ which is generally sufficient in our experiments to obtain accurate results. The tuning parameters of the Adam optimizer is set to the default values reported in Kingma and Ba (2014).

All results in this section, Section 4.4 and Appendix C.1.4 are reproducible with the demo `GMMN_QMC_paper` of the R package `gnn`. Our implementation utilizes the R packages `keras` and `tensorflow` which serve as R interfaces to the corresponding namesake Python libraries. Furthermore, all GMMN training is carried out on one NVIDIA Tesla P100 GPU. To generate the RQMC point set in Algorithm 4.2.3, we use scrambled nets (Owen, 1995); see also Appendix C.1.3. Specifically, we use the implementation `sobol(, randomize = "Owen")` from the R package `qrng`. Finally, our choice of R as programming language for this work was motivated by the fact that contributed packages providing functionality for copula modeling and quasi-random number generation — two of the three major fields of research (besides deep learning) this work touches upon — exist in R.

4.3.2 Visual assessments of GMMN samples

In this section we primarily focus on the bivariate case but include an example involving a trivariate copula; for higher-dimensional copulas, see Sections 4.3.3 and 4.4. For all one-parameter copulas considered, the single parameter will be chosen such that Kendall's tau, denoted by τ, is equal to 0.25 (weak dependence), 0.50 (moderate dependence) or 0.75 (strong dependence); clearly, this only applies to copula families where there is a one-to-one mapping between the copula parameter and τ.

t, Archimedean copulas and their associated mixtures

First, we consider Student t copulas, Archimedean copulas, and their mixtures.

Student t copulas are prominent members of the elliptical class of copulas and are given by $C(\boldsymbol{u}) = t_{\nu,P}(t_\nu^{-1}(u_1), \dots, t_\nu^{-1}(u_d))$, $\boldsymbol{u} \in [0,1]^d$, where $t_{\nu,P}$ denotes the distribution function of the d-dimensional t distribution with ν degrees of freedom, location vector $\mathbf{0}$ and correlation matrix P, and t_ν^{-1} denotes the quantile function of the univariate t distribution with ν degrees of freedom. For all t copulas considered in this work, we fix $\nu = 4$. Student t copulas have explicit inverse Rosenblatt transforms, so one can utilize the CDM for generating quasi-random samples from them Cambou et al. (2017).

Archimedean copulas are copulas of the form

$$C(\boldsymbol{u}) = \psi(\psi^{-1}(u_1) + \cdots + \psi^{-1}(u_d)), \quad \boldsymbol{u} \in [0,1]^d,$$

for an Archimedean generator ψ which is a continuous, decreasing function $\psi : [0, \infty] \to [0, 1]$ that satisfies $\psi(0) = 1$, $\psi(\infty) = \lim_{t\to\infty} \psi(t) = 0$ and that is strictly decreasing on $[0, \inf t : \psi(t) = 0]$. Examples of Archimedean generators include $\psi_C(t) = (1 + t)^{-1/\theta}$ (for $\theta > 0$) and $\psi_G(t) = \exp(-t^{1/\theta})$ (for $\theta \geq 1$), generating Clayton and Gumbel copulas, respectively. While the inverse Rosenblatt transform and thus the CDM is available analytically for Clayton copulas, this is not the case for Gumbel copulas; in Appendix C.2 we used numerical root finding to include the latter case for the purpose of timings only.

We additionally consider equally-weighted two-component mixture copulas in which one component is a 90-degree-rotated t_4 copula with $\tau = 0.50$ and the other component is either a Clayton copula ($\tau = 0.50$) or a Gumbel copula ($\tau = 0.50$). The two mixture copula models are referred to as Clayton-t(90) and Gumbel-t(90) copulas, respectively.

The top rows of Figures 4.2–4.4 display contour plots of true t, Clayton and Gumbel copulas respectively, with $\tau = 0.25$ (left), 0.50 (middle) and 0.75 (right) along with contours of empirical copulas based on GMMN pseudo-random and GMMN quasi-random samples corresponding to each true copula C. The top row of Figure 4.5 displays similar plots for Clayton-t(90) (left) and Gumbel-t(90) (right) copulas. In each plot, across all figures described above, we observe that the contour of the empirical copula based on GMMN pseudo-random samples is visually fairly similar to the contour of C, thus indicating that the 11 GMMNs have been trained sufficiently well. We also see that the contours of the empirical copulas based on GMMN quasi-random samples better approximate the contours of C than the contours of the empirical copulas based on the corresponding pseudo-random samples. This observation indicates that, at least visually, the 11 GMMN transforms

(corresponding to each C) have preserved the low-discrepancy of the input RQMC point sets.

The bottom rows of Figures 4.2–4.5 display Rosenblatt transformed GMMN quasi-random samples, corresponding to each of the 11 true copulas C under consideration. The Rosenblatt transform for a bivariate copula C maps $(U_1, U_2) \sim C$ to $(R_1, R_2) = (U_1, C_{2|1}(U_2 \,|\, U_1))$, where $C_{2|1}(u_2 \,|\, u_1)$ denotes the conditional distribution function of U_2 given $U_1 = u_1$ under C. We exploit the fact that $(R_1, R_2) \sim \mathrm{U}(0,1)^2$ if and only if $(U_1, U_2) \sim C$. Moreover, Rosenblatt-transforming the GMMN quasi-random samples should yield a more homogeneous coverage of $[0,1]^2$. From each of the scatter plots in Figures 4.2–4.5, we observe no significant departure from $\mathrm{U}(0,1)^2$, thus indicating that the GMMNs have learned sufficient approximations to the corresponding true copulas C. Furthermore, the lack of gaps or clusters in the scatter plots provides some visual confirmation of the low-discrepancy of the Rosenblatt-transformed GMMN quasi-random samples.

Nested Archimedean, Marshall–Olkin and mixture copulas

Next, we consider more complex copulas such as nested Archimedean copulas and Marshall–Olkin copulas. We also re-consider the two mixture copulas introduced in the previous section along with an additional mixture copula. To better showcase the complexity of these dependence structures, we use scatter plots instead of contour plots to display copula and GMMN-generated samples. We omit the plots containing the Rosenblatt transformed samples since they are harder to obtain for the copulas we investigate in this section.

Nested Archimedean copulas (NACs) are Archimedean copulas with arguments possibly replaced by other NACs; see McNeil (2008) or Hofert (2012). In particular, this class of copulas allows us to construct asymmetric extensions of Archimedean copulas. Important to note here is that NACs are copulas for which there is no known (tractable) CDM. To demonstrate the ability of GMMNs to capture such dependence structures, we consider the simplest three-dimensional copula for visualization and investigate higher-dimensional NACs in Sections 4.3.3 and 4.4. The three-dimensional NAC we consider here is

$$C(\boldsymbol{u}) = C_0(C_1(u_1, u_2), u_3), \quad \boldsymbol{u} \in [0,1]^3, \tag{4.10}$$

where C_0 is a Clayton copula with Kendall's tau $\tau_0 = 0.25$ and C_1 is a Clayton copula with Kendall's tau $\tau_1 = 0.50$. In Sections 4.3.3 and 4.4, we will present examples of five- and ten-dimensional NACs.

Bivariate Marshall–Olkin copulas are of the form

$$C(u_1, u_2) = \min\{u_1^{1-\alpha_1} u_2, u_1 u_2^{1-\alpha_2}\}, \quad u_1, u_2 \in [0,1],$$

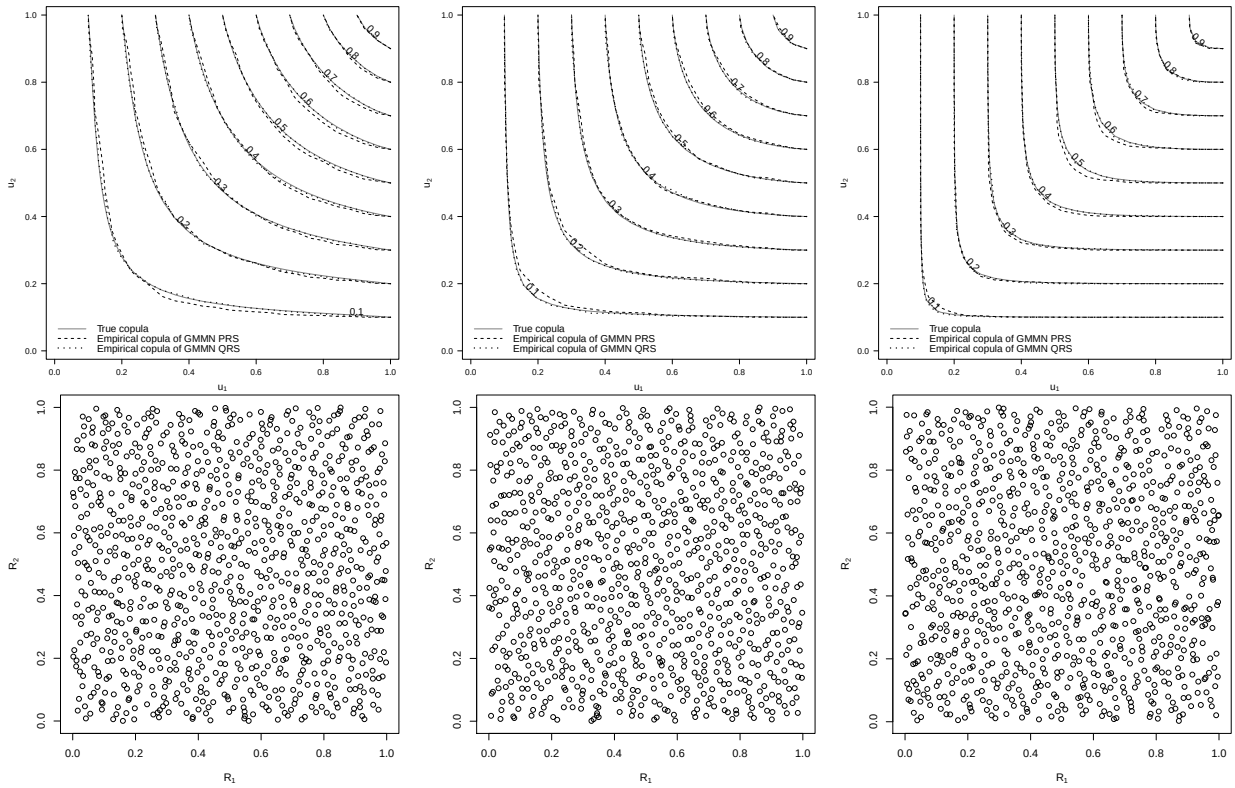

Figure 4.2: Top row contains contour plots of true t_4 copulas with $\tau = 0.25$ (left), 0.50 (middle) and 0.75 (right) along with the corresponding contour plots of empirical copulas based on GMMN pseudo-random and GMMN quasi-random samples (respectively, GMMN PRS and GMMN QRS), both of size $n_{\text{gen}} = 1000$. Bottom row contains Rosenblatt-transformed GMMN QRS corresponding to the same three t_4 copulas.

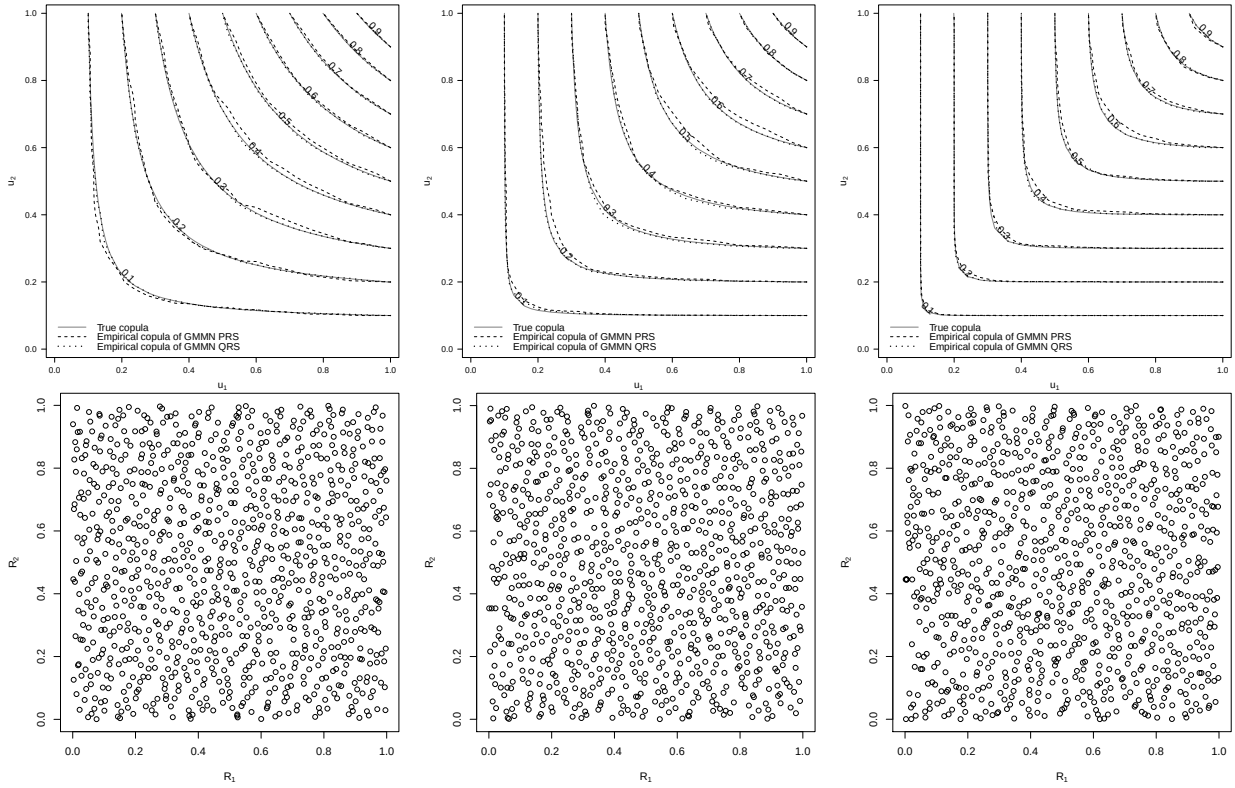

Figure 4.3: Top row contains contour plots of true Clayton copulas with $\tau = 0.25$ (left), 0.50 (middle) and 0.75 (right) along with the corresponding contour plots of empirical copulas based on GMMN PRS and GMMN QRS, both of size $n_{\text{gen}} = 1000$. Bottom row contains Rosenblatt-transformed GMMN QRS corresponding to the same three Clayton copulas.

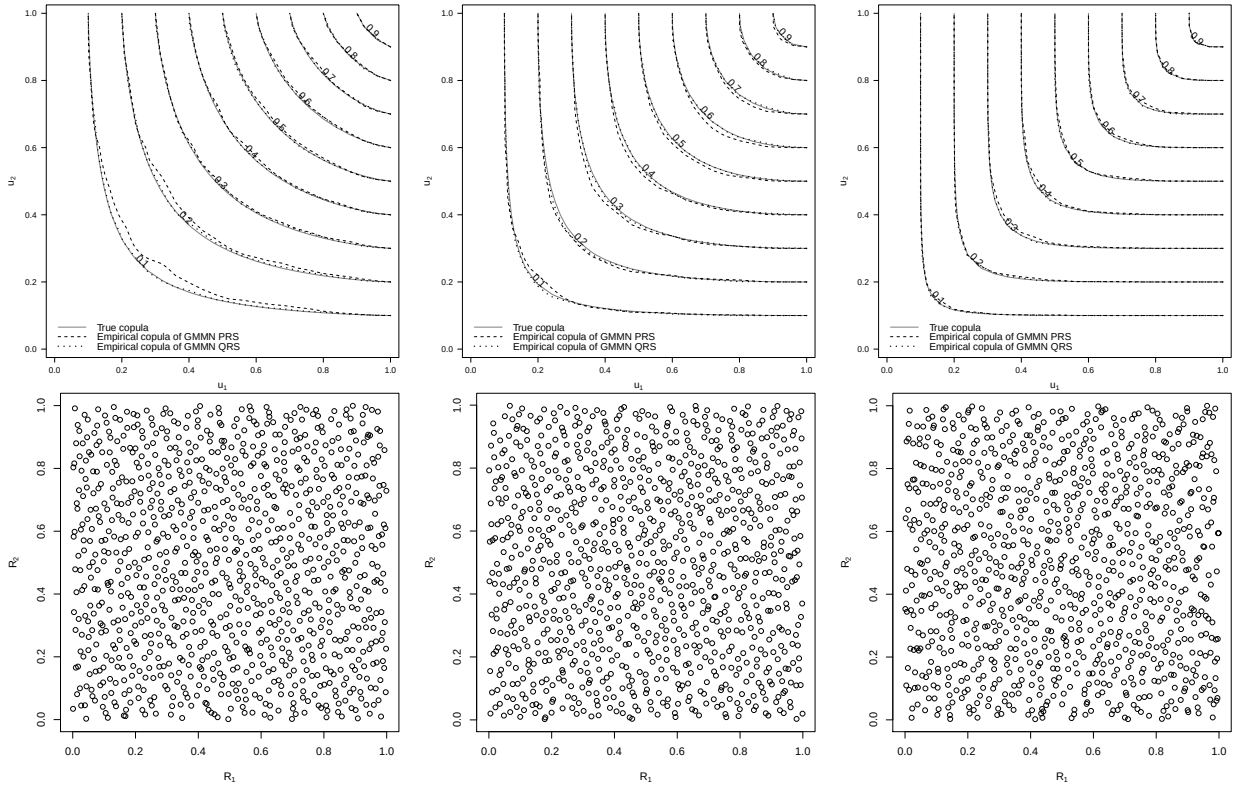

Figure 4.4: Top row contains contour plots of true Gumbel copulas with $\tau = 0.25$ (left), 0.50 (middle) and 0.75 (right) along with the corresponding contour plots of empirical copulas based on GMMN PRS and GMMN QRS, both of size $n_{\text{gen}} = 1000$. Bottom row contains Rosenblatt-transformed GMMN QRS corresponding to the same three Gumbel copulas.

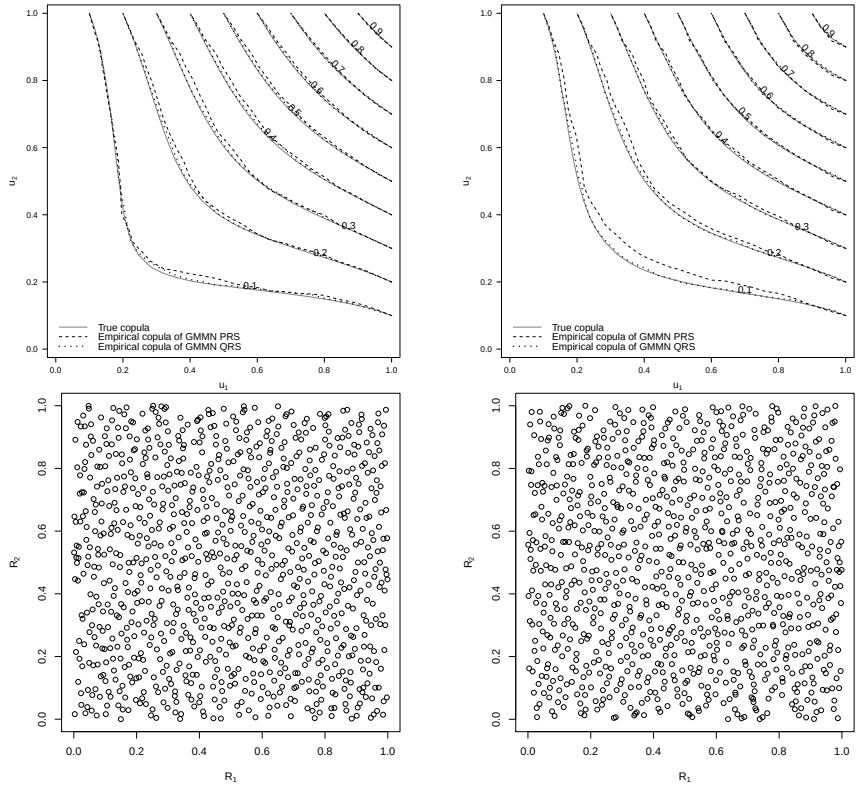

Figure 4.5: Top row contains contour plots of true Clayton-$t(90)$ (left) and Gumbel-$t(90)$ (right) mixture copulas along with the corresponding contour plots of empirical copulas based on GMMN PRS and GMMN QRS, both of size $n_{\text{gen}} = 1000$. Bottom row contains Rosenblatt-transformed GMMN QRS corresponding to the same two mixture copulas.

where $\alpha_1, \alpha_2 \in [0, 1]$. A notable feature of Marshall–Olkin copulas is that they have both an absolutely continuous component and a singular component. In particular, the singular component is determined by all points which satisfy $u_1^{\alpha_1} = u_2^{\alpha_2}$. Accurately capturing this singular component may present a different challenge for GMMNs, which is why we included this copula despite the fact that there also exists a CDM for this copula; see Cambou et al. (2017). As an example for visual assessment, we consider a Marshall–Olkin copula with $\alpha_1 = 0.75$ and $\alpha_2 = 0.60$.

We also consider three mixture models, all of which are equally weighted two-component mixture copulas with one component being a 90-degree-rotated t_4 copula with $\tau = 0.50$. The first two models are the Clayton-$t(90)$ and Gumbel-$t(90)$ mixture copulas as previously introduced. The second component in the third model is a Marshall–Olkin copula with parameters $\alpha_1 = 0.75$ and $\alpha_2 = 0.60$. We refer to this third model as the MO-$t(90)$ copula.

Figures 4.6–4.8 display pseudo-random samples (left column) from a $(2, 1)$-nested Clayton copula as in (4.10), a MO copula, and the three mixture copulas, respectively, along with GMMN pseudo-random samples (middle column) and GMMN quasi-random samples (right column) corresponding to each copula C. The similarity between the GMMN pseudo-random samples in the middle column and the pseudo-random samples in the left column indicate that the copulas C were learned sufficiently well by their corresponding GMMNs. Note that in the case of the nested Clayton copula, we can only comment on how well the bivariate margins of the copula C were learned. From the right columns, we can mainly observe that the GMMN quasi-random samples contain less gaps and clusters when compared with the corresponding pseudo-random and GMMN pseudo-random samples. The fact that GMMNs were capable of learning the main features of the MO copulas and the MO-$t(90)$ mixture copulas, including the singular components, is particularly noteworthy given how challenging it seems to be to learn a Lebesgue null set from a finite amount of samples.

4.3.3 Assessment of GMMN samples by the Cramér-von Mises statistic

After a purely visual inspection of the generated samples, we now assess the quality of GMMN pseudo-random and GMMN quasi-random samples more formally with the help of a goodness-of-fit statistic. Since bivariate copulas have been investigated in detail in the previous section, we focus on higher-dimensional copulas in this section.

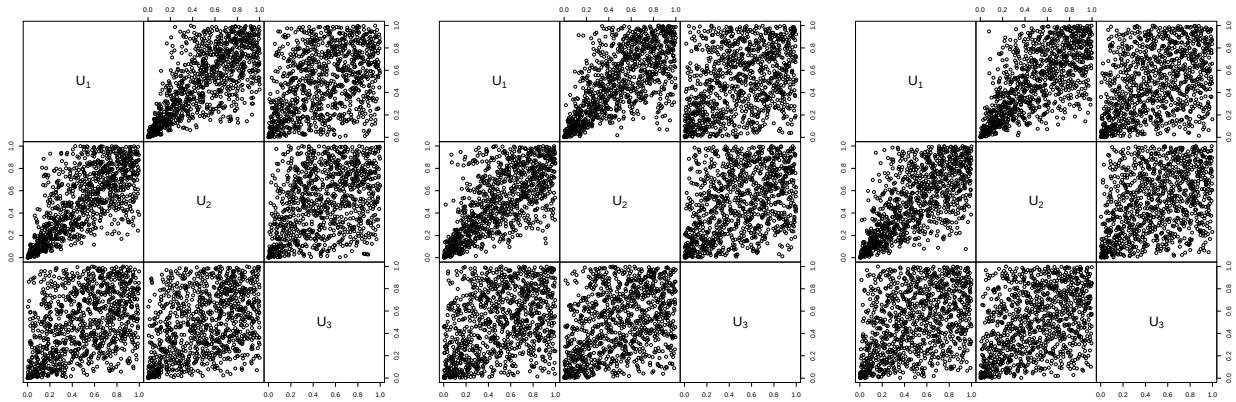

Figure 4.6: Pseudo-random samples (PRS; left), GMMN pseudo-random samples (GMMN PRS; middle) and GMMN quasi-random samples (GMMN QRS; right), all of size $n_{\text{gen}} = 1000$, from a (2,1)-nested Clayton copula as in (4.10) with $\tau_0 = 0.25$ and $\tau_1 = 0.50$.

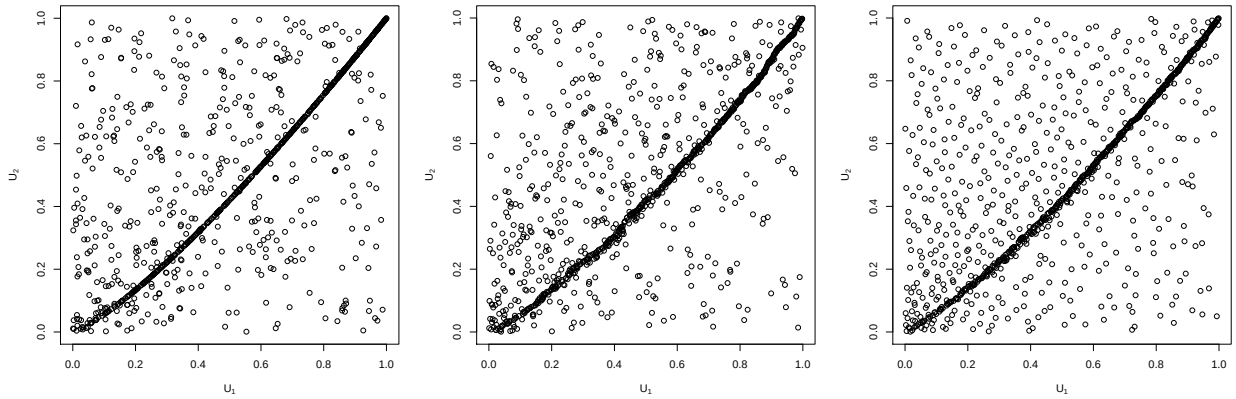

Figure 4.7: PRS (left), GMMN PRS (middle) and GMMN QRS (right), all of size $n_{\text{gen}} = 1000$, from a Marshall–Olkin copula with $\alpha_1 = 0.75$ and $\alpha_2 = 0.60$ (Kendall's tau equals 0.5).

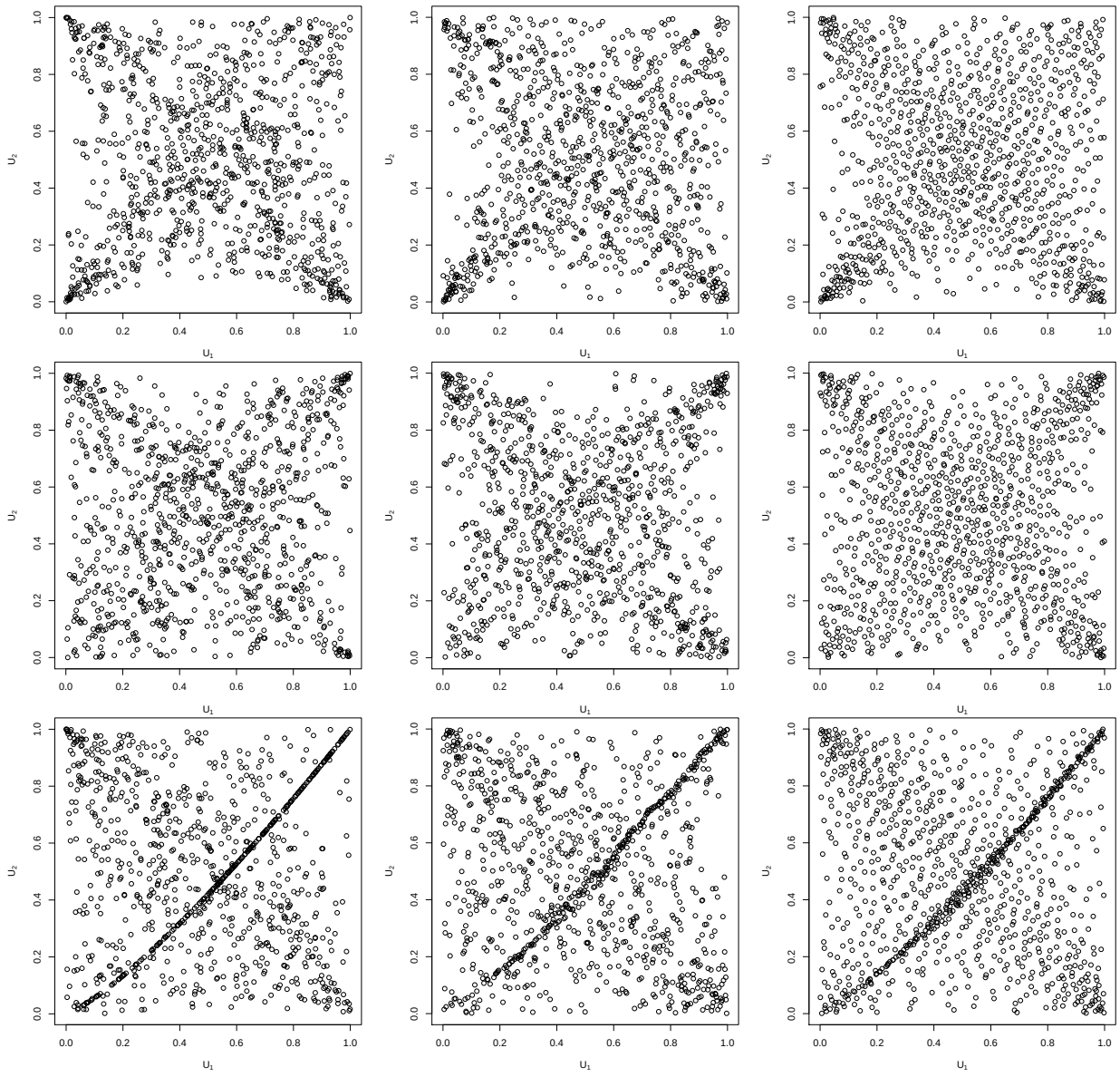

Figure 4.8: PRS (left column), GMMN PRS (middle column) and GMMN QRS (right column), all of size $n_{\text{gen}} = 1000$, from a Clayton-$t(90)$ (top row), Gumbel-$t(90)$ (middle row) and a MO-$t(90)$ mixture (bottom row) copula.

Specifically, we use the Cramér–von Mises statistic (Genest et al., 2009),

$$S_{n_{\text{gen}}} = \int_{[0,1]^d} n_{\text{gen}}(C_{n_{\text{gen}}}(\boldsymbol{u}) - C(\boldsymbol{u}))^2 \, \mathrm{d}C_{n_{\text{gen}}}(\boldsymbol{u}),$$

where the empirical copula

$$C_{n_{\text{gen}}}(\boldsymbol{u}) = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \mathbb{1}\{\hat{U}_{i1} \le u_1, \dots, \hat{U}_{id} \le u_d\}, \quad \boldsymbol{u} \in [0,1]^d, \tag{4.11}$$

is the empirical distribution function of the pseudo-observations. For $n_{\text{gen}} = 1000$ and each copula C, we compute $B = 100$ realizations of $S_{n_{\text{gen}}}$ three times — once for the case where $\hat{\boldsymbol{U}}_i$, $i = 1, \dots, n_{\text{gen}}$, are pseudo-observations of the true underlying copula (as benchmark), once for GMMN pseudo-random samples and once for GMMN quasi-random samples. We then use box plots to depict the distribution of $S_{n_{\text{gen}}}$ in each case. Figure 4.9 displays these box plots for t_4 (top row), Clayton (middle row) and Gumbel copulas (bottom row) of dimensions $d = 5$ (left column), $d = 10$ (right column) and $\tau = 0.50$. Similarly, Figure 4.10 displays such box plots for d-dimensional nested Clayton (left column) and nested Gumbel copulas (right column) for $d = 3$ (top row), $d = 5$ (middle row) and $d = 10$ (bottom row). The three-dimensional NACs have a structure as given by (4.10) with $\tau_0 = 0.25$ and $\tau_1 = 0.50$; the five-dimensional NACs have structure $C_0(C_1(u_1, u_2), C_2(u_3, u_4, u_5))$ with corresponding $\tau_0 = 0.25$, $\tau_1 = 0.50$ and $\tau_2 = 0.75$; and the ten-dimensional NACs have structure $C_0(C_1(u_1, \dots, u_5), C_2(u_6, \dots, u_{10}))$ with corresponding $\tau_0 = 0.25$, $\tau_1 = 0.50$ and $\tau_2 = 0.75$.

We can observe from both figures that the distributions of $S_{n_{\text{gen}}}$ for pseudo-random samples from C and from the GMMN are similar, with slightly higher $S_{n_{\text{gen}}}$ values for the GMMN pseudo-random samples, especially for $d = 10$. Additionally, we can observe that the distribution of $S_{n_{\text{gen}}}$ based on the GMMN quasi-random samples is closer to zero than that of the GMMN pseudo-random samples. This provides some evidence that the low-discrepancy of input RQMC points set has been preserved under the respective (trained) GMMN transforms.

We also see that $S_{n_{\text{gen}}}$ values based on the GMMN quasi-random samples are clearly lower than $S_{n_{\text{gen}}}$ values based on the copula pseudo-random samples, with the exception of some copulas for $d = 10$ where the distributions of $S_{n_{\text{gen}}}$ are more similar.

4.4 Convergence analysis of the RQMC estimator

In this section we numerically investigate the variance-reduction properties of the GMMN RQMC estimator $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ in (4.7) for two functions Ψ and transforms $q = f_{\hat{\boldsymbol{\theta}}} \circ \Phi^{-1}$ corresponding

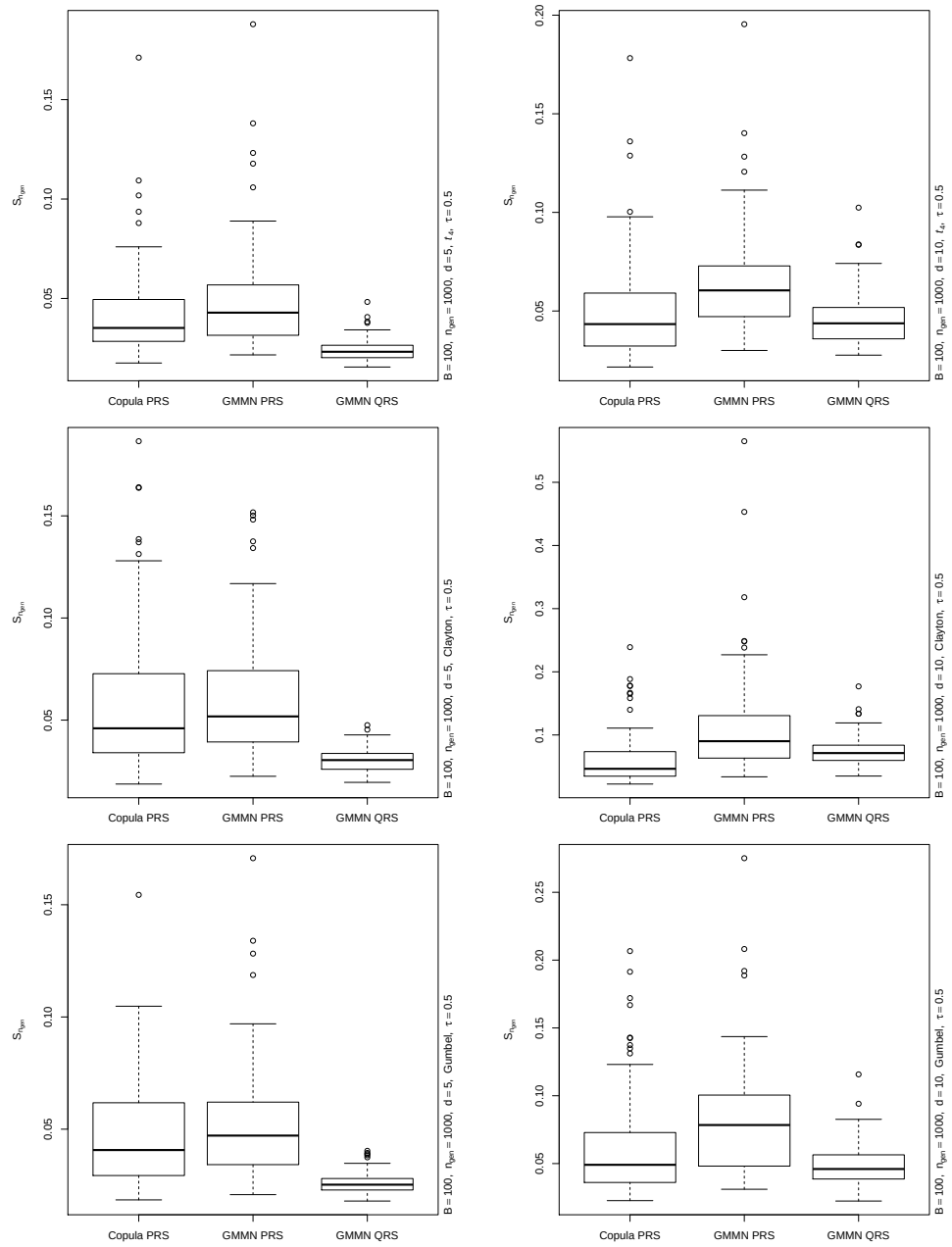

Figure 4.9: Box plots based on $B = 100$ realization of $S_{n_{\text{gen}}}$ computed from (i) a pseudo-random sample (PRS) of C (denoted by Copula PRS), (ii) a GMMN pseudo-random sample (denoted by GMMN PRS) and (iii) a GMMN quasi-random sample (denoted by GMMN QRS) — all of size $n_{\text{gen}} = 1000$ — for a t_4 (top row), Clayton (middle row) and Gumbel copulas (bottom row) with $\tau = 0.5$ as well as $d = 5$ (left column) and $d = 10$ (right column).

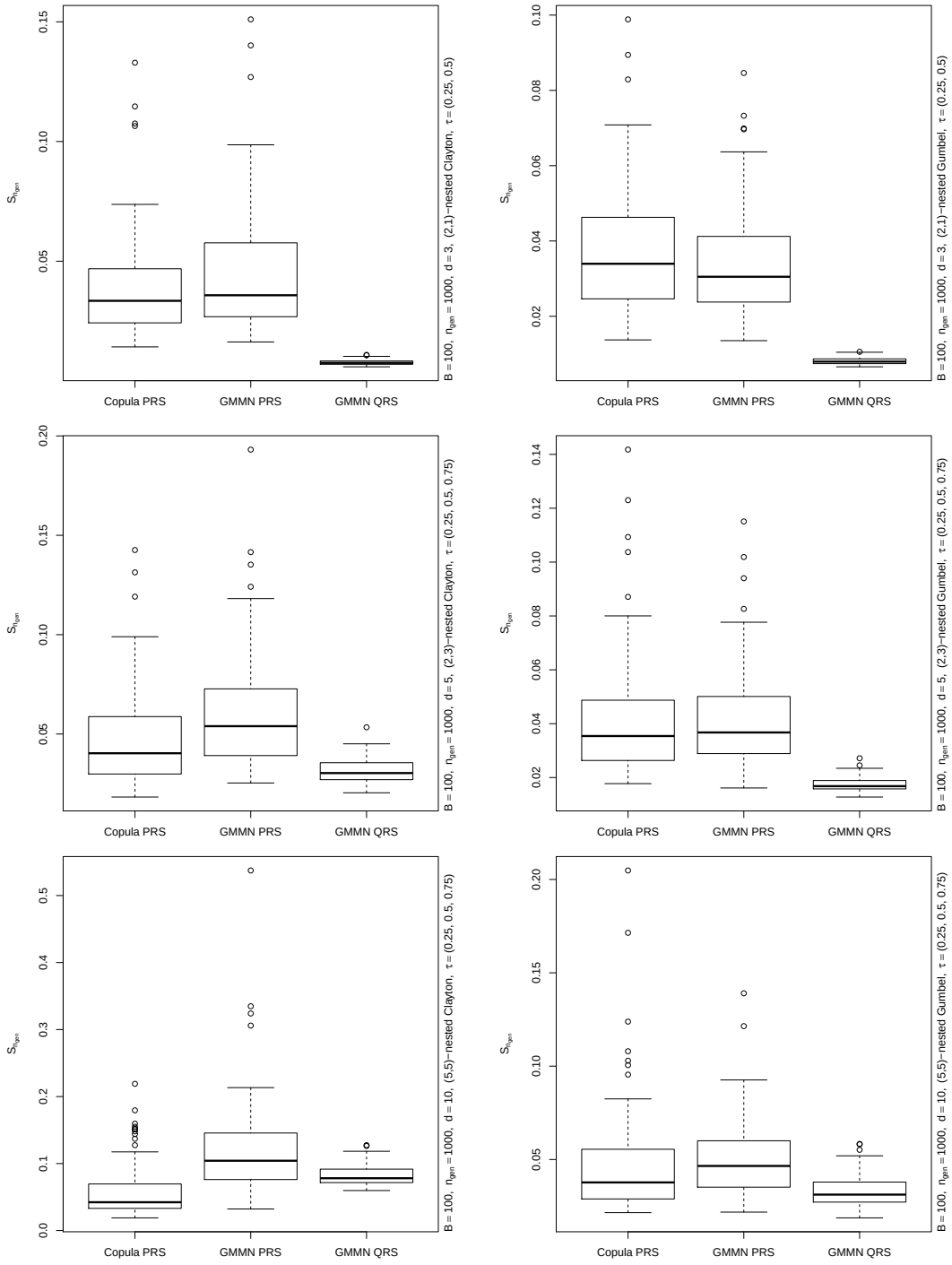

Figure 4.10: As Figure 4.9 but for nested Clayton (left column) and nested Gumbel copulas (right column) and for $d = 3$ (top row), $d = 5$ (middle row) and $d = 10$ (bottom row).

to different copulas C. We compare $\hat{\mu}^{\mathrm{NN}}_{n_{\mathrm{gen}}}$ with estimators based on standard copula pseudo-random and, where available, copula quasi-random samples. For the latter, we follow Cambou et al. (2017) but note that quasi-random sampling procedures are only available for some of the copulas we consider here; for others, the procedures are either too slow (e.g., for Gumbel copulas; see Appendix C.2) or not known at all (e.g., for nested Clayton or Gumbel copulas).

We consider two different types of functions Ψ. The first is a *test function* primarily used in the QMC literature to test the performance of $\hat{\mu}^{\mathrm{NN}}_{n_{\mathrm{gen}}}$ in terms of its ability to preserve the low-discrepancy of $\tilde{P}_{n_{\mathrm{gen}}}$. The second function Ψ is motivated from a practical application in risk management. For both functions, standard deviation estimates will be computed to compare convergence rates, based on $B = 25$ randomized point sets $\tilde{P}_{n_{\mathrm{gen}}}$ for each of $n_{\mathrm{gen}} \in \{2^{10}, 2^{10.5}, \dots, 2^{18}\}$ to help roughly gauge the convergence rate for all estimators. Furthermore, regression coefficients α (obtained by regressing the logarithm of the standard deviation on the logarithm of n_{gen}) are computed and displayed to allow for an easy comparison of the corresponding convergence rates $O(n_{\mathrm{gen}}^{-\alpha})$ with the theoretical convergence rate $O(n_{\mathrm{gen}}^{-0.5})$ of the Monte Carlo estimator's standard deviation. For RQMC estimators one can expect α to be larger than 0.5, but with an upper bound of $1.5 - \varepsilon$, where ε increases with dimension d; see Theorem C.1.3 for further details.

4.4.1 A test function

The test function we consider is the *Sobol' g* function (Radović et al., 1996) based on the Rosenblatt transform and is given by

$$\Psi_1(\boldsymbol{U}) = \prod_{j=1}^{d} \frac{|4R_j - 2| + j}{1 + j},$$

where $R_1 = U_1$ and, for $j = 2, \dots, d$ and if $\boldsymbol{U} \sim C$,

$$R_j = C_{j|1,\dots,j-1}(U_j \,|\, U_{j-1}, \dots, U_1)$$

denotes the conditional distribution function of U_j given $U_1, \dots, U_{j-1}$.

Figure 4.11 shows plots of standard deviation estimates for estimating $\mathbb{E}(\Psi_1(\boldsymbol{U}))$ for t_4 copulas (top row), Clayton (middle row) and Gumbel copulas (bottom row) in dimensions $d = 2$ (left column), $d = 5$ (middle column) and $d = 10$ (right column). For the t_4 and Clayton copulas we numerically compare the efficiency of the GMMN RQMC estimator (with legend label "GMMN QRS") with the copula RQMC estimator based on the CDM

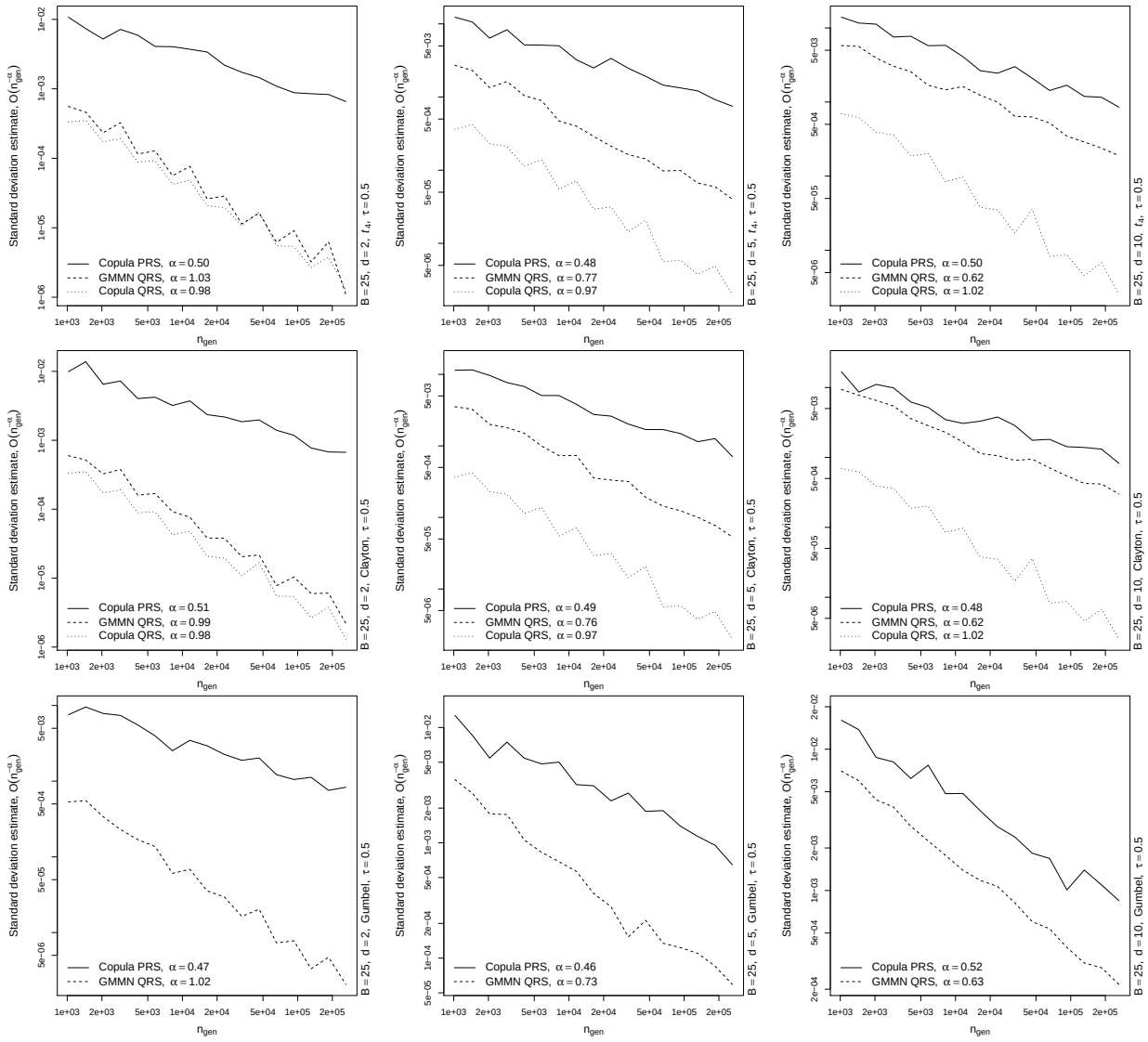

Figure 4.11: Standard deviation estimates based on $B = 25$ replications for estimating $\mathbb{E}(\Psi_1(\boldsymbol{U}))$, the expectation of the Sobol' g function, via MC based on a pseudo-random sample (PRS), via the copula RQMC estimator (whenever available; rows 1–2 only) and via the GMMN RQMC estimator. Note that each row has $d \in \{2, 5, 10\}$.

method (with legend label "Copula QRS") and the MC estimator (with legend label "Copula PRS"). For the Gumbel copula, however, the CDM approach ("Copula QRS") is computationally not feasible; see Section 4.3.2 and Appendix C.2. The legend of each plot also provides the regression coefficient α which indicates the convergence rate of each estimator.

From Figure 4.11, we observe that the GMMN RQMC estimator clearly outperforms the MC estimator. Naturally, so does the copula RQMC estimator for the copulas for which it is available. On the one hand, the rate of convergence of the GMMN RQMC estimator decreases with increasing copula dimensions; see also the decreasing regression coefficients α when moving from the two- to the ten-dimensional case. As a result, the copula RQMC estimator (when available) outperforms the GMMN RQMC estimator for five and ten dimensional copulas. On the other hand, the GMMN RQMC estimator still outperforms the MC estimator.

4.4.2 An example from risk management practice

Consider modeling the dependence of d risk-factor changes (for example, logarithmic returns) of a portfolio; see McNeil et al. (2015, Chapters 2, 6 and 7). We now demonstrate the efficiency of our GMMN RQMC estimator by considering the expected shortfall of the aggregate loss, a popular risk measure in quantitative risk management practice.

Specifically, if $\boldsymbol{X} = (X_1, \dots, X_d)$ denotes a random vector of risk-factor changes with $\mathrm{N}(0,1)$ margins, the aggregate loss is $S = \sum_{j=1}^d X_j$. The *expected shortfall* $\mathrm{ES}_{0.99}$ at level 0.99 of S is given by

$$\mathrm{ES}_{0.99}(S) = \frac{1}{1-0.99}\int_{0.99}^1 F_S^{-1}(u)\,\mathrm{d}u = \mathbb{E}\Big(S \,|\, S > F_S^{-1}(0.99)\Big) = \mathbb{E}(\Psi_2(\boldsymbol{X})),$$

where F_S^{-1} denotes the quantile function of S. As done previously, various copulas will be used to model the dependence between the components of $\boldsymbol{X}$.

Figure 4.12 shows plots of standard deviation estimates for estimating $\mathbb{E}(\Psi_2(\boldsymbol{X}))$. The first three rows contain results for the same copula models as considered in Section 4.4.1. The fourth row contains results for nested Gumbel copula models with dimension $d = 3$ (left column), $d = 5$ (middle column) and $d = 10$ (right column). The specific hierarchical structures and parameterization have been described earlier in Section 4.3.3; note that there is no quasi-random sampling procedure known for these copulas. We can observe from the plots that the GMMN RQMC estimator outperforms the MC estimator. Similar as before, we see a decrease in the convergence rate of the GMMN RQMC estimator as the copula dimension increases, although it still outperforms the MC estimator.

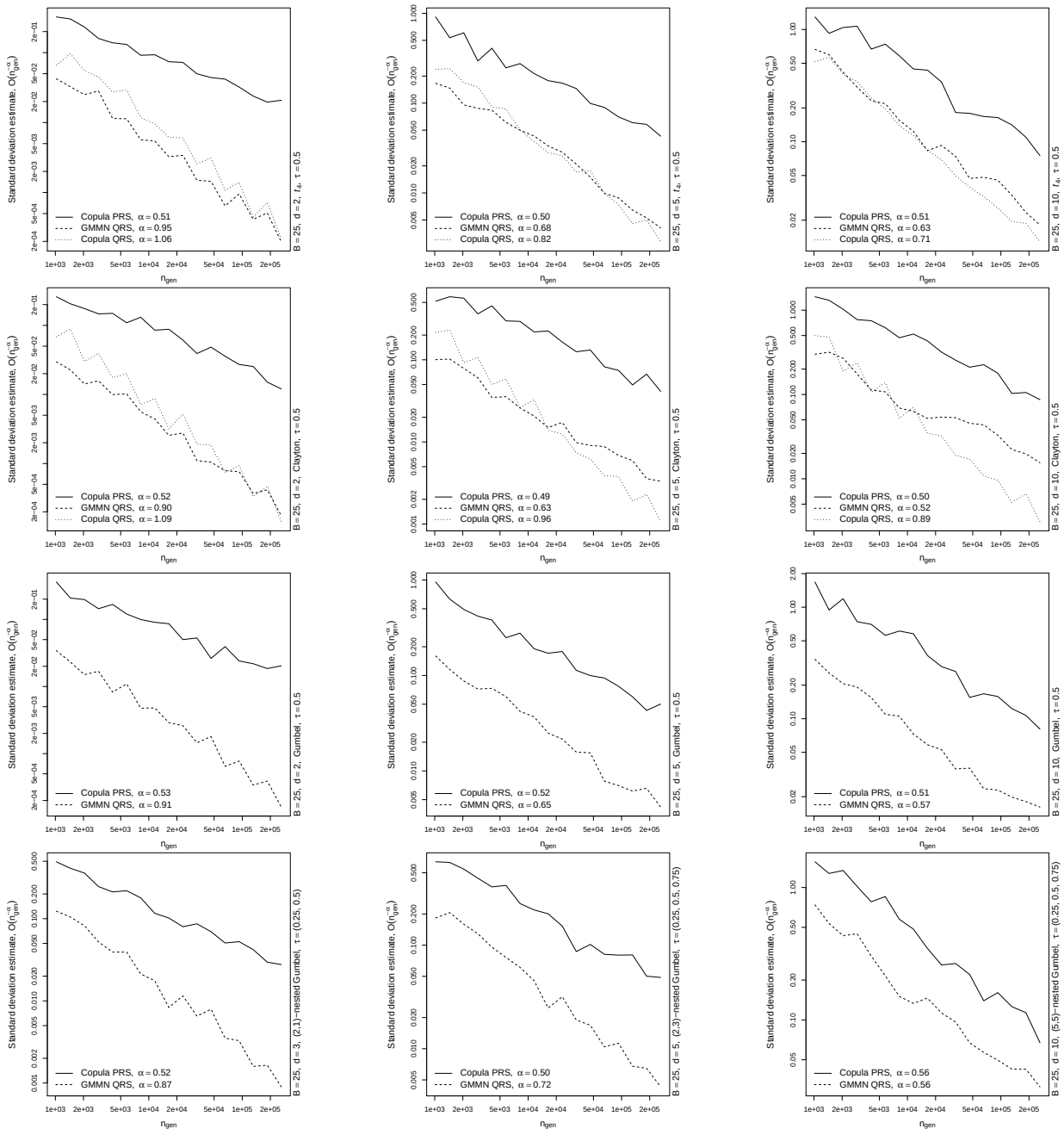

Figure 4.12: Standard deviation estimates based on $B = 25$ replications for estimating $\mathbb{E}(\Psi_2(\boldsymbol{X}))$, the expected shortfall $\mathrm{ES}_{0.99}(S)$, via MC based on a PRS, via the copula RQMC (whenever available; rows 1–2 only) and via the GMMN RQMC estimator. Note that in rows 1–3, $d \in \{2, 5, 10\}$, whereas in row 4, $d \in \{3, 5, 10\}$.

4.5 A financial data example

In this section, we present real-data examples to show how our method can be useful in practice. To this end, we consider applications from finance and risk management. Such applications often involve the modeling of dependent multivariate return data in order to estimate various quantities μ of interest. In this context, utilizing GMMNs for dependence modeling can yield two key advantages. Firstly, GMMNs are highly flexible and hence can model dependence structures not adequately captured by prominent parametric copula models; see, e.g., Hofert and Oldford (2018) for the latter point. Secondly, as demonstrated in Sections 4.3 and 4.4, one can readily generate GMMN quasi-random samples to achieve variance reduction when estimating μ; this is especially advantageous as oftentimes oversimplified parametric models are chosen just so that this can be achieved. In this section, we model asset portfolios consisting of S&P 500 constituents to showcase these advantages. All results are reproducible with the demo `GMMN_QMC_data` of the R package `gnn`.

4.5.1 Portfolios of S&P 500 constituents

We consider daily adjusted closing prices of 10 constituent time series from the S&P 500 in the time period from 1995-01-01 to 2015-12-31. The selected constituents include three stocks from the information technology sector — Intel Corp. (INTC), Oracle Corp. (ORCL) and International Business Machines Corp. (IBM); three stocks from the financial sector — Capital One Financial Corp. (COF), JPMorgan Chase & Co. (JPM) and American International Group Inc (AIG); and four stocks from the industrial sector — 3M Company (MMM), Boeing Company (BA), General Electric (GE) and Caterpillar Inc. (CAT). We also investigate sub-portfolios of stocks with dimensions $d = 5$ (consisting of INTC, ORCL, IBM, COF and AIG) and $d = 3$ (consisting of INTC, IBM and AIG). The data are obtained from the R package `qrmdata`.

To account for marginal temporal dependencies, we follow the copula–GARCH approach (Jondeau and Rockinger, 2006; Patton, 2006) and model each marginal time series of log-returns by an ARMA$(1, 1)$–GARCH$(1, 1)$ model with standardized t innovation distributions (*deGARCHing*). We then extract the marginal standardized residuals (i.e., the realizations of the standardized t innovations) and compute, for each of the three portfolios, their pseudo-observations for the purpose of modeling the cross-sectional dependence among the corresponding portfolio's log-return series.

4.5.2 Assessing the fit of the dependence models

As models for the pseudo-observations of each of the three portfolios we use prominent parametric copulas (Gumbel, Clayton, exchangeable normal, unstructured normal, exchangeable t and unstructured t) and GMMNs of the same architecture and with the same training setup as detailed in Section 4.3.1. The rather small number of training data points ($n_{\text{trn}} = 5287$) allows us to use $n_{\text{bat}} = n_{\text{trn}}$ here and hence directly train with the entire dataset. All parametric copulas are fitted using the maximum pseudo-likelihood method; see (Hofert et al., 2018b, Section 4.1.2).

To evaluate the fit of a dependence model, we use a Cramér-von-Mises type test statistic introduced by Rémillard and Scaillet (2009) to assess the equality of two empirical copulas. This statistic is defined as

$$S_{n_{\text{trn}},n_{\text{gen}}} = \int_{[0,1]^d} \left(\sqrt{\frac{1}{n_{\text{gen}}} + \frac{1}{n_{\text{trn}}}}\right)^{-1} \left(C_{n_{\text{gen}}}(\boldsymbol{u}) - C_{n_{\text{trn}}}(\boldsymbol{u})\right)^2 \mathrm{d}\boldsymbol{u},$$

where $C_{n_{\text{gen}}}(\boldsymbol{u})$ and $C_{n_{\text{trn}}}(\boldsymbol{u})$ are the empirical copulas, defined according to (4.11), of the n_{gen} samples generated from the fitted dependence model and the n_{trn} pseudo-observations used to fit the dependence model, respectively. For how $S_{n_{\text{trn}},n_{\text{gen}}}$ is evaluated, see Rémillard and Scaillet (2009, Section 2).

For each of the three portfolios and each of the seven dependence models considered, we compute B realizations of $S_{n_{\text{trn}},n_{\text{gen}}}$ based on $n_{\text{gen}} = 10\,000$ pseudo-random samples generated from the fitted dependence model under consideration and the $n_{\text{trn}} = 5287$ pseudo-observations of each portfolio considered. Figure 4.13 displays box plots depicting the distribution of $S_{n_{\text{trn}},n_{\text{gen}}}$ for each portfolio and dependence model. Across all three portfolios, we can observe that the distribution of $S_{n_{\text{trn}},n_{\text{gen}}}$ based on the GMMN models is concentrated closer to zero than those of each fitted parametric copula. In fact, the difference in distributions of $S_{n_{\text{trn}},n_{\text{gen}}}$ realizations between GMMN models and the best parametric copula model (a t-copula with unstructured correlation matrix) is most noticeable for $d = 10$, where an adequate fit becomes more challenging for the parametric copulas. For each of the three portfolios, a GMMN provides the best fit. Hence, we use these fitted GMMNs to model the underlying dependence structure for the three portfolios in each of three applications considered next.

4.5.3 Assessing the variance reduction effect

In three applications we study the variance reduction effect of our GMMN RQMC estimator $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ computed from quasi-random samples in comparison to the GMMN MC estimator

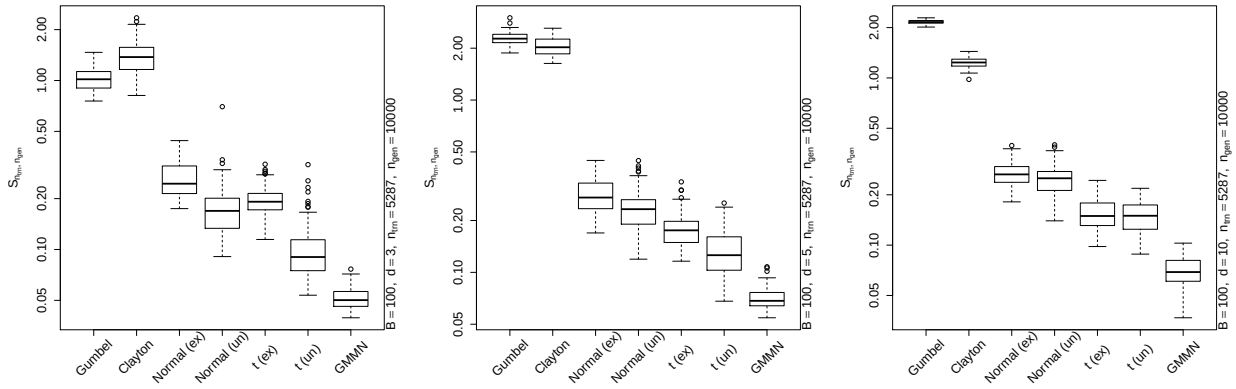

Figure 4.13: Box plots based on $B = 100$ realizations of $S_{n_{\text{trn}},n_{\text{gen}}}$ computed for portfolios of dimensions $d = 3$ (left), $d = 5$ (middle) and $d = 10$ (right) and for each fitted dependence model using a pseudo-random sample of size $n_{\text{gen}} = 10\,000$ from each corresponding fitted model.

$\hat{\mu}^{\text{NN,MC}}_{n_{\text{gen}}}$ computed from pseudo-random samples.

Our first application concerns the estimation of the expected shortfall $\mu = \text{ES}_{0.99}(S)$ for $S = \sum_{j=1}^{d} X_j$ as in Section 4.4.2, where the margins of $\boldsymbol{X} = (X_1, \dots, X_d)$ are now the fitted standardized t distributions as obtained by deGARCHing and the dependence structure is the previously fitted GMMN. This is a classical task in risk management practice according to the Basel guidelines. As a second application we consider a capital allocation problem which concerns estimating how to allocate an amount of risk capital (e.g., computed as $\text{ES}_{0.99}(S)$) to each of d business lines. Without loss of generality, we consider one business line, the first, and estimate the *expected shortfall contribution* $\mu = \text{AC}_{1,0.99} = \mathbb{E}(X_1 \,|\, S > F_S^{-1}(0.99))$ according to the Euler principle; see McNeil et al. (2015, Section 8.5). Our third application comes from finance and concerns the estimation of the expected payoff $\mu = \mathbb{E}(\exp(-r(T-t)) \max\{(\sum_{j=1}^{d} S_{T,j}) - K, 0\})$ of a European basket call option, where r denotes the continuously compounded annual risk-free interest rate, t denotes the current time point, T the maturity in years and K the strike price. We assume a Black–Scholes framework for the marginal stock prices $(S_{T,1}, \dots, S_{T,d})$ at maturity T, so $S_{T,j} \sim \text{LN}(\log(S_{t,j}) + (r - \sigma_j^2/2)(T-t),\ \sigma_j^2(T-t))$, where $S_{t,j}$ denotes the last available stock price of the jth constituent (i.e., the close price on 2015-12-31) and σ_j denotes the volatility of the jth constituent (estimated by the standard deviation of its log-returns over the time period from 2014-01-01 to 2015-12-31). The dependence structure of $(S_{T,1}, \dots, S_{T,d})$ is modeled by the previously fitted GMMN. Furthermore, we choose $t = 0$ to be the last available point in the data period considered (i.e., 2015-12-31), $T = 1$ and

$r = 0.01$. The strike prices K are chosen about 0.5% higher than the average stock price of all stocks in the respective portfolio at $t = 0$.

For each of the three portfolios and for each of the three expectations μ considered, we compute $B = 200$ realizations of the GMMN MC estimator $\hat{\mu}^{\text{NN,MC}}_{n_{\text{gen}}}$ and the GMMN RQMC estimator $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$, using $n_{\text{gen}} = 10^5$ samples for both estimators. Figure 4.14 displays box plots of these realizations of $\hat{\mu}^{\text{NN,MC}}_{n_{\text{gen}}}$ (with x-axis label "GMMN PRS") and of $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ (with x-axis label "GMMN QRS") for $\text{ES}_{0.99}$ (left column), $\text{AC}_{1,0.99}$ (middle column), the expected payoff of the basket call option (right column) and for the portfolio in dimension $d = 3$ (top row), $d = 5$ (middle row) and $d = 10$ (bottom row). Additionally, to quantify the variance reduction effect of $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ over $\hat{\mu}^{\text{NN,MC}}_{n_{\text{gen}}}$, we report in the secondary y-axis of each box plot the estimated variance reduction factor (VRF) — namely, the sample variance of $\hat{\mu}^{\text{NN,MC}}_{n_{\text{gen}}}$ over the sample variance of $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ — and the corresponding improvement in percentages.

From Figure 4.14, we observe that $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ is able to reduce the variance in all considered applications and across all dimensions. While variance reduction is diminished in higher dimensions ($d = 10$), the GMMN RQMC estimator is still immensely useful in estimating expectations μ for three reasons. Firstly, as demonstrated in the previous section, GMMNs best fit the underlying dependence structure of the data. Secondly, unlike many parametric copulas, we can generate quasi-random samples independently of the type of dependence structure observed in the data. Finally, we can generate GMMN quasi-random samples at no additional cost over GMMN pseudo-random samples; see also Appendix C.2.

4.6 Discussion

This work has been inspired by the simple question of how to obtain quasi-random samples for a large variety of multivariate distributions. Until recently, this was only possible for a few multivariate distributions with specific underlying copulas. In general, for the vast majority of multivariate distributions, obtaining quasi-random samples is a hard problem (Cambou et al., 2017). Our approach based on GMMNs provides a first universal method for doing so. It depends on first learning a generator $f_{\hat{\boldsymbol{\theta}}}$ such that, given $\boldsymbol{Z}$ (with independent components from some known distribution such as the standard uniform or standard normal), $f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z})$ follows the targeted multivariate distribution. Conditional on this first step being successful, we can then replace $\boldsymbol{Z}$ with $F^{-1}_{\boldsymbol{Z}}(\tilde{\boldsymbol{v}}_i)$, $i = 1, \dots, n_{\text{gen}}$, where $\{\tilde{\boldsymbol{v}}_1, \dots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$ is an RQMC point set, to generate quasi-random samples from $\boldsymbol{X}$.

It is generally difficult to assess the low-discrepancy property of non-uniform quasi-random samples. To evaluate the quality of our GMMN quasi-random samples, we used

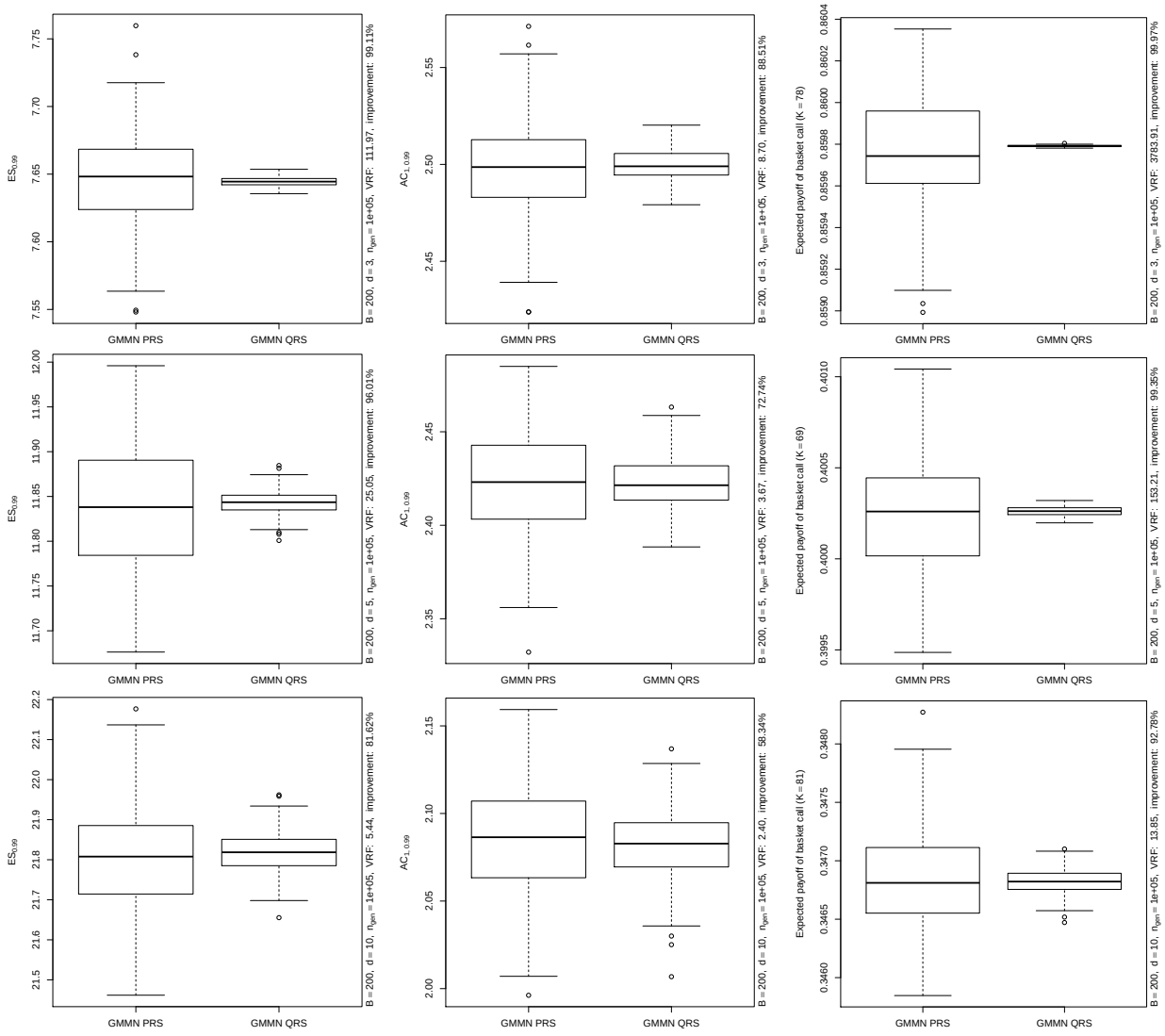

Figure 4.14: Box plots based on $B = 200$ realizations of the GMMN MC estimator $\hat{\mu}^{\mathrm{NN,MC}}_{n_{\mathrm{gen}}}$ (label "GMMN PRS") and the GMMN RQMC estimator $\hat{\mu}^{\mathrm{NN}}_{n_{\mathrm{gen}}}$ (label "GMMN QRS") of $\mathrm{ES}_{0.99}$ (left column), $\mathrm{AC}_{1,0.99}$ (middle column) and the expected payoff of a basket call (right column) for portfolios of dimensions $d = 3$ (top row), $d = 5$ (middle row) and $d = 10$ (bottom), using $n_{\mathrm{gen}} = 10^5$ samples for both estimators.

visualization tools (Section 4.3.2), goodness-of-fit statistics (Section 4.3.3), and investigated variance reduction effects (Section 4.4) when estimating $\mu = \mathbb{E}(\Psi(\boldsymbol{X}))$ for a test function and for expected shortfall. As dependence structures among the components of $\boldsymbol{X}$, we included various known copulas, some of which allowed for quasi-random sampling which allowed us to statistically assess the performance of our GMMN quasi-random samples. However, we emphasize that the key feature of our method is that, given a sufficiently large dataset with dependence structure not well described by any known parametric copula model for which quasi-random sampling is available, we are now able to generate quasi-random samples from its empirical distribution. We demonstrated this with a real dataset in Section 4.5. Not only does a GMMN provide the best fitting model in this application, allowing us to avoid the tedious and often computationally challenging search that is typically required in classical copula modeling for an adequate dependence model, we also obtain, at no additional cost, quasi-random samples from this GMMN — a whole other challenge in classical copula modeling. This universality and computability is an attractive feature of GMMNs for multivariate modeling.

However, this does not mean that the problem of quasi-random sampling for multivariate distributions is completely solved. In high dimensions learning an entire distribution is a hard problem, and so is learning the generator $f_{\hat{\boldsymbol{\theta}}}$. At a superficial level, the literature on generative NNs — and the many headlines covering them — may give the impression that such NNs are now capable of generating samples from very high-dimensional distributions. This, of course, is not true; see, for example, Arjovsky et al. (2017), Tolstikhin et al. (2017), or Arora et al. (2018). In particular, while available evidence is convincing that any *specific* generated sample $f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_1)$, typically an image, can be very realistic in the sense that it looks just like a typical training sample, this is not the same as saying that the *entire collection* of generated samples $\{f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_1), f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{Z}_2), \dots\}$ will have the same distribution as the training sample. The latter is a much harder problem to solve. Unlike widely cited generative NNs such as variational autoencoders and generative adversarial networks, GMMNs are capable of learning entire distributions, because they rely on the MMD-criterion as the loss function rather than, for example, the mean squared error which does not measure the discrepancy between entire distributions. Even so, this still does not mean GMMNs are practical for very high dimensions yet, simply because the fundamental curse of dimensionality cannot be avoided easily. At the moment, it is simply not realistic to hope that one can learn an entire distribution in high dimensions from a training sample of only moderate size.

Going forward there are two primary impediments to quasi-random sampling from higher-dimensional copulas and distributions. Firstly, the problem of distribution learning via generative NNs remains a challenging task. We may also consider using other goodness-of-fit statistics for multivariate distributions rather than the MMD as the loss function

(provided that the statistic is differentiable in order to train a generative NN). Secondly, we discovered from our empirical investigation in Section 4.4 that the convergence rates of GMMN RQMC estimators decrease with increasing dimension. Preserving the low-discrepancy of RQMC point sets upon transformations in high dimensions remains an open problem in this regard.

Chapter 5

Multivariate time-series modeling with generative neural networks

5.1 Introduction

The task of modeling multivariate time series (MTS) arises in a variety of applications in finance, economics and quantitative risk management. In many situations, a suitable model arises from breaking down this task into two key components: the modeling of serial dependence within each univariate time series and the modeling of cross-sectional dependence between the individual time series. There is a plethora of literature on univariate time series modeling with a wide range of models that are tailor-made for capturing various types of serial patterns such as seasonality, volatility clustering or regime switching. In the realm of financial econometrics, the class of generalized auto-regressive conditional heteroscedasticity (GARCH) models (Bollerslev, 1986) is a popular choice. GARCH-type models are designed to account for stylized facts (such as volatility clustering) that are often present in financial return series data; see McNeil et al. (2015, Chapter 3).

There have been numerous approaches proposed for extending univariate time series modeling approaches to the multivariate case. Within the broad GARCH framework, Bollerslev (1990) initially introduced a multivariate model characterized by the distributional assumption of multivariate normality with a constant conditional correlation structure. Dynamic conditional correlation (DCC)-GARCH models were then introduced by Engle (2002) and Tse and Tsui (2002). DCC-GARCH models relax the conditional correlation assumption but still utilize multivariate normal distributions to model the cross-sectional dependence between the univariate time series. Leveraging Sklar's Theorem (Sklar, 1959),

Jondeau and Rockinger (2006) and Patton (2006) presented a flexible family of multivariate GARCH models where the assumption of multivariate normality has been relaxed to allow for any copula of the joint innovation distribution. This popular modeling approach for MTS data is known as the *copula–GARCH* approach; see Patton (2012) for a brief overview in the context of finance and econometrics. It allows us to flexibly model joint innovation distributions with copulas, thereby decomposing the MTS modeling task into modeling of the (univariate) marginal time series and their cross-sectional dependence. There have been various research papers investigating the calibration of copula–GARCH models, for example the more recent work of Aas (2016), Almeida et al. (2016) or Oh and Patton (2017).

While there is a growing collection of copula models used to characterize complex dependence structures, most models are rather limited already in moderately large dimensions and often do not provide an adequate fit to given data (see, for example, Hofert and Oldford (2018)) or require sophisticated, model-specific algorithms for parameter estimation and model selection. In this chapter, we propose a framework for MTS modeling in which a classical copula model to account for cross-sectional dependence is replaced by a generative moment matching network (GMMN). In comparison to classical copulas, GMMNs can capture a large variety of complex dependence structures. For high-dimensional time series data, we incorporate principal component analysis (PCA) as an intermediate step to reduce the dimensionality. Our primary goal is to construct empirical predictive distributions, also known as probabilistic forecasts, rather than point forecasts. Additionally, these empirical predictive distributions can be utilized to further forecast various quantities of interest (e.g., quantiles) via simulation.

The chapter is organized as follows. In Section 5.2, we outline our framework for modeling MTS data. In particular, we focus on the novel integration of GMMNs within this framework. In Section 5.3, we showcase our GMMN-based multivariate time series models in applications to yield curve and exchange rate data. Section 5.4 provides concluding remarks. All results in this chapter can be reproduced with the demo `GMMN_MTS_paper` in the R package `gnn` (version 0.0-3).

5.2 A framework for multivariate time series modeling

Let $(\boldsymbol{X}_t)_{t\in\mathbb{Z}}$ denote a d-dimensional time series of interest, where $\boldsymbol{X}_t = (X_{t,1}, \dots, X_{t,d})$. Furthermore, consider a stretch of τ realizations from $(\boldsymbol{X}_t)_{t\in\mathbb{Z}}$ denoted by $\boldsymbol{X}_1, \dots, \boldsymbol{X}_\tau$. In applications in finance (risk management), these are often log-returns (negative log-returns)

of d asset prices; see Section 5.3 for more details and the pre-processing steps applied to each empirical dataset we consider.

Our suggested framework for modeling $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_\tau$ consists of three primary components:

1. marginal time series modeling — while many possibilities can be considered, we focus on ARMA–GARCH models;
2. dimension reduction — again, many tools are available, but we simply utilize PCA; and
3. dependence modeling — here, the typical approach is to choose a parametric copula, but we introduce the use of GMMNs, the main contribution of this chapter.

While Step 1 and Step 3 are essential, the dimension reduction component in Step 2 is optional and typically only used for high-dimensional time series which are amenable to good approximations by lower-dimensional representations.

5.2.1 Marginal time series modeling

The *ARMA–GARCH models* in Step 1 are ARMA models with GARCH errors; see McNeil et al. (2015, Section 4.2.3). These ARMA(p_{1j}, q_{1j})–GARCH(p_{2j}, q_{2j}) models have the form

$$\begin{aligned} X_{t,j} &= \mu_{t,j} + \sigma_{t,j} Z_{t,j}, \\ \mu_{t,j} &= \mu_j + \sum_{k=1}^{p_{1j}} \phi_{jk}(X_{t-k,j} - \mu_j) + \sum_{l=1}^{q_{1j}} \gamma_{jl}(X_{t-l,j} - \mu_{t-l,j}), \\ \sigma_{t,j}^2 &= \omega_j + \sum_{k=1}^{p_{2j}} \alpha_{jk}(X_{t-k,j} - \mu_{t-k,j})^2 + \sum_{l=1}^{q_{2j}} \beta_{jl}\sigma_{t-l,j}^2, \end{aligned}$$

where, for each component $j = 1, \ldots, d$, one has $\mu_j \in \mathbb{R}$, $\omega_j > 0$, and $\alpha_{jk}, \beta_{jl} \geq 0$ for all k, l. Some additional conditions on the coefficients ϕ_{jk}, γ_{jl}, α_{jk} and β_{jl} are necessary to ensure that all ARMA- and GARCH-processes are respectively causal and covariance stationary; for example see McNeil et al. (2015, Section 4.1.2–4.2.2).

For each $j = 1, \ldots, d$, the *innovations* $Z_{t,j}$ in the definition of the ARMA–GARCH model are independent and identically distributed (iid) random variables with $\mathbb{E}(Z_{t,j}) = 0$ and $\mathrm{Var}(Z_{t,j}) = 1$; their realizations after fitting marginal ARMA(p_{1j}, q_{1j})–GARCH(p_{2j}, q_{2j})

models are known as *standardized residuals* and denoted by $\hat{Z}_{t,j}$, $t = 1, \ldots, \tau$ and $j = 1, \ldots, d$. In financial time series applications, common choices of innovation distributions include the standard normal, the scaled t and the skewed t distribution.

Fitting the marginal time series models is typically done by fitting low-order models with likelihood-based methods and selecting the most adequate fit using the AIC/BIC model selection criterion among the candidate models. A popular broad-brush approach is to fit a GARCH$(1, 1)$ model for financial return series — specifically, an ARMA$(0, 0)$–GARCH$(1, 1)$ model in our context — and continue the modeling based on the standardized residuals $\hat{Z}_{1,j}, \ldots, \hat{Z}_{\tau,j}$; see McNeil et al. (2015, Chapter 4) or Hofert et al. (2018b, Section 6.2.3). This procedure is also referred to as *deGARCHing*. With the help of model diagnostic tools — for example, plots of the autocorrelation function (ACF) of $\hat{Z}_{1,j}, \ldots, \hat{Z}_{\tau,j}$ and that of their squared values, Ljung–Box tests or assessment of the innovation distribution through Q-Q plots — one can then assess the adequacy of each marginal time series model. In what follows we use $\hat{\mu}_{t,j}$ and $\hat{\sigma}^2_{t,j}$ to denote the estimated conditional mean and variance models for the jth marginal time series with corresponding chosen orders $\hat{p}_{1j}, \hat{q}_{1j}, \hat{p}_{2j}, \hat{q}_{2j}$ and fitted parameters $\hat{\phi}_{jk}, \hat{\gamma}_{jl}, \hat{\alpha}_{jk}, \hat{\beta}_{jl}$.

Having accounted for the marginal serial dependence in this way, the subsequent analysis in our modeling framework will operate on the standardized residuals $\hat{\boldsymbol{Z}}_t = (\hat{Z}_{t,1}, \ldots, \hat{Z}_{t,d})$, $t = 1, \ldots, \tau$, which are themselves realizations of the innovation random variables, $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_\tau$, assumed to be iid in the copula–GARCH approach.

Before we continue, we emphasize once again that any other adequate marginal time series modeling approach can be applied in our framework as long as the model's residuals can be considered to be iid from continuous marginal distributions. Our choice of ARMA–GARCH models is motivated only from the fact that these are the most popular marginal time series models used in practice.

5.2.2 Dimension reduction

Two popular dimension-reduction techniques for multivariate financial time series are factor models and PCA; see McNeil et al. (2015, Chapter 6) and the references therein for a brief summary. An approach that is perhaps less discussed in the financial econometrics literature involves using autoencoder neural networks for dimension reduction in which two separate neural network mappings are learned to and from the lower dimensional space; see Hinton and Salakhutdinov (2006). As dimension reduction is not our main contribution in this chapter, we simply utilize PCA in what follows.

Note that PCA is often applied to the original MTS data $\boldsymbol{X}_t$ in the literature; see, e.g., Alexander (2000) for an investigation of the so-called orthogonal GARCH model. Apart from reducing the burden of marginal time series modeling, there is no strong reason why PCA should be applied to potentially non-stationary data. If dimension reduction is necessary, we find it statistically more sound to apply PCA to the standardized residuals $\hat{\boldsymbol{Z}}_t$ *after* first accounting for any serial dependence in the marginal time series.

Let $\hat{\Sigma}$ denote the sample covariance matrix of the standardized residuals $\hat{\boldsymbol{Z}}_t$, $t = 1, \ldots, \tau$. The result from PCA is the matrix $\hat{\Gamma} \in \mathbb{R}^{d \times d}$ whose columns consist of the eigenvectors of $\hat{\Sigma}$, sorted according to decreasing eigenvalues $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_d \geq 0$. For the purposes of dimension reduction, $\hat{\boldsymbol{Z}}_t$, $t = 1, \ldots, \tau$, are transformed to $\hat{\boldsymbol{Y}}_t = \hat{\Gamma}_{\cdot,1:k}^\top \hat{\boldsymbol{Z}}_t$, where $\hat{\Gamma}_{\cdot,1:k} \in \mathbb{R}^{d \times k}$ represent the first k columns of $\hat{\Gamma}$ for some $1 \leq k < d$. As a result, the sample covariance matrix of $\boldsymbol{Y}_t$ is (approximately) diagonal, and the components of $\boldsymbol{Y}_t$ are (approximately) uncorrelated. The jth component series $Y_{t,j}$, $t = 1, \ldots, \tau$, forms realizations of the jth principal component, and the first k principal component series account for $\sum_{j=1}^k \hat{\lambda}_j / \sum_{j=1}^d \hat{\lambda}_j$ of the total variance.

As dimension reduction is an optional component in our modeling framework, the next step involves dependence modeling of either the standardized residuals $\hat{\boldsymbol{Z}}_1, \ldots, \hat{\boldsymbol{Z}}_\tau$ or their principal components $\hat{\boldsymbol{Y}}_1, \ldots, \hat{\boldsymbol{Y}}_\tau$. To unify the notation going forward, we define a d^*-dimensional time series $\hat{\boldsymbol{Y}}_t = \hat{\Upsilon}^\top \hat{\boldsymbol{Z}}_t$, where $\hat{\Upsilon} = \hat{\Gamma}_{\cdot,1:k}$ if dimension reduction is employed and $\hat{\Upsilon} = I_d$ (the identity matrix in $\mathbb{R}^{d \times d}$) otherwise; consequently, $d^* = k$ in the former case and $d^* = d$ in the latter. Furthermore, we treat $\hat{\boldsymbol{Y}}_1, \ldots, \hat{\boldsymbol{Y}}_\tau$ as realizations from $\boldsymbol{Y}_t$, where, naturally, $\boldsymbol{Y}_t = \Upsilon^\top \boldsymbol{Z}_t$ with $\Upsilon = \Gamma_{\cdot,1:k}$ if dimension reduction is used and $\Upsilon = I_d$ otherwise.

5.2.3 Dependence modeling

The final task in our framework involves the modeling of the iid series $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_\tau$. To account for cross-sectional dependence, we model the joint distribution function H of $\boldsymbol{Y}_t$ using Sklar's Theorem as

$$H(\boldsymbol{y}) = C(F_1(y_1), \ldots, F_{d^*}(y_{d^*})), \quad \boldsymbol{y} \in \mathbb{R}^{d^*},$$

where F_j, $j = 1, \ldots, d^*$, are the margins of H and $C : [0,1]^{d^*} \to [0,1]$ is the copula of $(Y_{t,1}, \ldots, Y_{t,d^*})$ for each t.

Following a classical copula modeling approach, one first builds the *pseudo-observations* $\hat{U}_{t,j} = R_{t,j}/(\tau + 1)$, $t = 1, \ldots, \tau$, $j = 1, \ldots, d^*$, where $R_{t,j}$ denotes the rank of $\hat{Y}_{t,j}$ among $\hat{Y}_{1,j} \ldots, \hat{Y}_{\tau,j}$. The pseudo-observations are viewed as realizations from C based on which

one would fit candidate copula models; see, for example, McNeil et al. (2015, Section 7.5.1) or Hofert et al. (2018b, Section 4.1.2). Note that by considering (non-parametric) pseudo-observations (even in the case when we do not apply a dimension reduction technique and thus know the (fitted) marginal innovation distributions), we reduce the risk of misspecifying one of the margins affecting the estimation of the copula C; see Genest and Segers (2010) for a theoretical justification of this approach. Therefore, going forward, we will use the pseudo-observations $\hat{\boldsymbol{U}}_t = (\hat{U}_{t,1}, \dots, \hat{U}_{t,d^*})$, $t = 1, \dots, \tau$, to model the cross-sectional dependence structure of $\hat{\boldsymbol{Y}}_t$.

Dependence modeling with parametric copulas

A traditional approach for modeling the cross-sectional dependence described by $\hat{\boldsymbol{U}}_1, \dots, \hat{\boldsymbol{U}}_\tau$ involves the fitting of parametric copula models, their goodness-of-fit assessment and finally, model selection. There are numerous families of copula models to consider depending on prominent features of the dependence structure present in $\hat{\boldsymbol{U}}_t$ such as (a)symmetries or a concentration of points in the lower/upper tail of the joint distribution (or pairs of such) which hints at an adequate model possessing tail dependence.

A problem with this approach is that it is often hard to find an adequate copula model for given real-life data, especially in higher dimensions where typically some pairwise dependencies contradict the corresponding model-implied marginal copulas; see, for example, Hofert and Oldford (2018). Another problem is that certain copula models are computationally expensive to fit and test for goodness-of-fit. In Section 5.3, we investigate whether (the much more flexible) GMMNs can outperform prominent elliptical and Archimedean copulas in the context of our framework. In what follows we thus shall denote by $\hat{C}_{\text{PM}}$ a (generic) parametric copula model fitted to the pseudo-observations $\hat{\boldsymbol{U}}_1, \dots, \hat{\boldsymbol{U}}_\tau$.

Dependence modeling with GMMNs

We propose to utilize generative neural networks (in particular, GMMNs) for modeling the cross-sectional dependence structure of the pseudo-observations $\hat{\boldsymbol{U}}_1, \dots, \hat{\boldsymbol{U}}_\tau$. In our framework, a generative neural network $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ learns the distribution of the pseudo-observations. Let $\hat{C}_{\text{NN}}$ denote the empirical copula based on a sample generated from a trained GMMN $f_{\hat{\boldsymbol{\theta}}}$.

As introduced in Sections 4.2.1 and 4.2.2, we work with GMMNs constructed using feed-forward neural networks. In the context of our MTS modeling framework, let $\hat{\boldsymbol{U}}_1, \dots, \hat{\boldsymbol{U}}_\tau$ denote the training data. Then, given a sample $\boldsymbol{V}_1, \dots, \boldsymbol{V}_{n_{\text{gen}}}$ from a p-dimensional input

distribution with independent components $F_{\boldsymbol{V}}$, the GMMN generates an output sample $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_{n_{\text{gen}}}$, where $\boldsymbol{U}_t = f_{\boldsymbol{\theta}}(\boldsymbol{V}_t)$, $t = 1, \ldots, n_{\text{gen}}$. For notational convenience, let us stack $\hat{\boldsymbol{U}}_1, \ldots, \hat{\boldsymbol{U}}_\tau$ into an $\tau \times d^*$ matrix $\hat{U}$ and likewise $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_{n_{\text{gen}}}$ into an $n_{\text{gen}} \times d^*$ matrix U.

To train the GMMN $f_{\boldsymbol{\theta}}$, we thus perform the optimization

$$\min_{\boldsymbol{\theta}} \ \text{MMD}(\hat{U}, (f_{\boldsymbol{\theta}}(V)),$$

where the MMD statistic is computed as defined in (4.6), the $n_{\text{gen}} \times p$ matrix V is obtained by stacking $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_{n_{\text{gen}}}$, and the NN transform $f_{\boldsymbol{\theta}}$ is understood to be applied row-wise. As discussed in Section 4.3.1, we work with a mixture of n_{krn} Gaussian kernels with different bandwidth parameters as our kernel function for the MMD statistic. Furthermore, we always simply set $n_{\text{gen}} = \tau$ when training the GMMN for sake of convenience. For details regarding the training procedure, see Algorithm 4.2.1. The resulting trained GMMN is denoted by $f_{\hat{\boldsymbol{\theta}}}$.

5.2.4 Simulating paths of dependent multivariate time series

After utilizing our framework for modeling multivariate time series, a typical next step is to simulate paths from the fitted/trained multivariate model. With these simulated paths we immediately obtain *empirical predictive distributions* at future time points. Additionally, we can forecast quantities of interest such as (confidence) intervals or risk-measures (for example value-at-risk or expected shortfall) based on the simulated paths. Some of these quantities will be discussed further in Section 5.3. In this section, we focus on how to simulate the required paths in our framework.

To fix ideas, suppose we are interested in future time points, $\tau + 1, \tau + 2, \ldots, T$. Furthermore, let $h \leq T - \tau$ denote the simulation horizon. Then, for every $t = \tau, \ldots, T - h$, once all realizations up to and including time t — namely, the entire sequence $(\boldsymbol{X}_s)_{s \leq t}$ — become available, we can simulate multiple paths,

$$\{\hat{\boldsymbol{X}}_{t+1}^{(i)}, \hat{\boldsymbol{X}}_{t+2}^{(i)}, \ldots, \hat{\boldsymbol{X}}_{t+h}^{(i)}\}_{i=1}^{n_{\text{pth}}},$$

going forward for a total of h time periods.

A key component for simulating these paths is the generation of samples from the estimated dependence model. For fitted parametric copulas $\hat{C}_{\text{PM}}$, one typically uses a model-specific stochastic representation to sample $\boldsymbol{U}_t$; see, for example, Hofert et al. (2018b, Chapter 3). Sampling from the fitted GMMN $f_{\hat{\boldsymbol{\theta}}}$ (with corresponding empirical copula $\hat{C}_{\text{NN}}$) can be done as follows.

Algorithm 5.2.1 (GMMN sampling)

1. Fix the number n_{gen} of samples to generate from $\hat{C}_{\text{NN}}$.
2. Draw $\boldsymbol{V}_1, \dots, \boldsymbol{V}_{n_{\text{gen}}} \overset{\text{ind.}}{\sim} F_{\boldsymbol{V}}$ from the input distribution.
3. Return $\boldsymbol{U}_s = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{V}_s)$, $s = 1, \dots, n_{\text{gen}}$.

Since copulas have $\mathrm{U}(0,1)$ margins, we typically equip Algorithm 5.2.1 with a post-processing step by returning the pseudo-observations based on $\boldsymbol{U}_1, \dots, \boldsymbol{U}_{n_{\text{gen}}}$ to remove any residual marginal non-uniformity from the GMMN samples.

For any given $t = \tau, \dots, T-h$, we can now utilize Algorithm 5.2.1 along with the fitted marginal time series models in our framework in order to simulate multiple paths $\{\hat{\boldsymbol{X}}_{t+1}^{(i)}, \hat{\boldsymbol{X}}_{t+2}^{(i)}, \dots, \hat{\boldsymbol{X}}_{t+h}^{(i)}\}_{i=1}^{n_{\text{pth}}}$ with a fixed simulation horizon h, as outlined in Algorithm 5.2.2 below.

Algorithm 5.2.2 (Simulating paths of dependent multivariate time series via GMMNs)

1. Fix the number of sample paths n_{pth} and the simulation horizon h.
2. For $t = \tau, \dots, T-h$ do:
   (a) Generate $\boldsymbol{U}_s^{(i)}$, $i = 1, \dots, n_{\text{pth}}$, $s = t+1, \dots, t+h$, from the fitted GMMN $\hat{C}_{\text{NN}}$ via Algorithm 5.2.1.
   (b) For every $\boldsymbol{U}_s^{(i)}$ in Step 2a, construct $\boldsymbol{Y}_s^{(i)} = (\hat{F}_1^{-1}(U_{s,1}^{(i)}), \dots, \hat{F}_{d^*}^{-1}(U_{s,d^*}^{(i)}))$. If no dimension reduction is utilized, the marginals $\hat{F}_j$, $j = 1, \dots, d^*$, are the fitted parametric innovation distributions selected as part of the ARMA–GARCH model setup; otherwise, they are the empirical distribution functions of $\hat{Y}_{1,j}, ..., \hat{Y}_{\tau,j}$, $j = 1, \dots, d^*$.
   (c) For every $\boldsymbol{Y}_s^{(i)}$ in Step 2b, construct samples from the fitted innovation distributions via the transform $\boldsymbol{Z}_s^{(i)} = \hat{\Upsilon}\boldsymbol{Y}_s^{(i)}$. (Note that $\boldsymbol{Y}_s^{(i)} \in \mathbb{R}^{d^*}$ whereas $\boldsymbol{Z}_s^{(i)} \in \mathbb{R}^d$.)
   (d) For each $j = 1, \dots, d$, compute $\hat{\sigma}_{s,j}^{2(i)}$, $\hat{\mu}_{s,j}^{(i)}$ and $\hat{X}_{s,j}^{(i)}$, for $i = 1, \dots, n_{\text{pth}}$ and

$s = t+1, \ldots, t+h$, via

$$\hat{\sigma}_{s,j}^{2^{(i)}} = \hat{\omega} + \sum_{k=1}^{\hat{p}_{2j}} \hat{\alpha}_{jk} (\hat{X}_{s-k,j}^{(i)} - \hat{\mu}_{s-k,j}^{(i)})^2 + \sum_{l=1}^{\hat{q}_{2j}} \hat{\beta}_{jl} \hat{\sigma}_{s-l,j}^{2^{(i)}},$$

$$\hat{\mu}_{s,j}^{(i)} = \hat{\mu}_j + \sum_{k=1}^{\hat{p}_{1j}} \hat{\phi}_{jk} (\hat{X}_{s-k,j}^{(i)} - \hat{\mu}_j) + \sum_{l=1}^{\hat{q}_{1j}} \hat{\gamma}_{jl} (\hat{X}_{s-l,j}^{(i)} - \hat{\mu}_{s-l,j}^{(i)}),$$

$$\hat{X}_{s,j}^{(i)} = \hat{\mu}_{s,j}^{(i)} + \hat{\sigma}_{s,j}^{2^{(i)}} Z_{s,j}^{(i)},$$

where, for $s \leq t$, set $\hat{X}_{s,j}^{(i)} = X_{s,j}$, $\hat{\sigma}_{s,j}^{2^{(i)}} = \hat{\sigma}_{s,j}^2$, and $\hat{\mu}_{s,j}^{(i)} = \hat{\mu}_{s,j}$ for all $i = 1, \ldots, n_{\text{pth}}$.

(e) Return $\hat{\boldsymbol{X}}_s^{(i)} = (\hat{X}_{s,1}^{(i)}, \ldots, \hat{X}_{s,d}^{(i)})$, $i = 1, \ldots, n_{\text{pth}}$, $s = t+1, \ldots, t+h$.

Note that Step 2a in Algorithm 5.2.2 can be replaced by sampling from the fitted parametric copula $\hat{C}_{\text{PM}}$ to obtain the classically applied approach for sampling paths in the copula–GARCH framework.

While Algorithm 5.2.2 describes how to simulate paths of multivariate time series for any simulation horizon h, we will focus on one-period-ahead ($h = 1$) empirical predictive distributions henceforth.

5.2.5 Assessing the quality of predictions of dependent multivariate time series models

In this section, we discuss the metrics we will use in all numerical investigations in this chapter to assess and compare various MTS models. Of particular interest is the comparison of GMMN–GARCH and copula–GARCH models. In practice, to assess the out-of-sample performance of our models, realizations of time series will naturally be divided into separate training and test datasets. To that end, suppose that we have realizations $(\boldsymbol{X}_t)_{t \in \mathcal{T}}$ from the *test period* $\mathcal{T} = \{\tau+1, \ldots, T\}$ that have been set aside (i.e., not used for training) as a separate test set.

Assessing the quality of dependence models in the test period

We can use the MMD statistic to measure how close the empirical distributions of a fitted GMMN $\hat{C}_{\text{NN}}$ and a fitted parametric copula $\hat{C}_{\text{PM}}$ match the cross-sectional dependence structure of the test set, $(\boldsymbol{X}_t)_{t \in \mathcal{T}}$. This cross-sectional dependence structure can be extracted

using the fitted (marginal) ARMA–GARCH models and the fitted PCA models (if dimension reduction is applied), as described in the following algorithm.

Algorithm 5.2.3 (Extracting underlying dependence structure of the test data set)

1. Compute $\hat{\sigma}_{t,j}^2$, $\hat{\mu}_{t,j}$ and $\hat{Z}_{t,j}$ for $t \in \mathcal{T}$ and $j = 1, \dots, d$ via

$$\begin{aligned}
\hat{\sigma}_{t,j}^2 &= \hat{\omega} + \sum_{k=1}^{\hat{p}_{2j}} \hat{\alpha}_{jk}(X_{t-k,j} - \hat{\mu}_{t-k,j})^2 + \sum_{l=1}^{\hat{q}_{2j}} \hat{\beta}_{jl}\hat{\sigma}_{t-l,j}^2, \\
\hat{\mu}_{t,j} &= \hat{\mu}_j + \sum_{k=1}^{\hat{p}_{1j}} \hat{\phi}_{jk}(X_{t-k,j} - \hat{\mu}_j) + \sum_{l=1}^{\hat{q}_{1j}} \hat{\gamma}_{jl}(X_{t-l,j} - \hat{\mu}_{t-l,j}), \\
\hat{Z}_{t,j} &= \frac{X_{t,j} - \hat{\mu}_{t,j}}{\hat{\sigma}_{t,j}}.
\end{aligned}$$

2. Obtain a sample from the underlying empirical stationary distribution via the transform $\hat{\boldsymbol{Y}}_t = \hat{\Upsilon}^\top \hat{\boldsymbol{Z}}_t$, $t \in \mathcal{T}$. (Note that $\hat{\boldsymbol{Z}}_t \in \mathbb{R}^d$ whereas $\hat{\boldsymbol{Y}}_t \in \mathbb{R}^{d^*}$.)

3. Return the pseudo-observations $\hat{\boldsymbol{U}}_t = (\hat{U}_{t,1}, \dots, \hat{U}_{t,d^*})$ of $\hat{\boldsymbol{Y}}_t$, for $t \in \mathcal{T}$.

Let us stack the pseudo-observations $\hat{\boldsymbol{U}}_{\tau+1}, \dots, \hat{\boldsymbol{U}}_T$ obtained from the test dataset via Algorithm 5.2.3 into an $(T - \tau) \times d^*$ matrix $\hat{U}$. Similarly, let U denote an $n_{\text{gen}} \times d^*$ matrix consisting of a sample generated from either $\hat{C}_{\text{NN}}$ or $\hat{C}_{\text{PM}}$, where we choose $n_{\text{gen}} = T - \tau$ (other choices are possible). We can then compute one realization of the MMD statistic $\text{MMD}(\hat{U}, U)$ as defined in (4.6). In our analysis in Section 5.3, we use an average MMD statistic based on n_{rep} repeated samples $U^{(i)} \in [0,1]^{n_{\text{gen}} \times d^*}$, $i = 1, \dots, n_{\text{rep}}$, given by

$$\text{AMMD} = \frac{1}{n_{\text{rep}}} \sum_{i=1}^{n_{\text{rep}}} \text{MMD}(\hat{U}, U^{(i)}). \tag{5.1}$$

For the MMD statistic, we use a mixture of $n_{\text{krn}} = 5$ Gaussian kernels with bandwidth parameters $\boldsymbol{\sigma} = (0.1, 0.3, 0.5, 0.7, 0.9)$. These bandwidth parameters are purposefully chosen to be different from the bandwidth parameters $\boldsymbol{\sigma}$ used in Section 5.3 below for the GMMN training procedure, to allow for a fairer out-of-sample assessment.

Assessing the quality of an empirical predictive distribution

While there exist numerous metrics to assess univariate or multivariate point forecasts, there are only a handful of metrics that can be utilized to evaluate the quality of dependent

multivariate empirical predictive distributions. We now present two such metrics we will use across all numerical examples.

Firstly, we use a version of the *mean squared error (MSE)* metric defined via the Euclidean norm to assess how well the empirical predictive distribution $\{\hat{\boldsymbol{X}}_t^{(i)} : i = 1, \dots, n_{\text{pth}}\}$ concentrates around each true value $\boldsymbol{X}_t$ in the test set, so for $t \in \mathcal{T}$. To obtain a single numerical value, we work with an average MSE metric computed over the entire test period $t \in \mathcal{T}$, defined by

$$\text{AMSE} = \frac{1}{T-\tau} \sum_{t=\tau+1}^{T} \left(\frac{1}{n_{\text{pth}}} \sum_{i=1}^{n_{\text{pth}}} \|\hat{\boldsymbol{X}}_t^{(i)} - \boldsymbol{X}_t\|_2^2 \right). \tag{5.2}$$

Secondly, we use the *variogram score* introduced by Scheuerer and Hamill (2015), which, in our context, assesses if the empirical predictive distribution is biased for the distance between any two component samples. For a single numeric summary, we work with an average variogram score (of order p) over the entire test period $t \in \mathcal{T}$,

$$\text{AVS}^p = \frac{1}{T-\tau} \sum_{t=\tau+1}^{T} \left(\sum_{j_1=1}^{d} \sum_{j_2=1}^{d} \left(|X_{t,j_1} - X_{t,j_2}|^p - \frac{1}{n_{\text{pth}}} \sum_{i=1}^{n_{\text{pth}}} |\hat{X}_{t,j_1}^{(i)} - \hat{X}_{t,j_2}^{(i)}|^p \right)^2 \right). \tag{5.3}$$

As numerically demonstrated by Scheuerer and Hamill (2015), by focusing on pairwise distances between component samples, this metric discriminates well between various dependence structures. Scheuerer and Hamill (2015) stated that a typical choice of the variogram order might be $p = 0.5$, but they also note in their concluding remarks that smaller values of p could potentially yield more discriminative metrics when dealing with non-Gaussian data which is why we choose to work with $p = 0.25$.

5.3 Applications

In this section, we demonstrate the flexibility of our GMMN–GARCH models when compared to copula–GARCH models. To that end, we focus on modeling multivariate yield curve and exchange rate time series.

Before delving into the two financial econometrics applications, we will first detail the selection and setup of component models within our framework that will be utilized for all examples in this section. Specifically, we will describe the choice of marginal time series models, the implementation details for GMMN models, and the choice of parametric copula models used for comparison.

Furthermore, note that all examples considered in this section were implemented in R and can be reproduced using the demo `GMMN_MTS_paper` in the R package `gnn`. As in Chapter 4, the R packages `keras` and `tensorflow` were used as R interfaces to the corresponding namesake Python libraries. All GMMN training was carried out on a single NVIDIA Tesla P100 GPU with 12GB RAM.

5.3.1 Multivariate time series modeling: setup and implementation details

Marginal models

For modeling the marginal time series, we take a broad-brush approach and choose to fit ARMA(1,1)–GARCH(1,1) models with scaled t innovation distributions $F_j(z_j) = t_{\nu_j}(z_j\sqrt{\nu_j/(\nu_j-2)})$, $j = 1, \ldots, d$, to each component sample. As mentioned earlier, these models are popular choices for modeling univariate financial time series. To fit them, we use the `fit_ARMA_GARCH(, solver="hybrid")` function from the R package `qrmtools` which relies on the `ugarchfit()` function from the R package `rugarch` (Ghalanos, 2019).

Dependence models: GMMN architecture and training setup

Taking into consideration that we are working with relatively small number of realizations of time series data in both applications, we find that a single hidden layer architecture ($L = 1$) provides sufficient flexibility. Given the single hidden layer, we experiment with three NN architectures with $d_1 = 100$ (*GMMN model 1*), $d_1 = 300$ (*GMMN model 2*) and $d_1 = 600$ (*GMMN model 3*), respectively, for all examples in this section. As in Chapter 4, we fix ϕ_1 to be ReLU since it offers computational efficiency via non-expensive and non-vanishing gradients and ϕ_2 to be sigmoid given that our target output lies in $[0,1]^{d^*}$.

As mentioned earlier in Section 5.2.3, we utilize a mixture of Gaussian kernels for the MMD statistic in (4.6). Following the setup in Section 4.3.1, we fix $n_{\text{krn}} = 6$ and choose $(\sigma_1, \ldots, \sigma_6) = (0.001, 0.01, 0.15, 0.25, 0.50, 0.75)$ as the bandwidth parameters. This hyperparameter setting is specifically suited for copula samples or pseudo-observations as they lie in $[0,1]^{d^*}$. Furthermore, as we demonstrated in Chapter 4, GMMNs trained with this particular specification of the loss function are capable of learning a wide variety of complex dependence structures.

We choose the dimension of the input distribution $F_{\boldsymbol{V}}$ to be $p = d^*$. As a result we obtain a natural d^*-to-d^* GMMN transform $f_{\boldsymbol{\theta}}$. Following common practice, we select

$\boldsymbol{V} \sim \mathrm{N}(\mathbf{0}, I_{d^*})$, where I_{d^*} denotes the identity matrix in $\mathbb{R}^{d^* \times d^*}$. Hence $\boldsymbol{V}$ consists of independent standard normal random variables. Since we are working with a modest number of training data points in each of the data sets considered, we opt for a batch optimization procedure presented as a special case ($n_{\text{bat}} = \tau$) of Algorithm 4.2.1. For the number of epochs, we choose $n_{\text{epo}} = 1000$ which ensures a sufficiently long training period to obtain accurate results. The tuning parameters of the Adam optimizer is set to the default values reported in Kingma and Ba (2014).

Dependence models: parametric copulas

For comparison with GMMN–GARCH models, we also present results for a number of different parametric copula models C_{PM}. These include Gumbel copulas, normal copulas with exchangeable correlation matrices and t copulas with exchangeable and with unstructured correlation matrices. We fit these copulas using the maximum pseudo-likelihood method via the function `fitCopula(, method="mpl")` from the R package `copula`. We can generate samples from the fitted copulas using the `rCopula()` function from the same R package. We also produce results for the independence copula which serves as a simple benchmark model.

5.3.2 Yield curve modeling

Analyzing and modeling *zero-coupon bond (ZCB) yield curves*, also referred to as the *term structure of interest rates*, is a critical task in various financial and economic applications. While early research in this area is often solely focused on constructing models of yield curves based on economic theory, the seminal work by Diebold and Li (2006) focused on the critical task of yield curve forecasting.

The primary approach showcased in Diebold and Li (2006) was the embedding of autoregressive models within the parametric structure of the three factor Nelson–Siegel model (Nelson and Siegel, 1987) which intuitively characterizes the level, slope and curvature of the yield curve. Since then various approaches for forecasting yield curves have been investigated; see Diebold and Rudebusch (2013) for an overview and Caldeira et al. (2016) for a recently proposed forecast combination approach. Most models proposed and reviewed in the literature are particularly designed towards constructing point forecasts for yield curves. Such point forecasts are typically useful in bond portfolio optimization and in the pricing of certain financial assets. Alternatively, distributional forecasts of ZCB yield curves could potentially be helpful in risk management applications, derivative pricing (via

simulation) and economic scenario generation. To that end, in this section, we consider modeling US and Canadian ZCB yield curves using MTS models. We then utilize our fitted GMMN–GARCH models to obtain empirical predictive distributions of these ZCB yield curves.

Modeling US and Canadian ZCB data

For US treasury ZCB data, we consider a 30-dimensional yield curve constructed from ZCBs with times to maturity ranging from 1 to 30 years in annual increments. For Canadian ZCB data, we consider a 120-dimensional yield curve constructed from ZCBs with times to maturity ranging from 0.25 to 30 years in quarterly increments. Refer to the R package `qrmdata` for further details about these data. In particular, we consider these multivariate time series in the time period from 1995-01-01 to 2015-12-31 (2015-08-31 for the Canadian data), treating data from 1995-01-01 to 2014-12-31 as the training set and the remainder as the test set.

As a pre-processing step, we begin by applying a simple difference transform to the original time series. We then take the transformed series to be the series $\boldsymbol{X}_t$ that we work with.

Following our framework, we first model the marginal time series using the ARMA–GARCH model setup described in Section 5.3.1 with $\mu_j = 0$, $j = 1, \dots, d$. Since these data are relatively high-dimensional ($d = 30$ for the US data and $d = 120$ for the Canadian data), we apply PCA to the standardized residuals $\hat{\boldsymbol{Z}}_t$ for dimension reduction. Yield curves are indeed amenable to good approximations via lower dimensional representations; various dimension reduction techniques such as factor models have been incorporated by various yield curve models (Diebold and Li, 2006). We choose the number of top principal components k to construct the lower dimensional representation for each dataset as follows. We select the smallest $k \geq 3$ such that the first k principal components account for at least 95% of the total variance in the standardized residuals $\hat{\boldsymbol{Z}}_t$. For the US data, this choice is $k = 3$; for the Canadian data, it is $k = 4$.

Assessment

We evaluate the performance of our models on the test set using the metrics discussed in Section 5.2.5. First, we compute the AMMD metric (5.1) using $n_{\text{rep}} = 100$ replications to assess the quality of the dependence models in the test period. Then, to assess if capturing the underlying cross-sectional dependence structure well translates to better

one-day-ahead empirical predictive distributions, we compute the AMSE metric (5.2) and the AVS^p metric (5.3) using $n_{\text{pth}} = 1000$ simulated paths.

Figure 5.1 displays scatter plots of AMSE (left) and $\text{AVS}^{0.25}$ (right) versus AMMD for the US (top) and Canadian (bottom) data. For both datasets, samples generated from the three GMMN models (see Section 5.3.1) more closely match the underlying cross-sectional dependence structure in their corresponding test sets than those generated from the four parametric copulas and the independence copula (see Section 5.3.1). Moreover, across the entire spectrum of GMMN–GARCH and copula–GARCH models being studied, it is also clear that better dependence modeling (as measured by the AMMD metric) does indeed translate into better one-day-ahead empirical predictive distributions (as measured by the AMSE and $\text{AVS}^{0.25}$ metrics). Specifically, all GMMN models clearly outperform the best copula model, i.e., a t-copula with unstructured correlation matrix, in all three metrics — although among the GMMN models themselves there is not a single best one.

5.3.3 Exchange rate modeling

The modeling and analysis of foreign exchange rate dependence is an important task in risk management applications involving a global portfolio of financial assets. As such, dependent multivariate time series of exchange rates have been previously studied in the copula literature; for example see Patton (2006) or Dias and Embrechts (2010). In this section, we consider modeling foreign exchange rate data with respect to the US dollar (USD) and Pound sterling (GBP) using MTS models. We then utilize our fitted GMMN–GARCH and copula–GARCH models to obtain empirical predictive distributions and Value-at-Risk (VaR) forecasts for portfolios of exchange rate assets.

Modeling USD and GBP exchange rate data

For the USD exchange rate data, we consider the daily exchange rates of Canadian dollar (CAD), Pound sterling (GBP), Euro (EUR), Swiss Franc (CHF) and Japanese yen (JPY) with respect to the USD. For the GBP exchange rate data, we consider the daily exchange rates of CAD, USD, EUR, CHF, JPY and the Chinese Yuan (CNY) with respect to the GBP. For further details regarding both the USD and GBP exchange rate data, see the R package `qrmdata`. In particular, we consider these multivariate time series in the time period from 2000-01-01 to 2015-12-31, treating data up to 2014-12-31 as the training set and the remainder as the test set. Due to the fixed peg of the CNY against the USD, particularly prior to August 2005, we do not include it in the USD dataset.

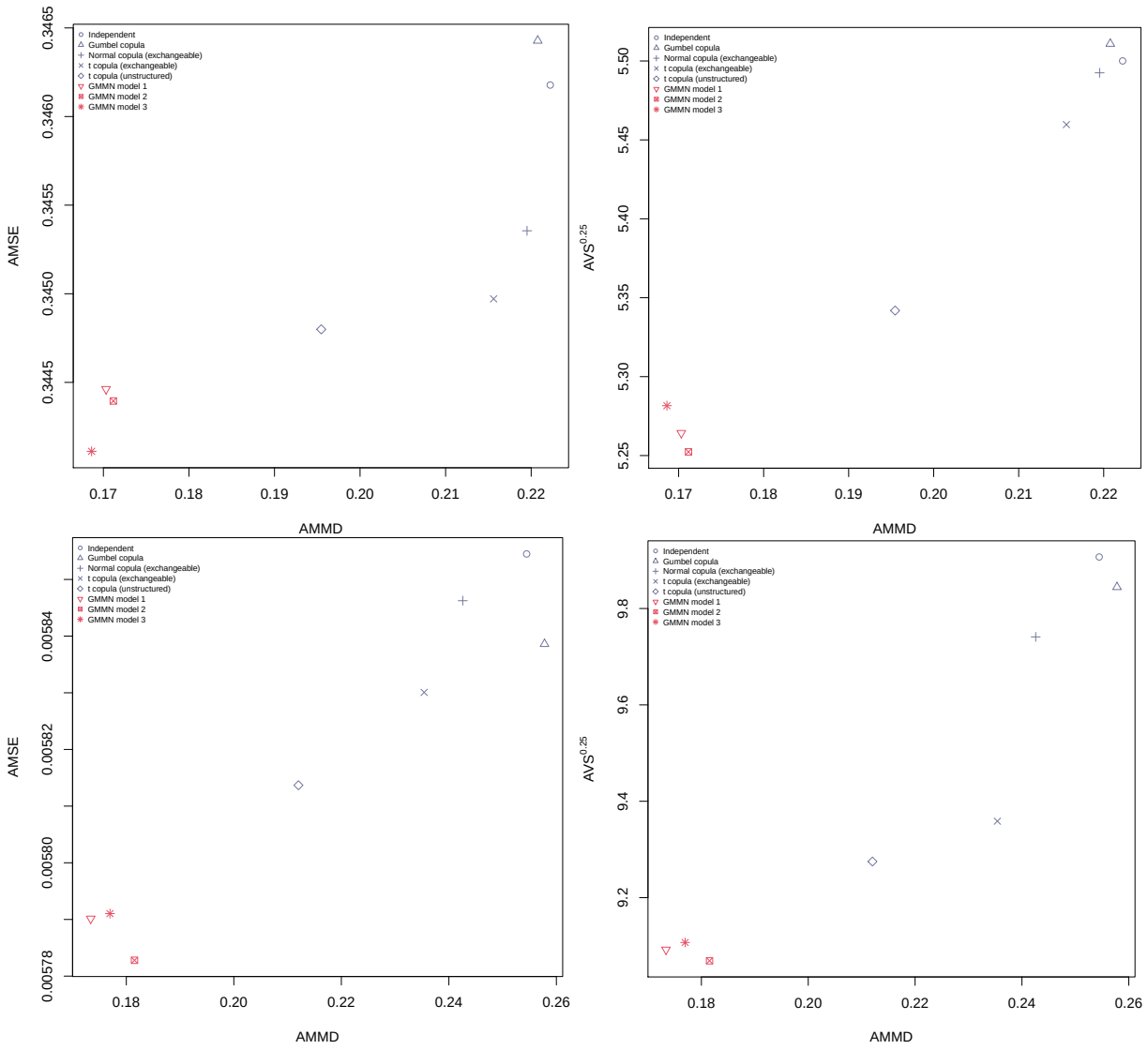

Figure 5.1: Model assessments for US (top) and Canadian (bottom) ZCB yield curve data. Scatter plots of AMSE (left) and $\text{AVS}^{0.25}$ (right) computed based on $n_{\text{pth}} = 1000$ simulated paths versus AMMD computed based on $n_{\text{rep}} = 100$ realizations. All models incorporate PCA with $k = 3$ (US) and $k = 4$ (Canadian) principal components.

To begin with, we apply the log-returns transformation to the nominal exchange rates and work with the resulting return series for modeling. Following our framework, we start by modeling the marginal time series using the ARMA–GARCH specification as detailed in Section 5.3.1. Since these datasets are relatively low-dimensional ($d = 5$ for the USD data and $d = 6$ for the GBP data), we do not incorporate any dimension reduction step in this analysis.

Assessment

Following the setup in Section 5.3.2, we evaluate the performance of our models with the AMMD, AMSE and $\text{AVS}^{0.25}$ metrics on the test set. Figure 5.2 displays scatter plots of AMSE (left) and $\text{AVS}^{0.25}$ (right) versus AMMD for the USD (top) and GBP (bottom) data. We can draw the same conclusions from this figure as those from Figure 5.1. In addition, here we also observe that the independence copula performs noticeably worse than all other models, whether capturing the dependence structure of the innovation distribution or making probabilistic forecasts.

Forecasting daily portfolio VaR

As demonstrated in the previous section, GMMN–GARCH models produce better one-day-ahead empirical predictive distributions when compared with various copula–GARCH models. We can utilize these one-day-ahead empirical predictive distributions to extract forecasts of various quantities of interest in risk management. One such popular quantity is the Value-at-Risk (VaR) of a portfolio.

To begin with, consider the portfolio aggregate return $S_t = \sum_{j=1}^{d} X_{t,j}$ at time t. Then, the (theoretical) VaR at confidence level α and time t is given by $\text{VaR}_\alpha(S_t) = F_{S_t}^{-1}(\alpha)$ where $F_{S_t}^{-1}$ denotes the quantile function of S_t. In practice, we can compute the empirical α-quantile of S_t from its empirical predictive distribution, $\{\hat{S}_t^{(i)} = \sum_{j=1}^{d} \hat{X}_{t,j}^{(i)} : i = 1, \ldots, n_{\text{pth}}\}$. We denote the corresponding forecast by $\widehat{\text{VaR}}_\alpha(\hat{S}_t)$. Thus, for each MTS model, we compute daily forecasts $\widehat{\text{VaR}}_\alpha(\hat{S}_t)$ for every $t \in \mathcal{T}$ in the test period. To assess the quality of these forecasts, we can compute the frequency with which S_t actually exceeds the daily forecast $\widehat{\text{VaR}}_\alpha(\hat{S}_t)$ over the entire test period $\mathcal{T}$. We expect this frequency to be α. Hence, we can evaluate our VaR forecasts by measuring the (absolute) error between the actual and the expected exceedance frequency, or simply the *VaR exceedance absolute error*, defined by

$$\text{VEAR}_\alpha = \left| \alpha - \frac{1}{T - \tau} \sum_{t=\tau+1}^{T} \mathbb{1}_{\{S_t < \widehat{\text{VaR}}_\alpha(\hat{S}_t)\}} \right|. \tag{5.4}$$

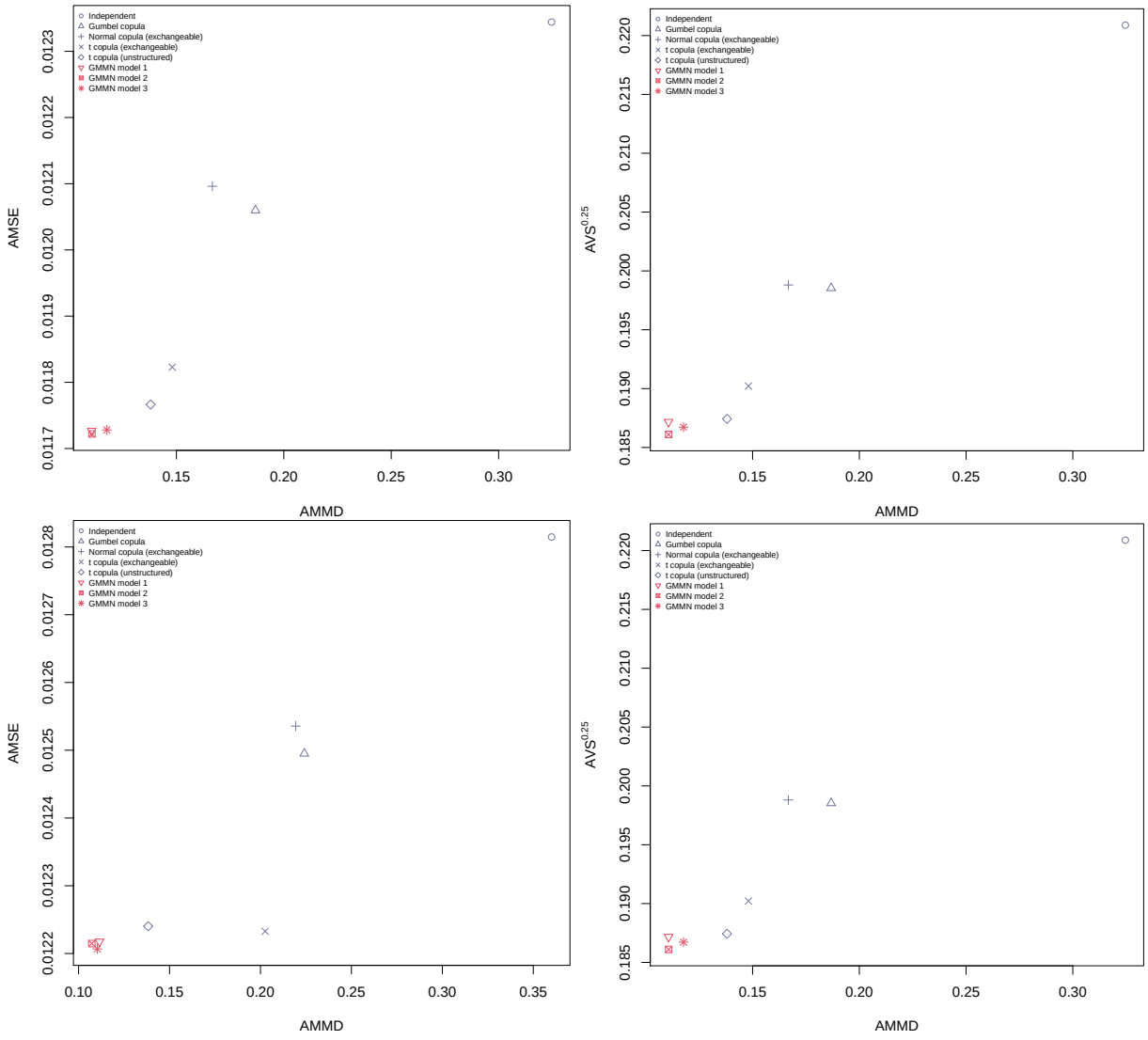

Figure 5.2: Model assessments for USD (top) and GBP (bottom) exchange rate data. Scatter plots of AMSE (left) and $\text{AVS}^{0.25}$ (right) computed based on $n_{\text{pth}} = 1000$ simulated paths versus AMMD computed based on $n_{\text{rep}} = 100$ realizations.

Figure 5.3 displays scatter plots of $\text{VEAR}_{0.05}$ versus AMMD for the USD (left) and GBP (right) exchange rates data. For both datasets, the three GMMN–GARCH models produce better daily forecasts of $\text{VaR}_{0.05}(S_t)$ than the five copula–GARCH models do. Again, there exists a clear general trend that fitted dependence models which more closely match the underlying dependence structures of the test datasets tend to yield better daily forecasts. Particularly, assuming independence amongst the exchange rate returns leads to notably poorer forecasts.

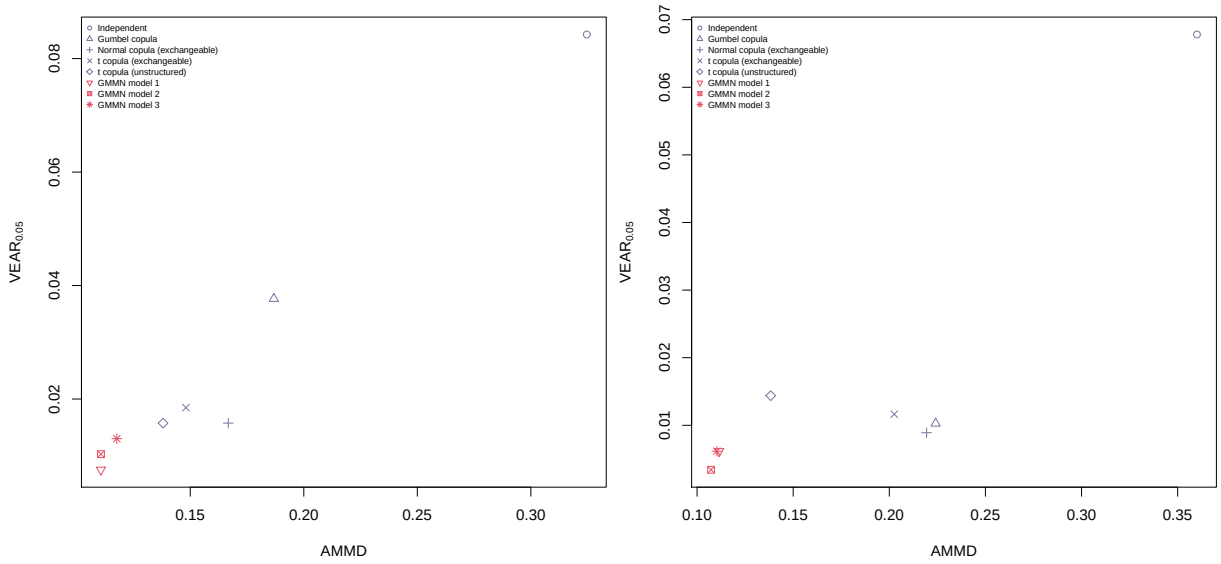

Figure 5.3: VaR forecast assessments for USD (left) and GBP (right) exchange rate data. Scatter plots of $\text{VEAR}_{0.05}$ computed based on $n_{\text{pth}} = 1000$ simulated paths versus AMMD computed based on $n_{\text{rep}} = 100$ realizations.

5.4 Conclusion

We introduced generative moment matching networks (GMMNs) for modeling the dependence in MTS data. First, ARMA–GARCH models are used to marginally model serial dependence. Second, for high-dimensional MTS data, a dimension reduction method can be applied. Last, the cross-sectional dependence is modeled by a GMMN. In the popular copula–GARCH approach, the latter step typically requires us to find a parametric copula model which fits the given data well. This can already be a challenging task in moderately large dimensions. By contrast, GMMNs are highly flexible and easy to simulate from, which

is a major advantage of our GMMN–GARCH approach. The primary objective of fitting these MTS models is to produce empirical predictive distributions, with which we can then forecast various quantities of interest in risk management such as VaR or expected shortfall.

To showcase the flexibility of our GMMN–GARCH framework, we considered modeling ZCB yield curves and foreign exchange rate returns. Across all the examples considered, we demonstrated that fairly simple GMMNs were able to better capture the underlying cross-sectional dependence than many well-known parametric copulas. Consequentially, we observed that the corresponding GMMN–GARCH models yielded superior one-period-ahead empirical predictive distributions. Additionally, for exchange rate data, we demonstrated that GMMN–GARCH models produced more accurate daily portfolio VaR forecasts as well.

Chapter 6

Summary and Future Research

6.1 Summary

In this thesis, we made several contributions to dependence modeling and its associated application areas. Broadly speaking, our contributions to the literature spanned three categories — dependence measures, dependence modeling with parametric copulas and dependence modeling with generative neural networks.

Firstly, in Chapter 2, we proposed a framework for measuring association of random vectors using collapsing functions. This framework yielded various tools to characterize the dependence between random vectors including numerous new measures of association along with the corresponding asymptotic results required to construct confidence intervals for these measures, collapsed copulas to analytically summarize the dependence between random vectors under certain setups and a graphical assessment of independence between groups of random variables. We showcased the applicability of these tools to detect and rank dependencies between random vectors in bioinformatics and finance applications. The *key takeaway* from our theoretical and empirical investigations is that there is no universal notion of a *best* collapsing function. Consequently, different collapsing functions were useful in different contexts and applications.

Secondly, in Chapter 3, we introduced a new class of flexible parametric copulas, the hierarchical Archimax copulas (HAXCs). Two ways of inducing hierarchies for AXCs were investigated — one at the level of EVCs and the other at the level of ACs. To that end, we presented a novel approach for constructing hierarchical EVCs which involved the connection between stable tail dependence functions and d-norms. We additionally imposed

hierarchical dependence structures at the level of frailties in the same vein as NACs were constructed from ACs. Since all of our methods of constructing HAXCs were based on stochastic representations, sampling algorithms were easy to formulate. Various examples of HAXCs were then presented along with their associated stochastic representations.

Thirdly, in Chapters 4 and 5, we investigated the use of generative neural networks for dependence modeling tasks. This investigation formed the second half of the thesis and involved two different application areas, quasi-random sampling and time series modeling. In both projects, we opted to work with a type of generative model known as the generative moment matching network (GMMN). In Chapter 4, we demonstrated how GMMNs can be utilized to generate quasi-random samples from a large variety of multivariate distributions. Utilizing GMMNs yielded a more *flexible* and *universal* approach for multivariate (dependent) quasi-random sampling compared to classical parametric copula methods. Furthermore, we showcased the benefits of utilizing GMMN quasi-random samples to approximate expectations of the form $\mu = \mathbb{E}(\Psi(\boldsymbol{X}))$ by theoretically establishing convergence rates for the corresponding GMMN RQMC estimators and numerically demonstrating the variance reduction effects achieved by these RQMC estimators. Finance and risk management applications where the objective was to approximate quantities of interest μ involving asset portfolios were then used to demonstrate the ability of GMMNs to more accurately capture complex dependence structures in real data compared to parametric copulas in addition to the variance reduction capabilities of GMMN RQMC estimators when estimating μ. In Chapter 5, we proposed a GMMN–GARCH framework for multivariate time series (MTS) modeling with the primary goal of constructing empirical predictive distributions (EPDs), also known as probabilistic forecasts. These EPDs are useful in forecasting risk measures, e.g., VaR. The flexibility of our GMMN–GARCH models to produce superior EPDs and VaR forecasts compared to well-known copula–GARCH models was showcased in the context of modeling ZCB yield curves and foreign exchange rate returns.

6.2 Future research

In this section, we discuss future research directions that involve both direct extensions of our work in Chapters 2–5 and potential ways to combine research topics across these chapters.

Extensions of the collapsing function framework

Firstly, while we focused on using collapsed measures of association for ranking dependencies in our numerical investigations, we could also utilize these measures to formulate tests of independence between random vectors. To this end, we could adapt the asymptotic results derived for various collapsing functions to develop *asymptotic tests* of independence. Additionally, we could create corresponding *permutation tests* of independence for the various collapsed measures considered. There exists some literature on this research topic, most notably the statistical tests based on the distance covariance coefficient (Székely et al., 2007), Hilbert Schmidt independence criterion (Gretton et al., 2008) and Cramér–von Mises functionals (Kojadinovic and Holmes, 2009). In addition, Josse and Holmes (2016) conducted a survey of various existing tests of independence between random vectors. Therefore, it would be interesting to compare the various asymptotic and permutation tests derived from our collapsing function framework, with existing tests in the literature, using simulation exercises and real data applications. Ideally, we would be searching for collapsed measures of association which yield tests with great statistical power and which are capable of detecting a wide range of complex non-linear dependencies.

Secondly, we could further investigate collapsed distribution functions $F_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ and collapsed copulas $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$. In Chapter 2, we primarily focused on deriving analytical forms for $F_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ and $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ in terms of the joint distribution of $(\boldsymbol{X}, \boldsymbol{Y})$ under certain setups, e.g., maximum and PIT collapsing functions. An interesting and challenging open problem would be to extend this analytical exercise to other collapsing functions such as the weighted sum collapsing function. Furthermore, we could construct a new type of hierarchical model which utilizes the collapsed copula $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}$ as a sufficient lower-dimensional proxy for the dependence structure between $\boldsymbol{X}$ and $\boldsymbol{Y}$, while modeling the dependence structures within $\boldsymbol{X}$ and within $\boldsymbol{Y}$ using higher dimensional copulas $C_{\boldsymbol{X}}$ and $C_{\boldsymbol{Y}}$. In practice, such a modeling effort would typically extend beyond the two groups of random variables and the one level of collapsing. While there exists some research in this direction for the PIT collapsing function (Brechmann, 2014) and the sum collapsing function (Arbenz et al., 2012; Côté and Genest, 2015), extensions to other collapsing functions remain an open problem. In particular, it would be useful to develop sampling algorithms and estimation procedures for such hierarchical dependence models.

Sampling protein conformations via generative neural networks

In Chapter 2, we discussed an application from bioinformatics that involved generating protein side chain conformations. Each protein has approximately between 160–190 residues,

which have side chains whose conformations are characterized by sets of dihedral angles, with lengths ranging between zero to four. In our work, we used the protein side chain conformations that were generated by Ghoraie et al. (2015a) using fast side-chain prediction (SCP) algorithms. These specialized algorithms were used as an efficient alternative to the popular Rosetta modeling suite (Kaufmann et al., 2010), which utilizes a knowledge-guided Metropolis Monte Carlo sampling algorithm. Even so, as Ghoraie et al. (2015a) noted, the efficient SCP approach still took approximately one second to generate one sample conformation; a marked improvement over the 40 seconds taken by the Rosetta modeling suite procedure.

As an interesting future research project, we could investigate the use of generative neural networks for sampling protein side-chain conformations. Once trained on a moderately-sized dataset of protein side-chain conformations obtained from either the SCP algorithm or the Rosetta modeling suite, the fitted NN would be capable of generating millions of novel conformations very efficiently. Naturally, due to the complex nature of the problem, we would have to explore more sophisticated NN architectures than those presented in this thesis. Also, we would potentially have to incorporate certain problem-specific constraints within the NN architecture and/or the loss function used to train the generative model, in order to ensure that the generated protein conformations are realistic, in a biological sense.

Fitting hierarchical Archimax copulas

In Chapter 3, we mainly discussed how to construct and sample from HAXCs. Hence, our proposed dependence models would currently be useful only in simulation studies. To utilize this flexible class of copulas for modeling real data, we would need to develop parameter estimation and model selection procedures. To this end, we have already derived some preliminary results concerning the density of AXCs and its numerical treatment, which could be useful in employing a maximum likelihood estimation procedure; see Appendix B.1. As an alternative to the full likelihood approach, we could also utilize a composite (pairwise) likelihood method (Varin et al., 2011) to achieve gains in efficiency at the expense of using a misspecified model. To fit HAXCs, we would then need to couple these estimation procedures for AXCs with appropriate tree structure selection techniques.

Very recently, Chatelain (2019) presented a fairly extensive treatment of inference techniques for AXCs which considered parametric, semi-parametric and non-parametric estimation procedures. Additionally, Chatelain (2019) also introduced the class of clustered Archimax copulas (CAXCs), which offers an alternative hierarchical extension for AXCs that are constructed based on the Williamson d-transform, along with corresponding inference

procedures. Going forward it would be interesting to compare the various estimation techniques proposed in Chatelain (2019) with the maximum (full and composite) likelihood methods. Furthermore, we could theoretically and empirically explore the similarities and differences between CAXCs and HAXCs.

Multivariate time series probabilistic forecasting with GMMN quasi-random samples

Following the investigation into the use of GMMNs as dependence models in Chapters 4 and 5, a natural subsequent research direction would be to study whether GMMN quasi-random samples could be used to produce better probabilistic forecasts (EPDs). As demonstrated in Chapter 4, GMMN transforms typically preserve the low-discrepancy of the input RQMC point set. Thus, the next step would be to theoretically and numerically analyze the extent to which the low-discrepancy properties observed in GMMN quasi-random samples propagate through the ARMA–GARCH and PCA components of our proposed MTS modeling framework. Provided that these composite transforms that arise from our framework are sufficiently smooth, we should be able to construct EPDs that possess low-discrepancy properties and consequently low-variance risk measure forecasts. In a broader context, it would be interesting to explore the impact that marginal time series models and dimension reduction techniques have on the preservation of the low-discrepancy observed in GMMN quasi-random samples upon transformation.

Extensions of the GMMN–GARCH framework

In our current GMMN–GARCH setup, once we account for temporal dependencies within each marginal time series using ARMA–GARCH models, we assume that the resulting joint innovation distribution is constant across time. Thus, a potential extension of our GMMN–GARCH framework is to incorporate time-varying cross-sectional dependence structures. However, it can be fairly challenging to train such time-varying GMMN dependence models. Given a fitted GMMN trained on sufficiently large training data, one approach would be to then re-train the GMMN after every subsequent τ_{rtn} time periods, while initializing the neural network with the previously fitted parameters. Adopting this approach, we can reduce the computational time and resources needed, i.e., using fewer epochs for the re-training, by leveraging features of the cross-sectional distribution that we have already learned from prior training. For example, if our fitted GMMN had captured the upper/lower tail dependencies, asymmetries or singular components present in the original (or previously

used) training data, we do not have to re-learn these features when re-training. Of course, if there is a drastic change in the nature of the dependence structure itself, we would need to expend greater computational resources to learn the new underlying cross-sectional dependence. However, in most real-data, we commonly observe a more gradual shift in the strength of dependencies across time rather than fundamental changes to the salient features of the underlying distribution. Furthermore, we could also explore utilizing change-point analysis to more strategically identify when we need to re-fit our GMMN–GARCH models.

In Chapter 5, we focused on financial time series applications with the goal of constructing EPDs and forecasting risk measures. Within the context of risk management and finance, we could also consider utilizing our GMMN–GARCH models for derivative pricing and portfolio-risk optimization. Furthermore, potential future research projects could involve exploring other application areas for GMMN–GARCH models such as weather, energy and demand probabilistic forecasting.

Finally, we could investigate using a variety of other marginal time series models and dimension reduction techniques in the first two modeling steps of our GMMN–GARCH framework. For modeling different types of temporal dependencies found in time series data, we could explore other models within the GARCH family such as IGARCH, EGARCH or GJR–GARCH. Alternatively, we could also more broadly fuse GMMN and (marginal) stochastic volatility models to create, for example, GMMN–Heston or GMMN–CEV models. Furthermore, while we only considered PCA as a dimension reduction technique in Chapter 5, it would be interesting to work with more sophisticated models such as auto-encoders to better characterize higher dimensional MTS data.

Multivariate time series modeling with generative neural networks: An alternate approach

In this thesis, we solely worked with generative neural networks that were constructed using feedforward neural networks. An alternate approach for MTS modeling would be to utilize generative models that are constructed based on recurrent neural networks (RNNs); for further details on RNNs see Goodfellow et al. (2016, Chapter 10). RNNs were specifically designed to model sequential data. So instead of relegating the task of modeling temporal dependence to classical time series models such as GARCH models, we could jointly model temporal and cross-sectional dependence using RNNs. However, since we are particularly interested in producing EPDs or probabilistic forecasts, we would have to adapt our RNN model setup and loss function appropriately. To that end, we could develop generative models with (embedded) RNNs by using either the average MSE metric, defined in (5.2), or

the average variogram score, defined in (5.3), as loss functions for training. These generative neural networks would then be geared towards producing good probabilistic forecasts.

References

K. Aas. Pair-copula constructions for financial applications: A review. *Econometrics*, 4(4): 43, 2016.

E. F. Acar, C. Czado, and M. Lysy. Flexible dynamic vine copula models for multivariate time series data. *Econometrics and Statistics*, 12:181–197, 2019.

C. Alexander. A primer on the orthogonal garch model. *manuscript ISMA Centre, University of Reading, UK*, 2, 2000.

A. Alfons, C. Croux, and P. Filzmoser. Robust maximum association estimators. *Journal of the American Statistical Association*, 112(517):436–445, 2017.

J. J. Allaire, Y. Tang, D. Eddelbuettel, N. Golding, and T. Kalinowski. *tensorflow: R Interface to "TensorFlow"*, 2017. URL http://cran.r-project.org/package=keras. R package version 2.1.6.

C. Almeida, C. Czado, and H. Manner. Modeling high-dimensional time-varying dependence using dynamic d-vine models. *Applied Stochastic Models in Business and Industry*, 32(5): 621–638, 2016.

G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013.

P. Arbenz, C. Hummel, and G. Mainik. Copula based hierarchical risk aggregation through sample reordering. *Insurance: Mathematics and Economics*, 51(1):122–133, 2012.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. 2016. URL https://arxiv.org/abs/1611.01491.

S. Arora, A. Risteski, and Y. Zhang. Do gans learn the distribution? some theory and empirics. In *International Conference on Learning Representations, ICLR*, 2018.

S. Aulbach, M. Falk, and M. Zott. The space of d-norms revisited. *Extremes*, 18(1):85–97, 2015.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

P. Barbe, C. Genest, K. Ghoudi, and B. Rémillard. On Kendall's process. *Journal of Multivariate Analysis*, 58:197–229, 1996.

J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels. *Statistics of Extremes: Theory and Applications*. Wiley, 2004.

L. Belzile, J. L. Wadsworth, P. J. Northrop, S. D. Grimshaw, and R. Huser. *mev: Multivariate Extreme Value Distributions*, 2017. URL https://CRAN.R-project.org/package=mev. R package version 1.10.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank, 1999–. In *International Tables for Crystallography Volume F: Crystallography of biological macromolecules*, pages 675–684. Springer, 2006.

S. N. Bernstein. Sur les fonctions absolument monotones. *Acta Mathematica*, 52:1–66, 1928.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

T. Bollerslev. Modelling the coherence in short-run nominal exchange rates: a multivariate generalized arch model. *Review of Economics and statistics*, 72(3):498–505, 1990.

P. Bougerol and N. Picard. Strict stationarity of generalized autoregressive processes. *The Annals of Probability*, pages 1714–1730, 1992.

E. C. Brechmann. Hierarchical Kendall copulas: Properties and inference. *Canadian Journal of Statistics*, 42(1):78–108, 2014.

F. X. Briol, A. Barp, A. B. Duncan, and M. Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, 2019.

P. J. Brockwell, R. A. Davis, and S. E. Fienberg. *Time Series: Theory and Methods: Theory and Methods*. Springer Science & Business Media, 1991.

A. Buja, D. Cook, H. Hofmann, M. Lawrence, E.K. Lee, D. F. Swayne, and H. Wickham. Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4361–4383, 2009.

R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

J. F. Caldeira, G. V. Moura, and A. A. P. Santos. Predicting the yield curve using forecast combinations. *Computational Statistics & Data Analysis*, 100:79–98, 2016.

M. Cambou, M. Hofert, and C. Lemieux. Quasi-random numbers for copula models. *Statistics and Computing*, 27(5):1307–1329, 2017.

P. Capéraà, A.L. Fougères, and C. Genest. Bivariate distributions with given extreme value distributions. *Journal of Multivariate Analysis*, 72:30–49, 2000.

S. Castruccio, R. Huser, and M. G. Genton. High-order Composite Likelihood Inference for Max-Stable Distributions and Processes. *Journal of Computational and Graphical Statistics*, 25:1212–1229, 2016.

A. Charpentier, A.L. Fougères, C. Genest, and J. Nešlehová. Multivariate Archimax copulas. *Journal of Multivariate Analysis*, 126:118–136, 2014a.

A. Charpentier, A.L. Fougères, C. Genest, and J. G. Nešlehová. Multivariate Archimax copulas. *Journal of Multivariate Analysis*, 126:118–136, 2014b.

S. Chatelain. *Modeling the dependence of pre-asymptotic extremes*. PhD thesis, Université de Lyon; McGill university (Montréal, Canada), 2019.

F. Chollet, J. J. Allaire, Y. Tang, D. Falbel, W. van der Bijl, and M. Studer. *Keras: R Interface to "keras"*, 2017. URL http://cran.r-project.org/package=keras. R package version 2.1.6.

S. Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 2001.

G. Constantine and T. Savits. A multivariate faa di bruno formula with applications. *Transactions of the American Mathematical Society*, 348(2):503–520, 1996.

M. Côté and C. Genest. A copula-based risk aggregation model. *Canadian Journal of Statistics*, 43(1):60–81, 2015.

R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914, 1976.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

M. Daily, T. Upadhyaya, and J. Gray. Contact rearrangements form coupled networks from local motions in allosteric proteins. *Proteins: Structure, Function, and Bioinformatics*, 71(1):455–466, 2008.

L. de Haan. A spectral representation for max-stable processes. *The Annals of Probability*, pages 1194–1204, 1984.

A. Dias and P. Embrechts. Modeling exchange rate dependence dynamics at different time horizons. *Journal of International Money and Finance*, 29(8):1687–1705, 2010.

F. X. Diebold and C. Li. Forecasting the term structure of government bond yields. *Journal of Econometrics*, 130(2):337–364, 2006.

F. X. Diebold and G. D. Rudebusch. *Yield curve modeling and forecasting: the dynamic Nelson-Siegel approach*. Princeton University Press, 2013.

A. B. Dieker and T. Mikosch. Exact simulation of Brown-Resnick random fields at a finite number of locations. *Extremes*, 18(2):301–14, 2015.

C. Doersch. Tutorial on variational autoencoders. 2016. URL https://arxiv.org/abs/1606.05908.

C. Dombry, S. Engelke, and M. Oesting. Exact simulation of max-stable processes. *Biometrika*, 103(2):303, 2016.

G. Doyon. On densities of extreme value copulas. Master's thesis, ETH Zürich, 2013.

R. Durrett. *Probability: Theory and Examples*. Duxbury Press, 3 edition, 2004.

G. K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 258–267. AUAI Press, 2015. URL http://www.auai.org/uai2015/proceedings/papers/230.pdf.

R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940, 2016.

P. Embrechts and A. Dias. Quantitative risk management: Concepts, techniques and tools. 2004. URL https://people.math.ethz.ch/~embrecht/ftp/quebec.pdf.

P. Embrechts, A. J. McNeil, and D. Straumann. Correlation and dependency in risk management: Properties and pitfalls. In M. Dempster, editor, *Risk Management: Value at Risk and Beyond*, pages 176–223. Cambridge University Press, 2002.

P. Embrechts, F. Lindskog, and A. J. McNeil. Modelling dependence with copulas and applications to risk management. In S. Rachev, editor, *Handbook of Heavy Tailed Distributions in Finance*, pages 329–384. Elsevier, 2003.

P. Embrechts, M. Hofert, and R. Wang. Bernoulli and tail-dependence compatibility. *The Annals of Applied Probability*, 26(3):1636–1658, 2016. doi: 10.1214/15-AAP1128.

R. Engle. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics*, 20(3):339–350, 2002.

H. Faure and C. Lemieux. Generalized Halton sequence in 2008: A comparative study. *ACM Transactions on Modeling and Computer Simulation*, 19:Article 15, 2009.

W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. Wiley, 2 edition, 1971.

M. Fischer, C. Köck, S. Schlüter, and F. Weigert. An empirical analysis of multivariate copula models. *Quantitative Finance*, 9(7):839–854, 2009.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

C. Genest and R. J. MacKay. Copules archimédiennes et familles de lois bidimensionnelles dont les marges sont données. *The Canadian Journal of Statistics*, 14:145–159, 1986.

C. Genest and L.P. Rivest. Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association*, 88(423):1034–1043, 1993.

C. Genest and J. Segers. On the covariance of the asymptotic empirical copula process. *Journal of Multivariate Analysis*, 101(8):1837–1845, 2010.

C. Genest, K. Ghoudi, and L.-P. Rivest. A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82(3): 543–552, 1995.

C. Genest, B. Rémillard, and D. Beaudoin. Goodness-of-fit tests for copulas: A review and a power study. *Insurance: Mathematics and economics*, 44(2):199–213, 2009.

A. Ghalanos. *rugarch: Univariate GARCH models.*, 2019. R package version 1.4-1.

L. S. Ghoraie, F. Burkowski, and M. Zhu. Using kernelized partial canonical correlation analysis to study directly coupled side chains and allostery in small g proteins. *Bioinformatics*, 31(12):i124–i132, 2015a.

S. Ghoraie, F. Burkowski, and M. Zhu. Sparse networks of directly coupled, polymorphic, and functional side chains in allosteric proteins. *Proteins: Structure, Function, and Bioinformatics*, 83(3):497–516, 2015b.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.

M. Grabisch, J. L. Marichal, R. Mesiar, and E. Pap. *Aggregation Functions (Encyclopedia of Mathematics and Its Applications)*. Cambridge University Press, New York, NY, USA, 1st edition, 2009. ISBN 0521519268, 9780521519267.

A. Gretton, K. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pages 513–520, 2007.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, pages 585–592, 2008.

A. Gretton, K. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012a.

A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems*, pages 1205–1213, 2012b.

O. Grothe, J. Schnieders, and J. Segers. Measuring association and dependence between random vectors. *Journal of Multivariate Analysis*, 123:96–110, 2014.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

M. Hardy. Combinatorics of partial derivatives. *The Electronic Journal of Combinatorics*, 13(1), 2006.

G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

E. Hlawka. Über die diskrepanz mehrdimensionaler folgen mod 1. *Mathematische Zeitschrift*, 77:273–284, 1961.

E. Hlawka and R. Mück. über eine transformation von gleichverteilten folgen ii. *Computing*, 9(2):127–138, 1972.

W. Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, pages 293–325, 1948.

M. Hofert. *Sampling Nested Archimedean Copulas with Applications to CDO Pricing*. Südwestdeutscher Verlag für Hochschulschriften AG & Co. KG, 2010. ISBN 978-3-8381-1656-3. PhD thesis.

M. Hofert. Efficiently sampling nested Archimedean copulas. *Computational Statistics & Data Analysis*, 55:57–70, 2011. doi: 10.1016/j.csda.2010.04.025.

M. Hofert. A stochastic representation and sampling algorithm for nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, 82(9):1239–1255, 2012. doi: 10.1080/00949655.2011.574632.

M. Hofert and K. Hornik. *`qrmtools`: Tools for Quantitative Risk Management*, 2016. R Package Version 0.0-6.

M. Hofert and R. W. Oldford. Visualizing Dependence in High-dimensional Data: An Application to S&P 500 Constituent Data. *Econometrics and Statistics*, 8:161–183, 2018. doi: 10.1016/j.ecosta.2017.03.007.

M. Hofert and W. Oldford. Visualizing dependence in high-dimensional data: An application to S&P 500 constituent data. *Econometrics and Statistics*, 2017.

M. Hofert and D. Pham. Densities of nested Archimedean copulas. *Journal of Multivariate Analysis*, 118:37–52, 2013. doi: 10.1016/j.jmva.2013.03.006.

M. Hofert and M. Scherer. CDO pricing with nested Archimedean copulas. *Quantitative Finance*, 11(5):775–787, 2011. doi: 10.1080/14697680903508479.

M. Hofert, I. Kojadinovic, M. Mächler, and J. Yan. *copula: Multivariate Dependence with Copulas*, 2005. URL http://CRAN.R-project.org/package=copula. R package version 0.999-19.

M. Hofert, M. Mächler, and A.J. McNeil. Likelihood inference for Archimedean copulas in high dimensions under known margins. *Journal of Multivariate Analysis*, 110:133–150, 2012. doi: 10.1016/j.jmva.2012.02.019.

M. Hofert, M. Mächler, and A.J. McNeil. Archimedean copulas in high dimensions: Estimators and numerical challenges motivated by financial applications. *Journal de la Société Française de Statistique*, 154(1):25–63, 2013.

M. Hofert, I. Kojadinovic, M. Maechler, and J. Yan. *copula: Multivariate Dependence with Copulas*, 2017. URL http://cran.r-project.org/package=copula. R package version 0.999-18.

M. Hofert, R. Huser, and A. Prasad. Hierarchical Archimax copulas. *Journal of Multivariate Analysis*, 167:195–211, 2018a.

M. Hofert, I. Kojadinovic, M. Mächler, and J. Yan. *Elements of Copula Modeling with R*. Springer Use R! Series, 2018b. ISBN 978-3-319-89635-9. doi: 10.1007/978-3-319-89635-9. URL http://www.springer.com/de/book/9783319896342.

M. Hofert, A. Prasad, and M. Zhu. Quasi-random sampling for multivariate distributions via generative neural networks. *arXiv preprint arXiv:1811.00683*, 2018c.

M. Hofert, W. Oldford, A. Prasad, and M. Zhu. A framework for measuring association of random vectors via collapsed random variables. *Journal of Multivariate Analysis*, 172: 5–27, 2019.

M. Hofert, A. Prasad, and M. Zhu. Multivariate time-series modeling with generative neural networks. *arXiv preprint arXiv:2002.10645*, 2020.

T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond, 2016.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

R. Huser and A. Davison. Composite likelihood estimation for the Brown–Resnick process. *Biometrika*, page ass089, 2013.

J. Hüsler and R.D. Reiss. Maxima of normal random vectors: Between independence and complete dependence. *Statistics & Probability Letters*, 7(4):283–286, 1989. doi: 10.1016/0167-7152(89)90106-5.

P. Jaworski, F. Durante, W. K. Härdle, and T. Rychlik, editors. *Copula Theory and Its Applications*, volume 198 of *Lecture Notes in Statistics – Proceedings*. Springer, 2010.

H. Joe. *Multivariate Models and Dependence Concepts*. Chapman & Hall/CRC, Dordrecht, 1997.

H. Joe. *Dependence modeling with copulas*. Chapman and Hall/CRC, 2014.

H. Joe and P. Sang. Multivariate models for dependent clusters of variables with conditional independence given aggregation variables. *Computational Statistics & Data Analysis*, 97: 114–132, 2016.

E. Jondeau and M. Rockinger. The copula–GARCH model of conditional dependencies: An international stock market application. *Journal of International Money and Finance*, 25:827–853, 2006.

J. Josse and S. Holmes. Measuring multivariate association and beyond. *Statistics surveys*, 10:132, 2016.

Z. Kabluchko, M. Schlather, and L. de Haan. Stationary max-stable fields associated to negative definite functions. *The Annals of Probability*, pages 2042–2065, 2009.

K. W. Kaufmann, G. H. Lemmon, S. L. DeLuca, J. H. Sheehan, and J. Meiler. Practically useful: what the rosetta protein modeling suite can do for you. *Biochemistry*, 49(14): 2987–2998, 2010.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2014. URL https://arxiv.org/abs/1412.6980.

D. P. Kingma and M. Welling. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.

J. B. Kinney and G. S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980, 2017.

I. Kojadinovic and M. Holmes. Tests of independence among continuous random vectors based on cramér–von mises functionals of the empirical copula process. *Journal of Multivariate Analysis*, 100(6):1137–1154, 2009.

R. Krzysztofowicz. The case for probabilistic forecasting in hydrology. *Journal of hydrology*, 249(1-4):2–9, 2001.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

P. L'Ecuyer. Randomized quasi-monte carlo: An introduction for practitioners. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 29–52. Springer, 2016.

A. J. Lee. *U-statistics: Theory and Practice*. Dekker, 1990.

D. Lee and H. Joe. Multivariate extreme value copulas with factor and tree dependence structures. *Extremes*, 2017. To appear.

C. Lemieux. *Monte Carlo and Quasi–Monte Carlo Sampling*. Springer, 2009.

C.L Li, W.C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015.

D. Lopez-Paz, P. Hennig, and B. Schölkopf. The randomized dependence coefficient. In *Advances in Neural Information Processing Systems*, pages 1–9, 2013.

K. V. Mardia, G. Hughes, C. C. Taylor, and H. Singh. A multivariate von Mises distribution with applications to bioinformatics. *Canadian Journal of Statistics*, 36(1):99–109, 2008.

A. W. Marshall and I. Olkin. Families of multivariate distributions. *Journal of the American Statistical Association*, 83(403):834–841, 1988.

D. McFadden. Modeling the choice of residential location. In A. Karlqvist, F. Snickars, and J. Weibull, editors, *Spatial Interaction Theory and Planning Models*, pages 75–96. Elsevier North Holland, 1978.

A. J. McNeil. Sampling nested Archimedean copulas. *Journal of Statistical Computation and Simulation*, 78(6):567–581, 2008.

A. J. McNeil and J. Nešlehová. Multivariate Archimedean copulas, d-monotone functions and l_1-norm symmetric distributions. *The Annals of Statistics*, 37(5b):3059–3097, 2009.

A. J. McNeil, R. Frey, and P. Embrechts. *Quantitative Risk Management: Concepts, Techniques, Tools*. Princeton University Press, 2 edition, 2015.

J. Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.

R. Mesiar and V. Jágr. d-dimensional dependence functions and Archimax copulas. *Fuzzy Sets and Systems*, 228:78–87, 2013.

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

A. Min and C. Czado. Scomdy models based on pair-copula constructions with application to exchange rates. *Computational Statistics & Data Analysis*, 76:523–535, 2014.

G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2924–2932, 2014.

R. B. Nelsen. *An Introduction to Copulas*. Springer-Verlag, 2006.

C. R. Nelson and A. F. Siegel. Parsimonious modeling of yield curves. *Journal of business*, pages 473–489, 1987.

H. Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63. SIAM, 1992.

M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com.

A. K. Nikoloulopoulos, H. Joe, and H. Li. Extreme value properties of multivariate t copulas. *Extremes*, 12:129–148, 2009.

J. P. Nolan. *Stable Distributions – Models for Heavy Tailed Data*. Birkhäuser, 2017. URL http://fs2.american.edu/jpnolan/www/stable/chap1.pdf.

D. H. Oh and A. J. Patton. Modeling dependence in high dimensions with factor copulas. *Journal of Business & Economic Statistics*, 35(1):139–154, 2017.

T. Opitz. Extremal t processes: Elliptical domain of attraction and a spectral representation. *Journal of Multivariate Analysis*, 122:409–413, 2013.

A. B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 299–317. Springer, 1995.

A. B. Owen. Scrambled net variance for integrals of smooth functions. *The Annals of Statistics*, 25(4):1541–1562, 1997a.

A. B. Owen. Monte carlo variance of scrambled net quadrature. *SIAM Journal on Numerical Analysis*, 34(5):1884–1910, 1997b.

A. B. Owen. Variance and discrepancy with alternative scramblings. *ACM Transactions of Modeling and Computer Simulation*, 13(4), 2003.

A. B. Owen. Local antithetic sampling with scrambled nets. *The Annals of Statistics*, 36 (5):2319–2343, 2008.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

A. J. Patton. Modelling asymmetric exchange rate dependence. *International Economic Review*, 47(2):527–556, 2006. URL http://public.econ.duke.edu/~ap172/Patton_IER_2006.pdf.

A. J. Patton. A review of copula models for economic time series. *Journal of Multivariate Analysis*, 110:4–18, 2012.

M. D. Penrose. Semi-min-stable processes. *Annals of Probability*, 20(3):1450–1463, 1992. doi: 10.1214/aop/1176989700.

T. Pillards and R. Cools. Using box-muller with low discrepancy points. In *International Conference on Computational Science and Its Applications*, pages 780–788. Springer, 2006.

G. Puccetti and M. Scarsini. Multivariate comonotonicity. *Journal of Multivariate Analysis*, 101:291–304, 2010.

I. Radović, I.M. Sobol, and R.F. Tichy. Quasi-monte carlo methods for numerical integration: Comparison of different low discrepancy sequences. *Monte Carlo Methods and Applications*, 2(1):1–14, 1996.

B. Rémillard and O. Scaillet. Testing for equality between two copulas. *Journal of Multivariate Analysis*, 100(3):377–386, 2009.

A. Rényi. On measures of dependence. *Acta Mathematica Hungarica*, 10:441–451, 1959.

D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E.S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

S. I. Resnick. *A Probability Path*. Birkhäuser, 2014.

P. Ressel. Homogeneous distributions – and a spectral representation of classical mean values and stable tail dependence functions. *Journal of Multivariate Analysis*, 117:246–256, 2013.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

M. Scarsini. On measures of concordance. *Stochastica*, 8(3):201–218, 1984.

M. Scheuerer and T. M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

M. Schlather. Models for stationary max-stable random fields. *Extremes*, 5(1):33–44, 2002.

F. Schmid, R. Schmidt, T. Blumentritt, S. Gaißer, and M. Ruppert. Copula-based measures of multivariate association. In *Copula theory and its applications*, pages 209–236. Springer, 2010.

B. Schweizer and E. F. Wolff. On nonparametric measures of dependence for random variables. *The Annals of Statistics*, 9:879–885, 1981.

J. Segers. Max-stable models for multivariate extremes. 2012. URL http://arxiv.org/abs/1204.0332.

N. Simon and R. Tibshirani. Comment on "detecting novel associations in large data sets" by Reshef Et Al, Science dec 16, 2011. *arXiv preprint arXiv:1401.7645*, 2014.

A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de L'Institut de Statistique de L'Université de Paris*, 8:229–231, 1959.

I. M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 7(4):784–802, 1967.

A. G. Stephenson. Simulating multivariate extreme value distributions of logistic type. *Extremes*, 6(1):49–59, 2003.

G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

J. A. Tawn. Modelling multivariate extreme value distributions. *Biometrika*, 77(2):245–253, 1990.

M. Telgarsky. benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539, 2016.

I. O. Tolstikhin, S. Gelly, O. Bousquet, C.J. Simon-Gabriel, and B. Schölkopf. Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems*, pages 5424–5433, 2017.

Y. K. Tse and A. K. C. Tsui. A multivariate generalized autoregressive conditional heteroscedasticity model with time-varying correlations. *Journal of Business & Economic Statistics*, 20(3):351–362, 2002.

C. Varin, N. Reid, and D. Firth. An overview of composite likelihood methods. *Statistica Sinica*, pages 5–42, 2011.

S. Vettori, R. Huser, and M. G. Genton. Bayesian clustering and dimension reduction in multivariate extremes. 2017. submitted.

C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong. Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems*, 29(3):1033–1044, 2013.

D. Wang. Visual inference of independence. Master's thesis, 2013. University of Waterloo.

S. Zhao, J. Song, and S. Ermon. Infovae: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

APPENDICES

Appendix A

Additional details for Chapter 2

A.1 Proofs and additional details for the asymptotic framework

A.1.1 Proof of Proposition 2.2.2

Proof. We begin by explicitly writing out the population version of our measure of association. For a general collapsing function S,

$$\chi(\boldsymbol{X}, \boldsymbol{Y}) = \rho\{S(\boldsymbol{X}), S(\boldsymbol{Y})\} = \frac{\mu_{xy} - \mu_x\mu_y}{\sqrt{\mu_{xx} - \mu_x^2}\sqrt{\mu_{yy} - \mu_y^2}}.$$

Case 1: S is a p-variate function

Based on the n independent random samples, define

$$m_x^{(1)} = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{X}_i), \quad m_y^{(1)} = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{Y}_i), \quad m_{xx}^{(1)} = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{X}_i)^2,$$
$$m_{yy}^{(1)} = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{Y}_i)^2, \quad m_{xy}^{(1)} = \frac{1}{n}\sum_{i=1}^{n} S(\boldsymbol{X}_i)S(\boldsymbol{Y}_i).$$

By Hoeffding (1948), $m_x^{(1)}$, $m_y^{(1)}$, $m_{xx}^{(1)}$, $m_{yy}^{(1)}$, $m_{xy}^{(1)}$ are U-statistics for μ_x, μ_y, μ_{xx}, μ_{yy}, μ_{xy} respectively. Following from Hoeffding's decomposition theorem, see Lee (1990, Chapter 3),

we can conclude that, as $n \to \infty$,

$$\sqrt{n}(m_x^{(1)} - \mu_x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{S(\boldsymbol{X}_i) - \mu_x\} + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_y^{(1)} - \mu_y) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{S(\boldsymbol{Y}_i) - \mu_y\} + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{xx}^{(1)} - \mu_{xx}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{S(\boldsymbol{X}_i)^2 - \mu_{xx}\} + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{yy}^{(1)} - \mu_{yy}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{S(\boldsymbol{Y}_i)^2 - \mu_{yy}\} + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{xy}^{(1)} - \mu_{xy}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \{S(\boldsymbol{X}_i)S(\boldsymbol{Y}_i) - \mu_{xy}\} + o_{\mathrm{p}}(1).$$

Combining all the terms, it follows that, for $n \to \infty$,

$$\sqrt{n}(m_x^{(1)} - \mu_x,\ m_y^{(1)} - \mu_y,\ m_{xx}^{(1)} - \mu_{xx},\ m_{yy}^{(1)} - \mu_{yy},\ m_{xy}^{(1)} - \mu_{xy})^\top \xrightarrow{\mathrm{d}} \mathrm{N}_5(\boldsymbol{0}, \Sigma_1),$$

where Σ_1 is the covariance matrix of the random vector $(S(\boldsymbol{X}), S(\boldsymbol{Y}), S(\boldsymbol{X})^2, S(\boldsymbol{Y})^2, S(\boldsymbol{X})S(\boldsymbol{Y}))$.

Then, we construct an estimator for the population version of the measure of association, $\chi(\boldsymbol{X}, \boldsymbol{Y})$, as a function of U-statistics.

$$\chi_n(\boldsymbol{X}, \boldsymbol{Y}) = f(m_x^{(1)}, m_y^{(1)}, m_{xx}^{(1)}, m_{yy}^{(1)}, m_{xy}^{(1)}) = \frac{m_{xy}^{(1)} - m_x^{(1)} m_y^{(1)}}{\sqrt{m_{xx}^{(1)} - (m_x^{(1)})^2}\sqrt{m_{yy}^{(1)} - (m_y^{(1)})^2}},$$

where $m_x^{(1)}$, $m_y^{(1)}$, $m_{xx}^{(1)}$, $m_{yy}^{(1)}$, and $m_{xy}^{(1)}$ are the sample quantities as previously defined. Then, by the delta method we have, as $n \to \infty$,

$$\sqrt{n}\{\chi_n(\boldsymbol{X}, \boldsymbol{Y}) - \chi(\boldsymbol{X}, \boldsymbol{Y})\} \xrightarrow{d} \mathrm{N}(0, \sigma_\chi^2),$$

where $\sigma_\chi^2 = (\nabla f_{5\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_1 (\nabla f_{5\times 1}|_{\boldsymbol{\mu}})$ and the gradient vector is evaluated at $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_{xx}, \mu_{yy}, \mu_{xy})$.

Case 2: S is a $2p$-variate function

Consider

$$m_x^{(2)} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j>i}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j), \quad m_y^{(2)} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j>i}^{n} S(\boldsymbol{Y}_i, \boldsymbol{Y}_j),$$
$$m_{xx}^{(2)} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j>i}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j)^2, \quad m_{yy}^{(2)} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j>i}^{n} S(\boldsymbol{Y}_i, \boldsymbol{Y}_j)^2,$$
$$m_{xy}^{(2)} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j>i}^{n} S(\boldsymbol{X}_i, \boldsymbol{X}_j) S(\boldsymbol{Y}_i, \boldsymbol{Y}_j).$$

Similar to the setup presented in Case 1, these sample quantities are natural U-statistics for their corresponding population versions. Again following from Hoeffding's decomposition theorem, we have that, as $n \to \infty$,

$$\sqrt{n}(m_x^{(2)} - \mu_x) = \frac{2}{\sqrt{n}} \sum_{i=1}^{n} [\mathbb{E}_{\boldsymbol{X}'}\Big\{S(\boldsymbol{X}_i, \boldsymbol{X}')\Big\} - \mu_x] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_y^{(2)} - \mu_y) = \frac{2}{\sqrt{n}} \sum_{i=1}^{n} [\mathbb{E}_{\boldsymbol{Y}'}\Big\{S(\boldsymbol{Y}_i, \boldsymbol{Y}')\Big\} - \mu_y] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{xx}^{(2)} - \mu_{xx}) = \frac{2}{\sqrt{n}} \sum_{i=1}^{n} [\mathbb{E}_{\boldsymbol{X}'}\Big\{S(\boldsymbol{X}_i, \boldsymbol{X}')^2\Big\} - \mu_{xx}] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{yy}^{(2)} - \mu_{yy}) = \frac{2}{\sqrt{n}} \sum_{i=1}^{n} [\mathbb{E}_{\boldsymbol{Y}'}\Big\{S(\boldsymbol{Y}_i, \boldsymbol{Y}')^2\Big\} - \mu_{yy}] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{xy}^{(2)} - \mu_{xy}) = \frac{2}{\sqrt{n}} \sum_{i=1}^{n} [\mathbb{E}_{(\boldsymbol{X}', \boldsymbol{Y}')}\Big\{S(\boldsymbol{X}_i, \boldsymbol{X}') S(\boldsymbol{Y}_i, \boldsymbol{Y}')\Big\} - \mu_{xy}] + o_{\mathrm{p}}(1),$$

where the conditional expectations in the expressions above represent the first-order Hoeffding decomposition of the corresponding U-statistic. Combining all the terms, it follows that

$$\sqrt{n}(m_x^{(2)} - \mu_x,\ m_y^{(2)} - \mu_y,\ m_{xx}^{(2)} - \mu_{xx},\ m_{yy}^{(2)} - \mu_{yy},\ m_{xy}^{(2)} - \mu_{xy})^\top \xrightarrow{\mathrm{d}} \mathrm{N}_5(\boldsymbol{0}, 4\Sigma_2),$$

where Σ_2 is the covariance matrix of the random vector

$$\Big(\mathbb{E}_{\boldsymbol{X}'}\Big\{S(\boldsymbol{X}, \boldsymbol{X}')\Big\}, \mathbb{E}_{\boldsymbol{Y}'}\Big\{S(\boldsymbol{Y}, \boldsymbol{Y}')\Big\}, \mathbb{E}_{\boldsymbol{X}'}\Big\{S(\boldsymbol{X}, \boldsymbol{X}')^2\Big\}, \mathbb{E}_{\boldsymbol{Y}'}\Big\{S(\boldsymbol{Y}, \boldsymbol{Y}')^2\Big\},$$
$$\mathbb{E}_{(\boldsymbol{X}', \boldsymbol{Y}')}\Big\{S(\boldsymbol{X}, \boldsymbol{X}') S(\boldsymbol{Y}, \boldsymbol{Y}')\Big\}\Big).$$

We can then construct an estimator for the population version of the measure of association $\chi(\boldsymbol{X}, \boldsymbol{Y})$ exactly as we did in Case 1 but instead with the use of the sample quantities $m_x^{(2)}$, $m_y^{(2)}$, $m_{xx}^{(2)}$, $m_{yy}^{(2)}$, $m_{xy}^{(2)}$. By the delta method we have that, as $n \to \infty$,

$$\sqrt{n}\{\chi_n(\boldsymbol{X}, \boldsymbol{Y}) - \chi(\boldsymbol{X}, \boldsymbol{Y})\} \xrightarrow{d} \mathrm{N}(0, \sigma_\chi^2),$$

where $\sigma_\chi^2 = (\nabla f_{5\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_2 (\nabla f_{5\times 1}|_{\boldsymbol{\mu}})$. □

A.1.2 Additional details for estimation of the asymptotic variance

Analytical forms of the components of the gradient vector are given below; note that $\boldsymbol{m} = (m_x, m_y, m_{xx}, m_{yy}, m_{xy})$ acts as a place holder for both $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ defined in Remark 2.2.3:

$$\begin{aligned}
\nabla f_1|_{\boldsymbol{m}} &= \frac{m_x(m_{xy} - m_x m_y)}{(m_{xx} - m_x^2)^{3/2}\sqrt{m_{yy} - m_y^2}} - \frac{m_y}{\sqrt{m_{xx} - m_x^2}\sqrt{m_{yy} - m_y^2}}, \\
\nabla f_2|_{\boldsymbol{m}} &= \frac{m_y(m_{xy} - m_x m_y)}{(m_{yy} - m_y^2)^{3/2}\sqrt{m_{xx} - m_x^2}} - \frac{m_x}{\sqrt{m_{xx} - m_x^2}\sqrt{m_{yy} - m_y^2}}, \\
\nabla f_3|_{\boldsymbol{m}} &= -\frac{m_{xy} - m_x m_y}{2(m_{xx} - m_x^2)^{3/2}\sqrt{m_{yy} - m_y^2}}, \quad \nabla f_4|_{\boldsymbol{m}} = -\frac{m_{xy} - m_x m_y}{2(m_{yy} - m_y^2)^{3/2}\sqrt{m_{xx} - m_x^2}}, \\
\nabla f_5|_{\boldsymbol{m}} &= \frac{1}{\sqrt{m_{xx} - m_x^2}\sqrt{m_{yy} - m_y^2}}.
\end{aligned}$$

A.1.3 Additional asymptotic results

An estimator τ_n of $\tau\{S(\boldsymbol{X}), S(\boldsymbol{Y})\} = \rho[\mathbb{1}\{S(\boldsymbol{X}) \le S(\boldsymbol{X}')\}, \mathbb{1}\{S(\boldsymbol{Y}) \le S(\boldsymbol{Y}')\}]$ can be constructed through the U-statistic framework with the corresponding asymptotic results following as a consequence of Proposition 1.

Corollary A.1.1 (Asymptotic distribution of τ_n)
Let $(\boldsymbol{X}', \boldsymbol{Y}')$, $(\boldsymbol{X}'', \boldsymbol{Y}'')$, and $(\boldsymbol{X}''', \boldsymbol{Y}''')$ be independent copies of $(\boldsymbol{X}, \boldsymbol{Y})$. Suppose $\tau_n(\boldsymbol{X}, \boldsymbol{Y})$ is constructed as a function of U-statistics. Then, as $n \to \infty$,

$$\sqrt{n}[\tau_n\{S(\boldsymbol{X}), S(\boldsymbol{Y})\} - \tau\{S(\boldsymbol{X}), S(\boldsymbol{Y})\}] \xrightarrow{d} \mathrm{N}(0, \sigma_\tau^2),$$

where

$$\sigma_\tau^2 = \begin{cases} 4(\nabla f_{3\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_1 (\nabla f_{3\times 1}|_{\boldsymbol{\mu}}), & \text{if } S \text{ is a } p\text{-variate function,} \\ 16(\nabla f_{3\times 1}|_{\boldsymbol{\mu}})^\top \Sigma_2 (\nabla f_{3\times 1}|_{\boldsymbol{\mu}}), & \text{if } S \text{ is a } 2p\text{-variate function.} \end{cases}$$

Here, $\nabla f_{3\times 1}|_{\boldsymbol{\mu}}$ denotes the gradient vector of the function

$$f(a,b,c) = \frac{c-ab}{\sqrt{a-a^2}\sqrt{b-b^2}},$$

evaluated at the population mean $\boldsymbol{\mu} = (\mu_x, \mu_y, \mu_{xy})$, where $\mu_x = \mathbb{P}\{S(\boldsymbol{X}) \leq S(\boldsymbol{X}')\}$, $\mu_y = \mathbb{P}\{S(\boldsymbol{Y}) \leq S(\boldsymbol{Y}')\}$ and $\mu_{xy} = \mathbb{P}\{S(\boldsymbol{X}) \leq S(\boldsymbol{X}'), S(\boldsymbol{Y}) \leq S(\boldsymbol{Y}')\}$. Furthermore, Σ_1 denotes the covariance matrix of

$$(\mathbb{P}\{S(\boldsymbol{X}) \leq S(\boldsymbol{X}') \,|\, \boldsymbol{X}\},\ \mathbb{P}\{S(\boldsymbol{Y}) \leq S(\boldsymbol{Y}') \,|\, \boldsymbol{Y}\},\ \mathbb{P}\{S(\boldsymbol{X}) \leq S(\boldsymbol{X}'),\ S(\boldsymbol{Y}) \leq S(\boldsymbol{Y}') \,|\, \boldsymbol{X}, \boldsymbol{Y}\})$$

and Σ_2 denotes the covariance matrix of

$$\begin{aligned} &(\mathbb{P}\{S(\boldsymbol{X}, \boldsymbol{X}') \leq S(\boldsymbol{X}'', \boldsymbol{X}''') \,|\, \boldsymbol{X}\},\ \mathbb{P}\{S(\boldsymbol{Y}, \boldsymbol{Y}') \leq S(\boldsymbol{Y}'', \boldsymbol{Y}''') \,|\, \boldsymbol{Y}\}, \\ &\mathbb{P}\{S(\boldsymbol{X}, \boldsymbol{X}') \leq S(\boldsymbol{X}'', \boldsymbol{X}'''),\ S(\boldsymbol{Y}, \boldsymbol{Y}') \leq S(\boldsymbol{Y}'', \boldsymbol{Y}''') \,|\, \boldsymbol{X}, \boldsymbol{Y}\}), \end{aligned} \tag{A.1}$$

where $\mathbb{P}\{\cdot \,|\, \cdot\}$ denotes a conditional probability.

Proof. We begin by writing the population version of our measure of association explicitly. For a general collapsing function S,

$$\tau\{S(\boldsymbol{X}), S(\boldsymbol{Y})\} = \rho[\mathbb{1}\{S(\boldsymbol{X}) \leq S(\boldsymbol{X}')\}, \mathbb{1}\{S(\boldsymbol{Y}) \leq S(\boldsymbol{Y}')\}] = \frac{\mu_{xy} - \mu_x \mu_y}{\sqrt{\mu_x - \mu_x^2}\sqrt{\mu_y - \mu_y^2}}.$$

Case 1: S is a p-variate function

Based on a random sample $(\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n)$, estimators $m_x^{(1)}, m_y^{(1)}$, and $m_{xy}^{(1)}$ can be constructed using the setup of the proof of Case 2 of Proposition 1. The convergence result follows from a similar delta-method argument.

Case 2: S is $2p$-variate function

The sample quantities

$$m_x^{(2)} = \frac{1}{\binom{n}{4}} \sum_{i<j<k<l} \mathbb{1}\{S(\boldsymbol{X}_i, \boldsymbol{X}'_j) \leq S(\boldsymbol{X}''_k, \boldsymbol{X}'''_l)\},$$
$$m_y^{(2)} = \frac{1}{\binom{n}{4}} \sum_{i<j<k<l} \mathbb{1}\{S(\boldsymbol{Y}_i, \boldsymbol{Y}'_j) \leq S(\boldsymbol{Y}''_k, \boldsymbol{Y}'''_l)\},$$
$$m_{xy}^{(2)} = \frac{1}{\binom{n}{4}} \sum_{i<j<k<l} \mathbb{1}\{S(\boldsymbol{X}_i, \boldsymbol{X}'_j) \leq S(\boldsymbol{X}''_k, \boldsymbol{X}'''_l), S(\boldsymbol{Y}_i, \boldsymbol{Y}'_j) \leq S(\boldsymbol{Y}''_k, \boldsymbol{Y}'''_l)\}$$

are natural U-statistics for their corresponding population versions. Then, following Hoeffding's decomposition theorem, we have that, as $n \to \infty$,

$$\sqrt{n}(m_x^{(2)} - \mu_x) = \frac{4}{\sqrt{n}} \sum_{i=1}^{n} \Big[\mathbb{P}\{S(\boldsymbol{X}_i, \boldsymbol{X}') \leq S(\boldsymbol{X}'', \boldsymbol{X}''') \,|\, \boldsymbol{X}\} - \mu_x\Big] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_y^{(2)} - \mu_y) = \frac{4}{\sqrt{n}} \sum_{i=1}^{n} \Big[\mathbb{P}\{S(\boldsymbol{Y}_i, \boldsymbol{Y}') \leq S(\boldsymbol{Y}'', \boldsymbol{Y}''') \,|\, \boldsymbol{Y}\} - \mu_y\Big] + o_{\mathrm{p}}(1),$$
$$\sqrt{n}(m_{xy}^{(2)} - \mu_{xy}) = \frac{4}{\sqrt{n}} \sum_{i=1}^{n} \Big[\mathbb{P}\{S(\boldsymbol{X}_i, \boldsymbol{X}') \leq S(\boldsymbol{X}'', \boldsymbol{X}'''),$$
$$S(\boldsymbol{Y}_i, \boldsymbol{Y}') \leq S(\boldsymbol{Y}'', \boldsymbol{Y}''') \,|\, \boldsymbol{X}, \boldsymbol{Y}\} - \mu_{xy}\Big] + o_{\mathrm{p}}(1),$$

where the conditional probabilities in the expressions above represent the first order Hoeffding decomposition of the corresponding U-statistic. Combining all the terms, it follows that

$$\sqrt{n}(m_x^{(2)} - \mu_x,\ m_y^{(2)} - \mu_y,\ m_{xy}^{(2)} - \mu_{xy})^\top \xrightarrow{\mathrm{d}} \mathrm{N}_3(\boldsymbol{0}, 16\Sigma_2),$$

where Σ_2 denotes the covariance matrix of the random vector defined in (A.1). One can then construct an estimator using $\tau_n\{S(\boldsymbol{X}), S(\boldsymbol{Y})\} = f(m_x^{(2)}, m_y^{(2)}, m_{xy}^{(2)})$ where f is defined as in the claim Using the delta method, the convergence result follows. □

Remark A.1.2

In the U-statistic framework, one usually works with symmetric kernels as noted in Lee (1990, Chapter 1). For choices of collapsing functions which would yield non-symmetric

kernels, one can easily replace the kernel with a symmetric variant. Suppose for example $\phi(X_1, \ldots, X_m)$ is a kernel of order m. Then, the symmetric variant can be constructed as

$$\phi(X_1, \ldots, X_m) = \frac{1}{m!} \sum_{\alpha_1, \ldots, \alpha_m} \phi(X_{\alpha_1}, \ldots, X_{\alpha_m}),$$

where the summation is taken over all permutations $(\alpha_1, \ldots, \alpha_m)$ of $(1, \ldots, m)$. By replacing any non-symmetric kernel with its symmetric variant, the rest of the derivation for the asymptotic distribution would then naturally follow.

A.2 Additional example of a collapsed copula for the maximum collapsing function

Example A.2.1 (Meta Archimax copula model and the maximum collapsing function) Let $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y}) \sim F_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) = C\Big\{F_X(x_1), \ldots, F_X(x_p), F_Y(y_1), \ldots, F_Y(y_q)\Big\}$, where $X_j \sim F_X$, $j \in \{1, \ldots, p\}$, and $Y_k \sim F_Y$, $k \in \{1, \ldots, q\}$, are continuously distributed. Furthermore, let C be an Archimax copula with generator ψ and stable tail dependence function ℓ, that is,

$$C(u_1, \ldots, u_p, v_1, \ldots, v_q) = \psi\Big[\ell\{\psi^{-1}(u_1), \ldots, \psi^{-1}(u_p), \psi^{-1}(v_1), \ldots, \psi^{-1}(v_q)\}\Big];$$

see Charpentier et al. (2014b) for more details. By Aulbach et al. (2015), ℓ allows for the representation

$$\ell(x_1, \ldots, x_p, y_1, \ldots, y_q) = \mathbb{E}[\max\{\max_{1 \le j \le p}(|x_j| W_{1j}), \max_{1 \le k \le q}(|y_k| W_{2k})\}],$$

where the generators $W_{11}, \ldots, W_{1p}, W_{21}, \ldots, W_{2q}$ satisfy $W_{1j} \ge 0$, $\mathbb{E}(W_{1j}) = 1$ for $j \in \{1, \ldots, p\}$ and $W_{2k} \ge 0$, $\mathbb{E}(W_{2k}) = 1$ for $k \in \{1, \ldots, q\}$.

Let $C_{\boldsymbol{X}}$ and $C_{\boldsymbol{Y}}$ denote the p- and q-dimensional marginal copulas of C corresponding to $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. Note that they are Archimax copulas with generator ψ and stable tail dependence functions

$$\ell_{\boldsymbol{X}} = \ell(\boldsymbol{x}, 0, \ldots, 0) = \mathbb{E}\{\max_{1 \le j \le p}(|x_j| W_{1j})\} \quad \text{and} \quad \ell_{\boldsymbol{Y}} = \ell(0, \ldots, 0, \boldsymbol{y}) = \mathbb{E}\{\max_{1 \le k \le q}(|x_k| W_{2k})\},$$

Consider the maximum collapsing function S. Then,

$$\begin{aligned}
S(\boldsymbol{X}) \sim F_{S(\boldsymbol{X})}(x) &= C_{\boldsymbol{X}}\{F_X(x), \ldots, F_X(x)\} = \psi\Big(\ell_{\boldsymbol{X}}[\psi^{-1}\{F_X(x)\}, \ldots, \psi^{-1}\{F_X(x)\}]\Big) \\
&= \psi\Big\{\mathbb{E}(\max_{1\le j\le p}[|\psi^{-1}\{F_X(x)\}|W_{1j}])\Big\} = \psi\Big[\psi^{-1}\{F_X(x)\}\mathbb{E}\{\max_{1\le j\le p}(W_{1j})\}\Big], \\
S(\boldsymbol{Y}) \sim F_{S(\boldsymbol{Y})}(y) &= C_{\boldsymbol{Y}}\{F_Y(y), \ldots, F_Y(y)\} = \psi\Big(\ell_{\boldsymbol{Y}}[\psi^{-1}\{F_Y(y)\}, \ldots, \psi^{-1}\{F_Y(y)\}]\Big) \\
&= \psi\Big\{\mathbb{E}(\max_{1\le k\le q}[|\psi^{-1}\{F_Y(y)\}|W_{2k}])\Big\} = \psi\Big[\psi^{-1}\{F_Y(y)\}\mathbb{E}\{\max_{1\le k\le q}(W_{2k})\}\Big].
\end{aligned}$$

For notational convenience let $c_1 = \mathbb{E}\{\max_{1\le j\le p}(W_{1j})\}$ and $c_2 = \mathbb{E}\{\max_{1\le k\le q}(W_{2k})\}$. The quantiles of $F_{S(\boldsymbol{X})}(x)$ and $F_{S(\boldsymbol{Y})}(y)$ are

$$F^-_{S(\boldsymbol{X})}(u) = F^-_X\Big[\psi_1\{\psi_1^{-1}(u)/c_1\}\Big] \quad \text{and} \quad F^-_{S(\boldsymbol{Y})}(v) = F^-_Y\Big[\psi_2\{\psi_2^{-1}(v)/c_2\}\Big],$$

respectively. Proposition 2.3.3 implies that the collapsed copula equals

$$\begin{aligned}
C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v) &= F_{\boldsymbol{X},\boldsymbol{Y}}\{F^-_{S(\boldsymbol{X})}(u), \ldots, F^-_{S(\boldsymbol{X})}(u), F^-_{S(\boldsymbol{Y})}(v), \ldots, F^-_{S(\boldsymbol{Y})}(v)\} \\
&= C\Big\{F_X\Big(F^-_X\Big[\psi\{\psi^{-1}(u)/c_1\}\Big]\Big), \ldots, F_X\Big(F^-_X\Big[\psi_1\{\psi^{-1}(u)/c_1\}\Big]\Big), \\
&\qquad F_Y\Big(F^-_Y\Big[\psi\{\psi^{-1}(v)/c_2\}\Big]\Big), \ldots, F_Y\Big(F^-_Y\Big[\psi\{\psi^{-1}(v)/c_2\}\Big]\Big)\Big\} \\
&= C\Big[\psi\{\psi^{-1}(u)/c_1\}, \ldots, \psi\{\psi_1^{-1}(u)/c_1\}, \psi\{\psi^{-1}(v)/c_2\}, \ldots, \psi\{\psi^{-1}(v)/c_2\}\Big] \\
&= \psi\Big[\ell\{\psi^{-1}(u)/c_1, \ldots, \psi^{-1}(u)/c_1, \psi^{-1}(v)/c_2, \ldots, \psi^{-1}(v)/c_2\}\Big] \\
&= \psi\Big\{\mathbb{E}\Big(\max\Big[\max_{1\le j\le p}\{|\psi^{-1}(u)/c_1|W_{1j}\}, \max_{1\le k\le q}\{|\psi^{-1}(v)/c_2|W_{2j}\}\Big]\Big)\Big\} \\
&= \psi\Big\{\mathbb{E}\Big(\max\Big[\psi^{-1}(u)\max_{1\le j\le p}(W_{1j})/\mathbb{E}\{\max_{1\le j\le p}(W_{1j})\}, \\
&\qquad \psi^{-1}(v)\max_{1\le k\le q}(W_{2k})/\mathbb{E}\{\max_{1\le k\le q}(W_{2k})\}\Big]\Big)\Big\} \\
&= \psi\Big(\mathbb{E}\Big[\max\Big\{\psi^{-1}(u)W_1^*, \psi^{-1}(v)W_2^*\Big\}\Big]\Big) \\
&= \psi[\ell^*\{\psi^{-1}(u), \psi^{-1}(v)\}],
\end{aligned}$$

where ℓ^* denotes the stable tail dependence function constructed with the d-norm generator

$$(W_1^*, W_2^*) = \left[\frac{\max_{1\le j\le p}(W_{1j})}{\mathbb{E}\{\max_{1\le j\le p}(W_{1j})\}}, \frac{\max_{1\le k\le q}(W_{2k})}{\mathbb{E}\{\max_{1\le k\le q}(W_{2k})\}}\right];$$

notice that $W_1^*, W_2^* \ge 0$ and $\mathbb{E}(W_1^*) = \mathbb{E}(W_2^*) = 1$.

We thus see that for the maximum collapsing function and groupwise equal margins, the collapsed copula $C_{S(\boldsymbol{X}),S(\boldsymbol{Y})}(u,v)$ is also an Archimax copula with generator ψ but stable tail dependence function ℓ^*. Moreover, the generators W_1^*, W_2^* arise from $W_{11},\dots,W_{1p},W_{21},\dots,W_{2q}$ by applying the maximum collapsing function to $W_{11},\dots,W_{1p}$ and $W_{21},\dots,W_{2q}$, respectively, and scaling them appropriately to satisfy the constraints $\mathbb{E}(W_1^*) = \mathbb{E}(W_2^*) = 1$.

A.3 Measures of association related to the multivariate Kendall distribution

In this section we provide examples which link measures of association the form $\chi(\boldsymbol{X},\boldsymbol{Y})$ with multivariate Kendall distributions. We start with the measure of association resulting from the PIT collapsing function.

Example A.3.1 (Correlation via the joint Kendall distribution)
Since $K_{\boldsymbol{X}}(t_1), K_{\boldsymbol{Y}}(t_2)$ are the distribution functions of $F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})$, respectively, and $K_{\boldsymbol{X},\boldsymbol{Y}}(t_1,t_2)$ is the joint distribution function of $(F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y}))$, Hoeffding's Identity implies that

$$\begin{aligned}\chi(\boldsymbol{X},\boldsymbol{Y}) &= \rho\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\} = \frac{\operatorname{Cov}\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\}}{\sqrt{\operatorname{Var}\{F_{\boldsymbol{X}}(\boldsymbol{X})\}\operatorname{Var}\{F_{\boldsymbol{Y}}(\boldsymbol{Y})\}}} \\ &= \frac{\iint_{[0,1]^2} K_{\boldsymbol{X},\boldsymbol{Y}}(t_1,t_2) - K_{\boldsymbol{X}}(t_1)K_{\boldsymbol{Y}}(t_2)\,\mathrm{d}t_1\mathrm{d}t_2}{\sqrt{\int_{[0,1]} K_{\boldsymbol{X}}(t_1) - K_{\boldsymbol{X}}^2(t_1)\,\mathrm{d}t_1 \int_{[0,1]} K_{\boldsymbol{Y}}(t_2) - K_{\boldsymbol{Y}}^2(t_2)\,\mathrm{d}t_2}}.\end{aligned}$$

Note that the numerator is the (integrated) difference between the joint Kendall distribution of $\boldsymbol{X}$ and $\boldsymbol{Y}$ and the joint Kendall distribution under independence of $\boldsymbol{X}$ and $\boldsymbol{Y}$; $\chi(\boldsymbol{X},\boldsymbol{Y})$ thus represents in some sense how far on average the random vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$ are from independence, thus mimicking the construction of standard bivariate measures of association.

Example A.3.2 (Spearman's rho via the joint Kendall distribution)
One drawback of the measure presented in Example A.3.1 is that it depends on the marginal distributions of the collapsed random variables. To rectify this, we can apply the marginal Kendall distributions $K_{\boldsymbol{X}}$ and $K_{\boldsymbol{Y}}$ to the collapsed random variables $F_{\boldsymbol{X}}(\boldsymbol{X})$ and $F_{\boldsymbol{Y}}(\boldsymbol{Y})$, respectively. The measure will then be a natural multivariate extension of Spearman's rho as it only depends on the Kendall copula. To this end, let $U = K_{\boldsymbol{X}}\{F_{\boldsymbol{X}}(\boldsymbol{X})\}$ and

$V = K_{\boldsymbol{Y}}\{F_{\boldsymbol{Y}}(\boldsymbol{Y})\}$. Then,

$$\begin{aligned}\chi(\boldsymbol{X}, \boldsymbol{Y}) &= \rho\Big[K_{\boldsymbol{X}}\{F_{\boldsymbol{X}}(\boldsymbol{X})\},\ K_{\boldsymbol{Y}}\{F_{\boldsymbol{Y}}(\boldsymbol{Y})\}\Big] = \rho(U, V) = \frac{\mathbb{E}[UV] - 1/4}{1/12} = 12\mathbb{E}[UV] - 3 \\ &= 12 \iint_{[0,1]^2} uv \, \mathrm{d}C_K(u, v) - 3 = \rho_S\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\},\end{aligned}$$

where $C_K(u, v)$ denotes the Kendall copula introduced in (2.6). Thus, $\chi(\boldsymbol{X}, \boldsymbol{Y})$ equals Spearman's rho of $F_{\boldsymbol{X}}(\boldsymbol{X})$ and $F_{\boldsymbol{Y}}(\boldsymbol{Y})$.

Example A.3.3 (Kendall's tau via the joint Kendall distribution)
Similarly, with U and V as defined in Example A.3.2 and that $(\boldsymbol{X}', \boldsymbol{Y}')$ is an independent copy of $(\boldsymbol{X}, \boldsymbol{Y})$, for Kendall's tau we have

$$\begin{aligned}\chi(\boldsymbol{X}, \boldsymbol{Y}) &= \rho[\mathbb{1}\{F_{\boldsymbol{X}}(\boldsymbol{X}) \leq F_{\boldsymbol{X}}(\boldsymbol{X}')\}, \mathbb{1}\{F_{\boldsymbol{Y}}(\boldsymbol{Y}) \leq F_{\boldsymbol{Y}}(\boldsymbol{Y}')\}] = \tau\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\} \\ &= 4 \iint_{[0,1]^2} C_K(u, v) \, \mathrm{d}C_K(u, v) - 1,\end{aligned}$$

where $C_K(u, v)$ denotes the Kendall copula as before and the last equality follows by definition of Kendall's tau of the collapsed random variables in the bivariate case. This measure forms a multivariate extension of Kendall's tau which only depends on the Kendall copula. Note that another multivariate extension of Kendall's tau as described in Section 2.2.3 is given by $\rho(\mathbb{1}\{\boldsymbol{X} \leq \boldsymbol{X}'\}, \mathbb{1}\{\boldsymbol{Y} \leq \boldsymbol{Y}'\})$ with the inequalities understood componentwise.

Example A.3.4 (Tail dependence via Kendall copulas)
In light of using (2.3) for measuring tail dependence between the collapsed random variables, it is easy to see that when using the PIT collapsing function, (2.3) as measure of association corresponds to computing (classical) coefficients of tail dependence of the underlying Kendall copula C_K. For example, if $\boldsymbol{X} \sim F_{\boldsymbol{X}}$, $\boldsymbol{Y} \sim F_{\boldsymbol{Y}}$ with Kendall distributions $K_{\boldsymbol{X}}$, $K_{\boldsymbol{Y}}$, respectively, and if $U = K_{\boldsymbol{X}}\{F_{\boldsymbol{X}}(\boldsymbol{X})\}$, $V = K_{\boldsymbol{Y}}\{F_{\boldsymbol{Y}}(\boldsymbol{Y})\}$ (note that $(U, V) \sim C_K$ in this case), then the coefficient of upper tail dependence can be expressed as

$$\begin{aligned}\lambda_U\{F_{\boldsymbol{X}}(\boldsymbol{X}), F_{\boldsymbol{Y}}(\boldsymbol{Y})\} &= \lim_{u \uparrow 1} \mathbb{P}\{F_{\boldsymbol{Y}}(\boldsymbol{Y}) > K_{\boldsymbol{Y}}^-(u) \,|\, F_{\boldsymbol{X}}(\boldsymbol{X}) > K_{\boldsymbol{X}}^-(u)\} = \lim_{u \uparrow 1} \mathbb{P}(V > u \,|\, U > u) \\ &= \lim_{u \uparrow 1} \frac{1 - 2u + C_K(u, u)}{1 - u}.\end{aligned}$$

Appendix B

Additional details for Chapter 3

B.1 Density of Archimax copulas

For likelihood-based inference on AXCs, it is important to know their density. In this section, we present the general form of the density of AXCs (if it exists) and address how it can be computed numerically.

Proposition B.1.1 (AXC density)
If the respective partial derivatives of ℓ exist and are continuous, the density c of a d-dimensional AXC C is given by

$$c(\boldsymbol{u}) = \left\{ \prod_{j=1}^{d} (\psi^{-1})'(u_j) \right\} \sum_{k=1}^{d} \psi^{(k)}\left[\ell\{\psi^{-1}(\boldsymbol{u})\} \right] \sum_{\pi \in \Pi : |\pi| = k} \prod_{B \in \pi} (\mathrm{D}_B\, \ell)\{\psi^{-1}(\boldsymbol{u})\}, \quad \boldsymbol{u} \in (0,1)^d,$$

where $\psi^{-1}(\boldsymbol{u}) = (\psi^{-1}(u_1), \ldots, \psi^{-1}(u_d))$, Π denotes the set of all partitions π of $\{1, \ldots, d\}$ (with $|\pi|$ denoting the number of elements of π) and $(\mathrm{D}_B\, \ell)(\psi^{-1}(\boldsymbol{u}))$ denotes the partial derivatives of ℓ with respect to the variables with index in B, evaluated at $\psi^{-1}(\boldsymbol{u})$.

Proof. By a multivariate version of Faà di Bruno's Formula, see Hardy (2006), the dth derivative of a composition of two functions $f : \mathbb{R} \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ is given by

$$\begin{aligned} \mathrm{D}\, f\{g(\boldsymbol{x})\} &= \sum_{\pi \in \Pi} \left[f^{(|\pi|)}\{g(\boldsymbol{x})\} \prod_{B \in \pi} \mathrm{D}_B\, g(\boldsymbol{x}) \right] = \sum_{k=1}^{d} \sum_{\pi \in \Pi : |\pi| = k} \left[f^{(|\pi|)}\{g(\boldsymbol{x})\} \prod_{B \in \pi} \mathrm{D}_B\, g(\boldsymbol{x}) \right] \\ &= \sum_{k=1}^{d} \sum_{\pi \in \Pi : |\pi| = k} \left[f^{(k)}\{g(\boldsymbol{x})\} \prod_{B \in \pi} \mathrm{D}_B\, g(\boldsymbol{x}) \right] = \sum_{k=1}^{d} f^{(k)}\{g(\boldsymbol{x})\} \sum_{\pi \in \Pi : |\pi| = k} \prod_{B \in \pi} \mathrm{D}_B\, g(\boldsymbol{x}), \end{aligned}$$

where $\mathrm{D} = \partial^d/(\partial x_d \dots \partial x_1)$, $\mathrm{D}_B = \partial^{|B|}/\prod_{j\in B}\partial x_j$, and $B \in \pi$ means that B runs through all partition elements of π. Assuming that the appearing derivatives exist and are continuous, we obtain from taking $f(x) = \psi(x)$ and $g(\boldsymbol{x}) = \ell\{\psi^{-1}(\boldsymbol{x})\}$ that

$$\begin{aligned} c(\boldsymbol{u}) &= \sum_{k=1}^{d} \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} \frac{\partial^{|B|}}{\prod_{j\in B}\partial u_j} \ell\{\psi^{-1}(\boldsymbol{u})\} \\ &= \left\{\prod_{j=1}^{d} (\psi^{-1})'(u_j)\right\} \sum_{k=1}^{d} \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} (\mathrm{D}_B\,\ell)\{\psi^{-1}(\boldsymbol{u})\}, \quad \boldsymbol{u}\in(0,1)^d, \end{aligned}$$

where the last equality holds since the derivatives of all of $\psi^{-1}(u_1), \dots, \psi^{-1}(u_d)$ (from applying the chain rule) appear in each summand of the sum $\sum_{\pi\in\Pi:|\pi|=k}$ and can thus be taken out of both summations. □

As a quick check of Proposition B.1.1, we can recover the density of ACs and EVCs.

Corollary B.1.2 (AC density as special case)
For $\ell(\boldsymbol{x}) = \sum_{j=1}^{d} x_j$, the density of ACs correctly follows from Proposition B.1.1 by noting that

$$\sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} (\mathrm{D}_B\,\ell)(\boldsymbol{x}) = \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} \mathbb{1}_{\{|B|=1\}} = \sum_{\pi\in\Pi:|\pi|=k} \mathbb{1}_{\{|B|=1 \text{ for all } B\in\pi\}} = \mathbb{1}_{\{k=d\}}.$$

Corollary B.1.3 (EVC density as special case)
For $\psi(t) = \exp(-t)$, $t \geq 0$, the density of EVCs correctly follows from Proposition B.1.1 as one has

$$c(\boldsymbol{u}) = \left\{\prod_{j=1}^{d}\left(-\frac{1}{u_j}\right)\right\} \sum_{k=1}^{d} \exp\Big[-\ell\{-\ln(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} [-(\mathrm{D}_B\,\ell)\{-\ln(\boldsymbol{u})\}], \quad \boldsymbol{u}\in(0,1)^d;$$

see, for example, Castruccio et al. (2016) or Doyon (2013).

The following result provides the general form of the density of AXCs based on the stable tail dependence function ℓ of a Gumbel copula.

Corollary B.1.4 (Density of AXCs with Gumbel stable tail dependence function as special case)
For the stable tail dependence function $\ell(\boldsymbol{x}) = (x_1^{1/\alpha} + \dots + x_d^{1/\alpha})^\alpha$, $\boldsymbol{x} \in [0,\infty)^d$, of a

Gumbel copula with parameter $\alpha \in (0,1]$, the density c of an AXC is given by

$$c(\boldsymbol{u}) = \frac{1}{\alpha^d}\left\{\prod_{j=1}^{d}(\psi^{-1})'(u_j)\psi^{-1}(u_j)^{\frac{1}{\alpha}-1}\right\} \cdot \sum_{k=1}^{d}\psi^{(k)}\left[\left\{\sum_{j=1}^{d}\psi^{-1}(u_j)^{\frac{1}{\alpha}}\right\}^{\alpha}\right]\left\{\sum_{j=1}^{d}\psi^{-1}(u_j)^{\frac{1}{\alpha}}\right\}^{\alpha k-d}\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}(\alpha)_{|B|}, \quad \boldsymbol{u}\in(0,1)^d,$$

where $(\alpha)_{|B|} = \prod_{l=0}^{|B|-1}(\alpha - l)$ denotes the falling factorial.

Proof. For the stable tail dependence function $\ell(\boldsymbol{x}) = (x_1^{1/\alpha} + \cdots + x_d^{1/\alpha})^\alpha$, $\boldsymbol{x} \in [0,\infty)^d$, $\alpha \in (0,1]$, one has

$$\mathrm{D}_B\,\ell(\boldsymbol{x}) = (\alpha)_{|B|}\left(\sum_{j=1}^{d}x_j^{1/\alpha}\right)^{\alpha-|B|}\left(\frac{1}{\alpha}\right)^{|B|}\prod_{j\in B}x_j^{1/\alpha-1}.$$

Since every index in $\{1,\ldots,d\}$ appears in precisely one $B \in \pi$,

$$\begin{aligned}\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}\mathrm{D}_B\,\ell(\boldsymbol{x}) &= \frac{1}{\alpha^d}\prod_{j=1}^{d}x_j^{1/\alpha-1}\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}(\alpha)_{|B|}\left(\sum_{j=1}^{d}x_j^{1/\alpha}\right)^{\alpha-|B|}\\ &= \frac{1}{\alpha^d}\prod_{j=1}^{d}x_j^{1/\alpha-1}\left(\sum_{j=1}^{d}x_j^{1/\alpha}\right)^{\alpha k-d}\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}(\alpha)_{|B|}.\end{aligned}$$

Using the general form of the density as given in Proposition B.1.1 and $\boldsymbol{x} = \psi^{-1}(\boldsymbol{u})$ leads to the result as stated. □

As we can see from Proposition B.1.1, the general form of the density of AXCs involves the (possibly high-order) derivatives $\psi^{(k)}$ and $\mathrm{D}_B\,\ell$. The former are well known to be numerically non-trivial; see, for example, Hofert et al. (2012) or Hofert et al. (2013). We therefore now address how the density of AXCs can be computed numerically. This is typically done by computing a *proper logarithm*, that is, a logarithm which is numerically more robust than just $\ln c$, and then returning the exponential (but only if required). As we will see, two nested proper logarithms can be used to evaluate the logarithmic density of AXCs, which is especially appealing.

Proposition B.1.5 (AXC logarithmic density evaluation)
If the respective partial derivatives of ℓ exist and are continuous, the logarithmic density

$\ln c$ of a d-dimensional AXC C is given by

$$\ln c(\boldsymbol{u}) = \sum_{j=1}^{d} \ln\{(-\psi^{-1})'(u_j)\} + b_{\max}^{\psi,\ell}(\boldsymbol{u}) + \ln \sum_{k=1}^{d} \exp\{b_k^{\psi,\ell}(\boldsymbol{u}) - b_{\max}^{\psi,\ell}(\boldsymbol{u})\}, \quad \boldsymbol{u} \in (0,1)^d,$$

where the notation is as in Proposition B.1.1 and

$$b_k^{\psi,\ell}(\boldsymbol{u}) = \ln\{(-1)^k \psi^{(k)}\}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] + a_{\max}^{\psi,\ell,k}(\boldsymbol{u}) + \ln \sum_{\pi\in\Pi:|\pi|=k} \exp\{a_\pi^{\psi,\ell,k}(\boldsymbol{u}) - a_{\max}^{\psi,\ell,k}(\boldsymbol{u})\},$$

$$b_{\max}^{\psi,\ell}(\boldsymbol{u}) = \max_k b_k^{\psi,\ell}(\boldsymbol{u})$$

for

$$a_\pi^{\psi,\ell,k}(\boldsymbol{u}) = \sum_{B\in\pi} \ln\{(-1)^{|B|-1} \mathrm{D}_B\, \ell\}\{\psi^{-1}(\boldsymbol{u})\}, \quad a_{\max}^{\psi,\ell,k}(\boldsymbol{u}) = \max_{\pi\in\Pi:|\pi|=k} a_\pi^{\psi,\ell,k}(\boldsymbol{u}).$$

Proof. Let $\boldsymbol{u} \in (0,1)^d$ and note that

$$\begin{aligned}
c(\boldsymbol{u}) &= \left\{\prod_{j=1}^{d} (\psi^{-1})'(u_j)\right\} \sum_{k=1}^{d} \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} (\mathrm{D}_B\, \ell)\{\psi^{-1}(\boldsymbol{u})\} \\
&= \left\{\prod_{j=1}^{d} (-\psi^{-1})'(u_j)\right\} \sum_{k=1}^{d} (-1)^k \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} (-1)^{d-k} \prod_{B\in\pi} (\mathrm{D}_B\, \ell)\{\psi^{-1}(\boldsymbol{u})\} \\
&= \left\{\prod_{j=1}^{d} (-\psi^{-1})'(u_j)\right\} \sum_{k=1}^{d} (-1)^k \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} \{(-1)^{|B|-1} \mathrm{D}_B\, \ell\}\{\psi^{-1}(\boldsymbol{u})\},
\end{aligned}$$

where the last equality follows from the fact that $\sum_{B\in\pi} |B| = d$ and $\prod_{B\in\pi} \mathrm{D}_B\, \ell$ is taken over those π for which $|\pi| = k$, so $\sum_{B\in\pi} 1 = k$; note that, as before, $|B|$ denotes the number of elements of B.

Since ψ has derivatives with alternating signs, $(-1)^k \psi^{(k)} > 0$ for all arguments; in particular, $(-\psi^{-1})' > 0$, too. By Ressel (2013, Theorem 6), ℓ is fully d-max decreasing which implies that, for all arguments of ℓ, $\operatorname{sign}(\mathrm{D}_B\, \ell) = (-1)^{|B|-1}$. This implies that $\operatorname{sign}\{(-1)^{|B|-1} \mathrm{D}_B\, \ell\} = 1$ and so all terms $a_\pi^{\psi,\ell,k}$ and $b_k^{\psi,\ell}$ as defined in the claim are well-defined.

Taking the logarithm, the first product in c becomes $\sum_{j=1}^{d} \ln\{(-\psi^{-1})'(u_j)\}$ as in the claim. By using the definitions in the claim, the logarithm of the remaining sum can be written as

$$\ln \sum_{k=1}^{d} \exp\left[\ln\left\{(-1)^k \psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big] \sum_{\pi\in\Pi:|\pi|=k} \prod_{B\in\pi} \{(-1)^{|B|-1} \mathrm{D}_B\, \ell\}\{\psi^{-1}(\boldsymbol{u})\}\right\}\right], \tag{B.1}$$

where

$$\begin{aligned}
&\ln\Big((-1)^k\psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big]\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}\{(-1)^{|B|-1}\,\mathrm{D}_B\,\ell\}\{\psi^{-1}(\boldsymbol{u})\}\Big)\\
&=\ln\Big((-1)^k\psi^{(k)}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big]\Big)+\ln\sum_{\pi\in\Pi:|\pi|=k}\prod_{B\in\pi}\{(-1)^{|B|-1}\,\mathrm{D}_B\,\ell\}\{\psi^{-1}(\boldsymbol{u})\}\\
&=\ln\{(-1)^k\psi^{(k)}\}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big]+\ln\sum_{\pi\in\Pi:|\pi|=k}\exp\Big[\sum_{B\in\pi}\ln\{(-1)^{|B|-1}\,\mathrm{D}_B\,\ell\}\{\psi^{-1}(\boldsymbol{u})\}\Big]\\
&=\ln\{(-1)^k\psi^{(k)}\}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big]+\ln\sum_{\pi\in\Pi:|\pi|=k}\exp\{a_\pi^{\psi,\ell,k}(\boldsymbol{u})\}\\
&=\ln\{(-1)^k\psi^{(k)}\}\Big[\ell\{\psi^{-1}(\boldsymbol{u})\}\Big]+a_{\max}^{\psi,\ell,k}(\boldsymbol{u})+\ln\sum_{\pi\in\Pi:|\pi|=k}\exp\{a_\pi^{\psi,\ell,k}(\boldsymbol{u})-a_{\max}^{\psi,\ell,k}(\boldsymbol{u})\}\\
&=b_k^{\psi,\ell}(\boldsymbol{u}).
\end{aligned}$$

We thus obtain that the term in (B.1) equals

$$\ln\sum_{k=1}^d\exp\{b_k^{\psi,\ell}(\boldsymbol{u})\}=b_{\max}^{\psi,\ell}(\boldsymbol{u})+\ln\sum_{k=1}^d\exp\{b_k^{\psi,\ell}(\boldsymbol{u})-b_{\max}^{\psi,\ell}(\boldsymbol{u})\}.$$

Putting the terms together, the logarithmic density has the form as in the claim. □

A couple of remarks are in order here. First, note that due to the signs of the involved terms, one can apply an exp − ln-trick twice (nested) for computing the logarithmic density of AXCs. The remaining logarithms of sums in the formula of the logarithmic density are typically numerically trivial, as all summands are bounded to lie in $[0,1]$. More importantly, the nested exp − ln-trick allows one to compute both (possibly high-order) derivatives $\psi^{(k)}$ and $\mathrm{D}_B\,\ell$ in logarithmic scale (see $b_k^{\psi,\ell}(\boldsymbol{u})$ and $a_\pi^{\psi,\ell,k}(\boldsymbol{u})$, respectively); the non-logarithmic values are never used. This is numerically an important result as the logarithmic terms can typically be implemented efficiently themselves; for $\ln\{(-1)^k\psi^{(k)}\}$ for well known Archimedean families see, for example, Hofert et al. (2012), Hofert et al. (2013) or the R package `copula` of Hofert et al. (2005).

B.2 On nested Archimax copulas

We now briefly explore the question whether, in principle, HAXCs can also be nested copulas so *nested Archimax copulas (NAXCs)*, that is, whether there are HAXCs C with

analytical form $C(\boldsymbol{u}) = C_0\{C_1(\boldsymbol{u}_1), \dots, C_S(\boldsymbol{u}_S)\}$, $\boldsymbol{u} \in [0,1]^d$. Note that the only known nontrivial class of copulas for which such *nesting* can be done (under the sufficient nesting condition) is the class of nested Archimedean copulas. To this end, we make the following assumption.

Assumption B.2.1 (Nested EVCs)
Assume that $D_0, \dots, D_S$ are EVCs such that $D(\boldsymbol{u}) = D_0\{D_1(\boldsymbol{u}_1), \dots, D_S(\boldsymbol{u}_S)\}$, $\boldsymbol{u} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_S) \in [0,1]^d$, is an EVC.

A D as in Assumption B.2.1 is referred to as *nested extreme-value copula (NEVC)*. The only known nontrivial copula family for which Assumption B.2.1 is known to hold is the nested Gumbel family (under the sufficient nesting condition). It thus remains an open question whether there are other families of EVCs or a general construction of NEVCs besides the Gumbel.

B.2.1 Based on nested extreme-value copulas or nested stable tail dependence functions

Our first result shows that Assumption B.2.1 is equivalent to the existence of a *nested stable tail dependence function*.

Lemma B.2.2 (Nesting correspondence)
An EVC D is a NEVC if and only if the stable tail dependence function ℓ of D is *nested*, that is,

$$\ell(\boldsymbol{x}) = \ell_0\{\ell_1(\boldsymbol{x}_1), \dots, \ell_S(\boldsymbol{x}_S)\}, \quad \boldsymbol{x} \in [0,\infty)^d. \tag{B.2}$$

Proof.

$$\begin{aligned}
D(\boldsymbol{u}) &= D_0\{D_1(\boldsymbol{u}_1), \dots, D_S(\boldsymbol{u}_S)\} = \exp[-\ell_0\{-\ln D_1(\boldsymbol{u}_1), \dots, -\ln D_S(\boldsymbol{u}_S)\}] \\
&= \exp\Big\{-\ell_0\Big(-\ln\Big[\exp\{-\ell_1(-\ln u_{11}, \dots, -\ln u_{1d_1})\}\Big], \dots, \\
&\qquad\qquad -\ln\Big[\exp\{-\ell_S(-\ln u_{S1}, \dots, -\ln u_{Sd_S})\}\Big]\Big)\Big\} \\
&= \exp\Big[-\ell_0\{\ell_1(-\ln u_{11}, \dots, -\ln u_{1d_1}), \dots, \ell_S(-\ln u_{S1}, \dots, -\ln u_{Sd_S})\}\Big] \\
&= \exp\{-\ell(-\ln u_{11}, \dots, -\ln u_{Sd_S})\}, \quad \boldsymbol{u} \in [0,1]^d,
\end{aligned}$$

if and only if $\ell(\boldsymbol{x}) = \ell_0\{\ell_1(\boldsymbol{x}_1), \dots, \ell_S(\boldsymbol{x}_S)\}, \quad \boldsymbol{x} \in [0,\infty)^d$. □

The following proposition is essentially a nested version of one of the two HAXC extensions suggested in Section 3.3.2 which, based on Assumption B.2.1 leads to *nested AXCs (NAXCs)* based on NEVCs or, equivalently, nested stable tail dependence functions; see Lemma B.2.2.

Proposition B.2.3 (NAXCs based on NEVCs or nested stable tail dependence functions)
Let D_s, $s \in \{0, \dots, S\}$, be as in Assumption B.2.1 with respective stable tail dependence functions ℓ_s, $s \in \{0, \dots, S\}$. Let $V \sim F = \mathcal{LS}^{-1}[\psi]$ and $\boldsymbol{Y} = (\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_S) = (Y_{11}, \dots, Y_{1d_1}, \dots, Y_{S1}, \dots, Y_{Sd_S}) \sim D$ be independent, where D is an EVC as in Assumption B.2.1. Then the copula C of

$$\begin{aligned}\boldsymbol{U} &= \Big(\psi\Big(\frac{-\ln \boldsymbol{Y}_1}{V}\Big), \dots, \psi\Big(\frac{-\ln \boldsymbol{Y}_S}{V}\Big)\Big) \\ &= \Big(\psi\Big(\frac{-\ln Y_{11}}{V}\Big), \dots, \psi\Big(\frac{-\ln Y_{1d_1}}{V}\Big), \dots, \psi\Big(\frac{-\ln Y_{S1}}{V}\Big), \dots, \psi\Big(\frac{-\ln Y_{Sd_S}}{V}\Big)\Big)\end{aligned}$$

is given, for all $\boldsymbol{u} \in [0,1]^d$, by

$$\begin{aligned}C(\boldsymbol{u}) &= \psi\Big(\ell_0\Big[\ell_1\{\psi^{-1}(\boldsymbol{u}_1)\}, \dots, \ell_S\{\psi^{-1}(\boldsymbol{u}_S)\}\Big]\Big) \\ &= \psi\Big(\ell_0\Big[\ell_1\{\psi^{-1}(u_{11}), \dots, \psi^{-1}(u_{1d_1})\}, \dots, \ell_S\{\psi^{-1}(u_{S1}), \dots, \psi^{-1}(u_{Sd_S})\}\Big]\Big);\end{aligned}$$

that is, C is an AXC with nested stable tail dependence function as given in (B.2).

Proof.

$$\begin{aligned}\mathbb{P}(\boldsymbol{U} \le \boldsymbol{u}) &= \mathbb{P}\{\boldsymbol{Y}_1 \le e^{-V\psi^{-1}(\boldsymbol{u}_1)}, \dots, \boldsymbol{Y}_S \le e^{-V\psi^{-1}(\boldsymbol{u}_S)}\} \\ &= \mathbb{E}[\mathbb{P}\{\boldsymbol{Y}_1 \le e^{-V\psi^{-1}(\boldsymbol{u}_1)}, \dots, \boldsymbol{Y}_S \le e^{-V\psi^{-1}(\boldsymbol{u}_S)} \mid V\}] \\ &= \mathbb{E}[D\{e^{-V\psi^{-1}(\boldsymbol{u}_1)}, \dots, e^{-V\psi^{-1}(\boldsymbol{u}_S)}\}] = \mathbb{E}[D^V\{e^{-\psi^{-1}(\boldsymbol{u}_1)}, \dots, e^{-\psi^{-1}(\boldsymbol{u}_S)}\}] \\ &= \mathbb{E}\Big(\exp\Big[-V\ell\{\psi^{-1}(\boldsymbol{u}_1), \dots, \psi^{-1}(\boldsymbol{u}_S)\}\Big]\Big) = \psi\Big[\ell\{\psi^{-1}(\boldsymbol{u}_1), \dots, \psi^{-1}(\boldsymbol{u}_S)\}\Big]\end{aligned}$$

The claim immediately follows from Lemma B.2.2 by noting that D is nested as of Assumption B.2.1. □

Corollary B.2.4 (Pairwise marginal copulas)
Under the setup of Proposition B.2.3 the bivariate marginal copulas of C satisfy

$$C(u_{si}, u_{tj}) = \begin{cases}\psi[\ell_s\{\psi^{-1}(u_{si}), \psi^{-1}(u_{sj})\}], & \text{if } t = s, \\ \psi[\ell_0\{\psi^{-1}(u_{si}), \psi^{-1}(u_{tj})\}], & \text{otherwise.}\end{cases}$$

Therefore, the bivariate marginal copulas of C are (possibly different) AXCs.

Proof. For a stable tail dependence function ℓ, one has that $\ell(\boldsymbol{x}) = x_j$ if all components except the jth of $\boldsymbol{x}$ are 0. As such, for any $s \in \{1, \dots, S\}$,

$$\ell_s\{\psi^{-1}(u_{s1}), \dots, \psi^{-1}(u_{sd_s})\} = \begin{cases} 0, & \text{if } u_{sj} = 1 \ \forall\, j \in \{1, \dots, d_s\}, \\ \psi^{-1}(u_{sk}), & \text{if } u_{sj} = 1 \ \forall\, j \in \{1, \dots, d_s\} \backslash \{k\}, \\ \ell_s\{\psi^{-1}(u_{sk}), \psi^{-1}(u_{sl})\}, & \text{if } u_{sj} = 1 \ \forall\, j \in \{1, \dots, d_s\} \backslash \{k, l\}, \end{cases}$$

from which the result follows. □

B.2.2 Additionally nesting frailties

As in the second method for introducing hierarchies on AXCs presented in Section 3.3.2, we could, additionally, impose a hierarchical structure on the underlying (multiple) frailties. We focus on the two-level case with S different frailties. Assume, as before, the sufficient nesting condition to hold, that is, $\psi_s \in \Psi$, $s \in \{0, \dots, S\}$, are Archimedean generators and, for all $s \in \{0, \dots, S\}$, the derivative of $\psi_0^{-1} \circ \psi_s$ is completely monotone.

Proposition B.2.5 (NAXCs based on nested frailties)
Let D_s, $s \in \{0, \dots, S\}$, be as in Assumption B.2.1 with respective stable tail dependence functions ℓ_s, $s \in \{0, \dots, S\}$. Furthermore, let $\psi_s \in \Psi$ be completely monotone, $s \in \{0, \dots, S\}$, and assume that the sufficient nesting condition holds. Assume $V_0 \sim F_0 = \mathcal{LS}^{-1}[\psi_0]$ and $V_{0s} \,|\, V_0 \sim F_{0s} = \mathcal{LS}^{-1}[\psi_{0s}(\cdot\,; V_0)]$, $s \in \{1, \dots, S\}$. Moreover, let $\boldsymbol{Y} = (\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_S) \sim D$ be independent of $V_0, V_1, \dots, V_S$ and assume that

$$\begin{aligned} &\mathbb{E}\Big\{\mathbb{E}\Big(D_0[D_1\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}\}, \dots, D_S\{e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\}] \,\Big|\, V_0\Big)\Big\} \\ &= \mathbb{E}\Big\{D_0\Big(\mathbb{E}[D_1\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}\} \,|\, V_0], \dots, \mathbb{E}[D_S\{e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\} \,|\, V_0]\Big)\Big\}. \end{aligned} \tag{B.3}$$

Then the copula C of

$$\boldsymbol{U} = \Big(\psi_1\Big(\frac{-\ln \boldsymbol{Y}_1}{V_{01}}\Big), \dots, \psi_S\Big(\frac{-\ln \boldsymbol{Y}_S}{V_{0S}}\Big)\Big)$$

is given by

$$C(\boldsymbol{u}) = C_0\{C_1(\boldsymbol{u}_1), \dots, C_S(\boldsymbol{u}_S)\}, \quad \boldsymbol{u} \in [0,1]^d,$$

where, for all $s \in \{0, \dots, S\}$, C_s is Archimax with generator ψ_s and stable tail dependence function ℓ_s.

Proof.

$$\begin{aligned}
\mathbb{P}(\boldsymbol{U} \leq \boldsymbol{u}) &= \mathbb{P}\{\boldsymbol{Y}_1 \leq e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}, \dots, \boldsymbol{Y}_S \leq e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\} \\
&= \mathbb{E}\Big(\mathbb{E}[\mathbb{P}\{\boldsymbol{Y}_1 \leq e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}, \dots, \boldsymbol{Y}_S \leq e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)} \,|\, V_{01}, \dots, V_{0S}\} \,|\, V_0]\Big) \\
&= \mathbb{E}\Big(\mathbb{E}[D\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}, \dots, e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\} \,|\, V_0]\Big) \\
&\underset{\text{(B.3)}}{=} \mathbb{E}\Big\{D_0\Big(\mathbb{E}[D_1\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}\} \,|\, V_0], \dots, \mathbb{E}[D_S\{e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\} \,|\, V_0]\Big)\Big\}.
\end{aligned}$$

Each component $\mathbb{E}[D_s\{e^{-V_{0s}\psi_s^{-1}(\boldsymbol{u}_s)}\} \,|\, V_0]$, $s \in \{1, \dots, S\}$, satisfies

$$\begin{aligned}
\mathbb{E}[D_s\{e^{-V_{0s}\psi_s^{-1}(\boldsymbol{u}_s)}\} \,|\, V_0] &= \mathbb{E}[D_s^{V_{0s}}\{e^{-\psi_s^{-1}(\boldsymbol{u}_s)}\} \,|\, V_0] = \mathbb{E}[e^{-V_{0s}\ell_s\{\psi_s^{-1}(\boldsymbol{u}_s)\}} \,|\, V_0] \\
&= \psi_{0s}[\ell_s\{\psi_s^{-1}(\boldsymbol{u}_s)\}; V_0],
\end{aligned}$$

thus

$$\begin{aligned}
\mathbb{P}(\boldsymbol{U} \leq \boldsymbol{u}) &= \mathbb{E}\Big\{D_0\Big(\psi_{01}[\ell_1\{\psi_1^{-1}(\boldsymbol{u}_1)\}; V_0], \dots, \psi_{0S}[\ell_S\{\psi_S^{-1}(\boldsymbol{u}_S)\}; V_0]\Big)\Big\} \\
&= \mathbb{E}\Big(D_0\Big[e^{-V_0\psi_0^{-1}\{C_1(\boldsymbol{u}_1)\}}, \dots, e^{-V_0\psi_0^{-1}\{C_S(\boldsymbol{u}_S)\}}\Big]\Big) \\
&= \mathbb{E}\Big(D_0^{V_0}\Big[e^{-\psi_0^{-1}\{C_1(\boldsymbol{u}_1)\}}, \dots, e^{-\psi_0^{-1}\{C_S(\boldsymbol{u}_S)\}}\Big]\Big) \\
&= \mathbb{E}\Big\{e^{-V_0(\ell_0[\psi_0^{-1}\{C_1(\boldsymbol{u}_1)\}, \dots, \psi_0^{-1}\{C_S(\boldsymbol{u}_S)\}])}\Big\} \\
&= \psi_0\Big(\ell_0\Big[\psi_0^{-1}\{C_1(\boldsymbol{u}_1)\}, \dots, \psi_0^{-1}\{C_S(\boldsymbol{u}_S)\}\Big]\Big) = C_0\{C_1(\boldsymbol{u}_1), \dots, C_S(\boldsymbol{u}_S)\}. \qquad \square
\end{aligned}$$

The following corollary provides a condition under which Assumption (B.3) holds. Note that this particular model can already be found in McFadden (1978).

Corollary B.2.6 (AC composed with AXCs)
If $D(\boldsymbol{u}) = \prod_{s=1}^S D_s(\boldsymbol{u}_s)$, (B.3) holds and $C(\boldsymbol{u}) = C_0\{C_1(\boldsymbol{u}_1), \dots, C_S(\boldsymbol{u}_S)\}$, where C_0 is Archimedean and $C_1, \dots, C_S$ are Archimax. In particular, if D is the independence copula, (B.3) holds and C is a NAC.

Proof. If $D(\boldsymbol{u}) = \prod_{s=1}^S D_s(\boldsymbol{u}_s)$, then, conditional on V_0, the sector components are independent and we obtain

$$\begin{aligned}
&\mathbb{E}\Big(D_0[D_1\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}\}, \dots, D_S\{e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\}] \,\Big|\, V_0\Big) \\
&= \mathbb{E}\Big[\prod_{s=1}^S D_s\{e^{-V_{0s}\psi_s^{-1}(\boldsymbol{u}_s)}\} \,\Big|\, V_0\Big] = \prod_{s=1}^S \mathbb{E}[D_s\{e^{-V_{0s}\psi_s^{-1}(\boldsymbol{u}_s)}\} \,|\, V_0] \\
&= D_0\Big(\mathbb{E}[D_1\{e^{-V_{01}\psi_1^{-1}(\boldsymbol{u}_1)}\} \,|\, V_0], \dots, \mathbb{E}[D_S\{e^{-V_{0S}\psi_S^{-1}(\boldsymbol{u}_S)}\} \,|\, V_0]\Big).
\end{aligned}$$

and thus (B.3) follows by taking the expectation. The rest follows immediately by noting that an EVC is the independence copula if and only if its stable tail dependence function is the sum of its components, so the Archimax (sector) copulas $C_s(\boldsymbol{u}_s) = \psi_s\Big[\ell_s\{\psi_s^{-1}(u_{s1}), \ldots, \psi_s^{-1}(u_{sd_s})\}\Big]$ are Archimedean generated by ψ_s, $s \in \{1, \ldots, S\}$. □

Appendix C

Additional details for Chapter 4

C.1 Analyzing GMMN QMC and GMMN RQMC estimators

C.1.1 QMC point sets

The idea behind quasi-random numbers is to replace pseudo-$U(0,1)^p$ random numbers with low-discrepancy points sets $P_{n_{\text{gen}}}$ to produce a more homogeneous coverage of $[0,1]^p$ in comparison to pseudo-random numbers. That is, with respect to a certain *discrepancy measure*, the empirical distribution of the $P_{n_{\text{gen}}}$ is closer to the uniform distribution $\mathrm{U}(0,1)^p$ than a pseudo-random sample.

Established notions of the discrepancy of a point set $P_{n_{\text{gen}}}$ are as follows. The *discrepancy function* of $P_{n_{\text{gen}}}$ in an interval $I = [\mathbf{0}, \boldsymbol{b}) = \prod_{j=1}^{p}[0, b_j)$, $b_j \in (0,1]$, $j = 1, \ldots, p$, is defined by

$$D(I; P_{n_{\text{gen}}}) = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \mathbb{1}_{\{\boldsymbol{v}_i \in I\}} - \lambda(I),$$

where $\lambda(I)$ is the p-dimensional Lebesgue measure of I. Thus the discrepancy function is the difference between the number of points of $P_{n_{\text{gen}}}$ in I and the probability of a p-dimensional standard uniform random vector to fall in I. For $\mathcal{A} = \{[\mathbf{0}, \boldsymbol{b}) : \boldsymbol{b} \in (0,1]^p\}$, the *star discrepancy* of $P_{n_{\text{gen}}}$ is defined by

$$D^*(P_{n_{\text{gen}}}) = \sup_{I \in \mathcal{A}} |D(I; P_{n_{\text{gen}}})|.$$

If $P_{n_{\text{gen}}}$ satisfies the condition $D^*(P_{n_{\text{gen}}}) \in O(n_{\text{gen}}^{-1}(\log n_{\text{gen}})^p)$, it is called a *low-discrepancy sequence* (Lemieux, 2009, p. 143).

There are different approaches to construct (deterministic) low-discrepancy sequences; see Lemieux (2009, Chapters 5–6). The two main approaches involve either lattices (grids which behave well under projections) or digital nets/sequences. In our numerical investigations presented in Sections 4.3–4.4, we worked with a type of digital net constructed using the Sobol' sequence; see Sobol' (1967).

C.1.2 Analyzing the GMMN QMC estimator

In this section, we will derive conditions under which the (non-randomized) GMMN QMC estimator

$$\frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(q(\boldsymbol{v}_i)) = \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} h(\boldsymbol{v}_i),$$

where $q = f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}$ and $h = \Psi \circ q = \Psi \circ f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}$, has a small error when approximating $\mathbb{E}(\Psi(\boldsymbol{Y}))$. In the following analysis, we need to further assume that $\text{supp}(F_{\boldsymbol{X}})$ and $\text{supp}(F_{\boldsymbol{Y}})$ are bounded.

The *Koksma–Hlawka inequality* (Niederreiter, 1992) for a function $g : [0,1]^p \to \mathbb{R}$ says that

$$\left| \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} g(\boldsymbol{v}_i) - \mathbb{E}(g(\boldsymbol{U}')) \right| \leq V(g) D^*(P_{n_{\text{gen}}}),$$

where $\boldsymbol{U}' \sim \text{U}(0,1)^p$ and the variation $V(g)$ is understood in the sense of Hardy and Krause; we refer to the right-hand side of the inequality as *Koksma–Hlawka bound*. From this Koksma–Hlawka inequality, we can establish that if g has finite bounded variation, that is $V(g) < \infty$, then the convergence rate for $\frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} g(\boldsymbol{v}_i)$ is determined by $D^*(P_{n_{\text{gen}}}) = O(n_{n_{\text{gen}}}^{-1}(\log n_{\text{gen}})^p)$.

We can use the Koksma–Hlawka inequality to analyze the convergence of the GMMN QMC estimator $\frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(\boldsymbol{y}_i)$ of $\mathbb{E}(\Psi(\boldsymbol{Y}))$, where $\boldsymbol{y}_i = q(\boldsymbol{v}_i)$, $i = 1, \ldots, n_{\text{gen}}$ and $\boldsymbol{Y} \sim F_{\boldsymbol{Y}}$, by establishing the conditions under which $V(h)$ is bounded. To that end, consider the following proposition.

Proposition C.1.1 (Sufficient conditions for finiteness of the Koksma–Hlawka bound) Assume that $\text{supp}(F_{\boldsymbol{Y}})$ is bounded and all appearing partial derivatives of q and Ψ exist

and are continuous. Consider $q = f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}$, the point set $P_{n_{\text{gen}}} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{n_{\text{gen}}}\} \subseteq [0,1)^p$ and let $\boldsymbol{y}_i = q(\boldsymbol{v}_i)$, $i = 1, \ldots, n_{\text{gen}}$, denote the GMMN quasi-random sample. Suppose that

1. $\Psi(\boldsymbol{y}) < \infty$ for all $\boldsymbol{y} \in \operatorname{supp}(F_{\boldsymbol{Y}})$ and
$$\frac{\partial^{|\boldsymbol{\beta}|_1} \Psi(\boldsymbol{y})}{\partial^{\beta_1} y_1 \ldots \partial^{\beta_d} y_d} < \infty, \quad \boldsymbol{y} \in \operatorname{supp}(F_{\boldsymbol{Y}}),$$
for all $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d) \subseteq \{0, \ldots, d\}^d$ and $|\boldsymbol{\beta}|_1 = \sum_{j=1}^d \beta_j \leq d$;
2. there exists an $M > 0$ such that $|\,\mathrm{D}^k F_{Z_j}^{-1}| \leq M$, for each $k, j = 1, \ldots, p$, where D^k denotes the k-fold derivative of its argument;
3. there exists, for each layer $l = 1, \ldots, L+1$ of the NN $f_{\hat{\boldsymbol{\theta}}}$, an $N_l > 0$ such that $|\,\mathrm{D}^k \phi_l| \leq N_l$ for all $k = 1, \ldots, p$; and
4. the parameter vector $\hat{\boldsymbol{\theta}} = (\widehat{W}_1, \ldots, \widehat{W}_{L+1}, \hat{\boldsymbol{b}}_1, \ldots, \hat{\boldsymbol{b}}_{L+1})$ of the fitted NN is bounded.

Then there exists a constant c independent of n_{gen}, but possibly depending on Ψ, $\hat{\boldsymbol{\theta}}$, M and $N_1, \ldots, N_{L+1}$, such that

$$\left| \frac{1}{n_{\text{gen}}} \sum_{i=1}^{n_{\text{gen}}} \Psi(\boldsymbol{y}_i) - \mathbb{E}(\Psi(\boldsymbol{Y})) \right| \leq c D^*(P_{n_{\text{gen}}}).$$

Proof. To begin with note that for any q such that $q(\boldsymbol{U}') \sim F_{\boldsymbol{Y}}$, we know that $\boldsymbol{Y}$ is in distribution equal to $q(\boldsymbol{U}')$ and thus $\mathbb{E}(\Psi(\boldsymbol{Y})) = \mathbb{E}\big(\Psi(q(\boldsymbol{U}'))\big) = \mathbb{E}(h(\boldsymbol{U}'))$. Based on this property, we can obtain the Koksma–Hlawka bound $V(h)D^*(P_{n_{\text{gen}}})$ for the change of variable h.

Following Lemieux (2009, Section 5.6.1), we can derive an expression for $V(h)$. To this end, let

$$V^{(j)}(h; \boldsymbol{\alpha}) = \int_{[0,1)^j} \left| \frac{\partial^j h^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j})}{\partial v_{\alpha_j} \ldots \partial v_{\alpha_1}} \right| \mathrm{d}v_{\alpha_1} \ldots \mathrm{d}v_{\alpha_j},$$

where $h^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j}) = h(\tilde{v}_1, \ldots, \tilde{v}_p)$ for $\tilde{v}_k = v_k$ if $k \in \{\alpha_1, \ldots, \alpha_j\}$ and $\tilde{v}_k = 1$ otherwise. Then

$$V(h) = \sum_{j=1}^{p} \sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|_1 = j} V^{(j)}(h; \boldsymbol{\alpha}), \tag{C.1}$$

where the inner sum is taken over all $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_j)$ with $\{\alpha_1, \ldots, \alpha_j\} \subseteq \{1, \ldots, p\}$ — see also Niederreiter (1992, pp. 19–20), Hlawka (1961, eq. (4)) and Hlawka and Mück (1972, eq. (4')). Following Hlawka and Mück (1972) and Constantine and Savits (1996, Theorem 2.1), we then have

$$\left| \frac{\partial^j h^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j})}{\partial v_{\alpha_j} \ldots \partial v_{\alpha_1}} \right| = \sum_{1 \leq |\boldsymbol{\beta}|_1 \leq j} \frac{\partial^{|\boldsymbol{\beta}|_1} \Psi}{\partial^{\beta_1} y_1 \ldots \partial^{\beta_d} y_d} \sum_{i=1}^{j} \sum_{(\boldsymbol{\kappa}, \boldsymbol{k}) \in \pi_i(\boldsymbol{\kappa}, \boldsymbol{k})} c_{\boldsymbol{\kappa}} \prod_{m=1}^{i} \frac{\partial^{|\boldsymbol{\kappa}_m|_1} q_{k_m}^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j})}{\partial^{\kappa_{mj}} v_{\alpha_j} \ldots \partial^{\kappa_{m1}} v_{\alpha_1}}, \tag{C.2}$$

where $\boldsymbol{\beta} \in \mathbb{N}_0^d$ and where $\pi_i(\boldsymbol{\kappa}, \boldsymbol{k})$ denotes the set of pairs $(\boldsymbol{\kappa}, \boldsymbol{k})$ such that $\boldsymbol{k} = (k_1, \ldots, k_i) \in \{1, \ldots, d\}^i$ and $\boldsymbol{\kappa} = (\boldsymbol{\kappa}_1, \ldots, \boldsymbol{\kappa}_i)$ with $\boldsymbol{\kappa}_m = (\kappa_{m1}, \ldots, \kappa_{mj}) \in \{0, 1\}^j$, $m = 1, \ldots, i$, and $\sum_{m=1}^{i} \kappa_{mi} = 1$ for $i = 1, \ldots, j$; see Constantine and Savits (1996) for more details on $\pi_i(\boldsymbol{\kappa}, \boldsymbol{k})$ and the constants $c_{\boldsymbol{\kappa}}$. Furthermore, for index $j = 1, \ldots, d$, $q_j^{(\alpha)}(v_{\alpha_1}, \ldots, v_{\alpha_j}) = q_j(\tilde{v}_1, \ldots, \tilde{v}_p)$ and $q_j(\tilde{v}_1, \ldots, \tilde{v}_p) = \phi_{L+1}(\widehat{W}_{L+1j\cdot} \boldsymbol{a}_L + \hat{\boldsymbol{b}}_{L+1})$, where $\boldsymbol{a}_l = \phi_l(\widehat{W}_l \boldsymbol{a}_{l-1} + \hat{\boldsymbol{b}}_l)$ for $l = 1, \ldots, L$ with $\boldsymbol{a}_0 = F_{\boldsymbol{Z}}^{-1}(\tilde{\boldsymbol{v}})$ and where $\widehat{W}_{L+1j\cdot}$ denotes the jth row of W_{L+1}.

Based on the decomposition in (C.2), a sufficient condition to ensure that $V(h) < \infty$ is that all products of the form

$$\frac{\partial^{|\boldsymbol{\beta}|_1} \Psi}{\partial^{\beta_1} y_1 \ldots \partial^{\beta_d} y_d} \prod_{m=1}^{i} \frac{\partial^{|\boldsymbol{\kappa}_m|_1} q_{k_m}^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j})}{\partial^{\kappa_{mj}} v_{\alpha_j} \ldots \partial^{\kappa_{m1}} v_{\alpha_1}}, \quad i = 1, \ldots, j,$$

are integrable.

To that end, Assumptions 2–4 imply that all mixed partial derivatives of $q = f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}$ are bounded. By the assumption of continuous partial derivatives of q, this implies that finite products of the form

$$\prod_{m=1}^{i} \frac{\partial^{|\boldsymbol{\kappa}_m|_1} q_{k_m}^{(\boldsymbol{\alpha})}(v_{\alpha_1}, \ldots, v_{\alpha_j})}{\partial^{\kappa_{mj}} v_{\alpha_j} \ldots \partial^{\kappa_{m1}} v_{\alpha_1}}, \quad i = 1, \ldots, j,$$

are integrable. By Assumption 1, decomposition (C.2) and Hölder's inequality, the quantity in (C.1) is bounded. This implies that h has bounded variation, so that the Koksma–Hlawka bound is finite.

□

The following remark provides insights into Assumptions 2–4 of Proposition C.1.1.

Remark C.1.2
$\mathrm{U}(a,b)^p$ for $a < b$, which is a popular choice for the input distribution, clearly satisfies Assumption 2 in Proposition C.1.1. Assumption 3 is satisfied for various commonly used activation functions, such as:

1. *Sigmoid.* If $\phi_l(x) = 1/(1+\mathrm{e}^{-x})$ for layer l, then $N_l = 1$.
2. *ReLU.* If $\phi_l(x) = \max\{0, x\}$ for layer l, then $N_l = 1$. In this case, only the first derivative is (partly) non-zero. Additionally, note that the ReLU activation function is not differentiable at $x = 0$. However, even if $\phi_l = \max\{0, x\}$ for all $l = 1, \ldots, L+1$, the set of all pointwise discontinuities of the mixed partial derivatives of q is a null set. Hence, the discontinuities do not jeopardize the proof of Proposition C.1.1.
3. *Softplus.* If $\phi_l(x) = \log(1+\mathrm{e}^x)$ for layer l, then $N_l = 1$. The Softplus activation function can be used as a smooth approximation of the ReLU activation function.
4. *Linear.* If $\phi_l(x) = x$ for layer l, then $N_l = 1$. Only the first derivative is non-zero.
5. *Tanh.* If $\phi_l(x) = \tanh(x)$ for layer l, then $N_l = 1$.
6. *Scaled exponential linear unit (SELU)*; see Klambauer et al. (2017). If, for layer l,

$$\phi_l(x) = \begin{cases} \lambda\alpha(\exp(-x) - 1), & \text{if } x < 0, \\ \lambda x, & \text{if } x \geq 0, \end{cases}$$

   where λ and α are prespecified constants, then $N_l = \max\{\lambda, \lambda\alpha, 1\}$. The same argument about discontinuities made with the ReLU activation function applies equally well to the case of the SELU activation function.

Assumption 4 of Proposition C.1.1 is satisfied in practice because NNs are always trained with regularization on the parameters, which means $\hat{\boldsymbol{\theta}}$ always lies in a compact set. Additionally note that in the general case where q is characterized by a composition of NN layers and $F_{\boldsymbol{Z}}^{-1}$ with a different (but standard) activation function in each layer, all partial derivatives of q exist and are continuous. Moreover, for the activation functions and input distributions listed above, all mixed partial derivatives of q are bounded.

C.1.3 RQMC point sets

In Monte Carlo applications, we need to randomize the low-discrepancy sequence $P_{n_{\text{gen}}}$ to obtain unbiased estimators and variance estimates. To that end, we can randomize $P_{n_{\text{gen}}}$

via a $\boldsymbol{U}' \sim \mathrm{U}(0,1)^p$ to obtain a randomized point set $\tilde{P}_{n_{\text{gen}}} = \tilde{P}_{n_{\text{gen}}}(\boldsymbol{U}') = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$, where $\tilde{\boldsymbol{v}}_i = r(\boldsymbol{U}', \boldsymbol{v}_i)$, $i = 1, \ldots, n_{\text{gen}}$, for a certain randomization function r. A simple randomization to obtain an RQMC point set is to consider $\tilde{\boldsymbol{v}}_i = (\boldsymbol{v}_i + \boldsymbol{U}') \bmod 1$, $i = 1, \ldots, n_{\text{gen}}$, for $\boldsymbol{U}' \sim \mathrm{U}(0,1)^p$, a so-called *random shift*; see Cranley and Patterson (1976).

In practice, more sophisticated alternatives to the random shift are often used. One such slightly more sophisticated randomization scheme is the *digital shift* method; see Lemieux (2009, Chapter 6) and Cambou et al. (2017). In the same vein as the random shift, one adds a random uniform shift to points in $P_{n_{\text{gen}}}$, but with operations in $\mathbb{Z}_b$, where b is the base in which the digital net is defined, rather than simply adding two real numbers. We use $\tilde{P}^{\text{ds}}_{n_{\text{gen}}}$ to denote the RQMC point set obtained using the digital shift method.

Another randomization approach is to *scramble* the digital net. This technique was originally proposed by Owen (1995). In particular, the type of scrambling we work with is referred to as the *nested uniform scrambling* (or *full random scrambling*) method; see Owen (2003). Since we primarily use this method throughout the chapter, $\tilde{P}_{n_{\text{gen}}}$ will denote specifically the RQMC point set obtained using scrambling. The digital shift method is more computationally efficient in comparison to scrambling but because the distortion of the deterministic point set is fairly simple in the digital shift method, there are *bad* functions one can construct such that the variance of the RQMC estimator is larger than that of the corresponding MC estimator; see Lemieux (2009, Chapter 6). Furthermore, when RQMC points are constructed with scrambling, we can justify (see Appendix C.1.4) that an improved rate of $O(n_{\text{gen}}^{-3}(\log n_{\text{gen}})^{p-1})$ is achievable for $\mathrm{Var}(\hat{\mu}^{\text{NN}}_{n_{\text{gen}}})$; this translates to $O(n_{\text{gen}}^{-3/2}(\log n_{\text{gen}})^{(p-1)/2})$ on the root mean squared error (RMSE) scale, which is more directly comparable to the convergence rate of $O(n_{\text{gen}}^{-1}(\log n_{\text{gen}})^{p})$ implied by the Koksma-Hlawka bound for the mean absolute error (MAE) of $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}$ using QMC points (see Appendix C.1.2). Hence, even though the aforementioned bad functions do not often arise in practice, we primarily work with the scrambling randomization method to construct our RQMC point sets. Both the scrambling and the digital shift methods are available in the R package `qrng` and can be accessed via `sobol(, randomize = "Owen")` and `sobol(, randomize = "digital.shift")` respectively.

The randomization schemes discussed above preserve the low-discrepancy property of $P_{n_{\text{gen}}}$ and the estimators of interest obtained using each type of RQMC point set are unbiased. Computing the estimator based on B such randomized point sets and computing the sample variance of the resulting B estimates then allows us to estimate the variance of the estimator of interest.

C.1.4 Analyzing the GMMN RQMC estimator

GMMN RQMC estimators constructed with scrambled nets

For RQMC estimators $\frac{1}{n_{\text{gen}}}\sum_{i=1}^{n_{\text{gen}}} g(\tilde{\boldsymbol{v}})$ based on scrambled nets, Owen (1997b) initially derived a convergence rate for the variance of the estimators under a certain smoothness condition on g, where $g : [0,1]^p \to \mathbb{R}$. Owen (2008) then generalized his earlier result to allow a weaker smoothness condition for a larger class of scrambled nets. Specifically, if $\tilde{P}_{n_{\text{gen}}} = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_{n_{\text{gen}}}\}$ is a so-called relaxed scrambled (λ, q, m, p)-net in base b with bounded gain coefficients — for example, Sobol' sequences randomized using nested uniform sampling belong to this class — then we have the following result as a direct consequence of Owen (2008).

Theorem C.1.3 (Owen (2008))
If all the mixed partial derivatives (up to order p) of g exist and are continuous, then

$$\operatorname{Var}\left(\frac{1}{n_{\text{gen}}}\sum_{i=1}^{n_{\text{gen}}} g(\tilde{\boldsymbol{v}}_i)\right) = O(n_{\text{gen}}^{-3}(\log n_{\text{gen}})^{p-1}).$$

Proof. See Owen (2008, Theorem 3). □

Now, for the GMMN RQMC estimator, $\hat{\mu}^{\text{NN}}_{n_{\text{gen}}} = \frac{1}{n_{\text{gen}}}\sum_{i=1}^{n_{\text{gen}}} \Psi(q(\tilde{\boldsymbol{v}}_i)) = \frac{1}{n_{\text{gen}}}\sum_{i=1}^{n_{\text{gen}}} h(\tilde{\boldsymbol{v}}_i)$, the corollary below naturally follows from Theorem C.1.3 with some added analysis of the composite function h.

Corollary C.1.4
If all the mixed partial derivatives (up to order p) of $h = \Psi \circ q = \Psi \circ f_{\hat{\boldsymbol{\theta}}} \circ F_{\boldsymbol{Z}}^{-1}$ exist and are continuous, then $\operatorname{Var}(\hat{\mu}^{\text{NN}}_{n_{\text{gen}}}) = O(n_{\text{gen}}^{-3}(\log n_{\text{gen}})^{p-1})$.

To analyze the mixed partial derivatives of h, it suffices to analyze each component function separately.

For popular choices of input distributions (such as $\mathrm{U}(a,b)^p$ for $a < b$ or $\mathrm{N}(0,1)^p$), the k-fold derivative $\mathrm{D}^k F_{Z_j}^{-1}$ exists and is continuous (on $[a,b]$ or $\mathbb{R}$ depending on the choice of input distribution) for each $k, j = 1, \ldots, p$.

For each layer $l = 1, \ldots, L+1$ of the NN $f_{\hat{\boldsymbol{\theta}}}$, $\mathrm{D}^k \phi_l$ exists and is continuous for $k = 1, \ldots, p$ — provided that we use (standard) activation functions; see Remark C.1.2 for further details on suitable activation functions. For NNs constructed using some popular activation functions such as the ReLU and SELU, note that the set of all pointwise discontinuities of

the mixed partial derivatives of $f_{\hat{\boldsymbol{\theta}}}$ is a set of Lebesgue measure zero and hence the proof of Theorem C.1.3 holds. Alternatively, we can use the softplus activation function as a smoother approximation of ReLU. Now in the most general case of NNs $f_{\hat{\boldsymbol{\theta}}}$ being composed of layers with different (but standard) activation functions, all mixed partial derivatives (up to order p) of $f_{\hat{\boldsymbol{\theta}}}$ exist and are continuous almost everywhere.

Finally, it is certainly true that, for many functionals Ψ that we care about in practice, such as those considered in Sections 4.4 and 4.5.3, all of its mixed partial derivatives (up to order p) exist and are continuous almost everywhere on $\mathbb{R}^d$.

GMMN RQMC estimators constructed with digitally shifted nets

For GMMN RQMC estimators $\hat{\mu}_{n_{\text{gen}}}^{\text{NN,ds}}$ constructed using digitally shifted RQMC point sets $\tilde{P}_{n_{\text{gen}}}^{\text{ds}}$, we can obtain an expression for $\text{Var}(\hat{\mu}_{n_{\text{gen}}}^{\text{NN,ds}})$ under the condition that the composite function h is square integrable; see Cambou et al. (2017, Proposition 6).

With added assumptions on the smoothness of h, one can obtain improved convergence rates (compared to MC estimators) for $\text{Var}(\hat{\mu}_{n_{\text{gen}}}^{\text{NN,ds}})$. For example, under the assumptions of Proposition C.1.1, h has finite bounded variation in the sense of Hardy–Krause, which implies that $\text{Var}(\hat{\mu}_{n_{\text{gen}}}^{\text{NN,ds}}) = O(n_{\text{gen}}^{-2}(\log n_{\text{gen}})^{2p})$; see L'Ecuyer (2016).

In practice, we observe that GMMN RQMC estimators constructed using both scrambled and digitally shifted nets achieve very similar convergence rates despite differences in the theoretical convergence rates. To that end, Figure C.1 shows plots of standard deviation estimates for estimating $\mathbb{E}(\Psi_2(\boldsymbol{X}))$ where we use the RQMC point sets $\tilde{P}_{n_{\text{gen}}}^{\text{ds}}$ for the same copula models as considered for Figure 4.12 (which is based on GMMN RQMC estimators constructed using scramble nets) in Section 4.4. The approximate convergence rates as implied by the regression coefficients α displayed in both figures are very similar across the various examples.

C.2 Run time

Run time depends on factors such as the hardware used, the current workload, the algorithm implemented, the programming language used, the implementation style, compiler flags, whether garbage collection was used, etc. There is not even a unique concept of time (system vs user vs elapsed time). Although none of our code was optimized for run time, we still report on various timings here, measured in elapsed time also known as wall-clock time.

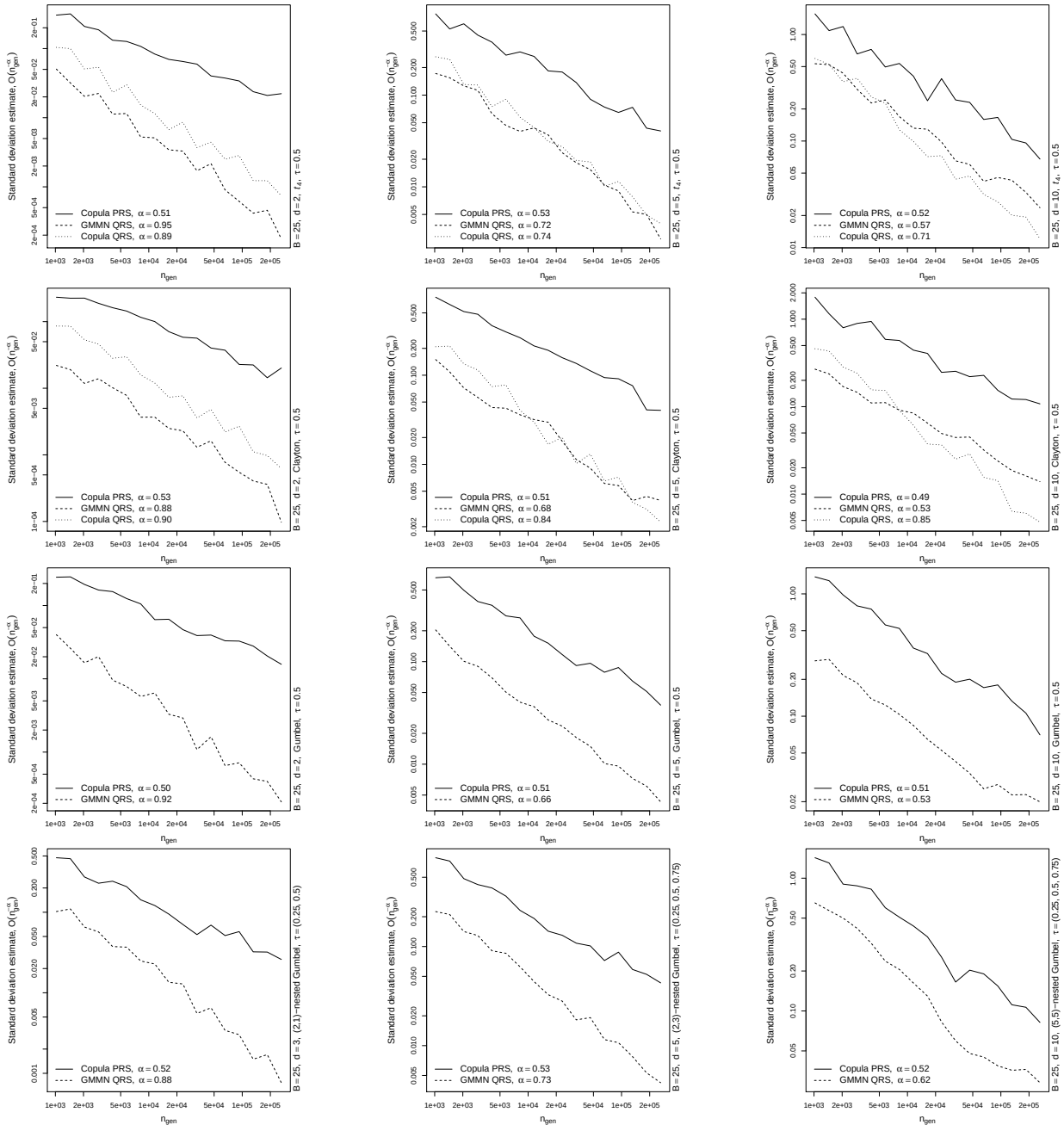

Figure C.1: Standard deviation estimates based on $B = 25$ replications for estimating $\mathbb{E}(\Psi_2(\boldsymbol{X}))$ via MC based on a pseudo-random sample (PRS), via the copula RQMC estimator (whenever available; rows 1–2 only) and via the GMMN RQMC estimator (based on digitally shifted nets). Note that in rows 1–3, $d \in \{2, 5, 10\}$, whereas in row 4, $d \in \{3, 5, 10\}$.

C.2.1 Training and sampling

The results in this section are reproducible with the demo `GMMN_QMC_timings` of the R package `gnn`.

Table C.1 shows elapsed times in minutes for training a GMMN on training data from t_4, Clayton (C), Gumbel (G) and nested Gumbel (NG) copulas in dimensions 2, 3, 5 and 10 as described in Sections 4.3.2 and 4.3.3. As is reasonable, the measured times are only affected by the dimension, not by the type of dependence model.

C	$d = 2, 3$	$d = 5$	$d = 10$
t_4	5.52	7.01	9.46
C	5.52	7.00	9.45
G	5.52	7.01	9.46
NG	6.01	7.01	9.44

Table C.1: Elapsed times in minutes for training GMMNs of the same architecture as used in Sections 4.3.2 and 4.3.3 with $n_{\text{epo}} = 300$, $n_{\text{trn}} = 60\,000$ and $n_{\text{bat}} = 5000$ on respective copula samples; training was done on one NVIDIA Tesla P100 GPU.

Table C.2 contains elapsed times for generating $n_{\text{gen}} = 10^5$ observations from the respective dependence model and sampling method on two different machines, once on the NVIDIA Tesla P100 GPU used for training and once locally on a 2018 2.7 GHz Quad-Core Intel Core i7 processor. The results for the copula-based pseudo-random sampling method are averaged over 100 repetitions. The results for the copula-based quasi-random sampling method are obtained as follows. If the conditional copulas involved in applying the inverse Rosenblatt transform of the respective copula model are not available analytically nor numerically, NA is reported; this applies to the nested Gumbel copula. And if they are only available numerically (by root finding), then a reduced sample size of 1000 is used and the reported run times were obtained by scaling up to n_{gen} by multiplication with 100; this applies to the Gumbel copula. We also measured run times for $n_{\text{gen}} = 10^6$ and $n_{\text{gen}} = 10^7$ and they scale proportionally as one would expect.

We see from Table C.2 that quasi-random sampling from specific copulas is available and can be fast, e.g., for t_4 and Clayton copulas. However, we already see that quasi-random sampling gets more time-consuming for larger d. For other copulas, such as Gumbel copulas, it can be much more time consuming. Furthermore, as seen from the nested case and as is currently the case for most copula models, a quasi-random sampling procedure is not even available. By contrast, on the same machine, GMMNs show very close run times, are

		2018 2.7 GHz Quad-Core Intel Core i7				NVIDIA Tesla P100 GPU			
		Copula		GMMN		Copula		GMMN	
d	C	PRS	QRS	PRS	QRS	PRS	QRS	PRS	QRS
2	t_4	0.0642	0.4420	1.2960	1.2720	0.1045	0.8210	3.6140	3.5820
2	C	0.0144	0.0230	1.3110	1.3140	0.0308	0.0400	3.6290	3.5820
2	G	0.0348	374.8000	1.3470	1.3310	0.0669	687.7000	3.6400	3.5750
(2,1)	NG	0.0633	NA	1.3360	1.3260	0.1369	NA	3.6560	3.6530
5	t_4	0.1410	1.4830	1.4490	1.3830	0.2567	3.0150	3.7330	3.6960
5	C	0.0425	0.0580	1.3890	1.5060	0.0936	0.1110	3.7670	3.6980
5	G	0.0523	1529.5000	1.3930	1.3860	0.1161	2939.9000	3.7380	3.7010
(2,3)	NG	0.0989	NA	1.4020	1.4070	0.2167	NA	3.9450	3.7210
10	t_4	0.2766	3.6080	1.5870	1.6720	0.4917	6.5290	3.9530	4.0990
10	C	0.0734	0.1190	1.6430	1.6630	0.1806	0.2320	3.9680	4.1910
10	G	0.0807	3579.6000	1.5500	1.5290	0.1984	7119.6000	4.0060	3.9370
(5,5)	NG	0.1324	NA	1.5470	1.5280	0.3087	NA	3.9740	3.9530

Table C.2: Elapsed times in seconds for generating samples of size $n_{\text{gen}} = 10^5$.

barely affected by the dimension and are not affected by the type of dependence model. For $d = 10$, the GMMN quasi-random sampling procedure even outperforms the t_4 quasi-random sampling procedure for which the conditional copulas are analytically available; for $d = 5$ the two procedures perform on par, depending on the machine used.

This highlights the universality of using neural networks for dependence modeling purposes. As an example, say a risk management application such as estimating expected shortfall with variance reduction is based on a t_4 copula and a regulator requires us to change the model to a Gumbel copula for stress testing purposes. Suddenly run time increases substantially. Also, if the regulator decides to incorporate hierarchies (as was more easily done for the t_4 model due to its correlation matrix) by utilizing a nested Gumbel copula, then there is suddenly no quasi-random sampling procedure known anymore. It is one of the biggest drawbacks of parametric copula models in applications that the level of difficulty of carrying out important statistical tasks such as sampling, fitting and goodness-of-fit can largely depend on the class of copulas used. These problems are eliminated with neural networks as dependence models.

C.2.2 Fitting and training times for data applications

We now briefly present the run times for fitting the parametric copula models and training the GMMNs used in Section 4.5. Recall that we considered three dimensions $d \in \{3, 5, 10\}$ and that fitting, respectively training, was only required once for each dimension, independently of the number of applications considered.

Table C.3 contains the elapsed times in seconds. Recall from Section 4.5 that GMMNs provided the best fit, followed by the unstructured t copula. The latter is in general a popular parametric copula model in practice; see Fischer et al. (2009). Comparing the last two columns of Table C.3, we see that fitting the t copula is comparably fast for $d = 3$, however, already for $d = 5$, run time for training a GMMN is on par. For $d = 10$, training a GMMN is significantly faster.

d	Gumbel	Clayton	Normal (ex)	Normal (un)	t (ex)	t (un)	GMMN
3	1.078	0.437	0.388	1.000	3.315	9.064	43.336
5	1.291	0.455	0.435	6.071	5.932	41.131	41.235
10	1.344	0.531	0.981	82.982	11.966	783.406	55.555

Table C.3: Elapsed times in seconds for fitting the respective parametric copula model and training the GMMN on one NVIDIA Tesla P100 GPU for the applications presented in Section 4.5.

C.2.3 TensorFlow vs R

Finally, let us stress again what we initially said, namely, that run time depends on many factors. In particular, one typically relies on TensorFlow for the feed-forward step of input through the GMMN, which creates overhead especially for smaller data size n. In the demo `GMMN_QMC_timings`, we also provide a pure R implementation for this step for GMMNs considered in this work.

For each of $d \in \{2, 5, 10\}$, we randomly initialize $B = 10$ GMMNs as in Algorithm 4.2.1 and average the elapsed times of their feed-forward steps when passing through data of size n (chosen equidistant in log-scale from 10 to 10^6) from the input distribution, once with TensorFlow, and once with our own R implementation. We then divide the averaged run times of the R implementation by the ones of the TensorFlow implementation. Whenever the ratio is smaller (larger) than one, the R implementation is faster (slower) than the

TensorFlow implementation. We ran this experiment once locally on the 2018 2.7 GHz Quad-Core Intel Core i7 processor and once on the NVIDIA Tesla P100 GPU. The results are depicted on the left and on the right plot in Figure C.2, respectively. Depending on the machine used, the R implementation can be significantly faster, especially for small n.

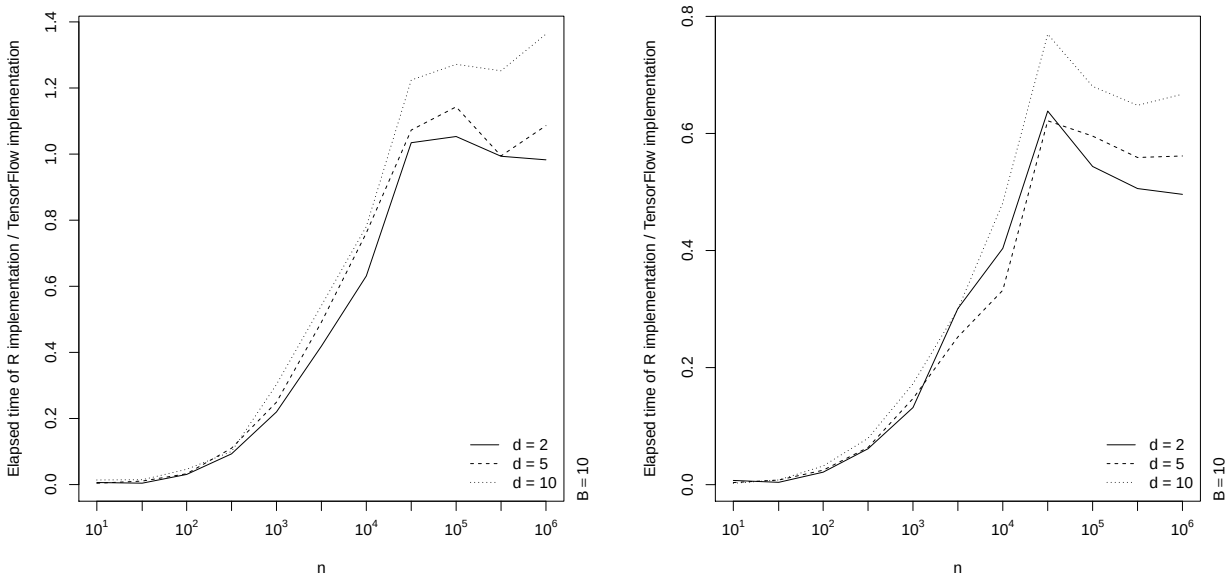

Figure C.2: Ratio of averaged elapsed times of an R implementation over the TensorFlow implementation when evaluating randomly initialized GMMNs, once run on a 2018 2.7 GHz Quad-Core Intel Core i7 processor (left) and once on an NVIDIA Tesla P100 GPU (right).