

University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Papers in Natural Resources

Natural Resources, School of

2018

Bayesian Model Selection of Regular Vine Copulas


Lutz F. Gruber

University of Nebraska-Lincoln, lutz.gruber@unl.edu

Claudia Czado

Technische Universitat Munchen

Follow this and additional works at: <https://digitalcommons.unl.edu/natrespapers>

 Part of the [Natural Resources and Conservation Commons](#), [Natural Resources Management and Policy Commons](#), and the [Other Environmental Sciences Commons](#)

Gruber, Lutz F. and Czado, Claudia, "Bayesian Model Selection of Regular Vine Copulas" (2018). *Papers in Natural Resources*. 1019.
<https://digitalcommons.unl.edu/natrespapers/1019>

This Article is brought to you for free and open access by the Natural Resources, School of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Papers in Natural Resources by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

Bayesian Model Selection of Regular Vine Copulas

Lutz F. Gruber^{*,†} and Claudia Czado[‡]

Abstract. Regular vine copulas are a flexible class of dependence models, but Bayesian methodology for model selection and inference is not yet fully developed. We propose sparsity-inducing but otherwise non-informative priors, and present novel proposals to enable reversible jump Markov chain Monte Carlo posterior simulation for Bayesian model selection and inference. Our method is the first to jointly estimate the posterior distribution of all trees of a regular vine copula. This represents a substantial improvement over existing frequentist and Bayesian strategies, which can only select one tree at a time and are known to induce bias. A simulation study demonstrates the feasibility of our strategy and shows that it combines superior selection and reduced computation time compared to Bayesian tree-by-tree selection. In a real data example, we forecast the daily expected tail loss of a portfolio of nine exchange-traded funds using a fully Bayesian multivariate dynamic model built around Bayesian regular vine copulas to illustrate our model's viability for financial analysis and risk estimation.

Keywords: multivariate analysis, dependence modeling, copula modeling, vine copulas, Bayesian inference, posterior simulation, importance sampling, simulation studies, financial analysis, risk forecasting.

1 Introduction

Multivariate models do not typically allow much customization of either marginal or dependence characteristics. Copula modeling is a more flexible approach to capture multivariate dependencies, in which univariate models describe marginal effects and the joint dependence effects are modeled by a copula (Nelsen, 2006; McNeil et al., 2005; Kurowicka and Cooke, 2006; Kurowicka and Joe, 2010). That marginal models are not required to be from the same family or to share any characteristics makes copulas a very convenient tool for statistical analyses. More specifically, a copula is a multivariate distribution function C with uniform marginals. It forms a multivariate distribution $F_{1:d}$ out of the univariate marginal distributions F_1, \dots, F_d by modeling the joint pattern of the transformed marginal data $u_i := F_i(x_i)$ (Sklar, 1959):

$$F_{1:d}(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)) := C(u_1, \dots, u_d).$$

A challenge of multivariate modeling is that multivariate models can not easily be constructed in higher dimensions. But there is a rich set of bivariate copulas available of

^{*}School of Natural Resources, University of Nebraska-Lincoln, Lincoln, Nebraska,
lutz.gruber@unl.edu

[†]QuantCo, Inc., Cambridge, Massachusetts

[‡]Center for Mathematics, Technische Universität München, Munich, Germany

which the theoretical properties are known and the densities are analytically tractable (Joe, 2001). This motivates the pair copula construction: combine a number of different bivariate—“pair”—copulas using nested conditioning to create a multivariate copula (Joe, 1996; Bedford and Cooke, 2001).

A regular vine copula pair copula construction uses a sequence of linked trees $\mathcal{V} = (T_1, \dots, T_{d-1})$, called the regular vine, to specify the building plan of the pair copula construction. Tree $T_j = (N_j, E_j)$, $1 \leq j \leq (d-1)$, corresponds each edge $e \in E_j$ to a parametric pair copula that is conditional on $d-j$ variables; \mathcal{B}_e denotes the copula family of this pair copula and $\theta_{e; \mathcal{B}_e}$ denotes its parameters.

Current literature treats tree-by-tree selection procedures that first estimate tree T_1 of the regular vine \mathcal{V} , then estimate the tree T_2 conditional on the estimate of tree T_1 , and so forth, until they estimate the last tree T_{d-1} conditional on the estimates of trees T_1, T_2, \dots, T_{d-2} ; examples are Gruber and Czado (2015)’s stepwise Bayesian approach and Dißmann et al. (2013)’s frequentist stepwise heuristics-based approach. These methods perform extremely well when the selection of the regular vine tree structure is not important; this is the case, for example, with Gaussian data. Selection methods that proceed tree-by-tree work less effectively with more complex data. Existing literature on Bayesian model selection of vine copulas is mostly limited to selecting pair copulas \mathcal{B}_e , $e \in E_k$ of trees $T_k \in \mathcal{V}$, conditional on a given regular vine tree structure \mathcal{V} : Smith et al. (2010); Min and Czado (2010, 2011) discuss the selection of pair copulas in drawable vine copulas, which are a subclass of regular vine copulas. Gruber and Czado (2015)’s is the only Bayesian strategy to also select the regular vine tree structure \mathcal{V} , however, as mentioned above, it proceeds tree-by-tree.

We propose sparsity-inducing but otherwise non-informative priors and provide a fully Bayesian extension of Gruber and Czado (2015)’s Bayesian tree-by-tree method that estimates all levels of a regular vine copula jointly. Our method is the first to select all levels of a regular vine copula simultaneously and it is the only fully Bayesian model selection strategy that applies to the general class of regular vine copulas. Our simultaneous selection method eliminates bias induced by tree-by-tree model selection procedures and outperforms existing procedures by a wide margin, as is shown in extensive simulation studies. Our implementation is based on reversible jump Markov chain Monte Carlo (MCMC; Green, 1995), and uses innovative ideas to generate proposals that facilitate rapid convergence to high posterior density regions of the huge model space. One of the approaches discussed employs a pre-MCMC importance resampling step that allows entire regular vine tree structures be sampled according to a newly introduced importance score.

This paper is organized as follows. Section 2 and Supplementary Appendix A (Gruber and Czado, 2017) describe regular vine copulas and reversible jump MCMC, the key model and method used in this paper. Sections 3 and 4 present our fully Bayesian strategies to select the pair copula families (Section 3) and regular vine and pair copula families jointly (Section 4). Both sections discuss our choice of priors and sampling algorithms; the simulation studies for both methods are presented in Section 5. Section 6 shows a real data example about forecasting value at risk and expected tail loss of a portfolio of nine exchange-traded funds, and Section 7 concludes this paper.

Copula	Notation	Parameters	Kendall's τ	Tail-dependence
Independence	I		$\tau = 0$	$\lambda^L = \lambda^U = 0$
Gaussian	N	$\rho \in (-1, 1)$	$\tau = \frac{2}{\pi} \arcsin(\rho)$	$\lambda^L = \lambda^U = 0$
t	T	$\rho \in (-1, 1)$ $\nu > 1$	$\tau = \frac{2}{\pi} \arcsin(\rho)$	$\lambda^L = \lambda^U$ $= 2T_{\nu+1} \left(-\sqrt{(\nu+1)\frac{1-\rho}{1+\rho}} \right)$
Double Clayton I	C	$\delta \in \mathbb{R}$	$\tau = \frac{\delta}{2+ \delta }$	$\lambda^L = 2^{-1/ \delta }, \lambda^U = 0$
Double Clayton II	C	$\delta \in \mathbb{R}$	$\tau = -\frac{\delta}{2+ \delta }$	$\lambda^L = 0, \lambda^U = 2^{-1/ \delta }$
Double Gumbel I	G	$\delta \in \mathbb{R}$	$\tau = \frac{\delta}{1+ \delta }$	$\lambda^L = 0, \lambda^U = 2 - 2^{1/(1+ \delta)}$
Double Gumbel II	G	$\delta \in \mathbb{R}$	$\tau = -\frac{\delta}{1+ \delta }$	$\lambda^L = 2 - 2^{1/(1+ \delta)}, \lambda^U = 0$

Table 1: Candidate pair copula families for use in the pair copula construction, and their parameter transformations.

2 Methods and Notation

Regular vine copulas are pair copula constructions that use a regular vine tree sequence as their building plan and bivariate copulas as their building blocks (Joe, 1996; Bedford and Cooke, 2001). We summarize key definitions and results from these papers in Appendix A.

2.1 Notation

Regular Vines For each level k , \mathbf{STP}_k denotes the set of all spanning trees $T_k = (N_k, E_k)$ that satisfy the proximity condition for a given tree T_{k-1} (see Definition A.1 in Appendix A; Appendix A also shows an example of a regular vine tree sequence and permissible vine trees). The edge set E_{k-1} of tree N_{k-1} constitute the node set N_k of tree T_k for $k > 1$; for tree T_1 , N_1 is the set of integers from 1 to d . Note that the set of permissible tree \mathbf{STP}_k changes when changes are made to tree T_{k-1} . Furthermore, \mathbf{B} shall denote the set of candidate pair copula families; in the remainder \mathbf{B} will consist of the pair copula families listed in Table 1.

Pair Copulas Table 1 lists candidate pair copula families for use in the pair copula construction. The Double Clayton and Double Gumbel copulas combine different rotations of the regular Clayton and Gumbel copulas to also allow modeling of negative dependence:

$$c_{\text{Double Clayton I}}(u_1, u_2; \delta) = \begin{cases} c_{\text{Clayton}}(u_1, u_2; \delta) & \text{if } \delta \geq 0, \\ c_{\text{Clayton}}(1 - u_1, u_2; -\delta) & \text{if } \delta < 0, \end{cases} \tag{1}$$

$$c_{\text{Double Clayton II}}(u_1, u_2; \delta) = c_{\text{Double Clayton I}}(u_1, 1 - u_2; \delta), \tag{2}$$

$$c_{\text{Double Gumbel I}}(u_1, u_2; \delta) = \begin{cases} c_{\text{Gumbel}}(u_1, u_2; \delta + 1) & \text{if } \delta \geq 0, \\ c_{\text{Gumbel}}(1 - u_1, u_2; -\delta + 1) & \text{if } \delta < 0, \end{cases} \tag{3}$$

$$c_{\text{Double Gumbel II}}(u_1, u_2; \delta) = c_{\text{Double Gumbel I}}(u_1, 1 - u_2; \delta). \tag{4}$$

The parameter transformations from the copulas' natural parameters to their Kendall's τ 's are provided in Tables 1 and 2 of Brechmann and Schepsmeier (2013).

Pair Copula Parameters Throughout this paper, all one-parameter pair copulas are parameterized in terms of their Kendall's τ , and the t copula is parameterized by its Kendall's τ and the logarithm of its degrees of freedom. The common parameterization makes it easier to compare different copulas' parameters and propose good parameter values when the pair copula families change. The notation of the parameter (vector) of the copula of edge e is $\theta_e \equiv \tau_e$ for a one-parameter copulas and $\theta_e = (\tau_e, \log \nu_e)$ for the t copula. The parameters of all pair copulas of the regular vine copula are collectively referred to by $\theta_{\mathcal{V}}$.

MCMC Iterates The following assumes the selection of an d -dimensional regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$. The r -th MCMC iterate of a variable, or other quantity of interest, is superscripted by r . Proposals are superscripted by an asterisk sign (*), and updated variables in the r -th iteration are superscripted by r, NEW . Proposal distributions are denoted by q and priors are denoted by π . Subscripts may be used to detail the affiliation of these quantities.

2.2 Reversible Jump MCMC

Reversible jump MCMC (Green, 1995) is an extension of the classic Metropolis-Hasting algorithm (Metropolis et al., 1953; Hastings, 1970) that enables simulation from target distributions of varying dimensionality, such as they appear in Bayesian model selection. Reversible jump MCMC has a long history of being used in Bayesian model selection of vine copulas, see, for example, Min and Czado (2011); Gruber and Czado (2015). Our approach is an evolution of Gruber and Czado (2015)'s to estimate all levels of the regular vine copula jointly. In our application, the model space is the set of all d -dimensional regular vine copulas $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \cdot)$. The parameter space is the set of all valid parameters $\theta_{\mathcal{V}}$ of the vine copula's pair copulas.

There are two kind of moves to update the sampling chain: within-model moves, which update only the parameters within a given model, and between-model moves, which update the model and its parameters simultaneously. We use standard Metropolis-Hastings updates for the within-model moves, and carefully designed between-model updates to achieve satisfactory exploration of the model space. Our between-model moves are detailed in Sections 3 and 4. Algorithm 2.1 shows the general reversible jump MCMC mechanism for posterior simulation.

Algorithm 2.1 (Reversible Jump MCMC for Posterior Simulation).

1: *Select starting values: set the regular vine*

$$\mathcal{V}^0 = (T_1^0 = (N_1^0, E_1^0), \dots, T_{d-1}^0 = (N_{d-1}^0 = E_{d-2}^0, E_{d-1}^0))$$

to an arbitrary regular vine on d dimensions; set all pair copula families $\mathcal{B}_{\mathcal{V}}^0$ to the independence copula, i.e., $c_{e; \mathcal{B}_e}(\cdot, \cdot) = 1$ for $e \in E_1^0, \dots, E_{d-1}^0$.

2: **for each** MCMC iteration $r = 1, \dots, R$ **do**

- 3: Perform a within-model move: update the parameters $\theta_{\mathcal{V}}$. Obtain updated parameters $\theta_{\mathcal{V}}^{r,NEW}$ through a Metropolis-Hastings step with random walk proposals from a mixture of normal distributions:

$$C^r = (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \theta_{\mathcal{V}}^r) := (\mathcal{V}^{r-1}, \mathcal{B}_{\mathcal{V}}^{r-1}, \theta_{\mathcal{V}}^{r,NEW}).$$

- 4: Perform a between-model move: update the regular vine \mathcal{V} along with, or only, the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$ to $\mathcal{V}^{r,NEW}$, $\mathcal{B}_{\mathcal{V}}^{r,NEW}$ and $\theta_{\mathcal{V}}^{r,NEW}$ (Algorithms B.1, B.2 of Appendix B):

$$C^r = (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \theta_{\mathcal{V}}^r) := (\mathcal{V}^{r,NEW}, \mathcal{B}_{\mathcal{V}}^{r,NEW}, \theta_{\mathcal{V}}^{r,NEW}).$$

5: **end for**

6: **return** the Bayesian posterior sample

$$(C^r)_{r=1,\dots,R} = (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \theta_{\mathcal{V}}^r)_{r=1,\dots,R}.$$

3 Posterior Inference for Pair Copula Families

Our Bayesian selection strategy extends Min and Czado (2011)'s, which can only select the pair copulas of drawable vine copulas, to select the pair copulas of general regular vine copulas. Furthermore, our method does not share Smith et al. (2010)'s limitation to only detect conditional independencies versus one global pair copula alternative: we can select different copula families for each pair, and the set of candidate copula families \mathbf{B} is not limited to one copula family.

A scenario in which the regular vine tree structure is known is when the modeler wishes to obtain a closed-form parametric (conditional) distribution for a specific subset of variables. For example, the modeler could want to know what the conditional distribution of variable 1 given variable 2 is; in this case, the modeler would select a regular vine tree structure that contains the respective edges, and only leaves open the selection of the pair copula families. In most cases, however, the regular vine tree structure is unknown and requires model selection, which is treated in Section 4.

3.1 Priors

We choose priors that induce model sparsity, but do not otherwise skew the posterior. Specifically, we assume

$$\pi(\mathcal{B}_{\mathcal{V}}) \propto \exp(-\lambda d_{\mathcal{B}_{\mathcal{V}}}), \tag{5}$$

$$\pi(\theta_e | \mathcal{B}_{\mathcal{V}}) \propto \begin{cases} \text{Uniform}_{(-1,1)}(\tau_e) & \text{if } \mathcal{B}_e \text{ is a one-parameter copula,} \\ \left(\text{Uniform}_{(-1,1)}(\tau_e) \cdot \frac{\mathbf{1}_{(1,30)}(\nu_e) \cdot \log(\nu_e)}{\int_1^{30} \log(x) dx} \right) & \text{if } \mathcal{B}_e \text{ is a t copula,} \end{cases} \tag{6}$$

where $d_{\mathcal{B}_{\mathcal{V}}}$ denotes the number of parameters of the regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$ or, equivalently, the dimension of the parameter vector $\theta_{\mathcal{V}}$. The prior on the parameters

$\theta_{\mathcal{V}}$ is uniform prior on the Kendall's τ 's and the log-degrees of freedom of the t copula. We limit the degrees of freedom from above by 30 to make the prior proper, and because the t copula becomes too similar to the Gaussian copula as ν increases.

When the effect of $\pi(\theta_e \mid \mathcal{B}_{\mathcal{V}})$ is neglected, the prior on the pair copula families $\mathcal{B}_{\mathcal{V}}$ has some appealing characteristics: for $\lambda = 0$, the posterior mode will be at the global maximum likelihood model; for $\lambda = 1$, the posterior mode will be at the global AIC (Akaike information criterion) optimum; while $\lambda > 1$ will provide even stronger shrinkage. Given that the independence pair copula is not likelihood-identifiable when compared with a Gaussian copula with Kendall's $\tau = 0$, shrinkage priors are an effective tool to reduce over-fitting. We find that $\lambda = 1$ yields a desirous degree of model parsimony without being too restrictive, and will use this value in the simulation study and data example below.

3.2 Between-Model Move to Update $(\mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$

This between-model move for our reversible jump MCMC sampler (Algorithm 2.1) updates the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$. It does not change the regular vine $\mathcal{V} = (T_1, \dots, T_{d-1})$, which specifies the building plan of the pair copula construction. This between-model move consists of a proposal step (Lines 1–13 of Algorithm B.1 of Appendix B) and an acceptance/rejection step (Line 14). The line numbers in this section refer to Algorithm B.1 of Appendix B.

Proposal Step The first step selects how many pair copulas are updated, N , and selects this many edges $E \subseteq E_1 \cup \dots \cup E_{d-1}$ as representatives of the pair copula families to be updated, $\mathcal{B}_E = (\mathcal{B}_e \mid e \in E)$ (Lines 1 and 2). After this, the algorithm iterates through each selected pair copula $e \in E$ to propose a new copula family $\mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r$ (Lines 4–13). Our proposal step guarantees that the proposal differs in exactly N pair copula families from the current state by excluding the current copula family from the set of qualifying candidate families. An auxiliary step evaluates the likelihood of each candidate pair copula family $\mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r$ with their parameters $\tilde{\theta}_{e;\mathcal{B}_e^*}$ chosen to match the current copula's Kendall's τ and tail-dependence coefficients λ^L or λ^U . If the parameters of the proposal copula are not identifiable by Kendall's τ and the upper and lower tail-dependence coefficients, λ^U and λ^L , respectively, (maximum likelihood) parameter estimation can be used to obtain $\tilde{\theta}_{e;\mathcal{B}_e^*}$. We align the proposal distribution closely with the posterior by making the proposal weights of each candidate family proportional to its likelihood (Line 6):

$$q_{\mathcal{B}}(\mathcal{B}_e^r \rightarrow \mathcal{B}_e^*) \propto L(\mathcal{B}_e^*; \tilde{\theta}_{e;\mathcal{B}_e^*} \mid \mathbf{U}) \text{ for } \mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r;$$

but re-weight small proposal probabilities to observe a lower bound that ensures that the acceptance probabilities of moves away from states with low proposal probabilities are large enough for good mixing behavior of the sampling chain (Line 7): if $q_{\mathcal{B}}(\mathcal{B}_e^r \rightarrow \mathcal{B}_e^*) < \kappa \max_{\mathcal{B} \in \mathbf{B} \setminus \mathcal{B}_e^r} q_{\mathcal{B}}(\mathcal{B}_e^r \rightarrow \mathcal{B})$ for some candidate pair copula \mathcal{B}_e^* , then we set $q_{\mathcal{B}}(\mathcal{B}_e^r \rightarrow \mathcal{B}_e^*) = \kappa \max_{\mathcal{B} \in \mathbf{B} \setminus \mathcal{B}_e^r} q_{\mathcal{B}}(\mathcal{B}_e^r \rightarrow \mathcal{B})$.

After the selected pair copulas' proposal families are drawn from these proposal distributions, a new parameter vector $\theta_{e;\mathcal{B}_e^*}^*$ is proposed for every pair copula of the regular

vine copula. For pair copulas $e \in E$ whose families were changed, new parameters $\theta_{e;\mathcal{B}_e^*}^*$ come from a truncated normal mixture distribution centered at the matched parameter $\tilde{\theta}_{e;\mathcal{B}_e^*}$ and truncated to the domain of the respective parameters; for all other pair copulas $e \notin E$, the new parameters $\theta_{e;\mathcal{B}_e^*}^*$ come from a truncated normal mixture distribution centered at the current parameters' values $\theta_{e;\mathcal{B}_e^r}^r$ (Line 8). The mixture proposals improve the acceptance rate of the proposals as follows. A small variance component tends to produce high posterior density proposals; and a high variance component increases the proposal probability of the return move $\phi(\theta_{e;\mathcal{B}_e^r,\Sigma_i}^r)(\theta_{e;\mathcal{B}_e^r}^r)$ in the numerator of the acceptance probability (7), especially when the current state $\theta_{e;\mathcal{B}_e^r}^r$ of the sampling chain is not close to the proposal mean $\theta_{e;\mathcal{B}_e^*}^*$. The covariance matrices Σ_i and mixture weights ω_i of the parameter proposal distribution are tuning parameters.

Acceptance Step The acceptance/rejection step uses the well-established Metropolis-Hastings acceptance probability of a proposal $\mathcal{C}^* := (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^*, \theta_{\mathcal{V}}^*)$ to ensure that the posterior distribution is the equilibrium distribution of the sampling chain (Line 14),

$$\alpha = \frac{L(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^*, \theta_{\mathcal{V}}^* | \mathbf{U})}{L(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \theta_{\mathcal{V}}^r | \mathbf{U})} \cdot \frac{\pi(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^*, \theta_{\mathcal{V}}^*)}{\pi(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \theta_{\mathcal{V}}^r)} \cdot \prod_{e \in E} \frac{q_{\mathcal{B}(\mathcal{B}_e^* \rightarrow \mathcal{B}_e^r)}}{q_{\mathcal{B}(\mathcal{B}_e^r \rightarrow \mathcal{B}_e^*)}} \cdot \sum_i \omega_i \left(\prod_{e \in E} \frac{\phi(\tilde{\theta}_{e;\mathcal{B}_e^r,\Sigma_i})(\theta_{e;\mathcal{B}_e^r}^r)}{\phi(\tilde{\theta}_{e;\mathcal{B}_e^*,\Sigma_i})(\theta_{e;\mathcal{B}_e^*}^*)} \cdot \prod_{e \notin E} \frac{\phi(\theta_{e;\mathcal{B}_e^r,\Sigma_i}^r)(\theta_{e;\mathcal{B}_e^r}^r)}{\phi(\theta_{e;\mathcal{B}_e^r,\Sigma_i}^r)(\theta_{e;\mathcal{B}_e^r}^r)} \right). \quad (7)$$

This representation of the acceptance probability uses the likelihood times prior proportionality of the posterior density. We write $\phi_{(\mu,\Sigma)}(\cdot)$ for the density of the multivariate normal distribution with mean μ and covariance matrix Σ . Equation (7) accounts for any birth/death moves by implicitly shrinking or expanding the interpretation of $\phi_{(\mu,\Sigma)}(\cdot)$ as the density function of a variable-dimension normal distribution.

Pseudo Code Pseudo code of this update procedure is provided in Appendix B.1.

4 Joint Posterior Inference for the Regular Vine and Pair Copulas

Selection of the regular vine tree structure \mathcal{V} is more complex and computationally intensive than selection of the pair copula families $\mathcal{B}_{\mathcal{V}}$. The main challenge for selection of the tree structure is the vast candidate model space: there are $\frac{d!}{2} \times 2^{\binom{d-2}{2}}$ different d -dimensional regular vine tree structures (Morales-Napoles, 2011). To illustrate this point, there are approximately 4.87e+14 different regular vines on 10 dimensions. This number is too high to expect posterior simulation to explore the full model space in feasible runtime. The key performance metric for any Bayesian model selection scheme thus becomes how quickly and reliably it moves towards high posterior density regions.

We present two different approaches for between-model moves to jointly update the tree structure \mathcal{V} and pair copula families $\mathcal{B}_{\mathcal{V}}$. The first between-model move (Section 4.3) is a local-search update that builds up the proposal for a new regular vine \mathcal{V} tree-by-

Dimension d	Vine Search Space		Vine Copula Search Space	
	Joint Selection	Stepwise Selection	Joint Selection	Stepwise Selection
2	1	1	7	7
3	3	3	1,029	154
4	24	< 20	2,823,576	< 5,642
5	480	< 145	1.3559e+11	< 305,767
6	23,040	< 1,441	1.0938e+17	< 22,087,639
7	2,580,480	< 18,248	1.4413e+24	< 1.9994e+9
8	660,602,880	< 280,392	3.0387e+32	< 2.1789e+11
9	3.8051e+11	< 5,063,361	1.0090e+42	< 2.7791e+13
10	4.8705e+14	< 105,063,361	5.2118e+52	< 4.0632e+15

Table 2: Size of the search space for vines \mathcal{V} and vine copulas $(\mathcal{V}, \mathcal{B}_{\mathcal{V}})$ with seven candidate families, i.e., $|\mathbf{B}| = 7$, by dimension d .

tree, starting from the current state vine; the second move (Section 4.4) uses importance sampling to draw entire regular vine tree sequences from a weighted distribution of vines.

4.1 Differences to Gruber and Czado (2015)’s Selection Method

Gruber and Czado (2015)’s method performs a Bayesian tree-by-tree posterior simulation of each level $k = 1, \dots, d - 1$ of a regular vine copula, conditional on already selected states of the previous levels $1, \dots, k - 1$. The posterior distribution of each level is collapsed into the posterior mode so that the model selection procedure can proceed to the next level $k + 1$. In the end, one obtains a level-by-level Bayesian procedure that produces a point estimate of the model. In this paper, we present a method to estimate the posterior distribution of all levels of a regular vine copula jointly—the output are many different regular vine copulas that represent draws from the posterior distribution of all regular vine copulas.

The key conceptual adjustment to change the equilibrium distribution of the sampling chain is to swap the two nested for-loops in the general sampling algorithm: in Gruber and Czado (2015), the outer for-loop iterates through the levels of the regular vine copula and the inner for-loop runs through the MCMC iterations; contrariwise, our outer for-loop runs through the MCMC iterations while the inner for-loop iterates through the levels of the regular vine copula.

This seemingly trivial swap of the nested for-loops comes with significant challenges for successful implementation in practice: the model search space of simultaneous selection of all levels is enormously larger than for tree-by-tree selection (Table 2 from (Gruber and Czado, 2015, Table 1)). We crafted a proposal mechanism for between-model moves that achieves quick convergence to desirable models in the large candidate space while only requiring few tuning parameters. That said, the acceptance probabilities of between-model moves are still sensitive to the choice of parameter proposal distributions, given each step updates a large number of parameters.

4.2 Priors

We choose priors that enforce model sparsity, but do not make structural assumptions about the vine copula. Specifically, we assume

$$\pi(\mathcal{V}) = \textit{discrete Uniform}(\cdot), \tag{8}$$

$$\pi(\mathcal{B}_{\mathcal{V}} \mid \mathcal{V}) \propto \exp(-\lambda d_{\mathcal{B}_{\mathcal{V}}}), \tag{9}$$

$$\pi(\theta_e \mid \mathcal{V}, \mathcal{B}_{\mathcal{V}}) \propto \begin{cases} \textit{Uniform}_{(-1,1)}(\tau_e) & \text{if } \mathcal{B}_e \text{ is a one-parameter copula,} \\ \left(\textit{Uniform}_{(-1,1)}(\tau_e) \cdot \frac{\mathbf{1}_{(1,30)}(\nu_e) \cdot \log(\nu_e)}{\int_1^{30} \log(x) dx} \right) & \text{if } \mathcal{B}_e \text{ is a t copula,} \end{cases} \tag{10}$$

where $d_{\mathcal{B}_{\mathcal{V}}}$ denotes the number of parameters of the regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$ or, equivalently, the dimension of the parameter vector $\theta_{\mathcal{V}}$. Conditionally on the regular vine \mathcal{V} , the priors on the pair copulas $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$ are the same as the ones used in Section 3.1. Again, we will use $\lambda = 1$ value in the simulation study and data example below.

4.3 Between-Model Move to Update $(\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$ (Version I)

The first step selects the lowest level K of the regular vine copula that will be changed in this iteration (Line 1 of Algorithm B.2 of Appendix B); all subsequent trees T_K, \dots, T_{d-1} will have to be adjusted, too, as the proximity condition (Definition A.1) ties them to the lower-level trees. This between-model move leaves the trees and pair copulas of the levels $k = 1, \dots, K - 1$ unchanged from the current state. The line numbers in this section refer to Algorithm B.2 of Appendix B.

Vine Proposal Step The proposal for tree $T_K^* = (N_K, E_K^*) \in \mathbf{STP}_K \setminus T_K^r$ of level K can come from a weighted distribution over the set of candidate trees \mathbf{STP}_K (Line 4). Possible versions of this include independent uniform proposals (11), Kendall’s τ -weighted proposals (12), or random walk proposals (13),

$$q_T(T_K^r \rightarrow T_K^*) = q_T(T_K^*) \propto 1, \tag{11}$$

$$q_T(T_K^r \rightarrow T_K^*) = q_T(T_K^*) \propto \prod_{e \in E_K^*} (\delta + |\tau_e|), \tag{12}$$

$$q_T(T_K^r \rightarrow T_K^*) \propto p^{|E_K^* \cap E_K^r|} \cdot (1 - p)^{|E_K^* \setminus E_K^r|}. \tag{13}$$

The parameters p and δ of the proposal distributions for tree T_K are MCMC tuning parameters. Values $p > 0.5$ increase the probability that the proposal tree T_K^* has many common edges with the current state tree T_K^r ; the situation is reversed for $p < 0.5$. Small values of δ skew the proposal distribution towards trees T_K^* with heavy edge weights (in absolute Kendall’s τ ’s), while large values of δ decrease the impact of edge weights on the proposal probabilities, which makes the proposal distribution more uniform.

The proposals for trees T_{K+1}, \dots, T_{d-1} are drawn from a discrete uniform distribution over all permissible trees, $q_T(T_k^r \rightarrow T_k^*) = q_T(T_k^*) = \frac{1}{|\mathbf{STP}_k|}$ (Line 6). Note that the

set \mathbf{STP}_k , and its cardinality, depend on the lower level trees T_1, \dots, T_{k-1} . The proposal probabilities will not generally cancel each other out in the acceptance probability, as a result (17).

Family and Parameter Proposal Step The proposals for the pair copula families $\mathcal{B}_k^* = \{\mathcal{B}_{e^*}^* \mid e^* \in E_k^*\}$ of levels $k = K:(d-1)$ are generated similar to Algorithm B.1 of Appendix B (Lines 4–13 in Algorithm B.1 of Appendix B; Lines 10–21 in Algorithm B.2 of Appendix B). However, the selection of the pair copula families is adapted in two ways: first, all pair copula families in \mathbf{B} are now permissible candidates; second, the parameters $\tilde{\boldsymbol{\theta}}_{e^*; \mathcal{B}_{e^*}^*}$ of each pair copula $e^* \in E_k^*$ are chosen such that the theoretical Kendall's τ , and tail-dependence coefficients λ^L and λ^U of the t copula, agree with the corresponding empirical quantities of the data $(\mathbf{u}_{i(e^*); D(e^*)}, \mathbf{u}_{j(e^*); D(e^*)})$. Alternatively, the parameters could be maximum likelihood-estimated, but this would be substantially more computationally expensive.

Once new trees and pair copula families are proposed, we propose new parameters $\boldsymbol{\theta}_{\mathcal{V}}^* = (\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{d-1}^*)$ for the pair copulas of all levels $k = 1, \dots, d-1$. The proposal parameters $\boldsymbol{\theta}_k^*$ for pair copulas e from levels $k = 1, \dots, K-1$ are drawn from a normal mixture distribution centered at the current parameters $\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^r$ (Line 20); the proposal parameters $\boldsymbol{\theta}_k^*$ for pair copulas e^* from levels $k = K, \dots, d-1$ are centered at the parameters $\tilde{\boldsymbol{\theta}}_{e^*; \mathcal{B}_{e^*}^*}$ (Line 15).

Proposal Summary The complete proposal state is $\mathcal{C}^* = (\mathcal{V}^*, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^*)$, where

$$\mathcal{V}^* = (T_1^r, \dots, T_{K-1}^r, T_K^*, \dots, T_{d-1}^*), \quad (14)$$

$$\mathcal{B}_{\mathcal{V}}^* = (\mathcal{B}_1^r, \dots, \mathcal{B}_{K-1}^r, \mathcal{B}_K^*, \dots, \mathcal{B}_{d-1}^*), \text{ and} \quad (15)$$

$$\boldsymbol{\theta}_{\mathcal{V}}^* = (\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{d-1}^*). \quad (16)$$

Acceptance Step The proposal $\mathcal{C}^* = (\mathcal{V}^*, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^*)$ is accepted with acceptance probability (Line 22)

$$\begin{aligned} \alpha = & \frac{L(\mathcal{V}^*, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^* \mid \mathbf{U})}{L(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \boldsymbol{\theta}_{\mathcal{V}}^r \mid \mathbf{U})} \cdot \frac{\pi(\mathcal{V}^*, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^*)}{\pi(\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \boldsymbol{\theta}_{\mathcal{V}}^r)} \\ & \cdot \prod_{k=K:(d-1)} \frac{q_T(T_k^* \rightarrow T_k^r)}{q_T(T_k^r \rightarrow T_k^*)} \cdot \prod_{k=K:(d-1)} \frac{\prod_{e \in E_k^r} q_{\mathcal{B}}(\mathcal{B}_e^r)}{\prod_{e^* \in E_k^*} q_{\mathcal{B}}(\mathcal{B}_{e^*}^*)} \\ & \cdot \sum_i \omega_i \left(\prod_{k=1:(K-1)} \prod_{e \in E_k^r} \frac{\phi(\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^r; \Sigma_i)(\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^r)}{\phi(\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^r; \Sigma_i)(\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^*)} \right. \\ & \left. \cdot \prod_{k=K:(d-1)} \frac{\prod_{e \in E_k^r} \phi(\tilde{\boldsymbol{\theta}}_{e; \mathcal{B}_e^r; \Sigma_i})(\boldsymbol{\theta}_{e; \mathcal{B}_e^r}^r)}{\prod_{e^* \in E_k^*} \phi(\tilde{\boldsymbol{\theta}}_{e^*; \mathcal{B}_{e^*}^*; \Sigma_i})(\boldsymbol{\theta}_{e^*; \mathcal{B}_{e^*}^*}^*)} \right). \quad (17) \end{aligned}$$

This Metropolis-Hastings acceptance probability ensures that the sampling chain will have the joint posterior distribution of the regular vine \mathcal{V} , pair copula families $\mathcal{B}_{\mathcal{V}}$ and

parameters $\theta_{\mathcal{V}}$ as its stationary distribution. Equation (17) uses the likelihood times prior proportionality of the posterior density.

Pseudo Code Pseudo code of this update procedure is provided in Appendix B.2.

4.4 Between-Model Move to Update $(\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$ (Version II)

This between-model move to update the regular vine \mathcal{V} generates a large importance sample of regular vines that it will draw its proposals from. The proposals for the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$ are generated as in the previous section.

Motivation We would like to generate proposals that lie in high-posterior density regions for most efficient MCMC sampling. We observe that the posterior density is typically dominated by the likelihood function of the model, unless overly informative priors are used. We propose a score $S_{\mathcal{V}}$ that approximates the expected log-likelihood of a regular vine copula with a given regular vine tree structure \mathcal{V} and will use importance re-sampling to generate proposals from a distribution $q_T(\mathcal{V}) \propto S_{\mathcal{V}}$.

Scoring We propose the sum of squared Kendall’s τ ’s of all pairs as an approximation of the expected log-likelihood of a regular vine tree structure \mathcal{V} :

$$S_{\mathcal{V}} := \sum_{k=1:(d-1)} \sum_{e \in E_k} \tau_e^2. \tag{18}$$

We exploit a few facts about regular vine copulas to back up our score approximation: 1) the log-likelihood of a regular vine copula can be obtained as the sum of the log-likelihoods of all pair copulas; 2) if all pair copula families are Gaussian, the resulting regular vine copula is a multivariate Gaussian copula irrespective of its regular vine tree structure; 3) the correlation parameter of each pair copula equals that pair’s partial correlation; 4) the correlation parameter Pearson’s ρ can be transformed to Kendall’s τ ; and 5) the expected likelihoods of the pair copula families listed in Table 1 tend to increase with the strength of association parameter Kendall’s τ .

Pre-MCMC Importance Sampling Before the start of our reversible jump MCMC sampler (Algorithm 2.1), we generate a large importance sample with different regular vine tree structures \mathcal{V} (Algorithm B.3 of Appendix B). We generate the samples \mathcal{V}^i , $i = 1, \dots, I$, from conditionally uniform tree-by-tree proposal distributions

$$q(\mathcal{V}) = \prod_{k=1:(d-1)} q(T_k | T_1, \dots, T_{k-1}), \text{ where} \tag{19}$$

$$q(T_k | T_1, \dots, T_{k-1}) = \frac{1}{|\mathbf{STP}_k|}. \tag{20}$$

Each sample \mathcal{V}^i is assigned importance weight $\alpha^i = \frac{S_{\mathcal{V}^i}}{q(\mathcal{V}^i)}$.

Algorithm	Parameter
B.1	$p(N = k) = \frac{1}{4} \log \left(1 - \frac{1-e^{-4}}{N_{\mathcal{V}}e^{-4}+k(1-e^{-4})} \right)$, where $N_{\mathcal{V}}$ is the number of pair copulas of the regular vine copula
B.1 and B.2	$\kappa = 0.05$
B.1 and B.2	$\omega_1 = 0.9$; $\Sigma_1 = 0.003^2$ for the Kendall's τ of one-parameter copulas; $\Sigma_1 = \begin{pmatrix} 0.003^2 & 0 \\ 0 & 0.03^2 \end{pmatrix}$ for the $(\tau, \log \nu)$ parameter vector of the t copula; $\omega_2 = 0.1$ and $\Sigma_2 = 10^2 \Sigma_1$
B.2	$q_T(T_K^r \rightarrow T_K^*) \propto \prod_{e \in E_K^*} (\delta + \tau_e)$ with $\delta = 0.2$; $q_T(T_k^r \rightarrow T_k^*) = \frac{1}{ \mathbf{STP}_k }$ for $k > K$

Table 3: MCMC tuning parameters for Algorithms B.1 and B.2 of Appendix B.

Vine Proposal Step At iteration r of Algorithm 2.1, a proposal regular vine \mathcal{V}^* is drawn from the importance sample $(\mathcal{V}^i, \alpha^i)$, $i = 1, \dots, I$ (Algorithm B.4 of Appendix B).

Pseudo Code Pseudo code detailing this importance sampling and update procedure is provided in Appendix B.3.

5 Simulation Studies

5.1 Estimation of $(\mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$

We generate multiple simulation data sets from different regular vine copulas to apply our Bayesian selection strategy to. Initial values for our MCMC simulation are regular vine copulas with the true models' tree structures but Gaussian pair copulas for all pairs.

Our reversible jump MCMC sampler was run with the tuning parameters described in Table 3. These settings were used throughout this paper: in the simulation study for the selection of the pair copula families for the 6-dimensional and 10-dimensional simulation data, for the joint selection of the regular vine structure and pair copula families for the 6-dimensional and 10-dimensional simulation data, and for the real data example with 9-dimensional data. The results are based on the last 15,000 MCMC iterations of a total of 20,000, and the analyses were replicated 100 times each.

Simulation Software

We implemented our model selection procedure in a proprietary C++ software package. Our software uses OpenMP for shared memory parallelization of the likelihood computation and parameter optimization. Our software uses the random spanning tree and minimum spanning tree algorithms provided by the boost graph library, and our numerical optimizer uses the CppAD library for automatic differentiation.

Scenario	1	2	3	4
# Pairs	15	15	15	15
# I pairs	0	0	10	0
# Selected as I	0	0	10	0
# T pairs	2	3	1	0
# Selected as T	2	3	1	0
# Shrunk to I, N, G, C	0	0	0	0
# N pairs	6	4	2	15
# Selected as N	6	4	2	15
# Shrunk to I	0	0	0	0
# G or C pairs	7	8	2	0
# Selected as G or C	7	8	2	0
# Shrunk to I	0	0	0	0

Table 4: Summary of the simulation study in 6 dimensions.

6-Dimensional Test Data

This analysis re-uses the simulation data sets from Gruber and Czado (2015), but with a different focus: here we assume the true models' regular vine tree structures as known and simultaneously select all pair copula families; Gruber and Czado (2015) selected the regular vine trees and the pair copula families sequentially by tree. All of the 100 simulation data sets for each scenario consist of 500 entries. The data generating regular vine copulas are shown in Tables 13–16 of Appendix D.

Table 4 summarizes the results of this study, based on the aggregated posterior distribution of the pair copula families across all 100 replications (see Tables 13–16 of Appendix D). The posterior mode family of every pair copula agrees with the one of the true model, providing empirical support of our selection strategy. The posterior modes of most pairs have empirical posterior probabilities in excess of 80%. Furthermore, the results of Scenario 3 also show that our shrinkage priors reliably detect independence pair copulas, which define sparsity patterns and can allow for model reduction.

It took about 50 minutes to generate 20,000 posterior samples for 10 parallel replications on a 32-core node.

10-Dimensional Test Data

Here we use small sample, 10-dimensional data with each simulation data set consisting of only 200 entries. We expect more widely-dispersed posteriors resulting from the combination of less information and a larger candidate model space. Again, the selection is replicated 100 times with independently drawn simulation data sets from each scenario to minimize sample bias.

There are seven main scenarios that cover three different regular vine structures (X1, X2, X3) that are truncated at different levels (T2, T3, etc.) to exhibit varying degrees of sparsity. The main goal of this simulation study is to establish that our Bayesian

Scenario	X1-T6	X1-T2	X2-T8	X2-T3	X3-T9	X3-T3	X3-T2
# Pairs	45	45	45	45	45	45	45
# I pairs	11	29	11	27	10	23	29
# Selected as I	11	29	10	27	6	23	29
# T pairs	6	3	5	3	8	6	5
# Selected as T	0	0	4	0	5	1	0
# Shrunk to N, G, C	6	3	1	3	2	5	5
# Shrunk to I	0	0	0	0	1	0	0
# N pairs	8	4	13	8	9	6	5
# Selected as N	7	4	8	6	7	6	5
# Shrunk to I	1	0	4	2	1	0	0
# G or C pairs	20	9	16	7	18	10	6
# Selected as G or C	15	9	12	6	15	9	6
# Shrunk to I	4	9	1	1	3	1	0

Table 5: Summary of the simulation study in 10 dimensions.

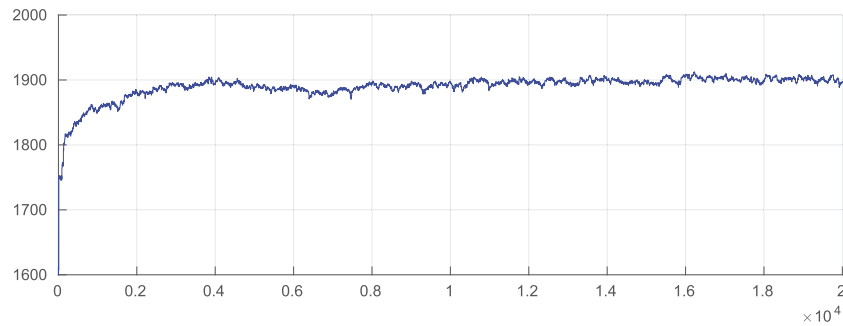


Figure 1: Log-likelihood trace plot of replication 1 of Scenario X3-T9.

selection method can converge quickly to high posterior density regions, shows good mixing behavior across different models, and identifies sparsity patterns.

Table 5 summarizes the results of this study. The data generating models as well as the complete aggregated posterior analysis is shown in Tables 17–23 of Appendix D. Detection of conditional independencies is excellent: in five scenarios, all conditional independencies are identified; in the remaining two scenarios 10 out of 11, and 6 out of 10 independence pairs are identified. Furthermore, most scenarios see additional pairs shrunk to (conditional) independence, and some t pair copulas shrunk to one-parameter copulas, as is expected from our combination of small sample size and a shrinkage prior. The vast majority of pair copulas selected for the remaining, not-shrunk, pairs (125 out of 131) preserve the original copulas' tail-dependence and symmetry characteristics.

The generation of 20,000 reversible jump MCMC iterations for 10 parallel scenarios took about 60 minutes to complete on a 32-core node. The log-likelihood trace plot of Figure 1 shows rapid convergence to a high-posterior density set; Scenario X3-T9 was

chosen for this illustration, because the smallest number of independence pairs should make it the most challenging for model selection. This suggests that our strategically designed proposals can quickly explore the candidate model space, which contained $|\mathbf{B}|^{45} = 7^{45}$ models. Our results illustrate that Bayesian model selection is possible in a very large discrete model space, and that the use of reversible jump MCMC can be suitable for such large-scale problems.

Summary Comments

Our fully Bayesian model selection strategy for the pair copula families of regular vine copulas extends beyond selection strategies discussed in existing literature. We showed that our proposed Bayesian selection method and reversible jump MCMC implementation work very well together in obtaining Bayesian posterior samples. Scalability in dimension d seems unproblematic, given that the computational run time and selection accuracy did not deteriorate significantly from dimension $d = 6$ to $d = 10$. Furthermore, our proposed shrinkage prior is effective in avoiding over-fitting, while it still lets the posterior sample retain all relevant copula pairs. As expected, there was no obvious effect of the tree structure on selection performance.

5.2 Joint Estimation of $(\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \theta_{\mathcal{V}})$

We will show empirical evidence that our reversible jump MCMC scheme selects suitable models and compare the results from our novel, fully Bayesian model selection method with those from selection methods suggested in existing literature (Dißmann et al. (2013) for frequentist and Gruber and Czado (2015) for Bayesian tree-by-tree selection) to provide context perspective and highlight the benefits of using our strategy.

The analysis will focus on evaluating the log-likelihoods as the main metric of model fit that separates regular vine copulas with different tree structures \mathcal{V} . Section 5.1 already evaluated sparsity detection and pair copula family selection of the pair copula family updates, which we will re-use from Section 3. Our analysis is based on the last 10,000 MCMC iterations out of 25,000. The quoted log-likelihoods of our fully Bayesian selection methods are the averages of the log-likelihoods from MCMC iterations $i = 15,001, \dots, 25,000$; the quoted log-likelihoods of the tree-by-tree methods are of these methods' point estimates. If the parameters from the fully Bayesian analysis were averaged to their posterior means, the log-likelihoods would increase by several units.

The between-model updates are generated from a 50%–50% mixture of Algorithms B.1 and B.2 of Appendix B. This means that in each iteration $r = 1, \dots, R$, with probability 50%, Algorithm 2.1 will update only the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$ (Algorithm B.1 of Appendix B), or jointly update the regular vine tree structure \mathcal{V} , pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\theta_{\mathcal{V}}$ (Algorithm B.2 of Appendix B). Given the increased complexity of this sampling scheme, all analyses are replicated only 50 times instead of 100 times as in Section 5.1.

Scenario	1	2	3	4
True model (MLE)	3782	3434	794	1390
Seq. frequentist selection	2883	2677	800	1383
Seq. Bayesian selection	3053	2916	796	1386
Fully Bayesian selection (I)	3661	3174	785	1382
Fully Bayesian selection (II)	3697	3226	785	1382

Table 6: Average log-likelihoods across all 50 replications; the log-likelihoods of the fully Bayesian procedures are the posterior means of the log-likelihoods, while all other log-likelihoods are those of the point estimates. The highest log-likelihoods are in bold.

6-Dimensional Test Data

This study uses the same 6-dimensional test data sets used in Section 5.1. Table 6 shows the comparative model fit of the selected models in terms of their log-likelihoods. Our proposed Bayesian selection strategies clearly outperform the existing methods in selecting regular vine copulas in Scenarios 1 and 2. The true model of Scenario 3 is truncated to the first level; this explains why the tree-by-tree selection methods perform on par with our fully Bayesian strategy here; Scenario 4 is of a multivariate Gaussian copula, so selection of the regular vine tree structure does not play a role and all methods perform on a level. Furthermore, our scoring-based proposals slightly outperform the Kendall's τ -based proposals.

An analysis of the sampling chains reveals that once the chain has converged to a local posterior mode, the regular vine tree structure \mathcal{V} tends to remain at the mode with further posterior variation only occurring in the pair copula families and parameters. Considering that different regular vine tree structures can lead to substantially different models, this is fully expected. The high log-likelihoods of Table 6 show that the tree structures \mathcal{V} occurring in the posterior sample represent suitable models, which suggests that the posterior simulation does not get stuck at premature levels.

It took about 2 hours to generate 25,000 posterior samples for eight replications in parallel on a 32-core node, and the preparation of the importance sample (size $I = 300,000$) was completed in several minutes.

10-Dimensional Test Data

This study uses the same 10-dimensional test data sets used in Section 5.1, and Table 7 compares the log-likelihoods of the selected models from different selection methods. We only compare against Dißmann et al. (2013)'s frequentist tree-by-tree selection method, given that Gruber and Czado (2015) only provided an analysis of 6-dimensional simulation data.

There are approximately $4.87e+14$ different regular vines on 10 dimensions, which makes exploration of the full model space practically impossible. Our carefully designed proposal mechanism enables quick convergence to high posterior density regions and succeeds in selecting more suitable models than Dißmann et al. (2013)'s frequentist strategy

Scenario	X3-T9	X2-T8	X1-T6	X2-T3	X3-T3	X1-T2	X3-T2
Number of non-trivial trees	9	8	6	3	3	2	2
True model (MLE)	1727	1707	1652	858	1256	1022	948
Seq. frequentist selection	1618	1339	1450	832	1265	1000	969
Fully Bayesian selection (I)	1717	1477	1480	822	1236	968	933
Fully Bayesian selection (II)	1729	1498	1477	825	1239	963	935

Table 7: Average log-likelihoods across all 50 replications; the log-likelihoods of the fully Bayesian procedures are the posterior means of the log-likelihoods, while all other log-likelihoods are those of the point estimates. The highest log-likelihoods are in bold.

in the most complex scenarios (X1-T6, X2-T8, and X3-T9). The superior performance of simultaneous selection in estimating complex dependence models is the most significant result of this study, given that we developed simultaneous selection specifically to estimate complex dependence structures, which were mis-selected by tree-by-tree methods. The results in Table 7 show not only substantial improvement of goodness of fit relative to tree-by-tree selection, but also excellent performance in absolute terms measured against the maximum likelihoods of the true models.

The models selected by the frequentist tree-by-tree selection method have slightly higher log-likelihoods than the average log-likelihood of our fully Bayesian posterior samples in Scenarios X1-T2, X2-T3, X3-T3, and X3-T2, which represent models of reduced complexity. In Scenarios X1-T2 and X3-T2, all pair copulas on levels greater or equal to 3 are independence copulas, and in Scenarios X2-T3 and X3-T3 or all pair copulas on levels greater or equal to 4 are independence copulas. Heavily truncated models benefit tree-by-tree selection and harm simultaneous selection of all trees because the latter still searches the enormously large model space of all 10-dimensional regular vine copula while tree-by-tree selection searches a much smaller model space, at each step conditional on assuming all higher order trees consist of independence pair copulas only. Even so, fully Bayesian selection is just narrowly behind tree-by-tree selection here, while tree-by-tree selection performed substantially weaker than simultaneous selection in selecting the complex models discussed above (Scenarios X1-T6, X2-T8, and X3-T9).

It took about 3–4 hours, depending on the scenario, to generate 25,000 posterior samples for three replications in parallel on a 32-core node; the score resampling based strategy required an additional 40–50 minutes to prepare the importance sample (size $I = 500,000$) before the start of reversible jump MCMC sampling.

Summary Comments

Our fully Bayesian model selection strategy for regular vine copulas is the first of its kind in two ways: it is the first selection method to estimate all levels of a regular vine copula jointly, and it is the first selection method to yield a fully Bayesian posterior sample. Our simulation study shows that our proposed Bayesian selection method and our reversible jump MCMC implementation work very well together in selecting superior models, and perform better than existing methods when working with complex dependence structures.

Fully Bayesian selection of the regular vine tree structure \mathcal{V} is challenged by the faster-than-exponential growth of the model space in dimension d . Our study showed that our Bayesian methods work extremely well in our $d = 6$ and $d = 10$ -dimensional simulation scenarios, especially when the data shows very complex dependence structures. Computing time restrictions mean that in practice only an increasingly small fraction of the total model space can be explored as the dimension of the problem is increased. With that in mind, we decided to run our MCMC sampler for “only” 25,000 iterations to highlight its quick convergence to high-posterior density regions, which is a key to successful application in practice.

Our fully Bayesian analysis is substantially faster than Gruber and Czado (2015)’s Bayesian tree-by-tree analysis, which performs a full posterior simulation for each level $k = 1, \dots, d - 1$. In contrast, our fully Bayesian strategy performs only one posterior simulation. The models selected by our method with pre-MCMC importance sampling (Version II) perform slightly better than the models from our sampler without that pre-processing step; however, this improvement in selection comes at the cost of additional computing time. It appears pragmatic to use the faster version without pre-MCMC importance sampling in most cases, given that it already provides all key benefits of simultaneous selection at less cost. Compared to tree-by-tree methods, both versions of our simultaneous selection method offer faster computation and better selection performance making them universally superior to previous approaches.

Our results suggest that full Bayesian analysis is most beneficial if there are substantial conditional dependencies in the data. If the variables are mostly conditionally independent, tree-by-tree selection methods are likely to perform just as well.

6 Example: Forecasting Portfolio Value at Risk and Expected Tail Loss

We provide a novel, and more extensive analysis of Gruber and Czado (2015)’s financial data set. Again, we set up a joint multivariate model through marginal time series DLMS and a copula dependence model. Our analysis will focus on out-of-sample forecasts of value at risk and expected tail loss (also called conditional value at risk). The value at risk at level $\alpha\%$ of a portfolio return r is just the $(1 - \alpha\%)$ -quantile of its distribution, and it gives a worst case estimate under the assumption that the realized outcome will be within the “good $\alpha\%$ of scenarios.” On the other hand, the expected tail loss is the conditional expectation of the return r , given the “bad $(1 - \alpha\%)$ of scenarios” (Acerbi and Tasche, 2003).

6.1 Description of the Data

The data contains 440 daily historical closing prices from January 2013 through September 2014 of nine exchange-traded funds (ETFs). The data were downloaded from <http://finance.yahoo.com>, and the nine selected ETFs are described in Table 8. These nine ETFs serve as an example of a diversified portfolio that the average retail investor could invest in and cover U.S. stocks, corporates, real estate, and commodities.

j	Symbol	Name	Exposure
1	IVV	iShares Core S&P 500 ETF	Large-cap U.S. stocks
2	IJH	iShares Core S&P Mid-Cap ETF	Mid-cap U.S. stocks
3	IJR	iShares Core S&P Small-Cap ETF	Small-cap U.S. stocks
4	HYG	iShares iBoxx \$ High Yield Corporate Bond ETF	High yield corporate bonds
5	LQD	iShares iBoxx \$ Investment Grade Corporate Bond ETF	U.S. investment grade corporate bonds
6	RTL	iShares Retail Real Estate Capped ETF	U.S. retail property real estate stocks and real estate investment trusts (REITs)
7	REZ	iShares Residential REIT Capped ETF	U.S. residential real estate stocks and real estate investment trusts (REITs)
8	SLV	iShares Silver Trust	Silver
9	IAU	iShares Gold Trust	Gold

Table 8: Details of the selected ETFs.

The daily prices are transformed to daily log-returns, and the log-returns of each ETF are modeled by a univariate dynamic linear model (DLM; West and Harrison (1997); Prado and West (2010)). Details of the marginal models are provided in Appendix C.

6.2 Joint Multivariate Model

The joint multivariate model is composed of all nine marginal univariate DLMS and a copula as the dependence model.

Learning and Forecasting The portfolio forecasting period begins in 2014, given that we use the 2013 data ($t = 1:252$) to estimate the posterior distribution of the regular vine copulas. We use Bayesian model averaging (BMA; Hoeting et al., 1990) over this posterior sample to generate the copula forecasts on the 9-dimensional unit cube, which then feed into the quantile function of the forecast t distributions of the marginal time series DLMS to generate sequential out-of-sample forecasts on the observation scale. While we do not change the copula over time, the univariate DLMS are updated sequentially on each day trading day $t = 1:440$; the out-of-sample prediction period contains 188 trading days, $t = 253:440$. This forecasting process is summarized below:

- 1: Select the copula using the u -data from 2013 ($t = 1:252$).
- 2: **for** each day $t = 253:440$ of 2014 **do**
- 3: Update the univariate DLMS using observation \mathbf{y}_t ;
- 4: Apply the time evolution step to get the step-ahead priors;
- 5: Sample $k = 1:N$ multivariate $\mathbf{u}^k = (u_1^k, \dots, u_9^k)$ -vectors from the posterior-weighted mixture of vine copulas (BMA);

	# N pairs	# T pairs	# G pairs	# C pairs	# I pairs
Seq. frequentist selection	5	11	7	6	7
Seq. Bayesian selection	10	0	3	0	23
Bayesian Family selection	11	6	10	2	6
Fully Bayesian selection (I)	11	4	9	1	11
Fully Bayesian selection (II)	10	6	5	4	11

Table 9: Summary of selected models for the financial data set.

6: Estimate the forecast distribution of \mathbf{y}_{t+1} from the Monte Carlo sample

$$\hat{\mathbf{y}}_{t+1}^k = (T_{\text{non std}}^{-1}(u_1^k; r_{1,t+1}, a_{1,t+1}, R_{1,t+1} + c_{1,t+1}), \dots, T_{\text{non std}}^{-1}(u_9^k; r_{9,t+1}, a_{9,t+1}, R_{9,t+1} + c_{9,t+1})). \quad (21)$$

7: **end for**

Selected Copulas We used Dißmann et al. (2013)'s frequentist tree-by-tree selection method, Gruber and Czado (2015)'s Bayesian tree-by-tree method and our three fully Bayesian methods to select the dependence models. The two tree-by-tree methods and our to fully Bayesian methods from Section 4 select regular vine copulas autonomously; our Bayesian family selection method from Section 3 requires us to provide a regular vine tree structure \mathcal{V} as a partly specified model input. We specify the regular vine structure as a drawable vine whose first tree T_1 is a path from nodes 1 to 9; this is a very simple and yet intuitive structure, given that ETFs from the same asset class are neighbors in the first tree. The copulas are estimated given the transformed u -data from January through December 2013, $t = 1:252$.

Table 9 shows summary statistics of the selected models. The models selected by the tree-by-tree methods can be found in Tables 7 and 8 of Gruber and Czado (2015); Tables 24–26 of Appendix E show the marginal posterior distributions of the pair copula families of our fully Bayesian methods. The posterior distributions are based on the last 10,000 MCMC iterations out of a total of 25,000. The posterior samples from both our fully Bayesian methods that also select the regular vine tree structure do not change the tree structure after burn-in, which allows the pair-based evaluation of the copula families. The model selected by Gruber and Czado (2015)'s Bayesian tree-by-tree method has the most independence pair copulas, and Dißmann et al. (2013)'s frequentist tree-by-tree model estimate has the most pair copulas with tail-dependence as well as the most asymmetric pair copulas.

6.3 Results

Analysis Method We investigate the expected tail loss forecasts of a portfolio to assess the adequacy of the estimated model. Our study assumes that the portfolio invests in the nine ETFs as to maximize the expected risk-adjusted portfolio return assuming that the returns of the ETFs are independent,

	# 10% VaR Hits	# 15% VaR Hits
Theoretical value	19	28
Seq. frequentist selection	17	24
Seq. Bayesian selection	17	24
Bayesian Family selection	20	28
Fully Bayesian selection (I)	20	28
Fully Bayesian selection (II)	20	26

Table 10: Number of value at risk hits under different dependence models.

	10% ETL	10% OL	15% ETL	15% OL
Seq. frequentist selection	-0.61%	-0.72%	-0.53%	-0.62%
Seq. Bayesian selection	-0.60%	-0.72%	-0.52%	-0.62%
Bayesian Family selection	-0.54%	-0.67%	-0.47%	-0.57%
Fully Bayesian selection (I)	-0.56%	-0.67%	-0.49%	-0.57%
Fully Bayesian selection (II)	-0.57%	-0.67%	-0.50%	-0.60%

Table 11: Forecast expected tail loss (ETL) and average observed losses (OL) under different dependence models. The forecasts with the smallest forecast error are in bold.

$$\mathbf{w}_t = \arg \max_{\mathbf{w}=(w_1, \dots, w_9)} \frac{\mathbf{w}' \hat{\boldsymbol{\mu}}_t}{\sqrt{\mathbf{w}' \hat{\boldsymbol{\Sigma}}_t \mathbf{w}_t}} \text{ subject to } \sum_{j=1:9} w_j = 1 \text{ and } 0 \leq w_j \leq 0.25, \quad (22)$$

where $\hat{\boldsymbol{\mu}}_t = (a_{1,t}, \dots, a_{9,t})$ denotes the forecast returns of the individual ETFs, and $\hat{\boldsymbol{\Sigma}}_t = \text{diag}((R_{1,t} + c_{1,t}) \cdot r_{1,t}/(r_{1,t} - 2), \dots, (R_{9,t} + c_{9,t}) \cdot r_{9,t}/(r_{9,t} - 2))$ denotes their forecast variances. We base the portfolio investment decisions on the independence model so that the same portfolio is used when we compare the portfolio risk forecasts from the different copula models.

Analysis of Forecasts Our out-of-sample analysis of the forecasts begins in January 2014 and ends in September 2014 ($t = 253:440$) and contains 188 trading days. Table 10 compares the number of value at risk hits under different dependence models; those from our fully Bayesian models are substantially closer to the theoretical values at 10% and 15% than those from the frequentist and Bayesian tree-by-tree methods. Table 11 compares the forecast expected tail losses with the observed quantities. Here we defined the observed losses as the realized portfolio returns on the days of value at risk hits. Again, the forecasts from our fully Bayesian models are closer to the observed values than the forecasts from the two treewise selected models. The forecast errors of all models are very similar and show a mean of about -0.10%. Lastly, Figure 2 illustrates the observed portfolio returns, forecast 90% value at risk and expected tail loss, and observed value at risk hits (observed losses) from our fully Bayesian model (selection method II).

Conclusions We showed that the combination of univariate time series DLMs with a Bayesian regular vine copula is a strong couple to forecast financial asset returns.

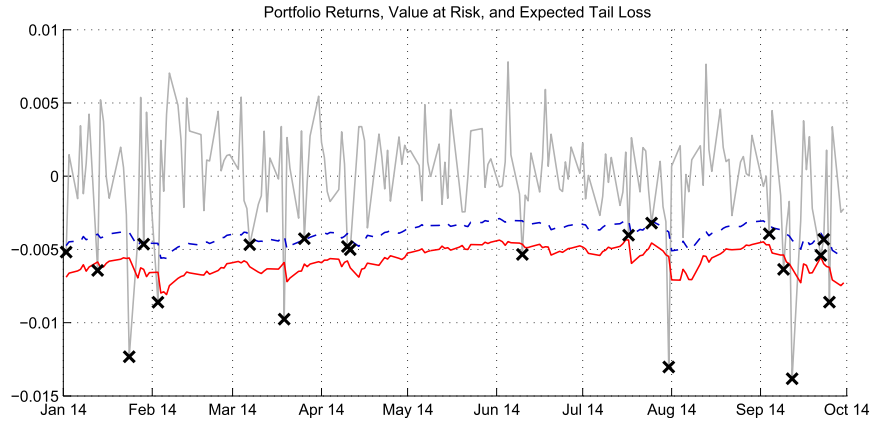


Figure 2: Observed portfolio returns (gray, solid), forecast 90% value at risk (blue, dashed) and expected tail loss (red, solid), and observed value at risk hits (black crosses). Forecasts are from our fully Bayesian model (selection method II).

The models selected by our fully Bayesian methods yielded superior forecasts than the treewise selected models. While the forecast accuracy for the expected tail losses were relatively similar, the value at risk forecasts of these fully Bayesian models were substantially more accurate than those from the treewise learned models. Our findings confirm that the quantification of model uncertainty through fully Bayesian selection contributes to more reliable risk forecasts.

7 Discussion and Outlook

Our algorithm estimates all levels of a regular vine copula simultaneously. This first-of-its-kind procedure represents a major improvement over current tree-by-tree selection procedures. Our simulation studies showed that our proposed method to select the pair copula families excels at selecting suitable pair copulas and identifying sparsity patterns through conditional independencies. Furthermore, our fully Bayesian strategies to select all levels of a regular vine copula jointly showed a major improvement of model fit compared to existing frequentist and Bayesian methods that proceed tree-by-tree. At the same time, computation time could be reduced substantially by fully Bayesian selection, which requires only one posterior simulation instead of one for each level $k = 1, \dots, d - 1$. Most important for application in practice, the sampling chains of our reversible jump MCMC samplers converge rapidly to high posterior density models, and the set of tuning parameters listed in Table 3 worked well without further tweaking for a broad range of dependence structures and datasets of different dimensions.

Eventually, the rapidly growing candidate space (as the dimension of the problem is increased) limits the scalability of any MCMC-based selection method. We view our method as one that is ideally suited for applications in low to medium dimensions

(up to around 10). Dißmann et al. (2013)'s frequentist heuristics-based procedure scales more easily to higher-dimensional problems; however, that procedure has only empirical and no proper probabilistic justification, and it does not provide model samples from a Bayesian posterior distribution over the candidate model space. Future research in scalable Bayesian methods may need to impose very restricting priors that keep the size of the candidate model space small enough for simulation-based exploration. Another avenue of future research could be an extension of Bayesian methods for factor vine copulas (such as Schamberger et al. (2017)) to higher dimensions.

Our real-data example proved the feasibility of using our proposed model selection strategies to specify a fully Bayesian multivariate time series model for forecasting risk metrics of a portfolio of financial assets. Our copula-based time series models produced highly accurate value at risk and expected tail loss forecasts at different levels, benefiting from inherent quantification of model uncertainty through Bayesian posterior analysis. We expect widespread adoption of our Bayesian selection methods, also in other contexts than financial risk modeling, given the improved out-of-sample forecasting performance and elimination of selection bias inherent in tree-by-tree methods.

Kim et al. (2013) illustrated in an example how the use of mixture pair copulas in pair copula constructions can improve the model fit. Incorporating mixture pair copulas in regular vine pair copula constructions comes with a host of challenges that open up avenues for future research regarding selection, sparsity constraints and conditions for identifiability of such models. Furthermore, the choice of alternative priors is mostly uninvestigated as of yet and can be another topic of future research.

Supplementary Material

Supplementary Appendix of Bayesian Model Selection of Regular Vine Copulas (DOI: [10.1214/17-BA1089SUPP](https://doi.org/10.1214/17-BA1089SUPP); .pdf).

References

- Acerbi, C. and Tasche, D. (2003). "Expected Shortfall: A Natural Coherent Alternative to Value at Risk." *Economic Notes*, 31: 379–388. [1128](#)
- Bedford, T. and Cooke, R. M. (2001). "Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines." *Annals of Mathematics and Artificial Intelligence*, 32: 245–268. [1112](#), [1113](#)
- Brechmann, E. C., Czado, C., and Aas, K. (2012). "Truncated regular vines in high dimensions with application to financial data." *Canadian Journal of Statistics*, 40(1): 68–85. [MR2896931](#). doi: <https://doi.org/10.1002/cjs.10141>.
- Brechmann, E. C. and Schepsmeier, U. (2013). "Modeling Dependence with C- and D-Vine Copulas: The R Package CDVine." *Journal of Statistical Software*, 52(3): 1–27. URL <http://www.jstatsoft.org/v52/i03> [MR3348832](#). doi: <https://doi.org/10.1016/j.jmva.2015.01.001>. [1114](#)

- Dißmann, J. F., Brechmann, E. C., Czado, C., and Kurowicka, D. (2013). “Selecting and estimating regular vine copulae and application to financial returns.” *Computational Statistics & Data Analysis*, 59(1): 52–69. MR3000041. doi: <https://doi.org/10.1016/j.csda.2012.08.010>. 1112, 1125, 1126, 1130, 1133
- Green, P. J. (1995). “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination.” *Biometrika*, 82: 711–732. 1112, 1114
- Gruber, L. F. and Czado, C. (2015). “Sequential Bayesian Model Selection of Regular Vine Copulas.” *Bayesian Analysis*, 10(4): 937–963. URL <http://projecteuclid.org/euclid.ba/1423083634> 1112, 1114, 1118, 1123, 1125, 1126, 1128, 1130
- Gruber, L. F. and Czado, C. (2017). “Supplementary Appendix of Bayesian Model Selection of Regular Vine Copulas.” *Bayesian Analysis*. doi: <https://doi.org/10.1214/17-BA1089SUPP>. 1112
- Hastings, W. K. (1970). “Monte Carlo sampling methods using Markov chains and their applications.” *Biometrika*, 57: 97–109. MR3363437. doi: <https://doi.org/10.1093/biomet/57.1.97>. 1114
- Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1990). “Bayesian Model Averaging: A Tutorial.” *Statistical Science*, 14(4): 382–417. 1129
- Joe, H. (1996). “Families of m -variate distributions with given margins and $m(m - 1)/2$ bivariate dependence parameters.” *Lecture Notes-Monograph Series*, 28: 120–141. 1112, 1113
- Joe, H. (2001). *Multivariate Models and Dependence Concepts*. Chapman & Hall. 1112
- Kim, D., Kim, J.-M., Liao, S.-M., and Jung, Y.-S. (2013). “Mixture of D-vine copulas for modeling dependence.” *Computational Statistics & Data Analysis*, 64: 1–19. 1133
- Kurowicka, D. and Cooke, R. M. (2006). *Uncertainty Analysis with High Dimensional Dependence Modelling*. John Wiley & Sons Ltd. MR2216540. doi: <https://doi.org/10.1002/0470863072>. 1111
- Kurowicka, D. and Joe, H. (eds.) (2010). *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co. Pte. Ltd. 1111
- McNeil, A. J., Frey, R., and Embrechts, P. (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press. 1111
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). “Equation of State Calculations by Fast Computing Machines.” *Journal of Chemical Physics*, 21: 1087–1092. 1114
- Min, A. and Czado, C. (2010). “Bayesian Inference for Multivariate Copulas Using Pair-Copula Constructions.” *Journal of Financial Econometrics*, 8(4): 511–546. 1112
- Min, A. and Czado, C. (2011). “Bayesian model selection for D-vine pair-copula constructions.” *Canadian Journal of Statistics*, 39(2): 239–258. MR2839479. doi: <https://doi.org/10.1002/cjs.10098>. 1112, 1114, 1115

- Morales-Napoles, O. (2011). “Counting vines.” In Kurowicka, D. and Joe, H. (eds.), *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co. 1117
- Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer. 1111
- Prado, R. and West, M. (2010). *Time Series: Modelling, Computation & Inference*. Chapman & Hall/CRC Press. URL <http://www.stat.duke.edu/~mw/tsbook/> 1129
- Schamberger, B., Gruber, L. F., and Czado, C. (2017). “Bayesian Inference for Latent Factor Copulas and Application to Financial Risk Forecasting.” *Econometrics*, 5(21). 1133
- Sklar, A. (1959). “Fonctions de répartition à n dimensions et leurs marges.” *Publications de l’Institut de Statistique de l’Université de Paris*, 8: 229–231. 1111
- Smith, M., Min, A., Almeida, C., and Czado, C. (2010). “Modeling Longitudinal Data Using a Pair-Copula Decomposition of Serial Dependence.” *Journal of the American Statistical Association*, 105(492): 1467–1479. MR2796564. doi: <https://doi.org/10.1198/jasa.2010.tm09572>. 1112, 1115
- Stöber, J., Joe, H., and Czado, C. (2013). “Simplified Pair Copula Constructions—Limitations and Extensions.” *Journal of Multivariate Analysis*, 119(0): 101–118.
- West, M. and Harrison, J. (1997). *Bayesian Forecasting & Dynamic Models*. Springer Verlag, 2nd edition. URL <http://www.stat.duke.edu/~mw/book.html> 1129

Acknowledgments

Research presented here was initiated while the first author was a researcher at the Center for Mathematics at Technische Universität München.

The numerical computations were performed on a Linux cluster supported by DFG grant INST 95/919-1 FUGG.

The authors are grateful to the Editor, Associate Editor and three anonymous referees for their detailed comments on the original version of this paper. Their suggestions and insights into the area were most relevant in revision and defining the final version.