Fakultät für Mathematik

Lehrstuhl für Mathematische Statistik

Bayesian Modeling of General Multivariate Problems and High-Dimensional Time Series

Lutz Fabian Gruber

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigten Dissertation.

Vorsitzende: Univ.-Prof. Donna Ankerst, Ph.D.
Prüfer der Dissertation:
1. Univ.-Prof. Claudia Czado, Ph.D.
2. Univ.-Prof. Dr. Sylvia Frühwirth-Schnatter
   Wirtschaftsuniversität Wien, Österreich
3. Prof. Mike West, Ph.D.
   Duke University, Durham, USA
   (schriftliche Beurteilung)

Die Dissertation wurde am 01.04.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 22.06.2015 angenommen.

Abstract

English

The field of multivariate statistics analyzes the joint behavior of several variables; this goes beyond an isolated analysis of every variable on its own. A key focus of investigation are dependence effects, for example: does stock A tend to gain or lose as stock B gains?

This dissertation presents Bayesian strategies to model statistical dependence in medium and high dimensions. The first part develops novel and powerful model selection strategies for regular vine copulas. These are very computationally intensive, however, so that application of these methods is restricted to problems of medium dimensionality (up to 10 or 20 variables). The second part develops a method to enable on-line analysis of high-dimensional time series (up to hundreds or thousands of variables) facilitated by sparse modeling and massively parallel computing on GPUs.

Extensive real-data examples and case studies illustrate application of our models and methods, and verify their suitability for application in practice. These case studies calculate portfolio risk and investment decisions using our novel models and historical price data on different stocks.

Deutsch

In der multivariaten Statistik werden das gemeinsame Verhalten mehrerer Variablen untersucht; dies ist im Unterschied zu einer isolierten Betrachtung jeder Variable. Insbesondere werden hier Abhängigkeitseffekte betrachtet, zum Beispiel: steigt oder fällt der Kurs von Aktie A, wenn der Kurs von Aktie B steigt?

Diese Dissertation präsentiert Bayesianische Lösungsansätze zur Modellierung statistischer Abhängigkeiten in mittleren und hohen Dimensionen. Der erste Teil entwickelt neue und leistungsfähigere Verfahren zur Modellwahl von Regular Vine Copulas, welche jedoch so rechenaufwändig sind, dass die Anwendung auf Probleme von mittlerer Dimension (bis ca. 10 oder 20 Variablen) beschränkt ist. Der zweite Teil entwickelt ein Verfahren zur Echtzeit Analyse hochdimensionaler dynamischer Zeitreihen (bis zu hunderten oder tausenden von Variablen). Dies wird durch statistische Methoden zur Reduktion der Modell Komplexität und den Einsatz massiv parallelisierter Rechnung auf Grafikkarten ermöglicht.

An die methodische Herleitung der entwickelten Modelle und Methoden anschließend werden diese in ausführlichen Anwendungsbeispielen illustriert und auf ihre Praxis Tauglichkeit untersucht. In diesen Anwendungen berechnen wir Portfolio Risiken und Investment Entscheidungen auf Grundlage unserer neuen Modelle und Verfahren, und unter Verwendung historischer Kursdaten von verschiedenen Aktien.

Contents

Acknowledgements

Firstly, I would like to thank my PhD advisor, Claudia Czado, who took me on as a PhD student after advising on my master's thesis, for her unparalleled enthusiasm about vine copulas, helpful discussions throughout the years, and generous funding for conference travels. Furthermore, I am grateful that Claudia let me follow my research interests, and my curiosity to go abroad.

Then I would like to thank Mike West for inviting me to the Department of Statistical Science at Duke University and offering his mentorship. Mike's mentoring during, and after, my visit has been exceptional, and I benefited greatly from our regular discussions; I always came out of these meetings bursting with new questions to investigate and ideas to explore.

My research visit was not only productive, but also a lot of fun. Faculty, staff and students at the Department of Statistical Science at Duke University made me feel welcome throughout my visit. In particular, I would like to thank my office mate Bruno Santos, as well as Maria Terres, Mary Beth Broadbent, Jacopo Soriano, Ted Pudlik, and Evan Calabrese for their friendship and numerous fun activities. Along with my friends from Duke, I thank my Munich-based friends Jakob Ameres, Clotaire Geffray, Julian Geiger, and Sebastian Mönch for fun times in swim practice and outside of the pool.

I would like to thank my parents, Arno and Irmengard Gruber, and my brother, Luis, for their love and encouragement. Finally, I thank Erica Stuber; my accomplishments would not have been possible without her love and support, nor would the last few years have been as happy.

Part I

Bayesian Model Selection of Regular Vine Copulas

Chapter 1

A Review of Dependence Modeling with Regular Vine Copulas and Current Techniques for Inference and Model Selection

1.1 Introduction

Copula modeling provides great flexibility in defining multivariate distributions that describe available data well, because the marginal models $F_1, \ldots, F_d$ are not required to be from the same family or to share any characteristics, and the copula, too, can be selected to best describe the data. This contrasts with most multivariate models, which do not allow much customization of either marginal or dependence characteristics. Specifically, Sklar [1959] showed that any multivariate distribution $\mathbf{X} = (X_1, \ldots, X_d) \sim F_{1:d}$ can be separated into a copula distribution C, which only models the joint multivariate dependence structure, and the marginal distributions $X_1 \sim F_1, \ldots, X_d \sim F_d$,

$$F_{1:d}(\mathbf{x}) = C\left(F_1(x_1), \ldots, F_d(x_d)\right), \mathbf{x} \in \mathbb{R}^d.$$

Furthermore, the combination of any copula C with any set of marginal distributions $F_1, \ldots, F_d$ results in a valid multivariate distribution. Copulas are discussed extensively in numerous textbooks [e.g. Nelsen, 2006, McNeil et al., 2005, Kurowicka and Cooke, 2006, Kurowicka and Joe, 2010, Joe, 2001, 2014, Mai and Scherer, 2012], with additional textbooks specifically on use of copulas in finance [e.g. Cherubini et al., 2004, 2011, Mai and Scherer, 2014].

Even though Sklar [1959]'s work dates back to 1959, most research on copulas is much more recent. Applications of copulas cover the entire spectrum of multivariate modeling, and examples include general multivariate time series [Chen and Fan, 2006b,a, Heinen and Rengifo, 2007, Lee and Long, 2009, Patton, 2009, Ausin and Lopes, 2010], hydrological engineering [Zhang and Singh, 2006, Grimaldi and Serinaldi, 2006, Chen et al., 2013, Madadgar and Moradkhani, 2013] and weather [Zhang and Singh, 2007, Renard and Lang, 2007, Cong and Brady, 2012, Erhardt et al., 2015a,b], energy price and demand [Gregoire et al., 2008, Smith et al., 2010], financial data [van den Goorbergh et al., 2005, Jondeau and Rockinger, 2006, Hu, 2006, Rodriguez, 2007, Bartram et al., 2007, Zhang and Guegan, 2008, Aas et al., 2009, Min and Czado, 2010, 2011, 2014, Patton, 2006, Hofert and Scherer, 2011, Brechmann et al., 2012, Dißmann et al., 2013, Boubaker and Sghaier, 2013, Gruber and Czado, 2015b,a], insurance [Frees and Wang, 2006, Sun et al., 2008, Yang et al., 2011, Krämer et al., 2013], marketing [Danaher and Smith, 2011], and many others.

Pair copula constructions are motivated by the observation that "one cannot just write down a parametric family of functions with the right boundary properties and expect them to satisfy the rectangle condition of a multivariate cdf" [Joe, 2001, Chapter 4, p. 83]. Instead, one combines several bivariate copulas, which are studied extensively and of which many exist, in a well-structured way that guarantees to result in a valid multivariate copula distribution [Joe, 1996, Bedford and Cooke, 2001]. This review covers basic concepts of bivariate dependence (Section 1.2) and popular bivariate copulas (Section 1.3), then presents a comprehensive and concise model specification of regular vine pair copula constructions (Section 1.4), and reviews current inference and model selection techniques (Section 1.5). This review has a stronger focus on bivariate dependence modeling methodology than Czado et al. [2013]'s earlier review, and also includes Gruber and Czado [2015b,a]'s Bayesian selection methods.

1.2 Measures of Bivariate Dependence

This section introduces several measures of bivariate statistical dependence that are commonly used in applications of pair copula constructions. We focus on measures of strength of association and tail-dependence; these bivariate dependence characteristics have very intuitive interpretations and are highly relevant in many applications. The measures discussed here are a key selection of the many discussed in [Joe, 2001, Chapter 2], and the equations below follow said reference.

Assume for the remainder of this section that a pair of random variable (X, Y) follows some general bivariate distribution F with marginal distributions $X \sim F_1$ and $Y \sim F_2$. Furthermore, define $U = F_1(X)$, $V = F_2(Y)$, and denote the associated copula of F by C, so $(U, V) \sim C$.

1.2.1 Measures of Association

Measures of association quantify some aspects of strength of dependence. While Pearson's correlation coefficient ρ is probably the most well-known and most widely used one, rank correlation coefficients such as Spearman's ρ and Kendall's τ have properties that make them more suitable to quantify association in general analyses.

Pearson's ρ. Pearson's correlation coefficient $\rho \in [-1, 1]$ is likely the most well-known measure of dependence. Defined as $\rho = \text{Cov}(X, Y)/\sqrt{\text{Var}(X)\text{Var}(Y)}$, Pearson's ρ measures linear dependence between X and Y. Pearson's correlation can be interpreted as the slope of the regression line of X and Y. Pearson's ρ between copula variables U and V is $\rho = 12 \int\int uv dC(u, v) - 3$.

It is easily demonstrated that the emphasis of *linear* dependence is important: suppose $U \sim U(0, 1)$ and $V = 2|U - 0.5|$. It is trivial to show that U and V are not independent; indeed, conditional on U, V can only take one value! However, Pearson's ρ of U and V is zero, which also conveniently shows that uncorrelatedness does not guarantee independence. These shortcomings motivate investigation of better dependence measures.

Spearman's ρ. Spearman's $\rho \in [-1, 1]$ between X and Y is a measure of monotone dependence that is defined as Pearson's ρ between the ranked variables $F_1(X)$ and $F_2(Y)$; it can be calculated as $\rho = 12 \int_x \int_y F_1(x)F_2(y)dF(x, y) - 3$. Spearman's ρ is invariant under strictly monotone transformations of X and Y, and it takes the limits if and only if one of U and V is an increasing or decreasing transform of the other. These properties set it apart from Pearson's ρ, for which neither holds. Spearman's ρ between two copula variables U and V is $\rho = 12 \int\int uv dC(u, v) - 3$.

Kendall's τ. Kendall's $\tau \in [-1, 1]$ is also a measure of monotone dependence and it is defined as the difference of the probability of two random concordant pairs and discordant pairs, $\tau = P((X - X')(Y - Y') > 0) - P((X - X')(Y - Y') < 0)$, where $(X, Y) \sim F$ and $(X', Y') \sim F$ are independent random pairs. Kendall's τ also satisfies invariance under strictly monotone transformations of X and Y, and it also takes the limits if and only if one of X and Y is an increasing or decreasing transform of the other. Kendall's τ between two copula variables U and V is $\tau = 4 \int\int C(u, v)dC(u, v) - 1$.

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	$\rho \in (-1, 1)$ $\nu > 1$	$\tau = \frac{2}{\pi}\arcsin(\rho)$	$\lambda^L = \lambda^U = 2T_{\nu+1}\left(-\sqrt{(\nu+1)\frac{1-\rho}{1+\rho}}\right)$
Clayton	C	$\delta \geq 0$	$\tau = \frac{\delta}{2+\delta}$	$\lambda^L = 2^{-1/\delta}$, $\lambda^U = 0$
Double Clayton I	C or DC I	$\delta \in \mathbb{R}$	$\tau = \frac{\delta}{2+\lvert\delta\rvert}$	$\lambda^L = 2^{-1/\lvert\delta\rvert}$, $\lambda^U = 0$
Double Clayton II	C or DC II	$\delta \in \mathbb{R}$	$\tau = -\frac{\delta}{2+\lvert\delta\rvert}$	$\lambda^L = 0$, $\lambda^U = 2^{-1/\lvert\delta\rvert}$
Gumbel	G	$\delta \geq 1$	$\tau = \frac{\delta-1}{\delta}$	$\lambda^L = 0$, $\lambda^U = 2 - 2^{1/\delta}$
Double Gumbel I	G or DG I	$\delta \in \mathbb{R}$	$\tau = \frac{\delta}{1+\lvert\delta\rvert}$	$\lambda^L = 0$, $\lambda^U = 2 - 2^{1/(1+\lvert\delta\rvert)}$
Double Gumbel II	G or DG II	$\delta \in \mathbb{R}$	$\tau = -\frac{\delta}{1+\lvert\delta\rvert}$	$\lambda^L = 2 - 2^{1/(1+\lvert\delta\rvert)}$, $\lambda^U = 0$

Table 1.1: Candidate pair copula families for use in the pair copula construction, and their parameter transformations.

1.2.2 Measures of Tail-Dependence

Measures of bivariate tail-dependence quantify the strength of dependence in the upper or lower-quadrant tails of a bivariate distribution.

Upper tail-dependence coefficient λ^U. The upper tail-dependence coefficient is the conditional tail probability $\lambda^U = \lim_{u\to 1} P(U > u \mid V > u) = \lim_{u \to 1} P(V > u \mid U > u)$. It can be rewritten as $\lambda^U = \lim_{u\to 1} \overline{C}(u,u)/(1-u)$, where $\overline{C}(u,v) = 1 - u - v + C(u,v)$ is the survival function of a copula C. If $\lambda^U > 0$, copula C is said to have upper tail-dependence; if $\lambda^U = 0$, C is said to have no upper tail-dependence.

Lower tail-dependence coefficient λ^L. The lower tail-dependence coefficient is the conditional tail probability $\lambda^L = \lim_{u\to 0} P(U < u \mid V < u) = \lim_{u\to 0} P(V < u \mid U < u)$, and it can be rewritten as $\lambda^L - \lim_{u\to 0} C(u,u)/u$. If $\lambda^L > 0$, copula C is said to have lower tail-dependence; if $\lambda^L = 0$, C is said to have no lower tail-dependence.

1.3 Bivariate Copulas

Bivariate copulas are extensively studied in existing literature, see, for example [Joe, 2001, 2014, Nelsen, 2006]. Among the large variety of bivariate copulas, we select a few that cover the most important dependence features: independence, negative dependence, positive dependence, tail-independence, tail-dependence, symmetry and asymmetry.

The transformations between Kendall's τ, the tail-dependence coefficients λ^L and λ^U, and the copula parameters in Table 1.1 are taken from Brechmann and Schepsmeier [2013]; the copula density functions below are taken from Gruber [2011]. Table 1.2 shows the pairs plots with normalized margins of several different copulas at Kendall's $\tau = 0.4$ and $\tau = 0.75$.

Independence copula. The independence copula describes two independent random variables U and V. Its copula density function $c(u,v) = 1$ is constant for $u \in [0,1]$, $v \in [0,1]$. The independence copula has no upper or lower tail-dependence.

Gaussian copula. The Gaussian copula is symmetric, has no tail-dependence, and its density function is

$$c(u, v; \rho) = \frac{1}{\sqrt{1-\rho^2}} \exp\left(-\frac{\rho^2\left(x^2+y^2\right) - 2\rho xy}{2(1-\rho^2)}\right), \tag{1.1}$$

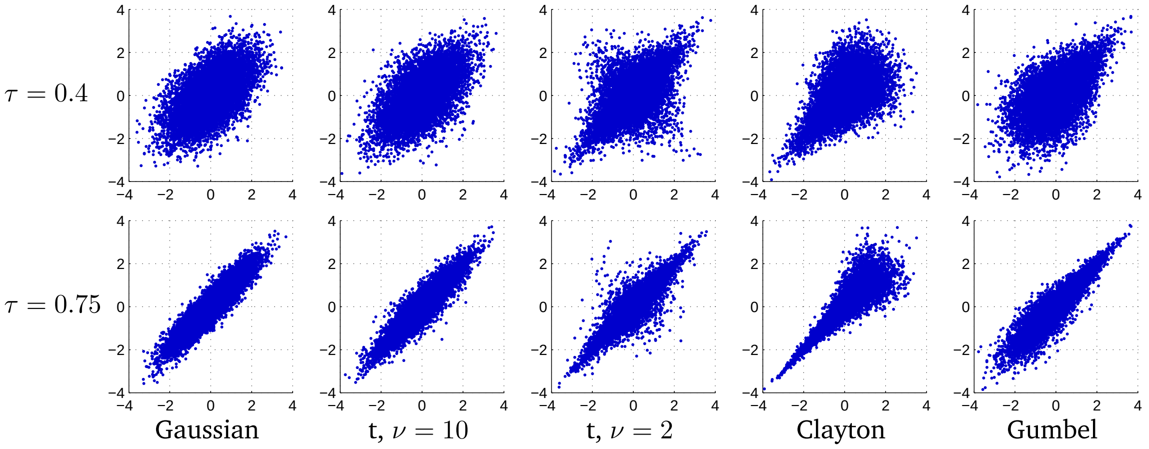


Table 1.2: Pairs plots of several different bivariate copulas at Kendall's $\tau = 0.4$ and $\tau = 0.75$.

where $x = \Phi^{-1}(u)$ and $y = \Phi^{-1}(v)$, and Φ^{-1} denotes the inverse of the cumulative distribution function of the univariate standard normal distribution. The Gaussian copula's correlation parameter $\rho \in [-1, 1]$ defines the strength and type of dependence: for $\rho \in \{-1, 1\}$, the Gaussian copula exhibits perfect monotonic dependence; for $\rho \in (-1, 0)$, it shows negative dependence; for $\rho = 0$, it simplifies to the independence copula; and for $\rho \in (0, 1)$, it shows positive dependence.

T copula. The T copula is symmetric, has upper and lower tail-dependence, and its density function is

$$c(u, v; \rho, \nu) = \frac{\left(1 + \frac{x^2+y^2-2\rho xy}{\nu(1-\rho^2)}\right)^{-\frac{\nu+2}{2}}}{2\pi t_\nu(x) t_\nu(y) \sqrt{1-\rho^2}}, \tag{1.2}$$

where $x = T_\nu^{-1}(u)$ and $y = T_\nu^{-1}(v)$, and T_ν^{-1} and t_ν denote the inverse cumulative distribution function and probability density function, respectively, of the T distribution with ν degrees of freedom. The T copula has two parameters: correlation $\rho \in [-1, 1]$ and degrees of freedom $\nu > 1$.

Clayton copula. The Clayton copula is asymmetric, has lower tail-dependence, but no upper tail-dependence, and its density function is

$$c(u, v; \delta) = (1 + \delta)(uv)^{-1-\delta}(u^{-\delta} + v^{-\delta} - 1)^{-\frac{1}{\delta}-2}. \tag{1.3}$$

The Clayton copula has one parameter, $\delta \geq 0$, and can only model non-negative monotone dependence. For $\delta = 0$, the Clayton copula simplifies to the independence copula; for $\delta > 0$, the Clayton copula shows positive dependence; and in the limit $\delta \to \infty$, the Clayton copula shows perfect monotone dependence.

Gumbel copula. The Gumbel copula is asymmetric, has no lower tail-dependence, but has upper tail-dependence, and its density function is

$$c(u, v; \delta) = \exp\left(x + y - (x^\delta + y^\delta)^{\frac{1}{\delta}}\right)(x^\delta + y^\delta)^{-2+\frac{2}{\delta}}(xy)^{\delta-1}\left(1 + (\delta - 1)(x^\delta + y^\delta)^{-\frac{1}{\delta}}\right), \tag{1.4}$$

where $x = -\log(u)$ and $y = -\log(v)$. The Gumbel copula has one parameter, $\delta \geq 1$, and can only model non-negative monotone dependence. For $\delta = 1$, the Gumbel copula simplifies to the independence copula; for $\delta > 0$, the Gumbel copula shows positive dependence; and in the limit $\delta \to \infty$, the Gumbel copula shows perfect monotone dependence.

Double Clayton and Gumbel copulas. The Double Clayton and Double Gumbel copulas combine different rotations of the regular Clayton and Gumbel copulas to also allow the modeling of negative dependence:

$$c_{\text{Double Clayton I}}(u_1, u_2; \delta) = \begin{cases} c_{\text{Clayton}}(u_1, u_2; \delta) \text{ if } \delta \geq 0 \\ c_{\text{Clayton}}(1 - u_1, u_2; -\delta) \text{ if } \delta < 0 \end{cases} \tag{1.5}$$

$$c_{\text{Double Clayton II}}(u_1, u_2; \delta) = c_{\text{Double Clayton I}}(u_1, 1 - u_2; \delta) \tag{1.6}$$

$$c_{\text{Double Gumbel I}}(u_1, u_2; \delta) = \begin{cases} c_{\text{Gumbel}}(u_1, u_2; \delta + 1) \text{ if } \delta \geq 0 \\ c_{\text{Gumbel}}(1 - u_1, u_2; -\delta + 1) \text{ if } \delta < 0 \end{cases} \tag{1.7}$$

$$c_{\text{Double Gumbel II}}(u_1, u_2; \delta) = c_{\text{Double Gumbel I}}(u_1, 1 - u_2; \delta). \tag{1.8}$$

In simplified notation, we may refer to the Double Clayton or Double Gumbel copula just as the Clayton or Gumbel copula.

1.4 Regular Vine Copulas

1.4.1 Model Structure

Joe [1996] presented the first construction of a multivariate copula using (conditional) bivariate copulas, and Bedford and Cooke [2001] developed a more general construction method of multivariate densities and introduced regular vines to organize different pair copula constructions. Definitions 1.1–1.4 and Theorem 1.1 follow Bedford and Cooke [2001].

Definition 1.1 (Regular Vine Tree Sequence). *A set of linked trees $\mathcal{V} = (T_1, T_2, ..., T_{d-1})$ is a regular vine on d elements if*

1. *T_1 is a tree with nodes $N_1 = \{1, ..., d\}$ and a set of edges denoted by E_1.*
2. *For $k = 2, ..., d-1$, T_k is a tree with nodes $N_k = E_{k-1}$ and edge set E_k.*
3. *For $k = 2, ..., d-1$, if $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$ are two nodes in N_k connected by an edge, then exactly one of the a_i equals one of the b_i (Proximity condition).*

Definition 1.2 (Conditioning Set, Conditioned Set). *Let A_e be the complete union of an edge $e = \{a, b\} \in E_k$ in tree T_k of a regular vine $\mathcal{V}$,*

$$A_e = \{v \in N_1 \mid \exists e_i \in E_i, i = 1, \ldots, k-1, \text{ such that } v \in e_1 \in \cdots \in e_{k-1} \in e\}.$$

The conditioning set associated with edge $e = \{a, b\}$ is $D(e) := A_a \cap A_b$ and the conditioned sets associated with edge e are $i(e) := A_a \setminus D(e)$ and $j(e) := A_b \setminus D(e)$. Here, $A \setminus B := A \cap B^c$ and B^c is the complement of B.

Bedford and Cooke [2001] showed that the conditioned sets are singletons and proved that a multivariate copula density is obtained by using each edge of a regular vine as a pair copula factor in the pair copula construction. The pair copula associated with an edge e, denoted $c_e(\cdot, \cdot; \boldsymbol{\theta}_{e;\mathcal{B}_e})$, describes the distribution of the transformed variables $u_{i(e)|D(e)}$ and $u_{j(e)|D(e)}$ (see Definition 1.4). We will use the notation e and $i(e), j(e); D(e)$ interchangeably (Example: Figure 1.1).

Definition 1.3 (Regular Vine Copula).

1. *Let $\mathcal{V} = (T_k = (N_k, E_k) \mid k = 1, \ldots, d-1)$ be a regular vine on d elements. $\mathcal{B}_e$ denotes the family and $\boldsymbol{\theta}_{e;\mathcal{B}_e}$ denotes the parameter vector of the pair copula $c_e(\cdot, \cdot; \boldsymbol{\theta}_{e;\mathcal{B}_e}); e \in T_k \in \mathcal{V}$, which is associated with an edge $e \in E_k$ of a regular vine tree $T_k \in \mathcal{V}$.*
2. *We write $\mathcal{B}_k := \{\mathcal{B}_e \mid e \in E_k\}$ for the pair copula families of level k and $\boldsymbol{\theta}_k := \{\boldsymbol{\theta}_{e;\mathcal{B}_e} \mid e \in E_k\}$ for their parameters.*
3. *The regular vine pair copula construction up to level $K \leq d-1$, denoted by $\mathcal{C}_K = (T_{1:K}, \mathcal{B}_{1:K}, \boldsymbol{\theta}_{1:K})$, consists of the regular vine trees $T_{1:K} = (T_1, \ldots, T_K)$, pair copula families $\mathcal{B}_{1:K} = (\mathcal{B}_1, \ldots, \mathcal{B}_K)$ and parameters $\boldsymbol{\theta}_{1:K} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$.*
4. *If $K = d-1$, we write $\mathcal{C}$ for $\mathcal{C}_{d-1}$, $\mathcal{B}_{\mathcal{V}}$ for $\mathcal{B}_{1:(d-1)}$ and $\boldsymbol{\theta}_{\mathcal{V}}$ for $\boldsymbol{\theta}_{1:(d-1)}$.*

Definition 1.4 (Density Function of a Regular Vine Copula). *The density function of the level K-regular vine copula $\mathcal{C}_K$, denoted by $c(\cdot; \mathcal{C}_K) : [0,1]^d \to [0, \infty)$, is*

$$c(\mathbf{u}; \mathcal{C}_K) = \prod_{k=1:K} \prod_{e \in E_k} c_{e;\mathcal{B}_e}\left(u_{i(e)|D(e)}, u_{j(e)|D(e)}; \boldsymbol{\theta}_{e;\mathcal{B}_e}\right). \tag{1.9}$$

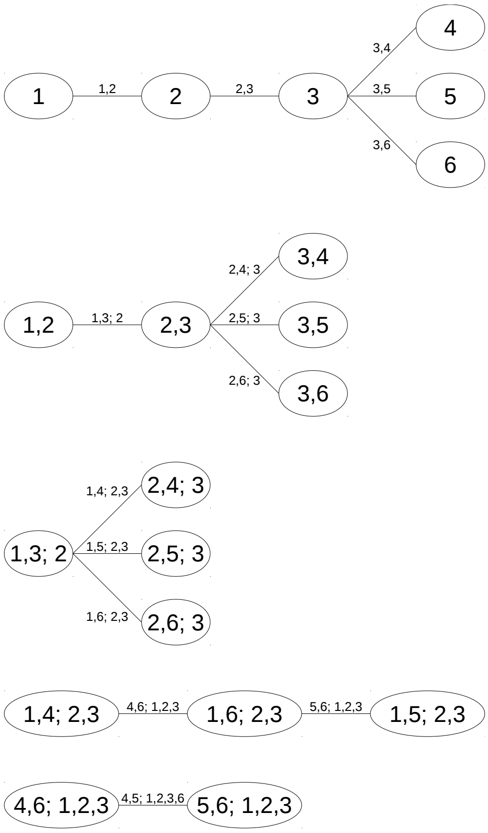


Figure 1.1: Six-dimensional regular vine copula.

Here the density argument $u_{i(e)|D(e)}$ is the value of a marginal copula distribution function,

$$u_{i(e)|D(e)} := C_{i(e)|D(e)}(u_{i(e)} \mid \mathbf{u}_{D(e)}; \mathcal{C}_{|D(e)|}) = \int_0^{u_{i(e)}} c_{i(e)|D(e)}(t \mid \mathbf{u}_{D(e)}; \mathcal{C}_{|D(e)|})\, dt, \tag{1.10}$$

where the conditional density function $c_{i(e)|D(e)}(\cdot \mid \cdot; \mathcal{C}_{|D(e)|})$ is obtained by integration and partial differentiation of the level $|D(e)|$-copula density function $c(\cdot; \mathcal{C}_{|D(e)|})$,

$$\begin{aligned} c_{i(e)|D(e)}(u \mid \mathbf{u}_{D(e)}; \mathcal{C}_{|D(e)|}) &= \frac{c_{i(e)\cup D(e)}(\mathbf{u}_{i(e)\cup D(e)}; \mathcal{C}_{|D(e)|})}{c_{D(e)}(\mathbf{u}_{D(e)}; \mathcal{C}_{|D(e)|})} &(1.11)\\ &= \frac{\frac{\partial}{\partial \mathbf{u}_{i(e)\cup D(e)}} \int_0^{g_{i(e)\cup D(e)}(\mathbf{u}_{i(e)\cup D(e)})} c(\mathbf{t}; \mathcal{C}_{|D(e)|})\, d\mathbf{t}}{\frac{\partial}{\partial \mathbf{u}_{D(e)}} \int_0^{g_{D(e)}(\mathbf{u}_{D(e)})} c(\mathbf{t}; \mathcal{C}_{|D(e)|})\, d\mathbf{t}}, &(1.12) \end{aligned}$$

where $g_J(\mathbf{u}_J) := (u_j$ if $j \in \{J\}; 1$ otherwise$) \in \mathbb{R}^d$ for some set $J \subseteq \{1, \ldots, d\}$. The same structure applies to $u_{j(e)|D(e)}$.

If $K < d-1$, the level K-regular vine copula $\mathcal{C}_K$ is said to be truncated at level K [Brechmann et al., 2012]. In the following, we will use of the terms *regular vine copula* and *regular vine pair copula construction* interchangeably.

Theorem 1.1 (Regular Vine Copula Density is a Probability Density). *For any $K \leq n-1$ (see Definition 1.3), the density function of the regular vine pair copula construction $\mathcal{C}_K$, $c(\cdot; \mathcal{C}_K)$ (see (1.9)), is an n-variate probability density function on $[0,1]^n$.*

Theorem 1.2 (Number of Regular Vines). *There exist*

$$\frac{d!}{2} \times 2^{\binom{d-2}{2}} \tag{1.13}$$

different regular vines on d dimensions.

Theorem 1.2 is due to Morales-Napoles [2011]; Table 1.3 illustrates the super-exponential growth of the number of regular vines for dimensions $d = 2, \ldots, 10$. In general, regular vine copulas that differ in the

Dimension d	# Regular Vines
2	1
3	3
4	24
5	480
6	23,040
7	2,580,480
8	660,602,880
9	3.8051e+11
10	4.8705e+14

Table 1.3: Number of regular vines on dimension d.

tree structure or in at least one pair copula family have different copula densities. Notable exceptions from this rule include the multivariate Gaussian, Student's t or Clayton copula, whose densities can be represented by different pair copula constructions [Stöber et al., 2013].

1.4.2 Likelihoods

Definition 1.3 specifies regular vine copulas tree-by-tree, which motivates the following break down into the likelihoods of individual trees and edges of a regular vine copula. All likelihoods are understood given data $\mathbf{U} = (\mathbf{u}^1, \ldots, \mathbf{u}^T) \in [0,1]^{T\times d}$.

The likelihood of level k depends only on the specification of the regular vine copula up to level k. The likelihoods of edge $e \in E_k$ and level k follow as

$$L(\mathcal{B}_e, \boldsymbol{\theta}_e \mid \mathbf{U}) := \prod_{t=1:T} c_{\mathcal{B}_e}(u^t_{i(e)|D(e)}, u^t_{j(e)|D(e)}; \boldsymbol{\theta}_e) \text{ and} \tag{1.14}$$

$$L(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) := \prod_{e\in E_k} L(\mathcal{B}_e, \boldsymbol{\theta}_e \mid \mathbf{U}), \tag{1.15}$$

where the $u^t_{i(e)|D(e)}$ and $u^t_{j(e)|D(e)}$ are obtained by transforming the t-th observation $\mathbf{u}^t$ as in (1.10). The likelihood of a regular vine copula $\mathcal{C}$ is then calculated tree-by-tree

$$L(\mathcal{C} \mid \mathbf{U}) = L(\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \boldsymbol{\theta}_{\mathcal{V}} \mid \mathbf{U}) = \prod_{T_k\in\mathcal{V}} L(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}). \tag{1.16}$$

1.4.3 Notation

Regular vines. For each level k, $\mathbf{STP}_k$ denotes the set of all spanning trees with node set N_k that satisfy the proximity condition for tree $T_k = (N_k, E_k)$ (see Definition 1.1). Furthermore, $\mathbf{B}$ shall denote the set of candidate pair copula families; in the remainder $\mathbf{B}$ consists of the pair copula families listed in Table 1.1.

Pair copulas. We will refer to the pair copula families by their identifiers as introduced in Table 1.1, and define the set of candidate pair copula families $\mathbf{B} = \{$I, N, T, DC I, DC II, DG I, DG II$\}$. We will also refer to the individual transformations by their degrees of rotation: rotating the first argument of the copula density function yields the 90° rotation, $c_{90}(u_1, u_2) = c_0(1-u_1, u_2)$; the 180° rotation rotates both arguments, $c_{180}(u, v) = c_0(1-u_1, 1-u_2)$; and the 270° rotation rotates only the second argument, $c_{270}(u, v) = c_0(u_1, 1-u_2)$. In some cases, we will amend the notations introduced in Table 1.1 by the degree of rotation and value of Kendall's τ; for example, C180(0.7) will indicate the 180° rotation of the Clayton copula with parameter δ such that Kendall's $\tau = 0.7$.

Reference	Dimension	Use of Copula
Aas et al. [2009], Min and Czado [2010]	4	Contemporaneous dependence of stock and bond indices
Min and Czado [2010, 2011]	5	Contemporaneous dependence of Euro swap rates
Min and Czado [2011]	6	Dependence of geometric dimensions of counterfeit bank notes
Czado et al. [2013]	7	Dependence of concentration of chemicals in water
Gruber and Czado [2015b,a]	9	Contemporaneous dependence of exchange-traded funds
Dißmann et al. [2013]	16	Contemporaneous dependence of stock, bond and commodity indices
Brechmann et al. [2012]	20	Contemporaneous dependence of foreign exchange rates and several market indices
Smith et al. [2010]	24	Serial time dependence of intraday electricity loads

Table 1.4: Example of data sets that were analyzed using regular vine copula models.

Pair copula parameters. In the remainder, 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. This uniform parameterization makes it easier to compare different copulas' parameters and propose good parameter values when the pair copula families change. The notation for one-parameter copulas is $\boldsymbol{\theta}_e \equiv \tau_e$, and $\boldsymbol{\theta}_e = (\tau_e, \log \nu_e)$ for the t copula. The parameters of all pair copulas of the regular vine copula are collectively referred to by $\boldsymbol{\theta}_{\mathcal{V}}$.

1.4.4 Applications

Vine copulas are used to benefit a wide field of multivariate modeling. Vine copulas can be utilized in a diverse set of applications, as illustrated by Table 1.4.

1.5 Inference and Model Selection of Vine Copulas

Current inference and model selection methods for regular vine copulas extend earlier methods on C-vine and D-vine copulas, both of which are subclasses of regular vine copulas. This review aims to provide more context to recent developments on model selection of regular vine copulas by highlighting the similarities and differences between existing methods. Furthermore, this review can be seen as an update of Czado et al. [2013]'s review, which did not yet include Gruber and Czado [2015a]'s new fully Bayesian selection strategy (see Chapter 3 of this dissertation).

Inference methods for regular vine copulas can be organized into parameter estimation and model selection, tree-by-tree sequential and simultaneous, and frequentist and Bayesian methods. We discuss tree-by-tree and simultaneous methods separately to show that all existing methods are just variants of these two different general approaches.

1.5.1 Sequential Tree-by-Tree Inference

All sequential methods select a vine copula and estimate its parameters one tree after the after. In the first step, select tree T_1 and the unconditional pair copulas of level $k = 1$, $\mathcal{B}_1$ (model selection only), and estimate the parameters $\boldsymbol{\theta}_1$. Subsequent steps, $k = 2, \ldots, d-1$, select tree T_k and the pair copulas $\mathcal{B}_k$ (model selection only), and estimate the parameters $\boldsymbol{\theta}_k$. These steps are conditional on the selected levels up to $k-1$. Algorithm 1.1 shows pseudo code for the general tree-by-tree inference approach.

Pseudo Code
Algorithm 1.1 (Tree-by-Tree Selection and Estimation).

```
1: for level $k = 1, \ldots, d-1$ do
2:     For structure selection, select the regular vine tree of level $k$, $\widehat{T}_k$.
3:     For copula selection, select the pair copula families of level $k$, $\widehat{\mathcal{B}}_k$.
4:     Estimate the parameters of level $k$, $\widehat{\boldsymbol{\theta}}_k$.
5: end for
6: return the sequential estimate $\widehat{\mathcal{C}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{B}}_{\mathcal{V}}, \widehat{\boldsymbol{\theta}}_{\mathcal{V}})$, where $\widehat{\mathcal{V}} = (\widehat{T}_1, \ldots, \widehat{T}_{d-1})$, $\widehat{\mathcal{B}}_{\mathcal{V}} = (\widehat{\mathcal{B}}_1, \ldots, \widehat{\mathcal{B}}_{d-1})$, and
   $\widehat{\boldsymbol{\theta}}_{\mathcal{V}} = (\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_{d-1})$.
```

Frequentist parameter estimation. Sequential tree-by-tree inference for vine copulas was first proposed in [Aas et al., 2009]. While applicable to the general class of regular vine copulas, the original work provides a detailed description of the tree-wise likelihoods for C-vine and D-vine copulas only, and Dißmann et al. [2013] presented likelihood-based inference for the entire class of regular vine copulas. Both papers suggest maximum likelihood estimation of the parameters of each copula $e \in E_k$ individually (see (1.14) for the likelihood function), and suggest that sequential estimation can be used on its own, or as a way to find starting values for full maximum likelihood estimation.

Frequentist model selection. Sequential tree-by-tree inference can be used for model selection similar to parameter estimation. Dißmann et al. [2013] and Brechmann et al. [2012] propose such a scheme for model selection of the regular vine $\mathcal{V}$ and pair copula families $\mathcal{B}_{\mathcal{V}}$. For each level $k = 1, \ldots, d-1$, Dißmann et al. [2013]'s method selects T_k as the maximum spanning tree that satisfies the proximity condition for that level using the absolute value of each pair's Kendall's τ as the edge weights for optimization. Once tree $T_k = (N_k, E_k)$ is selected, the pair copula families $\mathcal{B}_k$ are selected individually for each edge $e \in E_k$, using AIC to select the most suitable from a set of candidate families $\mathbf{B}$. Czado et al. [2013] discuss modifications of Dißmann et al. [2013]'s selection strategy that use AIC or p-values of goodness-of-fit tests as the edge weights for selecting the regular vine tree structure, and Brechmann et al. [2012] proposes an extension of Dißmann et al. [2013]'s selection method for shrinkage modeling: after completing the selection of level $k = 1, \ldots, d-1$, compare AIC or BIC of the level k-truncated model to that of the level $(k-1)$-truncated model; if the latter has a better AIC or BIC, end the selection procedure and return the level $(k-1)$-truncated model as the model estimate.

Bayesian model selection. Gruber and Czado [2015b]'s Bayesian selection strategy (see Chapter 2) performs Bayesian posterior simulation from the joint posterior distribution of the the regular vine tree T_k of level k, the pair copula families $\mathcal{B}_k$ of level k, and the parameters $\boldsymbol{\theta}_{1:k}$ of all levels up to k, sequentially for each level $k = 1, \ldots, d-1$. The posterior simulation follows a reversible jump MCMC scheme [Green, 1995] with Metropolis-Hastings updates of the parameters [Metropolis et al., 1953, Hastings, 1970]. Parsimonious model estimates are obtained through a shrinkage prior that penalizes additional parameters. This method is not strictly sequential, given that the parameters of previous levels keep being updated; Algorithm 1.1 provides a mostly accurate depiction of Gruber and Czado [2015b]'s selection method with the only caveat that line 4 estimates all parameters $\boldsymbol{\theta}_{1:k}$.

1.5.2 Simultaneous Inference

Simultaneous selection of all levels massively increases the search space compared to sequential selection, see Table 1.5. Present literature only discusses Bayesian techniques for simultaneous selection of all levels, most of which implement a reversible jump MCMC approach [Green, 1995] consisting of a between-model move and within-model move as outlined in Algorithm 1.2.

Pseudo Code
Algorithm 1.2 (Joint Bayesian Selection and Estimation).

```
1: for iteration $r = 1, \ldots, R$ do
```

	Vine Search Space		Vine Copula Search Space	
	Joint	Stepwise	Joint	Stepwise
Dimension d	Selection	Selection	Selection	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 1.5: 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.

```
    Perform a between-model move to update the current regular vine $\mathcal{V}^{r-1}$ and/or pair copula families $\mathcal{B}_\mathcal{V}^{r-1}$, and parameters $\boldsymbol{\theta}_\mathcal{V}^{r-1}$ to obtain new states $\mathcal{V}^r$, $\mathcal{B}_\mathcal{V}^r$, and $\boldsymbol{\theta}_\mathcal{V}^r$.
    Perform a within-model move to update only the parameters $\boldsymbol{\theta}_\mathcal{V}^r$.
end for
return the posterior sample $(\mathcal{C}^1, \ldots, \mathcal{C}^R)$, where $\mathcal{C}^r = (\mathcal{V}^i, \mathcal{B}_\mathcal{V}^i, \boldsymbol{\theta}_\mathcal{V}^r)$.
```

Frequentist parameter estimation. Joint maximum likelihood estimation of the parameter vector $\boldsymbol{\theta}_\mathcal{V}$ is suggested for frequentist parameter estimation across all levels $k = 1, \ldots, d-1$ simultaneously in Aas et al. [2009] and Dißmann et al. [2013].

Bayesian parameter estimation. Min and Czado [2010] discuss Bayesian inference for D-vine copulas based on Metropolis-Hastings sampling [Metropolis et al., 1953, Hastings, 1970], and their proposed algorithm estimates all parameters of the vine copula simultaneously. An extension of Min and Czado [2010]'s Metropolis-Hastings method for Bayesian posterior inference of the parameters of a regular vine copula is presented in Gruber [2011].

Origins of Bayesian model selection. Min and Czado [2011] and Smith et al. [2010] present competing approaches to select the pair copula families of D-vine copulas. Min and Czado [2011] propose a reversible jump MCMC sampler to decide between an independence copula or a non-independent copula for each pair $e \in \bigcup_{k=1:(d-1)} E_k$; Smith et al. [2010] propose Metropolis-Sampling on an extended state space that combines the regular vine copula parameters with a model indicator matrix that identifies non-independent pairs. Both strategies operate simultaneously on all levels $k = 1, \ldots, d-1$ of a vine copula to produce a Bayesian posterior sample that shows the posterior probability of (conditional) independence for each pair.

Current Bayesian model selection. Modern Bayesian model selection of regular vine copulas is discussed in Gruber and Czado [2015a] (see Chapter 3). We present two fully Bayesian strategies: one to select the pair copula families $\mathcal{B}_\mathcal{V}$ of all levels simultaneously, and another to jointly select the regular vine $\mathcal{V}$ and the pair copula families $\mathcal{B}_\mathcal{V}$ simultaneously across all levels. The former strategy is a major extension of Min and Czado [2011]'s selection method that applies to full class of regular vine copulas, enables selection among several candidate pair copula families $\mathbf{B}$ (Min and Czado [2011], Smith et al. [2010]'s methods can only select between the independence copula and one global alternative), and introduces a variable-strength shrinkage prior to induce selection of parsimonious models; the latter strategy is the first to enable selection of the regular vine $\mathcal{V}$ simultaneously across all levels. Rapid convergence to high posterior density regions is achieved through dynamic likelihood-based proposal distributions. Furthermore, we propose an approximating score of the likelihood of regular vine tree structures $\mathcal{V}$ that is used to generate proposals of $\mathcal{V}$ through importance resampling.

Scenario	1	2	X1-T6	X2-T8	X3-T9
Dimension	$d = 6$			$d = 10$	
Seq. frequentist selection	2883	2677	1450	1339	1618
Seq. Bayesian selection	3053	2916	n/a	n/a	n/a
Fully Bayesian selection (II)	**3697**	**3226**	**1477**	**1498**	**1729**

Table 1.6: Average log-likelihoods of models selected by different selection strategies.

1.5.3 Comparison of Selection Performance

Gruber and Czado [2015b,a] (see Chapters 2 and 3) present extensive simulation studies to quantify the performance of the three major selection methods for regular vine copulas with unknown regular vines $\mathcal{V}$: Dißmann et al. [2013]'s sequential frequentist selection, Gruber and Czado [2015b]'s sequential Bayesian selection, and Gruber and Czado [2015a]'s fully Bayesian selection.

Table 1.6 restates their most significant results: Scenarios 1 and 2 from their simulation studies in $d = 6$ dimensions using sample size $n = 500$ simulation data, and Scenarios X1-T6, X2-T8 and X3-T9 from the simulation study in $d = 10$ dimension using sample size $n = 200$ simulation data of Gruber and Czado [2015a]. The selected scenarios are these that have the fewest independence pairs, which gives the simulation data the most complex dependence structure, and the data are from 50 independent replications of each simulation. Across all selected scenarios, fully Bayesian selection performs best, followed by sequential Bayesian selection, and sequential frequentist selection.

Simultaneous vs. sequential selection. Simultaneous selection performs substantially better than any of the two sequential selection methods. In heavily truncated scenarios (X1-T2, X2-T3, X3-T2, X3-T3 of Gruber and Czado [2015a]), simultaneous selection becomes less efficient, because it searches a much bigger model space than is necessary; if hard truncation is desired by the modeler, theoretical considerations suggest that it should be beneficial to run simultaneous selection on the truncated model space, but there are no simulation results yet.

Sequential Bayesian vs. frequentist selection. The six-dimensional analysis shows that sequential Bayesian selection performs better than sequential frequentist selection. Furthermore, this is confirmed in Gruber and Czado [2015b]'s test of the detection of sparsity patterns through independence pairs in Scenario 3: their sequential Bayesian selection method selects, on average, only 0.9 out of 10 independence pairs as non-independent, while Dißmann et al. [2013]'s sequential frequentist method selects, on average, 3.3 pairs as non-independent.

Computation time. Dißmann et al. [2013]'s sequential frequentist selection is substantially faster than either Bayesian method. In Gruber and Czado [2015b]'s 6-dimensional simulation study, the frequentist method could be completed within several seconds, while sequential Bayesian selection took about 10 hours to complete. Fully Bayesian selection offers a dramatic reduction in computing time by only simulating one MCMC sampling chain instead of one for each level $k = 1, \ldots, d - 1$. In Gruber and Czado [2015a]'s study (see Chapter 3), fully Bayesian simulation completes in about 2 hours.

1.6 Discussion

Regular vine pair copula constructions are a very flexible class of multivariate dependence models. In spite of the model's recent development, it has been quickly embraced by researchers from many areas that work to develop inference and sampling methodology to enable widespread application.

The much-praised flexibility of pair copula constructions makes model selection a challenge. Initial research focused on C-vine and D-vine copulas, while more recent research generalized these findings to apply to, and developed new methods for, the entire class of regular vine copulas. This survey organizes these selection methods along algorithmic-conceptual lines to point out similarities and differences in strategy and performance of proposed methods.

Most researchers wishing to use regular vine copulas need a fully autonomous model selection strategy that can select the regular vine $\mathcal{V}$, and does not require the modeler to specify it. This leaves the choice between sequential frequentist selection [Dißmann et al., 2013, Brechmann et al., 2012], sequential Bayesian selection [Gruber and Czado, 2015b], and simultaneous Bayesian selection [Gruber and Czado, 2015a]. A comparison of these three methods suggests that the most recent, Gruber and Czado [2015a]'s fully Bayesian method, is also the most capable. By selecting all levels simultaneously, this strategy avoids selection bias found in sequential methods.

Furthermore, Gruber and Czado [2015a]'s fully Bayesian method achieves superior results, and is substantially faster than Gruber and Czado [2015b]'s sequential Bayesian strategy. In practice, researchers will most likely choose between almost instantaneous sequential frequentist selection, or more accurate, but also slower, fully Bayesian selection.

Chapter 2

Sequential Bayesian Selection

The contents of this chapter have already been published in Gruber and Czado [2015b]. This chapter is a lightly edited reproduction of selected contents of the accepted manuscript.

2.1 Bayesian Estimation of Regular Vine Copulas Using Reversible Jump MCMC

Bayesian selection of regular vine copulas aims at estimating the joint posterior distribution of the regular vine $\mathcal{V}$, pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\boldsymbol{\theta}_{\mathcal{V}}$.

The multi-layered composition of a regular vine copula and its density function makes analytical inference infeasible. Instead, we use reversible jump MCMC [Green, 1995], which is an extension the Metropolis–Hastings algorithm [Metropolis et al., 1953, Hastings, 1970] to include the selection of models with different numbers of parameters in the scope of inference, as a simulation-based approach to estimate the posterior distribution. Convergence of the sampling chain to the target distribution, here to the posterior distribution, is theoretically established under regularity conditions.

2.1.1 General Tree-by-Tree Model Selection

Our tree-by-tree model selection strategy first estimates the first level of the regular vine copula, which consists of tree $T_1 = (N_1, E_1)$ and the pair copula families $\mathcal{B}_1$ with parameters $\boldsymbol{\theta}_1$. For each higher level $k = 2, \ldots, d-1$, the density factorization $T_k = (N_k, E_k)$ and pair copula families $\mathcal{B}_k$ with parameters $\boldsymbol{\theta}_k$ are selected conditionally on the estimates of the lower levels $(T_1, \mathcal{B}_1, \boldsymbol{\theta}_1)$ to $(T_{k-1}, \mathcal{B}_{k-1}, \boldsymbol{\theta}_{k-1})$, which remain unchanged from the previous steps.

Motivation of tree-by-tree estimation. In the context of model selection for regular vine copulas, sequential approaches exhibit distinct strengths that make them more tractable than joint approaches.

Sequential approaches are much faster than joint approaches, as they break the overall problem into a sequence of smaller problems that can be solved more quickly. Table 1.5 shows the enormous reduction of the regular vine search space, if a sequential procedure is followed. The entries of Table 1.5 follow Morales-Napoles [2011]'s calculation of the number of vines and use the sum of the number of spanning trees with k nodes, $\sum_{k=2}^{d} k^{k-2}$, as an upper bound of the size of the sequential search space. Here the number of spanning trees is calculated using Cayley [1889]'s formula. Furthermore, the reduced number of model alternatives improves the convergence behavior of MCMC samplers as it allows for a quicker exploration of the search space.

Furthermore, a tree-by-tree approach avoids a regular vine copula-specific model identification issue. Different regular vine copulas can be representatives of the same multivariate copula, the most prominent example of which is the multivariate Gaussian copula [Kurowicka and Cooke, 2006]. The tree-by-tree approach is characterized by leaving previously selected trees unchanged and modifying only one tree at a time. Under the tree-by-tree paradigm, there is only one scenario in which the copula of the current state and proposed state are the same with a non-zero probability: all pair copulas—those on all previously selected trees and those on the current tree—are either Gaussian or independent. These states can be easily detected and collapsed into one state.

Priors. Following our tree-by-tree estimation approach, the priors are specified for each level $k = 1, \ldots, d-1$. Given that the proximity condition restricts which trees T_k are allowed for a level $k > 1$, these priors are inherently conditional on the selection on the previous trees $T_1, \ldots, T_{k-1}$.

We choose a noninformative yet proper prior over the set $\mathbf{STP}_k$ of all spanning trees that satisfy the proximity condition for level k for tree T_k, a sparsity-enforcing prior for the pair copula families $\mathcal{B}_k$ and proper noninformative priors for the parameters $\boldsymbol{\theta}_k$. We combine flat $(-1, 1)$-priors for the Kendall's τ parameters with flat $(0, \log(30))$-priors for the logarithm of the degrees of freedom ν of Student's t pair copulas.

$$\begin{aligned}
\pi(T_k) &\propto discrete\, Uniform(\mathbf{STP}_k), \\
\pi(\mathcal{B}_k \mid T_k) &= \frac{\exp(-\lambda d_k)}{\sum_{i=1}^{|E_k|} \sum_{d=0}^{2} \exp(-\lambda d)} \propto \exp(-\lambda d_k), \\
\pi(\boldsymbol{\theta}_e \mid T_k, \mathcal{B}_e) &\propto \begin{cases} Uniform_{(-1,1)}(\tau_e) \text{ if } \mathcal{B}_e \text{ is a single parameter copula} \\ Uniform_{(-1,1)}(\tau_e) \cdot \frac{\mathbf{1}_{(1,30)}(\nu_e) \cdot \log(\nu_e)}{\int_1^{30} \log(x) dx} \text{ if } \mathcal{B}_e \text{ is the Student's t copula} \end{cases} ,
\end{aligned}$$

where d_k denotes the dimension of the parameter vector $\boldsymbol{\theta}_k = (\boldsymbol{\theta}_{e;\mathcal{B}_e} \mid e \in E_k)$ of the pair copula families $\mathcal{B}_k$ of level k. Analogously, d_e denotes the dimension of the parameter vector of the pair copula family $\mathcal{B}_e$ of edge $e \in E_k$ and it holds that $d_k = \sum_{e \in E_k} d_e$. Our prior on the pair copula families $\mathcal{B}_k$ depends solely on the size d_k of their parameter vectors $\boldsymbol{\theta}_k$; if $\mathcal{B}_e$ is the independence copula, it holds that $d_e = 0$.

The prior density π of state $(T_k = (N_k, E_k), \mathcal{B}_k, \boldsymbol{\theta}_k)$ results as

$$\pi(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k) \propto \prod_{e \in E_k} \exp(-\lambda d_e) \pi(\boldsymbol{\theta}_e \mid T_k, \mathcal{B}_e). \tag{2.1}$$

This prior gives the posterior distribution the following form:

$$\begin{aligned}
p(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) &\propto \pi(\boldsymbol{\theta}_k \mid T_k, \mathcal{B}_k) \cdot \exp\left(\ell(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) - \lambda d_k\right) \\
&\overset{\propto}{\sim} \exp\left(\ell(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) - \lambda d_k\right),
\end{aligned}$$

where ℓ denotes the log likelihood function and $\overset{\propto}{\sim}$ means "approximately proportional." At $\lambda = 0$, no shrinkage occurs and the posterior mode estimate of level k will approximate that level's maximum likelihood estimate, while at $\lambda = 1$, the posterior mode estimate of level k will approximately minimize the Akaike Information Criterion (AIC).

Posterior distribution. The posterior distribution of level k given observed data $\mathbf{U}$ factorizes into the likelihood L and prior density π:

$$p(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) \propto L(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k \mid \mathbf{U}) \cdot \pi(T_k, \mathcal{B}_k, \boldsymbol{\theta}_k).$$

The tree-by-tree procedure requires the Bayesian posterior sample of each tree to be collapsed into a single model estimate. We choose the empirical mode of the sampled models $(T_k, \mathcal{B}_k)$ as the model estimate, given that we chose our priors for their effects on the posterior mode. The parameters are set to the means of the MCMC posterior iterates of the selected model. Other centrality estimates may be used as well.

Implementation. At each iteration $r = 1, \ldots, R$, the sampling mechanism performs a within-model move and a between-models move. The within-model move updates all parameters $\boldsymbol{\theta}_{1:k}$ of the regular vine copula, but leaves the pair copula families $\mathcal{B}_{1:k}$ and tree structure $T_{1:k}$ unchanged. The between-models move operates only on level k and updates the tree structure T_k, pair copula families $\mathcal{B}_k$ along with the parameters $\boldsymbol{\theta}_k$.

The between-models move is implemented as a 50:50 mixture of two mutually exclusive, collectively exhaustive (MECE) sub-routines: with a 50% probability, a local between-models move updates only the pair copula families $\mathcal{B}_k$ but leaves the tree structure T_k unchanged (Algorithm 2.2). With the remaining 50% probability, a global between-models move updates the tree structure T_k along with the pair copula families $\mathcal{B}_k$ (Algorithm 2.3). Algorithm 2.2 guarantees that the proposal state differs in at least one pair copula family from the current state; Algorithm 2.3 guarantees that the proposal state differs in at least one edge of tree T_k from the current state. This makes the proposals of the two sub-routines mutually exclusive and gives the acceptance probability a tractable analytical form that can be easily evaluated.

The between-models move is into two sub-routines, because this allows an intuitive interpretation of a local search (Algorithm 2.2) and a global search (Algorithm 2.3) as well as optimizes the computational cost of these updates by containing between-models moves that leave the tree structure unchanged to a dedicated sub-routine.

Algorithm 2.1 (Tree-by-Tree Bayesian Model Selection)**.**

```
for each level $k = 1, \dots, d-1$ do
    Choose starting values: set tree $T_k = (N_k, E_k)$ to an arbitrary tree that fulfills the proximity condition for level $k$; set all pair copula families $\mathcal{B}_k$ of level $k$ to the independence copula, i.e., $c_e(\cdot,\cdot) = 1$ for $e \in E_k$ and set the parameter vector $\boldsymbol{\theta}_k$ of level $k$ to an empty vector.
    for each MCMC iteration $r = 1, \dots, R$ do
        Perform a within-model move: update all parameters $\boldsymbol{\theta}_{1:k}$. Obtain $\boldsymbol{\theta}_{1:k}^{r,\text{NEW}}$ through a Metropolis–Hastings step with random walk proposals:
            $(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_{1:k}^r) = (T_k^{r-1}, \mathcal{B}_k^{r-1}, \boldsymbol{\theta}_{1:k}^{r,\text{NEW}}).$
        Perform a between-models move: update the tree structure $T_k$ along with, or only, the pair copula families $\mathcal{B}_k$ and parameters $\boldsymbol{\theta}_k$ (Algorithms 2.2, 2.3):
            $(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r) = (T_k^{r,\text{NEW}}, \mathcal{B}_k^{r,\text{NEW}}, \boldsymbol{\theta}_k^{r,\text{NEW}}).$
    end for
    Set the level $k$-estimate $(\hat{T}_k, \hat{\mathcal{B}}_k, \hat{\boldsymbol{\theta}}_k)$ to the empirical mode of the posterior sample $((T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r), r = 1, \dots, R)$:
        • Set $\hat{T}_k$ and $\hat{\mathcal{B}}_k$ to the most frequently sampled combination of $T_k$ and $\mathcal{B}_k$ in $((T_k^r, \mathcal{B}_k^r), r = 1, \dots, R)$.
        • Set $\hat{\boldsymbol{\theta}}_k$ to the sample mean of $\left(\boldsymbol{\theta}_k^r, r \in \{1, \dots, R\} \text{ with } T_k^r = \hat{T}_k \text{ and } \mathcal{B}_k^r = \hat{\mathcal{B}}_k\right)$.
    For all levels $l = 1, \dots, k-1$, update $\hat{\boldsymbol{\theta}}_l$ and set it to the sample mean of $\left(\boldsymbol{\theta}_l^r, r \in \{1, \dots, R\} \text{ with } T_k^r = \hat{T}_k \text{ and } \mathcal{B}_k^r = \hat{\mathcal{B}}_k\right)$.
end for
return the stepwise Bayesian model estimate $(\hat{\mathcal{V}}, \hat{\mathcal{B}}_{\mathcal{V}}, \hat{\boldsymbol{\theta}}_{\mathcal{V}})$, where $\hat{\mathcal{V}} = (\hat{T}_1, \dots, \hat{T}_{d-1})$, $\hat{\mathcal{B}}_{\mathcal{V}} = (\hat{\mathcal{B}}_1, \dots, \hat{\mathcal{B}}_{d-1})$, and $\hat{\boldsymbol{\theta}}_{\mathcal{V}} = (\hat{\boldsymbol{\theta}}_1, \dots, \hat{\boldsymbol{\theta}}_{d-1})$.
```

2.1.2 Update of the Pair Copulas of Level k

This section describes a sub-routine of Algorithm 2.1 to update the pair copula families $\mathcal{B}_k$ and parameters $\boldsymbol{\theta}_k$ of level k of a regular vine copula. This updating step leaves the density factorization $\mathcal{V}$ unchanged.

This sub-routine first selects how many pair copulas will be updated (Line 1 of Algorithm 2.2) and then randomly selects which pair copulas will be updated—denoted by $E \subseteq E_k$ in the remainder (Line 2). Next, it generates a proposal that updates the selected pair copulas (Lines 3–9), and, lastly, accepts or rejects the proposal based on a Metropolis–Hastings updating rules (Line 10).

The proposal step (Lines 3–9) iterates through all selected pair copulas $e \in E$. It first estimates the parameters $\boldsymbol{\theta}_{e;\mathcal{B}_e^*}$ of each candidate pair copula family $\mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r$, where the estimates are denoted by $\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}$. The likelihoods of the different candidate copulas, $L(\mathcal{B}_e^*, \hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*} \mid \mathbf{U})$, are then used as the proposal probability weights of the respective copula families: $q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*) \propto L(\mathcal{B}_e^*, \hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*} \mid \mathbf{U})$. After selecting a pair copula family, the proposal parameters $\boldsymbol{\theta}_e^*$ are drawn from a normal distribution centered at the parameter estimate $\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}$. The proposal distribution q_N from which N is drawn (Line 1), the parameter estimation procedure (Line 4) and the covariance matrix Σ of the parameters' proposal distribution (Line 6) are MCMC tuning parameters.

Pair copula families that can model only positive or negative Kendall's τ's such as the Clayton copula or Gumbel copula are extended to cover the entire range $[-1, 1]$. This is implemented by replacing the first argument u_1 of the copula density function $c(u_1, u_2)$ by $1 - u_1$ whenever the dependence parameter τ changes signs.

As this sub-routine and the one from Section 2.1.3 produce non-overlapping proposals, the acceptance probability follows as

$$\boldsymbol{\alpha} = \frac{L(T_k^r, \mathcal{B}_k^*, \boldsymbol{\theta}_k^* \mid \mathbf{U})}{L(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r \mid \mathbf{U})} \cdot \frac{\pi(T_k^r, \mathcal{B}_k^*, \boldsymbol{\theta}_k^*)}{\pi(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r)} \cdot \prod_{e \in E} \frac{q_{\mathcal{B}}(\mathcal{B}_e^* \to \mathcal{B}_e^r) \cdot \phi_{(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^r}, \Sigma)}(\boldsymbol{\theta}_e^r)}{q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*) \cdot \phi_{(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}, \Sigma)}(\boldsymbol{\theta}_e^*)}, \tag{2.2}$$

where $\phi_{\boldsymbol{\mu},\Sigma}(\cdot)$ denotes the density function of the truncated multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ; the truncation is assumed at the bounds of the respective parameters. Both the numerator and denominator of the acceptance probability contain $q_N(N)$ as a factor that cancels out and does not appear in Equation 2.2, given that the return move of any update must change the same number N of pair copulas as the outbound move.

Algorithm 2.2 (Between-Models Move to Update the Pair Copula Families $\mathcal{B}_k$ and Parameters $\boldsymbol{\theta}_k$)**.** *This is for the r-th iteration of line 5 of Algorithm 2.1.*

```
1: Select how many pair copulas are updated: $N \sim q_N(\cdot)$; $N \in \{1, \dots, |E_k|\}$.
2: Select which pair copulas are updated: $E \subseteq E_k$ with $|E| = N$.
3: for each selected pair copula $e \in E$ do
4:     For each candidate pair copula family $\mathcal{B}_e \in \mathbf{B} \setminus \mathcal{B}_e^r$ estimate the copula parameter $\boldsymbol{\theta}_{e;\mathcal{B}_e}$ given the
       transformed data $(\mathbf{u}_{i(e)|D(e)}^{t=1:T}, \mathbf{u}_{j(e)|D(e)}^{t=1:T})$ and denote the parameter estimate by $\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e}$.
5:     Draw a new copula family $\mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r$ from the proposal distribution
           $$q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*) \propto L(\mathcal{B}_e^*, \hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*} \mid \mathbf{U}).  \tag{2.3}$$
6:     Draw new parameters $\boldsymbol{\theta}_e^* \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}, \Sigma)$ from a normal distribution.
7:     The proposal family for pair copula $e \in E$ is $\mathcal{B}_e^*$ and the proposal parameter is $\boldsymbol{\theta}_e^*$.
8: end for
9: The proposal families for level $k$ are $\mathcal{B}_k^*$ and the proposal parameters are $\boldsymbol{\theta}_k^*$, where
       $$\mathcal{B}_k^* = (\mathcal{B}_e^* \text{ for } e \in E \text{ and } \mathcal{B}_e^r \text{ for } e \in E_k \setminus E)$$
       $$\boldsymbol{\theta}_k^* = (\boldsymbol{\theta}_e^* \text{ for } e \in E \text{ and } \boldsymbol{\theta}_e^r \text{ for } e \in E_k \setminus E).$$
10: Accept the proposal and set $(T_k^{r,NEW}, \mathcal{B}_k^{r,NEW}, \boldsymbol{\theta}_k^{r,NEW}) = (T_k^r, \mathcal{B}_k^*, \boldsymbol{\theta}_k^*)$ with probability $\boldsymbol{\alpha}$ (Eq. 2.2). If
    rejected, set $(T_k^{r,NEW}, \mathcal{B}_k^{r,NEW}, \boldsymbol{\theta}_k^{r,NEW}) = (T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r)$.
```

2.1.3 Joint Update of the Regular Vine and Pair Copulas of Level k

This section presents a sub-routine of Algorithm 2.1 to update the regular vine at level k—that is, tree T_k—and the pair copula families $\mathcal{B}_k$ and parameters $\boldsymbol{\theta}_k$ of that level. Definition 1.1 requires that the lower level trees $T_1, \dots, T_{k-1}$ of the regular vine are specified before tree T_k is estimated.

Algorithm 2.3 describes our joint update procedure of tree $T_k = (N_k, E_k)$ and the corresponding pair copula families $\mathcal{B}_k$ and parameters $\boldsymbol{\theta}_k$. We denote the set of all spanning trees with node set N_k that satisfy the proximity condition by $\mathbf{STP}_k$. The cardinality of this set is computed using Kirchhoff's matrix tree theorem [Kirchhoff, 1847] to obtain the normalizing constants of the proposal and prior distributions. In a first step, this sub-routine draws a new spanning tree $T_k^* = (N_k, E_k^*) \in \mathbf{STP}_k \setminus T_k^r$ from the proposal distribution $q_T(T_k^r \to T_k^*) \propto p^{|E_k^* \cap E_k^r|} \cdot (1-p)^{|E_k^* \setminus E_k^r|}$ (Line 1); this is just a random walk distribution on the set of allowable regular vine trees of level k! Then, the algorithm generates a proposal for the pair copula families $\mathcal{B}_k^*$ and parameters $\boldsymbol{\theta}_k^*$ of this level as in Algorithm 2.2 (Lines 3–9 in Algorithm 2.2; Lines 2–8 in Algorithm 2.3). The only difference is that all pair copula families in $\mathbf{B}$ are permissible candidates here and the edges e are different. We use the notation $q_{\mathcal{B}}(\mathcal{B}_e^*)$ instead of $q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*)$ to indicate the slightly different proposal distributions. The entire proposal for level k of the regular vine copula consists of a new tree T_k^*, pair copula families $\mathcal{B}_k^*$ and parameters $\boldsymbol{\theta}_k^*$, and is accepted or rejected based on Metropolis–Hastings updating rules (Line 9).

This sub-routine has three MCMC tuning parameters. The first is the parameter p of the proposal distribution for tree T_k: values $p > 0.5$ make tree proposals T_k^* similar to the current tree T_k^r more likely than proposals that are less similar to the current state. The situation is reversed for values $p < 0.5$. The second tuning parameter is the choice of the estimation procedure for the pair copula parameter vectors (Line 3) and the last is the covariance matrix Σ of the proposal distribution of the parameters (Line 5).

The proposal mechanism of this update routine guarantees that the proposed regular vine tree T_k^* is different from the current state T_k. This ensures that the proposals of this sub-routine and the one of Section 2.1.2 are mutually exclusive. Furthermore, the proposal probability of the reverse move from tree T_k^* to T_k is the same as the proposal probability of the away move, given that the number of shared edges as well as differing edges is the same. As a result, the acceptance probability of a proposal of this algorithm can be easily obtained as

$$\alpha = \frac{L(T_k^*, \mathcal{B}_k^*, \boldsymbol{\theta}_k^* \mid \mathbf{U})}{L(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r \mid \mathbf{U})} \cdot \frac{\pi(T_k^*, \mathcal{B}_k^*, \boldsymbol{\theta}_k^*)}{\pi(T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r)} \cdot \frac{\prod_{e \in E_k^r} q_{\mathcal{B}}(\mathcal{B}_e^r) \cdot \phi_{(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^r}, \Sigma)}(\boldsymbol{\theta}_e^r)}{\prod_{e \in E_k^*} q_{\mathcal{B}}(\mathcal{B}_e^*) \cdot \phi_{(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}, \Sigma)}(\boldsymbol{\theta}_e^*)}. \tag{2.4}$$

Algorithm 2.3 (Between-Models Move for a Joint Update of Tree $T_k = (N_k, E_k)$ and the Pair Copula Families $\mathcal{B}_k$ and Parameters $\boldsymbol{\theta}_k$)**.**
This is for the r-th iteration of line 5 of Algorithm 2.1.

```
1: Draw a new spanning tree $T_k^* = (N_k, E_k^*) \in \mathbf{STP}_k \setminus T_k^r$ that satisfies the proximity condition from the
   proposal distribution
       $q_T(T_k^r \to T_k^*) \propto p^{|E_k^* \cap E_k^r|} \cdot (1-p)^{|E_k^* \setminus E_k^r|}.$   (2.5)
2: for each pair copula $e \in E_k^*$ do
3:     For each candidate pair copula family $\mathcal{B}_e \in \mathbf{B}$ estimate the copula parameter $\boldsymbol{\theta}_{e;\mathcal{B}_e}$ given the trans-
       formed data $(\mathbf{u}_{i(e)|D(e)}^{t=1:T}, \mathbf{u}_{j(e)|D(e)}^{t=1:T})$ and denote the parameter estimate by $\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e}$.
4:     Draw a new copula family $\mathcal{B}_e^* \in \mathbf{B}$ from the proposal distribution
           $q_{\mathcal{B}}(\mathcal{B}_e^*) \propto L(\mathcal{B}_e^*, \hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*} \mid \mathbf{U}).$   (2.6)
5:     Draw new parameters $\boldsymbol{\theta}_e^* \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}, \Sigma)$ from a normal distribution.
6:     The proposal family for pair copula $e \in E_k^*$ is $\mathcal{B}_e^*$ and has proposal parameter $\boldsymbol{\theta}_e^*$.
7: end for
8: The proposal state is $(T_k^*, \mathcal{B}_k^*, \boldsymbol{\theta}_k^*)$, where
       $\mathcal{B}_k^* = (\mathcal{B}_e^* \mid e \in E_k^*)$ and $\boldsymbol{\theta}_k^* = (\boldsymbol{\theta}_e^* \mid e \in E_k^*).$
9: Accept the proposal and set $(T_k^{r,NEW}, \mathcal{B}_k^{r,NEW}, \boldsymbol{\theta}_k^{r,NEW}) = (T_k^*, \mathcal{B}_k^*, \boldsymbol{\theta}_k^*)$ with probability $\alpha$ (Eq. 2.4). If
   rejected, set $(T_k^{r,NEW}, \mathcal{B}_k^{r,NEW}, \boldsymbol{\theta}_k^{r,NEW}) = (T_k^r, \mathcal{B}_k^r, \boldsymbol{\theta}_k^r)$.
```

2.1.4 Implementation in C++

The model selection algorithms presented in this section are implemented in a proprietary C++ software package. As the computational cost of evaluating the likelihood and calculating parameter estimates increases linearly with the number of observations in the data set, these tasks are parallelized onto multiple CPU cores using OpenMP to help reduce overall computing time. Furthermore, our software package relies heavily on the tools and functionality provided by the boost [Boost Community, 2014] and CppAD [COIN-OR Foundation, 2014] libraries: the boost library contains a function that generates random spanning trees from a product probability distribution based on edge weights, which we employ in our implementation of Algorithm 2.3; the CppAD library allows for automatic differentiation, which we use for parameter estimation.

Scenario 1	Scenario 2	Scenario 3	Scenario 4
$c_{1,2}$ N(0.59)	$c_{1,2}$ T(0.54, 5)	$c_{1,2}$ N(0.41)	$c_{1,2}$ N(0.41)
$c_{2,3}$ C(0.71)	$c_{1,3}$ C90(-0.67)	$c_{2,3}$ C(0.50)	$c_{2,3}$ N(0.49)
$c_{3,4}$ C180(0.80)	$c_{1,4}$ C180(0.64)	$c_{3,4}$ C180(0.50)	$c_{2,4}$ N(-0.33)
$c_{3,5}$ N(-0.71)	$c_{1,5}$ N(-0.59)	$c_{3,5}$ N(-0.33)	$c_{3,5}$ N(-0.26)
$c_{3,6}$ T(0.65, 3)	$c_{1,6}$ T(0.54, 6)	$c_{3,6}$ T(0.49, 5)	$c_{3,6}$ N(0.13)
$c_{1,3\|2}$ G(0.75)	$c_{2,3\|1}$ G(0.71)		$c_{1,3\|2}$ N(0.59)
$c_{2,4\|3}$ N(0.41)	$c_{2,4\|1}$ G270(-0.71)		$c_{2,5\|3}$ N(0.13)
$c_{2,5\|3}$ C270(-0.60)	$c_{2,5\|1}$ C270(-0.60)		$c_{3,4\|2}$ N(0.41)
$c_{2,6\|3}$ N(-0.37)	$c_{2,6\|1}$ N(-0.45)		$c_{5,6\|3}$ N(-0.33)
$c_{1,4\|2,3}$ T(0.26, 5)	$c_{3,4\|1,2}$ T(0.30, 8)		$c_{1,5\|2,3}$ N(0.26)
$c_{1,5\|2,3}$ N(-0.26)	$c_{3,5\|1,2}$ N(-0.30)		$c_{2,6\|3,5}$ N(-0.41)
$c_{1,6\|2,3}$ C90(-0.56)	$c_{3,6\|1,2}$ C90(-0.43)		$c_{4,5\|2,3}$ N(0.19)
$c_{4,6\|1,2,3}$ N(0.13)	$c_{4,5\|1,2,3}$ N(0.19)		$c_{1,6\|2,3,5}$ N(0.49)
$c_{5,6\|1,2,3}$ C(0.20)	$c_{4,6\|1,2,3}$ C(0.43)		$c_{4,6\|2,3,5}$ N(0.41)
$c_{4,5\|1,2,3,6}$ G180(0.52)	$c_{5,6\|1,2,3,4}$ G180(0.50)		$c_{1,4\|2,3,5,6}$ N(-0.33)
17 parameters	18 parameters	6 parameters	15 parameters

Table 2.1: The vine copulas used in the simulation study. The parameters shown are the Kendall's τ and the degrees of freedom ν.

2.2 Simulation Study

We present a simulation study that compare our sequentially Bayesian strategy with Dißmann et al. [2013]'s frequentist model selection algorithm, the independence model and the maximum likelihood estimate (MLE) of the multivariate Gaussian copula. The comparisons with the independence model and Gaussian copula illustrate that vine copulas are relevant dependence models that significantly improve model fit over simpler standard models, while the comparison with Dißmann's vine copula estimates highlights the improved model selection capabilities of our method.

Our simulation study uses copula data generated from four different six-dimensional vine copulas (Table 2.1). These scenarios cover a wide range of dependence structures: the first two cover general cases of multivariate dependence, while the third and fourth scenario are special cases to investigate detailed characteristics of our model selection method. Scenario 3 consists of only one level, which means that all variables are conditionally independent. It also means that the true model lies in the search space of the first level of our selection procedure, so that this scenario can be used to validate our proposed scheme empirically. Scenario 4 is has only Gaussian pair copulas, which makes it a vine copula-representation of the multivariate Gaussian copula. As a result, this scenario allows for an isolated evaluation of the pair copula family selection aspect of our method, given that the multivariate Gaussian copula results from any vine density factorization $\mathcal{V}$ as long as all pair copula families are Gaussian.

We generate 100 data sets consisting of 500 independent samples from the respective copula distribution of each scenario and allow the pair copula families listed in Section 1.3 as candidates.

2.2.1 Choice of the Benchmark Algorithm

The models selected by Dißmann et al. [2013]'s algorithm serve as a benchmark. This algorithm follows a stepwise frequentist approach that selects each tree T_k, $k = 1, \ldots, 5$ as the maximum spanning tree using absolute values of Kendall's τ of the variable pairs as edge weights. The pair copula families are selected to optimize the AIC copula-by-copula and the parameters are set to their individual maximum likelihood estimates.

Dißmann's algorithm and ours share their tree-by-tree selection strategy. However, there are two major differences between our approaches: firstly, Dißmann follows a heuristic scheme to select the tree structure $\mathcal{V}$, while we follow a proper Bayesian selection scheme on each level k; secondly, Dißmann selects the pair copula families on an edge-by-edge basis, whereas we place priors on the distribution of the pair copula families across an entire level k to simultaneously select of all edges of that level.

Algorithm	Tuning Parameters
2.2	$q_N(N=k) = \frac{1}{3.5}\log\left(1 - \frac{1-e^{-3.5}}{\lvert E_k\rvert e^{-3.5}+k(1-e^{-3.5})}\right)$, where $\lvert E_k\rvert$ denotes the number of pair copulas of level k
2.2, 2.3	Parameter estimation is done by matching the Kendall's τ parameter to the sample Kendall's τ. The degrees of freedom ν of a Student's t pair copula is maximum likelihood estimated on a discrete grid.
2.2, 2.3	$\Sigma = 0.0125^2$ for the Kendall's τ of single-parameter copulas; $\Sigma = \begin{pmatrix} 0.0125^2 & 0 \\ 0 & 0.1^2 \end{pmatrix}$ for the $(\tau, \log\nu)$ parameter vector of the Student's t copula
2.3	$p = 0.667$

Table 2.2: MCMC tuning parameters used in the simulation study and real data example.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Gruber > Dißmann (out of 100)	97	98	0	86
Gruber rel. loglik (in %)	**81.0**	**84.9**	100.6	**100.3**
Dißmann rel. loglik (in %)	76.6	77.6	**101.3**	100.1
Gaussian MLE rel. loglik (in %)	64.0	67.6	84.7	100.1
Independence rel. loglik (in %)	0	0	0	0

Table 2.3: Number of replications in which our algorithm's estimate has a higher likelihood than Dißmann's; average percentage of the true log likelihood of the estimated vine copulas, the multivariate Gaussian copula and the independence model.

2.2.2 Configuration of Our Reversible Jump MCMC Sampler

We use the shrinkage prior introduced in Section 2.1.1 with shrinkage parameter $\lambda = 1$. The posterior mode estimates of each level k will then be approximately AIC-optimal.

We use our reversible jump MCMC Algorithm 2.1 from Section 2.1 to generate $R = 50{,}000$ posterior samples for each level $k = 1, \ldots, 5$ of the 6-dimensional regular vine copula. The MCMC tuning parameters are summarized in Table 2.2. Furthermore, we apply a re-weighting on the proposal probabilities (2.3) and (2.6) of the pair copula families in the sub-routines of Algorithms 2.2 and 2.3 to improve the mixing behavior of the sampling chain. This is achieved by ensuring that the ratio of smallest and biggest the proposal probabilities is bounded from below by 0.05,

$$\frac{\min_{\mathcal{B}^* \in \mathbf{B} \setminus \mathcal{B}^r} q_{\mathcal{B}}(\mathcal{B}^r \to \mathcal{B}^*)}{\max_{\mathcal{B}^* \in \mathbf{B} \setminus \mathcal{B}^r} q_{\mathcal{B}}(\mathcal{B}^r \to \mathcal{B}^*)} \geq 0.05 \text{ and } \frac{\min_{\mathcal{B}^* \in \mathbf{B}} q_{\mathcal{B}}(\mathcal{B}^*)}{\max_{\mathcal{B}^* \in \mathbf{B}} q_{\mathcal{B}}(\mathcal{B}^*)} \geq 0.05, \text{ respectively.}$$

2.2.3 Evaluation of the Results

The results are based on 100 replications of the estimation procedures with independently generated data sets of size 500 from the four scenarios and are summarized in Table 2.3. The fitting capabilities of our algorithm and Dißmann et al. [2013]'s are measured by the log likelihood of the estimated models. Knowing the underlying "true" models, we can also calculate the ratio of the estimated log likelihoods and the true log likelihoods to evaluate how well the selection methods perform in absolute terms. Figure 2.1 compares the performance of our Bayesian strategy with Dißmann's heuristic: markers above the diagonal line indicate replications in which our Bayesian model estimate has a higher likelihood than Dißmann's.

Scenarios 1, 2. The log likelihoods of the models selected by our algorithm average 81% and 85% of the log likelihoods of the true models, Dißmann's model estimates average 77% and 78%, and the multivariate Gaussian copula averages 64% and 68%, respectively (Table 2.3). While neither Dißmann's method nor ours selects the correct tree T_1 in any replication, these numbers still show that model fit is

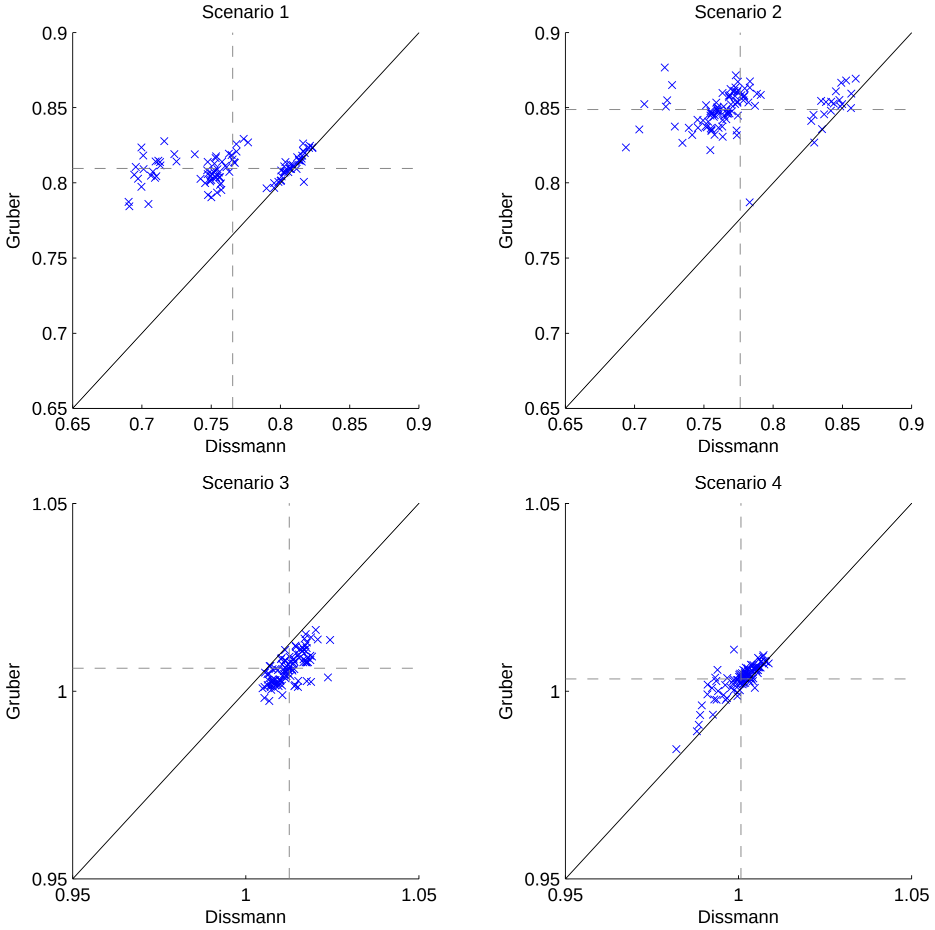


Figure 2.1: Comparison of relative log likelihoods of our method and Dißmann's. The dashed lines indicate the respective averages across all 100 replications.

Posterior Model	**29**	30	31
Posterior probability (in %)	**58.4**	32.1	4.0
Average relative log likelihood (in %)	**100.2**	100.1	100.4
Number of parameters	**5**	5	6
Correct tree T_1	**yes**	yes	yes

Table 2.4: Scenario 3, Replication 1. Histogram table of selected models with an empirical posterior probability of at least 1%. The posterior probabilities and relative log likelihoods are quoted in percentage points.

improved significantly by our approach, and that Dißmann's estimates are already more suitable models than the multivariate Gaussian copula.

Figure 2.1 shows that within each scenario, the relative log likelihoods of our model selection procedure are distributed much more narrowly about the mean compared with Dißmann's. Furthermore, in 97 (Scenario 1) and 98 (Scenario 2) out of 100 replications, our method's estimates perform better than Dißmann's (Table 2.3). Together, this shows that our model selection strategy is more robust and performs consistently better.

Scenario 3. The regular vine copula of Scenario 3 is truncated to the first level. As a result, the true model lies in the search space of the first step of our tree-by-tree model selection procedure, which makes this a test case to validate our implementation.

Our shrinkage prior effectively avoids over-fitting, given that, on average, only 0.8 of the 10 pair copulas on levels $k = 2{:}5$ are selected as non-independence copulas, and in 43 out of 100 replications, all pair copulas on levels $k = 2{:}5$ are selected as independence copulas. Furthermore, in all 100 replications, the posterior mode estimate has the true model's tree structure T_1. Dißmann's procedure is more prone to over-fitting with, on average, 3.3 out of the 10 pair copulas on levels $k = 2{:}5$ being non-independence copulas and only 2 out of 100 replications selecting all pair copulas as independence copulas.

The log likelihoods of the estimated models by our algorithm average 101% as do the log likelihoods from Dißmann's models. This is an excellent result that confirms the validity of our model selection scheme and shows that it is implemented correctly. The consistently slightly higher log likelihoods of Dißmann's model estimates are based on over-fitting. This scenario confirms our method as superior to Dißmann's, as it is important for an effective selection method to identify sparse patterns. The multivariate Gaussian copula lags behind with an average relative log likelihood of 85% even though it is the model that has the most parameters.

Detailed Analysis of the MCMC Output of Level 1 of Replication 1: After discarding the first 2,500 iterations as burn-in, the posterior mode model has a posterior probability of 58% (Model 29; Table 2.4) and all posterior samples, after burn-in, have the correct tree structure T_1. The selected pair copula families agree with the correct pair copula families, except for the family of edge $3, 6$: Model 29 selects a Gaussian pair copula, Model 30 selects the 180 degree rotation of the Gumbel copula, and Model 31 selects the Student's t copula. The fact that the correct model, Model 31, has only 4% posterior probability can be attributed to our shrinkage prior, given that the log likelihoods of these three models are nearly identical. Figure 2.2 illustrates the MCMC mixing behavior using the model index and log likelihood trace plots.

Scenario 4. Both model selection procedures select models that average about 100% of the log likelihoods of the true model. This extraordinary performance can be explained by a peculiarity of vine copulas: if all pair copulas are Gaussian or independence copulas, the vine copula equals a multivariate Gaussian copula irrespective of the density factorization $\mathcal{V}$. As a result, the selection of the density factorization $\mathcal{V}$ does not play a role in selecting suitable vine copula models here. Our sequential Bayesian procedure selects, on average, 13.8 out of the 15 pair copulas as either Gaussian or independence copulas, while Dißmann's procedures comes in second at 12.7 out of 15. This result, together with the high relative log likelihoods, suggests that both algorithms perform similarly well at selecting suitable pair copula families.

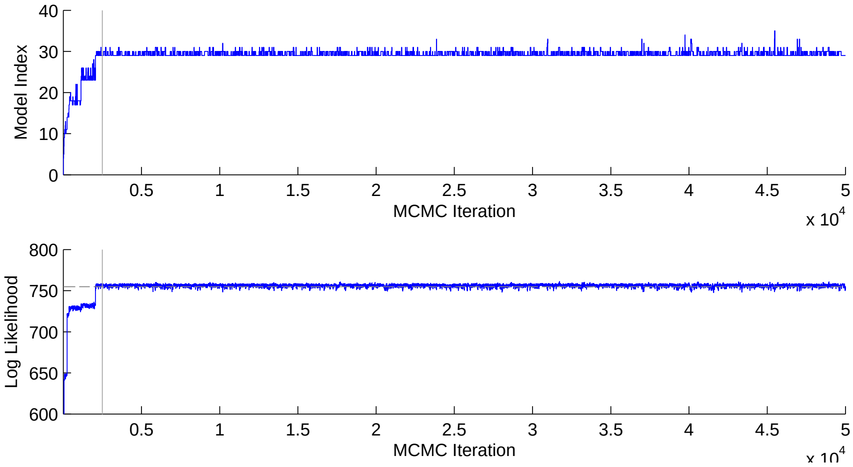


Figure 2.2: Model index and log likelihood trace plot of Replication 1 of Scenario 3. The horizontal line in the lower plot indicates the true model's log likelihood; the vertical lines show the burn-in period.

Conclusion. Both algorithms perform equally well in fitting a vine copula to Gaussian data. Our tree-level Bayesian approach improves model selection of general regular vine copulas, which are not independent of the selected tree structure $\mathcal{V}$. The large performance gap between Scenarios 1, 2 and the special cases of Scenarios 3, 4 shows the limits of our tree-by-tree approach towards the selection of the tree structure $\mathcal{V}$. We acknowledge that our model selection scheme does not yet represent the definitive answer to the model selection challenge. Nevertheless, our proposed selection scheme consistently selects better-fitting models than existing selection strategies and is better at detecting sparsity patterns for model reduction than Dißmann et al. [2013]'s frequentist method.

2.2.4 Analysis of the Computational Complexity and Runtime

Computational complexity. The computation of a single model estimate by our algorithm using the set-up described in Section 2.2.2 consists of 50,000 MCMC updates for each level of the vine copula. These sum up to 250,000 MCMC updates for the five levels of a six-dimensional regular vine copula. Each MCMC update consists of a between-models move and a within-model move: the between-models move consists of estimating the parameters and calculating the likelihood of each pair copula and each candidate pair copula family, and an additional evaluation of the likelihood after drawing a proposal parameter; the within-model move brings another evaluation of the likelihood of each pair copula.

Computing facilities and runtime. The simulation study was performed on a Linux cluster with AMD Opteron-based 32-way nodes using 2.6 GHz dual core CPUs for parallel processing. The Linux cluster is hosted by the Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften near Munich, Germany. It took approximately 10 hours to execute our stepwise Bayesian selection strategy for a six-dimensional data set of size 500, while it took 5-6 seconds to execute Dißmann et al. [2013]'s heuristic and less than 0.1 seconds to estimate the correlation matrix of the multivariate Gaussian copula. It may be noted that the runtime of our procedure could be cut significantly by reducing the number of MCMC iterations. Our analyses suggest that convergence is achieved quickly so $R \in [15{,}000, 30{,}000]$ will be adequate choices in practice.

Recommendations for researchers. In most studies, researchers have to strike a balance between getting quick, or getting more accurate results. With that in mind, we propose the following approach to decide which dependence model to use. In a quick first analysis, estimate a multivariate Gaussian copula and select a regular vine copula using Dißmann et al. [2013]'s heuristic, which can be completed within a few seconds. If the log likelihoods of both models are similar, use the multivariate Gaussian copula as a "good enough" standard model (see Scenario 4). However, if the log likelihood of the selected

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 REITs
7	REZ	iShares Residential REIT Capped ETF	U.S. residential real estate stocks and REITs
8	SLV	iShares Silver Trust	Silver
9	IAU	iShares Gold Trust	Gold

Table 2.5: Overview of the ETFs selected for the real data example. The exposure information is taken from the iShares homepage.

vine copula is substantially higher than the one of the Gaussian copula, perform a sequential Bayesian analysis using our method for more accurate and more robust results, and better sparsity detection (see Scenarios 1–3).

2.3 Example: Portfolio Asset Returns

We consider a diversified portfolio that invests in multiple asset classes using iShares exchange-traded funds (ETFs) and commodity trusts. The daily log returns of each investment are modeled by a univariate time series. The joint multivariate characteristics are modeled by a regular vine copula and a multivariate Gaussian copula.

We learn the copulas using one year's worth of data and then use the selected copulas together with the marginal time series to obtain joint multivariate step-ahead forecasts for six months. The quality of the forecasts is measured by comparing the forecast accuracy of various portfolio metrics with the actual realizations.

2.3.1 Description of the Data

The data set contains adjusted daily closing prices of nine iShares ETFs, $j = 1{:}9$, and covers the time period from January 2013 through June 2014[1]. The training set consists of 252 observations from January through December 2013 ($t = 1{:}252$); the test set consists of 124 observations from January through June 2014 ($t = 253{:}376$). The nine ETFs form a well-diversified portfolio that invests in multiple asset classes and can be easily replicated by retail investors (Table 2.5). Three of the funds invest in U.S. equities ($j = 1, 2, 3$), two funds in U.S. treasuries ($j = 4, 5$), two funds in U.S. real estate through real estate investment trusts (REITs, $j = 6, 7$), and two funds are commodity trusts investing in gold and silver ($j = 8, 9$)[2].

2.3.2 Marginal Time Series

We model the daily log returns $y_{j,t}$, $t = 1, 2, \ldots$ of each series $j = 1{:}9$ using a variance discounting dynamic linear model (DLM; [West and Harrison, 1997, Chapter 10.8]). The DLM is a fully Bayesian time series model that has closed-form posterior and forecast distributions, and the parameters are learned on-line. The following updating equations are adapted from Table 10.4 of West and Harrison [1997].

[1] The data were downloaded from http://finance.yahoo.com

[2] More details on the selected funds can be found on the iShares homepage at http://www.ishares.com/us/index

Model structure. The general DLM models each time series $y_{j,t}$, $j = 1{:}9$, by

$$y_{j,t} = \mathbf{F}'_{j,t}\boldsymbol{\theta}_{j,t} + \nu_{j,t} \qquad \nu_{j,t} \sim N(0, \lambda_{j,t}^{-1}), \tag{2.7}$$

$$\boldsymbol{\theta}_{j,t} = \mathbf{G}_{j,t}\boldsymbol{\theta}_{j,t-1} + \boldsymbol{\omega}_{j,t} \qquad \boldsymbol{\omega}_{j,t} \sim N(\mathbf{0}, \mathbf{W}_{j,t}/(c_{j,t}\lambda_{j,t})), \tag{2.8}$$

$$\lambda_{j,t} = \lambda_{j,t-1}\frac{\eta_{j,t}}{\beta_j} \qquad \eta_{j,t} \sim Beta(\frac{\beta_j n_{j,t-1}}{2}, \frac{(1-\beta_j)n_{j,t-1}}{2}), \tag{2.9}$$

with observation equation (2.7). Equations (2.8) and (2.9) describe the evolutions of the states $\boldsymbol{\theta}_{j,t}$, a vector with p_j entries, and $\lambda_{j,t}$, a positive scalar, where the innovations $\nu_{j,t}$, $\boldsymbol{\omega}_{j,t}$ and $\eta_{j,t}$ are assumed mutually independent and independent over time. The predictors $\mathbf{F}_{j,t}$ are a vectors of size p_j and the state evolution matrices $\mathbf{G}_{j,t}$ are of dimensions $p_j \times p_j$. The parameters $\mathbf{W}_{j,t}$, $n_{j,t-1}$ and β_j of the state evolutions (2.8, 2.9) are explained in the next paragraph.

Forward filtering. The information set at time t is denoted by $\mathcal{D}_t$. Suppose that at time $t-1$, a normal-gamma prior for $(\boldsymbol{\theta}_{j,t}, \lambda_{j,t})$, given information $\mathcal{D}_{t-1}$ has density

$$\pi_{j,t}(\boldsymbol{\theta}_{j,t}, \lambda_{j,t}) := N(\boldsymbol{\theta}_{j,t} \mid \mathbf{a}_{j,t}, \mathbf{R}_{j,t}/(c_{j,t}\lambda_{j,t})) \cdot G(\lambda_{j,t} \mid r_{j,t}/2, r_{j,t}c_{j,t}/2), \tag{2.10}$$

and parameters $\mathbf{a}_{j,t} \in \mathbb{R}^{p_j}$, $\mathbf{R}_{j,t} \in \mathbb{R}^{p_j \times p_j}$, $r_{j,t} > 0$ and $c_{j,t} > 0$; at $t = 0$, the initial prior parameters are $\mathbf{a}_{j,1}$, $\mathbf{R}_{j,1}$, $r_{j,1}$ and $c_{j,1}$. As $y_{j,t}$ is observed at time t, the information set is updated to $\mathcal{D}_t$ and the posterior distribution of $(\boldsymbol{\theta}_{j,t}, \lambda_{j,t})$, given information $\mathcal{D}_t$ follows as a normal-gamma

$$p_{j,t}(\boldsymbol{\theta}_{j,t}, \lambda_{j,t}) := N(\boldsymbol{\theta}_{j,t} \mid \mathbf{m}_{j,t}, \mathbf{C}_{j,t}/(s_{j,t}\lambda_{j,t}))G(\lambda_{j,t} \mid n_{j,t}/2, n_{j,t}s_{j,t}/2) \tag{2.11}$$

with parameters $\mathbf{m}_{j,t} = \mathbf{a}_{j,t} + \mathbf{A}_{j,t}e_{j,t} \in \mathbb{R}^{p_j}$, $\mathbf{C}_{j,t} = (\mathbf{R}_{j,t} - \mathbf{A}_{j,t}\mathbf{A}'_{j,t}q_{j,t})z_{j,t} \in \mathbb{R}^{p_j \times p_j}$, $n_{j,t} = r_{j,t} + 1 > 0$ and $s_{j,t} = z_{j,t}c_{j,t} > 0$, where $e_{j,t} = y_{j,t} - \mathbf{F}'_{j,t}\mathbf{a}_{j,t} \in \mathbb{R}$ is the forecast error, $q_{j,t} = c_{j,t} + \mathbf{F}'_{j,t}\mathbf{R}_{j,t}\mathbf{F}_{j,t} > 0$ is the forecast variance factor, $\mathbf{A}_{j,t} = \mathbf{R}_{j,t}\mathbf{F}_{j,t}/q_{j,t} \in \mathbb{R}^{p_j}$ is the adaptive coefficient vector, and $z_{j,t} = (r_{j,t} + e_{j,t}^2/q_{j,t})/n_{j,t} > 0$ is the volatility update factor (see Table 10.4 of West and Harrison [1997]). The step-ahead priors $(\boldsymbol{\theta}_{j,t+1}, \lambda_{j,t+1} \mid \mathcal{D}_t)$ at time t follow from the system equations (2.8, 2.9) as evolutions of the posterior states $(\boldsymbol{\theta}_{j,t}, \lambda_{j,t} \mid \mathcal{D}_t)$. The normal-gamma step-ahead prior density $\pi_{j,t+1}$ is as in Eq. 2.10 with t evolved to $t+1$ and parameters $r_{j,t+1} = \beta_j n_{j,t}$, $c_{j,t+1} = s_{j,t}$, $\mathbf{a}_{j,t+1} = \mathbf{G}_{j,t+1}\mathbf{m}_{j,t}$, $\mathbf{R}_{j,t+1} = \mathbf{G}_{j,t+1}\mathbf{C}_{j,t}\mathbf{G}'_{j,t+1} + \mathbf{W}_{j,t+1}$ and $\mathbf{W}_{j,t+1} = \frac{1-\delta_j}{\delta_j}\mathbf{G}_{j,t+1}\mathbf{C}_{j,t}\mathbf{G}'_{j,t+1}$. The discount factors $\beta_j, \delta_j \in (0,1)$ inflate the prior variances in the state evolution steps and determine the model's responsiveness to new observations.

Forecasting. The forecast distribution of $y_{j,t+1}$ at time t and with information $\mathcal{D}_t$ follows as a non-standardized Student's t distribution $T_{\text{non std}}(\nu, \mu, \sigma^2)$ with $\nu = r_{j,t+1}$ degrees of freedom, mean $\mu = \mathbf{F}'_{j,t+1}\mathbf{a}_{t+1}$ and variance $\sigma^2 = \mathbf{F}'_{j,t+1}\mathbf{R}_{j,t+1}\mathbf{F}_{j,t+1} + c_{j,t+1}$ by integration of the observation equation (2.7) over the prior distributions of the states $(\boldsymbol{\theta}_{j,t+1}, \lambda_{j,t+1})$. The non-standardized t distribution is a location-scale transformation

$$T_{\text{non std}}(\nu, \mu, \sigma^2) = \mu + \sqrt{\sigma^2} \cdot T_\nu \tag{2.12}$$

of a t distribution T_ν with ν degrees of freedom. In the remainder, we will denote the forecast distribution of $y_{j,t+1}$ at time t by $T_{j,t+1}$.

Model choice. We use a local-level DLM that assumes $\mathbf{F}_{j,t} = 1$ and has random walk evolutions $\mathbf{G}_{j,t} = 1$ for all j and t. The discount factors are set to $\beta_j = 0.96$ and $\delta_j = 0.975$ for all j to balance responsiveness to new observations with sufficient robustness for reliable forecasts. We start the analysis with the initial prior parameters of each series $j = 1{:}9$ set to $\mathbf{a}_{j,1} = 0$, $\mathbf{R}_{j,1} = 10^{-6}$, $r_{j,1} = 10$, $c_{j,1} = 10^{-5}$. Figure 2.3 shows the sequential step-ahead forecasts and realized daily log returns as well as the 10% quantiles of the forecast distributions as the daily value at risk of each series $j = 1{:}9$.

2.3.3 Estimation of the Dependence Models

Copula modeling is a two-step process: first, marginal models remove within-series effects from the data $y_{j,t}$ to obtain i.i.d.—within each series j—uniform noise $u_{j,t} := T_{j,t}(y_{j,t})$; second, a copula is selected to describe across-series dependence effects of the multivariate transformed $U(0,1)$ data $\mathbf{u}_t = (u_{1,t}, \ldots, u_{9,t})'$, $t = 1, 2, \ldots$.

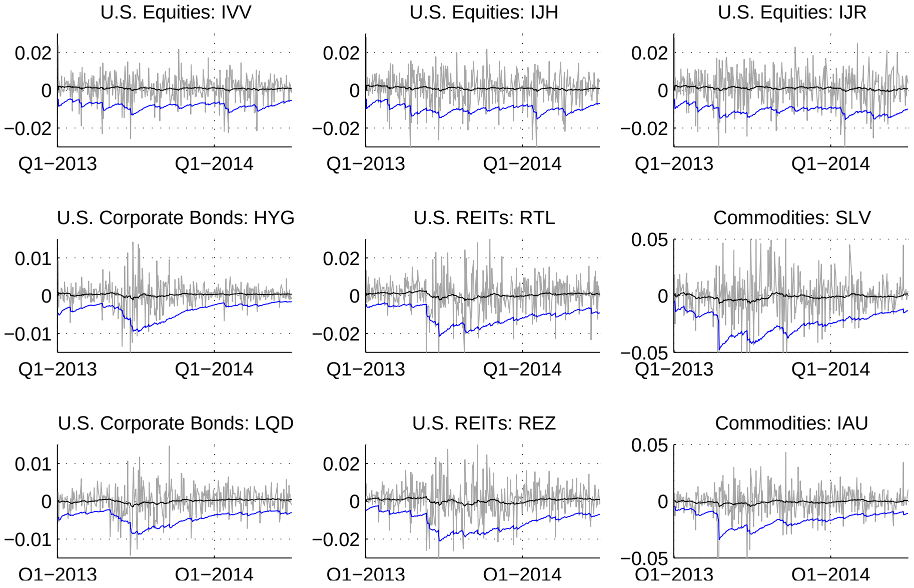


Figure 2.3: Realized returns (gray), forecast means (black) and 10% value at risk (blue).

Tree T_1	Tree T_2	Tree T_3	Tree T_4
$c_{1,2}$ N(0.74)	$c_{1,3;2}$ N(0.07)	$c_{1,9;5,7}$ I	$c_{1,8;6,7,9}$ G180(0.06)
$c_{1,4}$ N(0.46)	$c_{1,5;7}$ G270(-0.10)	$c_{2,6;1,7}$ G180(0.08)	$c_{2,9;1,6,7}$ I
$c_{1,7}$ N(0.39)	$c_{1,6;7}$ N(0.14)	$c_{3,7;1,2}$ I	$c_{3,6;1,2,7}$ I
$c_{2,3}$ N(0.79)	$c_{2,7;1}$ G(0.12)	$c_{4,5;1,7}$ N(0.27)	$c_{4,6;1,5,7}$ I
$c_{5,7}$ G(0.19)	$c_{4,7;1}$ G180(0.13)	$c_{5,6;1,7}$ N(0.12)	$c_{5,9;1,6,7}$ G(0.12)
$c_{6,7}$ N(0.51)	$c_{6,8;9}$ I	$c_{7,8;6,9}$ I	
$c_{6,8}$ G(0.09)	$c_{7,9;6}$ I		
$c_{8,9}$ N(0.71)			

Table 2.6: Sequential Bayesian estimate of the regular vine copula, given the training data $t = 1{:}252$. This tables shows the Kendall's τ parameters of the pair copulas.

Sequential Bayesian selection. We use our model selection scheme to estimate a 9-dimensional regular vine copula using the $t = 1{:}252$ observations from 2013. We apply the same priors and configuration of the sample as in Section 2.2. The selected model is shown in Table 2.6: it has 10 Gaussian pair copulas, 0 Student's t copulas, 8 Gumbel copulas, 0 Clayton copulas and 18 Independence copulas. The pair copulas of levels $k \geq 5$ are selected as Independence copulas and omitted in Table 2.6.

Dißmann's frequentist selection. We compare against our Bayesian tree-by-tree strategy to Dißmann et al. [2013]'s frequentist heuristic as we did in the simulation study of Section 2.2. Dißmann's vine copula is noticeably less parsimonious with only 13 Independence pair copulas and 5 Gaussian copulas, 7 Student's t copulas, 7 Gumbel copulas and 4 Clayton copulas. The selected model is shown in Table 2.7.

Multivariate Gaussian copula. For reference, we also included a maximum likelihood estimate of the multivariate Gaussian copula in our comparison. The estimated correlation matrix is shown in Table 2.8.

2.3.4 Analysis of Portfolio Forecasts

Sampling from the joint forecast distribution. Samples $\hat{\mathbf{y}}_t^{n=1:N} = (\hat{y}_{1,t}^n, \ldots, \hat{y}_{9,t}^n)'$ from the joint forecast distribution are generated by transforming samples $\mathbf{u}^n = (u_1^n, \ldots, u_9^n)'$ from the copula to the observation scale through inverse cdfs of the marginal forecast distributions, $\hat{y}_{j,t}^n := T_{j,t}^{-1}(u_j^n)$.

Tree T_1	Tree T_2	Tree T_3
$c_{1,2}$ N(0.74)	$c_{1,5;4}$ T(-0.18, 12.8)	$c_{1,3;2,7}$ N(0.06)
$c_{1,4}$ N(0.46)	$c_{1,7;2}$ G180(0.08)	$c_{1,9;4,5}$ G180(0.03)
$c_{2,3}$ T(0.79, 8.67)	$c_{2,4;1}$ I	$c_{2,5;1,4}$ I
$c_{2,7}$ T(0.41, 8.58)	$c_{2,6;7}$ N(0.16)	$c_{3,6;2,7}$ I
$c_{4,5}$ T(0.39, 4.93)	$c_{3,7;2}$ G270(-0.05)	$c_{4,7;1,2}$ G180(0.13)
$c_{5,9}$ G(0.16)	$c_{4,9;5}$ C(0.05)	$c_{4,8;5,9}$ C180(0.05)
$c_{6,7}$ T(0.51, 14.5)	$c_{5,8;9}$ C270(-0.05)	
$c_{8,9}$ N(0.70)		

Tree T_4	Tree T_5	Tree T_6
$c_{1,6;2,3,7}$ I	$c_{2,8;1,4,5,9}$ T(-0.04, 16.7)	$c_{3,9;1,2,4,5,7}$ I
$c_{1,8;4,5,9}$ G(0.06)	$c_{2,5;1,2,4,7}$ I	$c_{5,6;1,2,3,4,7}$ T(0.10, 14.3)
$c_{2,9;1,4,5}$ C(0.04)	$c_{4,6;1,2,3,7}$ I	$c_{7,8;1,2,4,5,9}$ I
$c_{3,4;1,2,7}$ I	$c_{7,9;1,2,4,5}$ I	
$c_{5,7;1,2,4}$ G(0.17)		

Table 2.7: Regular vine copula selected by Dißmann's heuristic, given the training data $t = 1{:}252$. This tables shows the Kendall's τ parameters of the pair copulas.

$$\Sigma = \begin{pmatrix} \mathbf{1} & 0.92 & 0.89 & 0.64 & 0.02 & 0.58 & 0.61 & 0.21 & 0.16 \\ 0.92 & \mathbf{1} & 0.95 & 0.61 & 0.03 & 0.59 & 0.61 & 0.21 & 0.17 \\ 0.89 & 0.95 & \mathbf{1} & 0.58 & 0.00 & 0.54 & 0.57 & 0.20 & 0.15 \\ 0.64 & 0.61 & 0.58 & \mathbf{1} & 0.37 & 0.53 & 0.57 & 0.23 & 0.23 \\ 0.02 & 0.03 & 0.00 & 0.37 & \mathbf{1} & 0.32 & 0.32 & 0.14 & 0.20 \\ 0.58 & 0.59 & 0.54 & 0.53 & 0.32 & \mathbf{1} & 0.74 & 0.20 & 0.18 \\ 0.61 & 0.61 & 0.57 & 0.57 & 0.32 & 0.74 & \mathbf{1} & 0.18 & 0.16 \\ 0.21 & 0.21 & 0.20 & 0.23 & 0.14 & 0.20 & 0.18 & \mathbf{1} & 0.90 \\ 0.16 & 0.17 & 0.15 & 0.23 & 0.20 & 0.18 & 0.16 & 0.90 & \mathbf{1} \end{pmatrix}$$

Table 2.8: Maximum likelihood estimate of the correlation matrix of the multivariate Gaussian copula, given the training data $t = 1{:}252$.

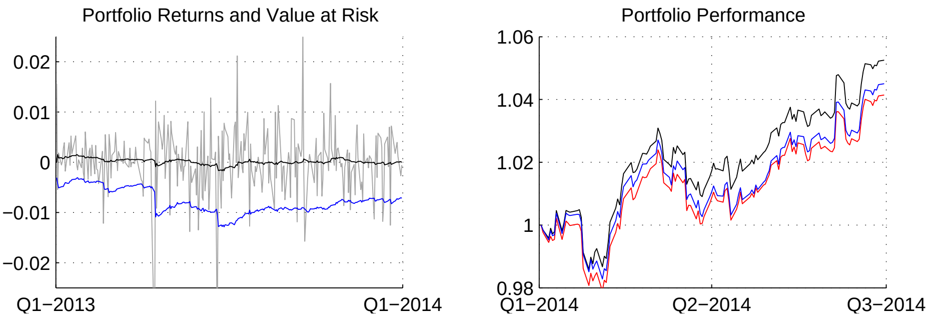


Figure 2.4: **Left:** realized portfolio returns (gray) vs. forecast means (black) and 10% value at risk (blue) from our sequential Bayesian vine copula model during the training date range $t = 1{:}252$. **Right:** Portfolio performance of investment strategy (2.13) under our Bayesian vine copula model (black; highest line), Dißmann's vine model (red; lowest line) and the Gaussian model (blue; middle line) during the test date range $t = 253{:}376$.

In-sample analysis. Consider a portfolio that invests equally in the ETFs from Table 2.5 and the weights $\mathbf{w}_t := (w_{1,t}, \ldots, w_{9,t})' = (\frac{1}{9}, \ldots, \frac{1}{9})'$ are maintained throughout the time period $t = 1{:}252$. For each t, we draw $N = 10{,}000$ samples $\hat{\mathbf{y}}_t^{n=1:N}$ from the joint forecast distribution to simulate the portfolio returns $\hat{r}_t^n := \mathbf{w}_t'\hat{\mathbf{y}}_t^n$. We compute the 10% value at risk as the 10% sample quantile and the expected portfolio return as the sample mean of $\hat{r}_t^{n=1:N}$. This allows an evaluation of the adequacy of the joint multivariate model, which consists of the nine marginal DLMs and the selected copula.

Figure 2.4 (left) shows that the predicted quantities from our sequential Bayesian vine copula model are in line with the actual portfolio returns. The actual portfolio return is under the predicted 10% quantile of the simulated portfolio return distribution that uses our sequential Bayesian vine copula on 8.7%, or 22 out of 252 days; if Dißmann's vine copula or the multivariate Gaussian copula are used in conjunction with the same marginal DLMs, the actual portfolio return is under the predicted 10% quantile on 8.3%, or 21 out of 252 days; if the independence copula is used, the 10% value at risk is exceeded 20%, or 50 out of 252 days.

Out-of-sample analysis. During the out-of-sample period from January through June 2014, $t = 253:376$, we investigate the performance of a dynamic portfolio whose weights $\mathbf{w}_t$ are updated daily to maximize the predicted Sharpe ratio [Sharpe, 1966]:

$$\max_{\mathbf{w}_t} \widehat{SR}_t(\mathbf{w}_t) \text{ subject to } \sum_{j=1:9} w_{j,t} = 1 \text{ and } w_{j,t} \in (0.05, 0.25) \text{ for all } j = 1{:}9. \tag{2.13}$$

Here $\widehat{SR}_t(\mathbf{w}_t)$ (Eq. 2.14) is the estimate of the annualized Sharpe ratio of a portfolio with investment weights $\mathbf{w}_t$, where $\hat{\boldsymbol{\mu}}_t$ is the sample mean and $\widehat{\Sigma}_t$ is the sample covariance matrix of the simulated joint forecasts $\hat{\mathbf{y}}_t^{n=1:N}$. Again, $N = 10{,}000$ samples were used.

$$\widehat{SR}_t(\mathbf{w}_t) := \frac{252 \cdot \mathbf{w}_t'\hat{\boldsymbol{\mu}}_t}{\sqrt{252 \cdot \mathbf{w}_t'\widehat{\Sigma}_t\mathbf{w}_t}}. \tag{2.14}$$

Our calculation of the Sharpe ratio in Eq. 2.14 is under the assumption of a zero-return risk-free asset. The optimization constraints $w_{j,t} \in (0.05, 0.25)$ in Eq. 2.13 refer to minimum and maximum weights of individual assets and are typical restrictions that aim at protecting investors from undue accumulation of risk.

When the regular vine copula selected by our sequential Bayesian procedure is used as the joint model's dependence model, the realized annualized Sharpe ratio of the investment strategy (2.13) during the out-of-sample period $t = 253{:}376$ is $SR = 1.95$; if the multivariate Gaussian copula is used to inform the investment decisions, the realized Sharpe ratio is $SR = 1.67$; if Dißmann's frequentist vine model is used, the realized Sharpe ratio is $SR = 1.53$. In addition, the realized nominal return of the portfolio driven by our sequential Bayesian vine copula is higher than the returns of the portfolios using the Gaussian copula or Dißmann's copula (Figure 2.4 (right)). This example provides additional evidence of our sequential Bayesian vine model as the most reliable model for use in a real-life context.

2.3.5 Considerations on Use in Practice

The computing time of updating the univariate DLMs is a few milliseconds, which is negligible in the context of daily portfolio risk rebalancing. In contrast, the computational burden of estimating the regular vine copula is much higher—in our example, our Bayesian model selection could take as long as up to a day to complete.

We suggest that the dependence model be update in monthly, quarterly, or semi-annually intervals only. Under the assumption that the dependence structure of financial asset returns is only slowly changing, this is a prudent way to proceed. Even though the long computation time of our Bayesian strategy might be a deterrent to implementing our approach, the benefits of increasing a portfolio's performance in nominal as well as risk-adjusted terms will quickly pay for the investment in computing time.

The combination of univariate DLMs with a regular vine copula as the dependence model provides a robust framework for forecasting, yet is highly responsive to new observations. This can be seen, for example, in the way the value at risk changes instantly on the day of a large negative market move. Furthermore, it is a distinct strength of regular vine copulas to be able to model asymmetric dependence characteristics along with various tail dependence characteristics in one model. Our example shows that a regular vine copula-driven model can help in decision making to achieve superior investment performance as well as improved risk forecasts.

2.4 Concluding Remarks

We discussed a Bayesian approach to model selection of regular vine copulas and presented a reversible jump Markov chain Monte Carlo-based algorithm to facilitate posterior sampling. A key feature of our approach, sequential model selection in the levels k reduces the search space for candidate models to a fraction of the search space for joint selection to keep the computational run time at an acceptable level.

A simulation study (Section 2.2) demonstrated that our Bayesian model selection approach is superior to Dißmann et al. [2013]'s frequentist one. The better performance of our Bayesian selection scheme can be attributed to its simultaneous and prior-informed selection of the pair copula families $\mathcal{B}_k$ of a given level k, while Dißmann's algorithm selects them one-by-one. In addition to the simulation study, a real data example (Section 2.3) illustrated how regular vine copulas can be used to achieve superior portfolio risk forecasts and investment decisions.

Our estimation procedure extends previously available inference methods for regular vine copulas in two significant ways. Our Bayesian tree-by-tree strategy allows the selection of the pair copula families $\mathcal{B}_\mathcal{V}$ from an arbitrary set of candidate families $\mathbf{B}$, which is a non-trivial extension of Smith et al. [2010]'s indicator-based approach that can only detect (conditional) pairwise independencies. Furthermore, we present the first Bayesian inference method for selecting the regular vine $\mathcal{V}$ as the building plan of the pair copula construction jointly with the pair copula families $\mathcal{B}_\mathcal{V}$. A major selling point of our approach is that we demonstrated its superiority to existing procedures in a simulation study under controlled conditions (see Section 2.2) as well as in an application study using real data (see Section 2.3).

Sequential model selection schemes can fail to select the correct model. This is illustrated, e.g., in Section 2.2 by the failure of the selected models to have relative log likelihoods close to 100% in Scenarios 1 and 2. Current research aims at developing a fully Bayesian model selection scheme to estimate all levels of a regular vine copula jointly as well as allowing for time-varying dependence effects.

Chapter 3

Simultaneous Bayesian Selection

The contents of this chapter have been submitted for publication in Gruber and Czado [2015a]. This chapter is a lightly edited reproduction of selected contents of the submitted manuscript.

3.1 Posterior Sampling Using 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 [2015b]. Our approach is an evolution of Gruber and Czado [2015b]'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 $\mathcal{V}(\mathcal{B}_\mathcal{V}(\cdot))$. The parameter space is the set of all valid parameters $\boldsymbol{\theta}_\mathcal{V}$ of the vine copula's pair copulas.

Reversible jump MCMC updates. 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 tuned between-model updates to achieve satisfactory exploration of the model space. Our between-model moves are detailed in Sections 3.2 and 3.3. Algorithm 3.1 shows the general reversible jump MCMC mechanism for model selection.

Pseudo Code
Algorithm 3.1 (Reversible Jump MCMC for Model Selection)**.**

```
Select starting values: set the regular vine
    $\mathcal{V}^0 = (T_1^0 = (N_1^0, E_1^0), \ldots, T_{d-1} = (N_{d-1}^0 = E_{d-2}^0, E_{d-1}^0))$
  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 $e \in E_1^0, \ldots, E_{d-1}^0$.
for each MCMC iteration $r = 1, \ldots, R$ do
    Perform a within-model move: update the parameters $\boldsymbol{\theta}_\mathcal{V}$. Obtain updated parameters $\boldsymbol{\theta}_\mathcal{V}^{r,NEW}$ through
    a Metropolis-Hastings step with random walk proposals from a mixture of normal distributions:
        $\mathcal{C}^r = (\mathcal{V}^r, \mathcal{B}_\mathcal{V}^r, \boldsymbol{\theta}_\mathcal{V}^r) := (\mathcal{V}^{r-1}, \mathcal{B}_\mathcal{V}^{r-1}, \boldsymbol{\theta}_\mathcal{V}^{r,NEW})$.
    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 $\boldsymbol{\theta}_\mathcal{V}$ to $\mathcal{V}^{r,NEW}$, $\mathcal{B}_\mathcal{V}^{r,NEW}$ and $\boldsymbol{\theta}_\mathcal{V}^{r,NEW}$ (Algorithms 3.2, 3.3):
        $\mathcal{C}^r = (\mathcal{V}^r, \mathcal{B}_\mathcal{V}^r, \boldsymbol{\theta}_\mathcal{V}^r) := (\mathcal{V}^{r,NEW}, \mathcal{B}_\mathcal{V}^{r,NEW}, \boldsymbol{\theta}_\mathcal{V}^{r,NEW})$.
```

```
5: end for
6: return the Bayesian posterior sample
```

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

Notation. The following assumes the selection of an n-dimensional regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \boldsymbol{\theta}_{\mathcal{V}})$. 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 will be denoted by q and priors are denoted by π. Subscripts may be used to detail the affiliation of these quantities.

3.2 Selection of the Pair Copula Families

Our Bayesian selection strategy extends Min and Czado [2011]'s, which can only select the pair copulas of D-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.

3.2.1 Priors

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

$$\pi(\mathcal{B}_{\mathcal{V}}) \propto \exp(-\lambda d_{\mathcal{B}_{\mathcal{V}}}), \tag{3.1}$$

$$\pi(\boldsymbol{\theta}_e \mid \mathcal{B}_{\mathcal{V}}) \propto \begin{cases} Uniform_{(-1,1)}(\tau_e) \text{ if } \mathcal{B}_e \text{ is a one-parameter copula} \\ Uniform_{(-1,1)}(\tau_e) \cdot \frac{\mathbf{1}_{(1,30)}(\nu_e)\cdot\log(\nu_e)}{\int_1^{30}\log(x)dx} \text{ if } \mathcal{B}_e \text{ is a t copula} \end{cases}, \tag{3.2}$$

where $d_{\mathcal{B}_{\mathcal{V}}}$ denotes the number of parameters of the regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \boldsymbol{\theta}_{\mathcal{V}})$ or, equivalently, the dimension of the parameter vector $\boldsymbol{\theta}_{\mathcal{V}}$. The prior on the parameters $\boldsymbol{\theta}_{\mathcal{V}}$ 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 $\pi(\boldsymbol{\theta}_e \mid \mathcal{B}_{\mathcal{V}})$ 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 optimum; while $\lambda > 1$ will provide even stronger shrinkage.

3.2.2 Between-Model Move to Update $(\mathcal{B}_{\mathcal{V}}, \boldsymbol{\theta}_{\mathcal{V}})$

This between-model move for our reversible jump MCMC sampler (Algorithm 3.1) updates the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\boldsymbol{\theta}_{\mathcal{V}}$. 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 3.2) and an acceptance/rejection step (Line 14).

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 $\mathcal{B}^*_e \in \mathbf{B} \setminus \mathcal{B}^r_e$ (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 $\mathcal{B}^*_e \in \mathbf{B} \setminus \mathcal{B}^r_e$ when their parameters $\boldsymbol{\theta}_{e;\mathcal{B}^*_e}$ are chosen to match the current copula's Kendall's τ and tail-dependence coefficients λ^L or λ^U; these matched, or augmented, parameters are denoted by

$\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}$. We align the proposal distribution closely with the posterior by making the proposal weights of each candidate family proportional to its likelihood (Line 6):

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

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): $\frac{\min_{\mathcal{B}\in\mathbf{B}} q_{\mathcal{B}}(\mathcal{B})}{\max_{\mathcal{B}\in\mathbf{B}} q_{\mathcal{B}}(\mathcal{B})} \geq \kappa$.

After the selected pair copulas' proposal families are drawn from these proposal distributions, a new parameter vector $\boldsymbol{\theta}^*_{e;\mathcal{B}_e^*}$ is proposed for every pair copula of the regular vine copula. For pair copulas $e \in E$ whose families were changed, new parameters $\boldsymbol{\theta}^*_{e;\mathcal{B}_e^*}$ come from a normal mixture distribution centered at the matched, or augmented, parameter $\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}$; for all other pair copulas $e \notin E$, the new parameters $\boldsymbol{\theta}^*_{e;\mathcal{B}_e^*}$ come from a normal mixture distribution centered at the current parameters' values $\boldsymbol{\theta}^r_{e;\mathcal{B}_e}$ (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 $\phi_{(\boldsymbol{\theta}^*_{e;\mathcal{B}_e^r},\Sigma_i)}(\boldsymbol{\theta}^r_{e;\mathcal{B}_e^r})$ in the numerator of the acceptance probability (3.3) high, especially when the current state $\boldsymbol{\theta}^r_{e;\mathcal{B}_e^r}$ of the sampling chain is not close to the proposal mean $\boldsymbol{\theta}^*_{e;\mathcal{B}_e^r}$. 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 $\mathcal{C}^* := (\mathcal{V}^r, \mathcal{B}^*_{\mathcal{V}}, \boldsymbol{\theta}^*_{\mathcal{V}})$ to ensure that the posterior distribution is the equilibrium distribution of the sampling chain (Line 14),

$$\begin{aligned}\alpha =& \frac{L(\mathcal{V}^r, \mathcal{B}^*_{\mathcal{V}}, \boldsymbol{\theta}^*_{\mathcal{V}} \mid \mathbf{U})}{L(\mathcal{V}^r, \mathcal{B}^r_{\mathcal{V}}, \boldsymbol{\theta}^r_{\mathcal{V}} \mid \mathbf{U})} \cdot \frac{\pi(\mathcal{V}^r, \mathcal{B}^*_{\mathcal{V}}, \boldsymbol{\theta}^*_{\mathcal{V}})}{\pi(\mathcal{V}^r, \mathcal{B}^r_{\mathcal{V}}, \boldsymbol{\theta}^r_{\mathcal{V}})} \cdot \prod_{e\in E} \frac{q_{\mathcal{B}}(\mathcal{B}_e^* \to \mathcal{B}_e^r)}{q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*)} \\ &\cdot \sum_i \omega_i \left(\prod_{e\in E} \frac{\phi_{(\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^r},\Sigma_i)}(\boldsymbol{\theta}^r_{e;\mathcal{B}_e^r})}{\phi_{(\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*},\Sigma_i)}(\boldsymbol{\theta}^*_{e;\mathcal{B}_e^*})} \cdot \prod_{e\notin E} \frac{\phi_{(\boldsymbol{\theta}^*_{e;\mathcal{B}_e^r},\Sigma_i)}(\boldsymbol{\theta}^r_{e;\mathcal{B}_e^r})}{\phi_{(\boldsymbol{\theta}^r_{e;\mathcal{B}_e^r},\Sigma_i)}(\boldsymbol{\theta}^*_{e;\mathcal{B}_e^r})} \right). \end{aligned} \tag{3.3}$$

This representation of the acceptance probability uses the likelihood times prior proportionality of the posterior density. We write $\phi_{(\boldsymbol{\mu},\Sigma)}(\cdot)$ for the density of the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. Equation (3.3) accounts for any birth/death moves by implicitly shrinking or expanding the interpretation of $\phi_{(\boldsymbol{\mu},\Sigma)}(\cdot)$ as the density function of a variable-dimension normal distribution.

Pseudo Code

Algorithm 3.2 (Update of the Pair Copula Families $\mathcal{B}_{\mathcal{V}}$).

```
1: Select how many pair copulas are updated: $N \in \{1, \dots, \frac{d(d-1)}{2}\}$.
2: Select which pair copulas are updated: $E \subseteq \cup_{k=1,\dots,d-1} E_k^r$ with $|E| = N$. Denote the corresponding
   pair copula families by $\mathcal{B}_E$ and the corresponding parameters by $\boldsymbol{\theta}_E$.
3: Select an active parameter proposal variance: $\Sigma = \Sigma_i$ with probability $\omega_i$.
4: for each selected pair copula $e \in E$ do
5:     Calculate the likelihoods $L(\mathcal{B}_e, \widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e} \mid \mathbf{U})$ for each candidate pair copula family $\mathcal{B}_e \in \mathbf{B} \setminus \mathcal{B}_e^r$.
6:     Propose a new copula family $\mathcal{B}_e^* \in \mathbf{B} \setminus \mathcal{B}_e^r$ from the proposal distribution

           $$q_{\mathcal{B}}(\mathcal{B}_e^r \to \mathcal{B}_e^*) \propto L(\mathcal{B}_e^*, \widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*} \mid \mathbf{U}).$$

7:     Re-weight the proposal probabilities so that

           $$\frac{\min_{\mathcal{B}\in\mathbf{B}} q_{\mathcal{B}}(\mathcal{B})}{\max_{\mathcal{B}\in\mathbf{B}} q_{\mathcal{B}}(\mathcal{B})} \geq \kappa.$$

8:     Propose a new parameter vector $\boldsymbol{\theta}^*_{e;\mathcal{B}_e^*} \sim \mathcal{N}(\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}, \Sigma)$ from a normal distribution centered at $\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e^*}$.
```

Algorithm	Parameter
3.2	$p(N = k) = \frac{1}{4} \log \left(1 - \frac{1-e^{-4}}{N_{\mathcal{V}} e^{-4} + k(1-e^{-4})}\right)$, where $N_{\mathcal{V}}$ is the number of pair copulas of the regular vine copula
3.2 and 3.3	$\kappa = 0.05$
3.2 and 3.3	$\omega_1 = 0.9$; $\Sigma_1 = 0.003^2$ 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 $\Sigma_2 = 10^2 \Sigma_1$
3.3	$q_T(T_K^r \to T_K^*) \propto \prod_{e \in E_K^*} (\delta + \lvert\tau_e\rvert)$ with $\delta = 0.2$; $q_T(T_k^r \to T_k^*) = \frac{1}{\lvert\mathbf{STP}_k\rvert}$ for $k > K$

Table 3.1: MCMC tuning parameters for Algorithms 3.2 and 3.3.

```
end for
for each not-to-be-changed pair copula $e \notin E$ do
    Leave the pair copula family $\mathcal{B}_e^r$ unchanged; set $\mathcal{B}_e^* = \mathcal{B}_e^r$.
    Propose a new parameter vector $\boldsymbol{\theta}_{e;\mathcal{B}_e^r}^* \sim \mathcal{N}(\boldsymbol{\theta}_{e;\mathcal{B}_e^r}, \Sigma_i)$ from a normal distribution centered at the
    current parameter vector $\boldsymbol{\theta}_{e;\mathcal{B}_e^r}^r$.
end for
Accept the proposal $\mathcal{C}^* = (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^*)$ and set
    $\mathcal{C}^{r,NEW} = (\mathcal{V}^{r,NEW}, \mathcal{B}_{\mathcal{V}}^{r,NEW}, \boldsymbol{\theta}_{\mathcal{V}}^{r,NEW}) := (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^*, \boldsymbol{\theta}_{\mathcal{V}}^*)$
with probability $\alpha$ (Equation (3.3)). If rejected, set
    $\mathcal{C}^{r,NEW} = (\mathcal{V}^{r,NEW}, \mathcal{B}_{\mathcal{V}}^{r,NEW}, \boldsymbol{\theta}_{\mathcal{V}}^{r,NEW}) := (\mathcal{V}^r, \mathcal{B}_{\mathcal{V}}^r, \boldsymbol{\theta}_{\mathcal{V}}^r)$.
return the updated state $\mathcal{C}^{r,NEW} = (\mathcal{V}^{r,NEW}, \mathcal{B}_{\mathcal{V}}^{r,NEW}, \boldsymbol{\theta}_{\mathcal{V}}^{r,NEW})$.
```

3.2.3 Simulation Study

We generate multiple simulation data sets from different regular vine copulas to apply our Bayesian selection strategy to. Starting values include Gaussian regular vine copulas with the true model's tree structure as well as with the tree structures selected by Dißmann et al. [2013]'s frequentist and Gruber and Czado [2015b]' sequential Bayesian selection methods.

Our reversible jump MCMC sampler was run with the tuning parameters described in Table 3.1. 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.

6-Dimensional Test Data

This analysis re-uses the simulation data sets from Gruber and Czado [2015b], 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 [2015b] selected the regular vine trees and the pair copula families

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 3.2: Summary of the simulation study in 6 dimensions.

sequentially, tree-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 3.11–3.14 of Appendix 3.A.

Table 3.2 summarizes the results of this study, based on the aggregated posterior distribution of the pair copula families across all 100 replications (see Tables 3.11–3.14 of Appendix 3.A). 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 selection method can converge quickly to high posterior density regions, shows good mixing behavior across different models, and identifies sparsity patterns.

Table 3.3 summarizes the results of this study. The data generating models as well as the complete aggregated posterior analysis is shown in Tables 3.15–3.21 of Appendix 3.A. 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) share 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 3.1 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 carefully calibrated 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.

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 I, N, G, C	6	3	1	3	3	5	5
# 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	0	1	1	3	1	0

Table 3.3: Summary of the simulation study in 10 dimensions.

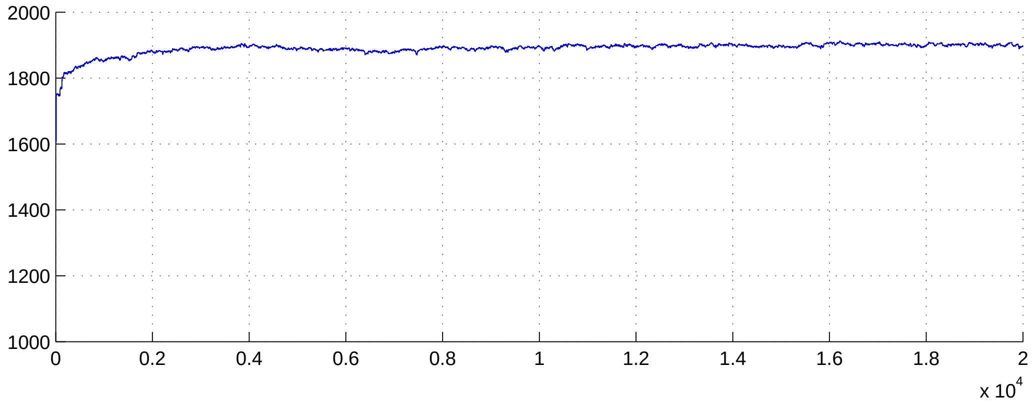


Figure 3.1: Log-likelihood trace plot of replication 1 of Scenario X3-T9.

Discussion

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 $d = 10$.

3.3 Joint Selection of 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 $\mathcal{B}_{\mathcal{V}}$. 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 3.3.3) is a local-search update that builds up the proposal for a new regular vine $\mathcal{V}$ tree-by-tree, starting from the current state vine; the second move (Section 3.3.4) uses importance sampling to draw entire regular vine tree sequences from a weighted distribution of vines.

3.3.1 Differences to Sequential Bayesian Selection

Gruber and Czado [2015b]'s method performs a sequential Bayesian posterior simulation of each level $k = 1, \ldots, d-1$ of a regular vine copula, conditional on already selected states of the previous levels $1, \ldots, k-1$. 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 [2015b], 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 sequential selection (Table 1.3 from [Gruber and Czado, 2015b, Table 1]). Very carefully tuned proposals for between-model moves are required to achieve convergence to desirable models in the enlarged candidate space. Furthermore, the acceptance probabilities of between-model moves are very sensitive to the choice of parameter proposal distributions, given each step updates a large number of parameters.

3.3.2 Priors

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

$$\pi(\mathcal{V}) = discrete\ Uniform(\cdot), \tag{3.4}$$

$$\pi(\mathcal{B}_{\mathcal{V}} \mid \mathcal{V}) \propto \exp(-\lambda d_{\mathcal{B}_{\mathcal{V}}}), \tag{3.5}$$

$$\pi(\boldsymbol{\theta}_e \mid \mathcal{V}, \mathcal{B}_{\mathcal{V}}) \propto \begin{cases} Uniform_{(-1,1)}(\tau_e) \text{ if } \mathcal{B}_e \text{ is a one-parameter copula} \\ Uniform_{(-1,1)}(\tau_e) \cdot \frac{\mathbf{1}_{(1,30)}(\nu_e) \cdot \log(\nu_e)}{\int_1^{30} \log(x) dx} \text{ if } \mathcal{B}_e \text{ is a t copula} \end{cases}, \tag{3.6}$$

where $d_{\mathcal{B}_{\mathcal{V}}}$ denotes the number of parameters of the regular vine copula $\mathcal{C} = (\mathcal{V}, \mathcal{B}_{\mathcal{V}}, \boldsymbol{\theta}_{\mathcal{V}})$ or, equivalently, the dimension of the parameter vector $\boldsymbol{\theta}_{\mathcal{V}}$. Conditionally on the regular vine $\mathcal{V}$, the priors on the pair copulas $\mathcal{B}_{\mathcal{V}}$ and parameters $\boldsymbol{\theta}_{\mathcal{V}}$ are the same as the ones used in Section 3.2.1.

3.3.3 Between-Model Move to Update $(\mathcal{V}, \mathcal{B}_\mathcal{V}, \boldsymbol{\theta}_\mathcal{V})$ (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 3.3); all subsequent trees $T_K, \dots, T_{d-1}$ will have to be adjusted, too, as the proximity condition (Definition 1.1) 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.

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 (3.7), Kendall's τ-weighted proposals (3.8), or random walk proposals (3.9),

$$q_T(T_K^r \to T_K^*) = q_T(T_K^*) \propto 1, \tag{3.7}$$

$$q_T(T_K^r \to T_K^*) = q_T(T_K*) \propto \prod_{e \in E_K^*} (\delta + |\tau_e|), \tag{3.8}$$

$$q_T(T_K^r \to T_K^*) \propto p^{|E_K^* \cap E_K^r|} \cdot (1-p)^{|E_K^* \setminus E_K^r|}. \tag{3.9}$$

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, $q_T(T_k^r \to T_k^*) = q_T(T_k^*) = \frac{1}{|\mathbf{STP}_k|}$ (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 (Equation (3.13)).

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 3.2 (Lines 4–13 in Algorithm 3.2; Lines 10–21 in Algorithm 3.3). 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 $\widetilde{\boldsymbol{\theta}}_{e^*;\mathcal{B}_{e^*}^*}$ 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 copula 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 $\widetilde{\boldsymbol{\theta}}_{e^*;\mathcal{B}_{e^*}^*}$ (Line 15).

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

$$\mathcal{V}^* = (T_1^r, \dots, T_{K-1}^r, T_K^*, \dots, T_{d-1}^*), \tag{3.10}$$

$$\mathcal{B}_\mathcal{V}^* = (\mathcal{B}_1^r, \dots, \mathcal{B}_{K-1}^r, \mathcal{B}_K^*, \dots, \mathcal{B}_{d-1}^*), \text{ and} \tag{3.11}$$

$$\boldsymbol{\theta}_\mathcal{V}^* = (\boldsymbol{\theta}_1^*, \dots, \boldsymbol{\theta}_{d-1}^*). \tag{3.12}$$

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

$$\alpha = \frac{L(\mathcal{V}^*, \mathcal{B}^*_\mathcal{V}, \boldsymbol{\theta}^*_\mathcal{V} \mid \mathbf{U})}{L(\mathcal{V}^r, \mathcal{B}^r_\mathcal{V}, \boldsymbol{\theta}^r_\mathcal{V} \mid \mathbf{U})} \cdot \frac{\pi(\mathcal{V}^*, \mathcal{B}^*_\mathcal{V}, \boldsymbol{\theta}^*_\mathcal{V})}{\pi(\mathcal{V}^r, \mathcal{B}^r_\mathcal{V}, \boldsymbol{\theta}^r_\mathcal{V})} \cdot \prod_{k=K:(d-1)} \frac{q_T(T^*_k \to T^r_k)}{q_T(T^r_k \to T^*_k)} \cdot \prod_{k=K:(d-1)} \frac{\prod_{e \in E^r_k} q_\mathcal{B}(\mathcal{B}^r_e)}{\prod_{e^* \in E^*_k} q_\mathcal{B}(\mathcal{B}^*_{e^*})}$$
$$\cdot \sum_i \omega_i \left(\prod_{k=1:(K-1)} \prod_{e \in E^r_k} \frac{\phi_{(\boldsymbol{\theta}^*_{e;\mathcal{B}^r_e}, \Sigma_i)}(\boldsymbol{\theta}^r_{e;\mathcal{B}^r_e})}{\phi_{(\boldsymbol{\theta}^r_{e;\mathcal{B}^r_e}, \Sigma_i)}(\boldsymbol{\theta}^*_{e;\mathcal{B}^r_e})} \cdot \prod_{k=K:(d-1)} \frac{\prod_{e \in E^r_k} \phi_{(\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}^r_e}, \Sigma_i)}(\boldsymbol{\theta}^r_{e;\mathcal{B}^r_e})}{\prod_{e^* \in E^*_k} \phi_{(\widetilde{\boldsymbol{\theta}}_{e^*;\mathcal{B}^*_{e^*}}, \Sigma_i)}(\boldsymbol{\theta}^*_{e^*;\mathcal{B}^*_{e^*}})} \right). \qquad (3.13)$$

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 $\boldsymbol{\theta}_\mathcal{V}$ as its stationary distribution. Equation (3.13) uses the likelihood times prior proportionality of the posterior density.

Pseudo Code

Algorithm 3.3 (Joint Update of the Regular Vine $\mathcal{V}$ and Pair Copula Families $\mathcal{B}_\mathcal{V}$).

This is for the r-th iteration in Algorithm 3.1.

```
1:  Select the lowest level of the vine that will be updated in this step; denote it by $K$.
2:  for each level $k = K, \dots, d-1$ do
3:    if $k = K$ then
4:      Draw a new spanning tree $T^*_k = (N_k, E^*_k) \in \mathbf{STP}_k \setminus T^r_k$ that satisfies the proximity condition from
        a proposal distribution $q_T(T^r_k \to T^*_k)$.
5:    else
6:      Draw a new spanning tree $T^*_k = (N_k, E^*_k) \in \mathbf{STP}_k$ that satisfies the proximity condition from the
        proposal distribution $q_T(T^r_k \to T^*_k) = q_T(T^*_k) = \frac{1}{|\mathbf{STP}_k|}$.
7:    end if
8:  end for
9:  Select an active parameter proposal variance: $\Sigma = \Sigma_i$ with probability $\omega_i$.
10: for each level $k = K, \dots, d-1$ do
11:   for each pair copula $e^* \in E^*_k$ do
12:     Calculate the likelihoods $L(\mathcal{B}_e, \widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}_e} \mid \mathbf{U})$ for each candidate pair copula family $\mathcal{B}_e \in \mathbf{B}$.
13:     Propose a new copula family $\mathcal{B}^*_e \in \mathbf{B}$ from the proposal distribution

            $$q_\mathcal{B}(\mathcal{B}^*_e) \propto L(\mathcal{B}^*_e, \widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}^*_e} \mid \mathbf{U}).$$

14:     Re-weight the proposal probabilities so that

            $$\frac{\min_{\mathcal{B} \in \mathbf{B}} q_\mathcal{B}(\mathcal{B})}{\max_{\mathcal{B} \in \mathbf{B}} q_\mathcal{B}(\mathcal{B})} \geq \kappa.$$

15:     Propose a new parameter vector $\boldsymbol{\theta}^*_{e;\mathcal{B}^*_e} \sim \mathcal{N}(\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}^*_e}, \Sigma_i)$ from a normal distribution centered at $\widetilde{\boldsymbol{\theta}}_{e;\mathcal{B}^*_e}$.
16:   end for
17: end for
18: for each pair copula $e \in E^r_1 \cup \cdots E^r_{K-1}$ from levels $k = 1, \dots, K-1$ do
19:   Leave the pair copula family $\mathcal{B}^r_e$ unchanged; set $\mathcal{B}^*_e = \mathcal{B}^r_e$.
20:   Propose a new parameter vector $\boldsymbol{\theta}^*_{e;\mathcal{B}^r_e} \sim \mathcal{N}(\boldsymbol{\theta}_{e;\mathcal{B}^r_e}, \Sigma_i)$ from a normal distribution centered at the
      current parameter vector $\boldsymbol{\theta}^r_{e;\mathcal{B}^r_e}$.
21: end for
22: Accept the proposal $\mathcal{C}^* = (\mathcal{V}^*, \mathcal{B}^*_\mathcal{V}, \boldsymbol{\theta}^*_\mathcal{V})$ and set

        $$\mathcal{C}^{r,NEW} = (\mathcal{V}^{r,NEW}, \mathcal{B}^{r,NEW}_\mathcal{V}, \boldsymbol{\theta}^{r,NEW}_\mathcal{V}) := (\mathcal{V}^*, \mathcal{B}^*_\mathcal{V}, \boldsymbol{\theta}^*_\mathcal{V})$$

    with probability $\alpha$ (Equation (3.13)). If rejected, set

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

23: return the updated state $\mathcal{C}^{r,NEW} = (\mathcal{V}^{r,NEW}, \mathcal{B}^{r,NEW}_\mathcal{V}, \boldsymbol{\theta}^{r,NEW}_\mathcal{V})$
```

3.3.4 Between-Model Move to Update $(\mathcal{V}, \mathcal{B}_\mathcal{V}, \boldsymbol{\theta}_\mathcal{V})$ (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 $\boldsymbol{\theta}_\mathcal{V}$ 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 $\mathcal{V}$:

$$S_\mathcal{V} := \sum_{k=1:(d-1)} \sum_{e \in E_k} \tau_e^2. \tag{3.14}$$

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 pairs partial correlation; 4) the correlation parameter Pearson's ρ can be transformed to Kendall's τ; and 5) the expected likelihood of the pair copula families listed in Table 1.1 tends to increase with its strength of association parameter Kendall's τ.

Pre-MCMC importance sampling. Before the start of our reversible jump MCMC sampler (Algorithm 3.1), we generate a large importance sample with different regular vine tree structures $\mathcal{V}$ (Algorithm 3.4). We generate the samples $\mathcal{V}^i$, $i = 1, \ldots, I$, from conditionally uniform tree-by-tree proposal distributions

$$q(\mathcal{V}) = \prod_{k=1:(d-1)} q(T_k | T_1, \ldots, T_{k-1}), \text{ where} \tag{3.15}$$

$$q(T_k | T_1, \ldots, T_{k-1}) = \frac{1}{|\mathbf{STP}_k|}. \tag{3.16}$$

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

Vine proposal step. At iteration r of Algorithm 3.1, a proposal regular vine $\mathcal{V}^*$ is drawn from the importance sample $(\mathcal{V}^i, \alpha^i)$, $i = 1, \ldots, I$ (Algorithm 3.5).

Pseudo Code
Algorithm 3.4 (Importance Sampling for Proposals for $\mathcal{V}$).
Execute this preparatory step before the start of reversible jump MCMC sampling (Algorithm 3.1).

```
Draw $I$ proposals $\mathcal{V}^i$, $i = 1, \ldots, I$, from the tree-wise proposal distribution
    $q(\mathcal{V}) = \prod_{k=1:(d-1)} q(T_k | T_1, \ldots, T_{k-1})$, where
    $q(T_k | T_1, \ldots, T_{k-1}) = \frac{1}{|\mathbf{STP}_k|}$.
Assign each proposal $\mathcal{V}^i$ the importance weight
    $\alpha^i = \frac{S_{\mathcal{V}^i}}{q(\mathcal{V}^i)}$.
```

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 3.4: Average log-likelihoods of the selected models across all 50 replications. The highest log-likelihoods are in bold.

```
3: If a proposal $\mathcal{V}^i$ appears $n_i \geq 2$ times, remove the additional copies and adjust its importance weight to $n_i\alpha^i$.
```

Algorithm 3.5 (Importance Resampling-Based Proposals for $\mathcal{V}$).
The following lines replace lines 1-8 of Algorithm 3.3.

```
1: Set $K = 1$.
2: Draw a new proposal vine $\mathcal{V}^* \sim q_{\mathcal{V}}(\mathcal{V}^i) = \alpha^i$ from the importance sample $(\mathcal{V}^i, \alpha^i)$, $i = 1, \ldots, I$.
```

3.3.5 Simulation Study

We will show empirical evidence that our reversible jump MCMC scheme selects suitable models. We will also 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 sequential frequentist selection and Gruber and Czado [2015b] for sequential Bayesian 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 3.2.3 already evaluated sparsity detection and pair copula family selection of the pair copula family updates, which we will re-use from Section 3.2. 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 $i = 15{,}001, \ldots, 25{,}000$; the quoted log-likelihoods of the sequential selection 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 points.

The between-model updates are generated from a 50%–50% mixture of Algorithms 3.2 and 3.3. This means that in each iteration $r = 1, \ldots, R$, with probability 50%, Algorithm 3.1 will update only the pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\boldsymbol{\theta}_{\mathcal{V}}$ (Algorithm 3.2), or jointly update the regular vine tree structure $\mathcal{V}$, pair copula families $\mathcal{B}_{\mathcal{V}}$ and parameters $\boldsymbol{\theta}_{\mathcal{V}}$ (Algorithm 3.3). Given the increased complexity of this sampling scheme, all analyses are replicated only 50 times instead of 100 times as in Section 3.2.3.

6-Dimensional Test Data

This study uses the same 6-dimensional test data sets used in Section 3.2.3. Table 3.4 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 get stuck there. Considering the vast discrete model space, this is fully expected. The high log-likelihoods of Table 3.4 show that the selected tree structures $\mathcal{V}$ make for suitable regular vine copulas, and suggest that lack of free mixing of regular vines is not detrimental to performance.

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

Table 3.5: Average log-likelihoods of the selected models across all 50 replications. The highest log-likelihoods are in bold.

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 3.2.3, and Table 3.5 compares the log-likelihoods of the selected models from different selection methods. We only compare against Dißmann et al. [2013]'s sequential frequentist selection method, given that Gruber and Czado [2015b] 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. This limits any Bayesian selection scheme to only explore a very small fraction of the entire space in finite time and is likely to push the limits of application of reversible jump MCMC. That said, our carefully-tuned proposals allow quick convergence to high posterior density regions and succeed in selecting more suitable models than Dißmann et al. [2013]'s frequentist strategy in the most complex scenarios (X1-T6, X2-T8, and X3-T9). The models selected by the sequential frequentist 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 these scenarios, all pair copulas on levels greater or equal to 3 (Scenarios X1-T2, X3-T2) or 4 (Scenarios X2-T3, X3-T3) are independence copulas, which benefits sequential selection because there are fewer higher-order levels that will be mis-specified by sequential selection and then lower overall model fit. Furthermore, our shrinkage prior may have affected the selection of sparse models that describe any systematic dependence characteristics without over-fitting.

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.

Discussion

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 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. 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 feature for application in practice.

Our fully Bayesian analysis is substantially faster than Gruber and Czado [2015b]'s sequential Bayesian, which performs a full posterior simulation for each level $k = 1, \ldots, d-1$. In contrast, our fully Bayesian

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 3.6: Details of the selected ETFs.

strategy performs only one posterior simulation. Faster computation and better selection performance make our fully Bayesian strategy the universally superior method.

Our results suggest that, if the regular vine tree structure is unknown, full Bayesian analysis will be most beneficial if there are substantial conditional dependencies in the data. If the variables are mostly conditionally independent, sequential selection methods are likely to perform just as well.

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

We provide a novel, and more extensive analysis of Gruber and Czado [2015b]'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 α% 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 that the "bad $(1 - \alpha\%)$ of scenarios" happens [Acerbi and Tasche, 2003].

3.4.1 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 3.6. 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.

3.4.2 Marginal Models

The daily prices are transformed to daily log-returns. The log-returns of each ETF are modeled by a univariate dynamic linear model (DLM; West and Harrison [1997], Prado and West [2010]).

Model choice. The DLM is a fully Bayesian model for time series data, which makes it particularly appealing to use for financial data. The DLM has time-varying parameters that are updated at every time point t. This extends to the volatility parameters, which, coupled with a suitable time evolution step, will exhibit the same volatility clusters observed in the underlying data. Furthermore, there are closed-form

Initial prior	$r_{j,1} = 10$	$c_{j,t} = 1e-5$
	$a_{j,1} = 0$	$R_{j,1} = 1e-6$
Time t posterior	$e_{j,t} = y_{j,t} - a_{j,t}$	$n_{j,t} = r_{j,t} + 1$
	$s_{j,t} = c_{j,t}(r_{j,t} + e^2_{j,t}/(R_{j,t} + c_{j,t}))/n_{j,t}$	
	$m_{j,t} = a_{j,t} + R_{j,t}/(R_{j,t} + c_{j,t}) \cdot e_{j,t}$	
	$C_{j,t} = (R_{j,t} - R^2_{j,t}/(R_{j,t} + c_{j,t}))s_{j,t}/c_{j,t}$	
Time $t+1$ step-ahead prior	$r_{j,t+1} = \beta n_{j,t}$	$c_{j,t+1} = s_{j,t}$
	$a_{j,t+1} = m_{j,t}$	$R_{j,t+1} = C_{j,t}/\delta$
Discount factors	$\beta = 0.96$	$\delta = 0.975$

Table 3.7: Prior and posterior parameters of the univariate local-trend DLM.

analytical expressions for the parameter updates and observation-level forecasts, which makes learning as well as forecasting extremely fast and efficient.

There are three distinct advantages of using the DLM to describe the marginal data over a GARCH model: learning of the DLM is much faster than estimating a GARCH-model, given the DLM's analytical forward filtering equations; the sequential forward filtering approach to learning, which updates the estimates at every time point, makes the parameter estimates robust to the selection of the time period of the data used for parameter learning; and the DLM allows systematic effects other than historical data to be included in the model as covariates.

Model structure. West and Harrison [1997] and Prado and West [2010] provide extensive literature on the DLM. We provide only a minimum set of key equations here that are required to understand the remainder of this example. Our basic DLM consists of a local trend parameter $\theta_{j,t}$ and innovations $\nu_{j,t} \sim N(0, \lambda_{j,t}^{-1})$ to describe the time t observation $y_{j,t}$ of series $j = 1{:}9$ by

$$y_{j,t} = \theta_{j,t} + \nu_{j,t}.$$

The time t information set $\mathcal{D}_t$ includes all previous observations $y_{j,1}, \ldots, y_{j,t}$. We choose conjugate normal gamma priors for the parameters $\theta_{j,t+1}$ and $\lambda_{j,t+1}$ of the form

$$p(\theta_{j,t+1}, \lambda_{j,t+1} \mid \mathcal{D}_t) = \phi_{(a_{j,t+1}, R_{j,t+1}/(c_{j,t+1}\lambda_{j,t+1}))}(\theta_{j,t+1}) \cdot f_{G(r_{j,t+1}/2, r_{j,t+1}c_{j,t+1}/2)}(\lambda_{j,t+1}),$$

where $f_{G(\alpha,\beta)}(\cdot)$ is the $Gamma(\alpha, \beta)$ density function. The time t posterior follows as

$$p(\theta_{j,t}, \lambda_{j,t} \mid \mathcal{D}_t) = \phi_{(m_{j,t}, C_{j,t}/(s_{j,t}\lambda_{j,t}))}(\theta_{j,t}) \cdot f_{G(n_{j,t}/2, n_{j,t}s_{j,t}/2)}(\lambda_{j,t}).$$

Table 3.7 lists the initial prior parameters as well as the prior and posterior evolutions.

Forecasting. The forecasting distribution of $y_{j,t+1}$ given the information available up to time t, $\mathcal{D}_t$, is a non-standardized t distribution $T_{\text{non std}}(r_{j,t+1}, a_{j,t+1}, R_{j,t+1} + c_{j,t+1})$. A location-scale transformation $T_{\text{non std}}(\nu, \mu, \sigma^2) = \mu + \sqrt{\sigma^2} \cdot T_\nu$ of a t distribution $T_{r_{j,t+1}}$ with $r_{j,t+1}$ degrees of freedom yields the non-standardized version.

Model fit. Figure 3.2 shows the realized daily log-returns versus the forecast mean and 90% value at risk of all series. The forecasts are sequential step-ahead forecasts at each time point t, which do not use any future information.

The trend forecasts show local trends while effectively eliminating daily fluctuations in the time series. The value at risk forecasts react without any delay to outsize market moves. The empirical probabilities of observed returns less than the 90% value at risk forecasts are close to the theoretical value of 10%, but the models tend to provide conservative risk forecasts that slightly over-estimate future volatility. In a last step we transform the realized returns into uniform u-data for copula analysis using the forecast distribution,

$$u_{j,t} = F_{T_{\text{non std}}(r_{j,t}, a_{j,t}, R_{j,t}+c_{j,t})}(y_{j,t}). \tag{3.17}$$

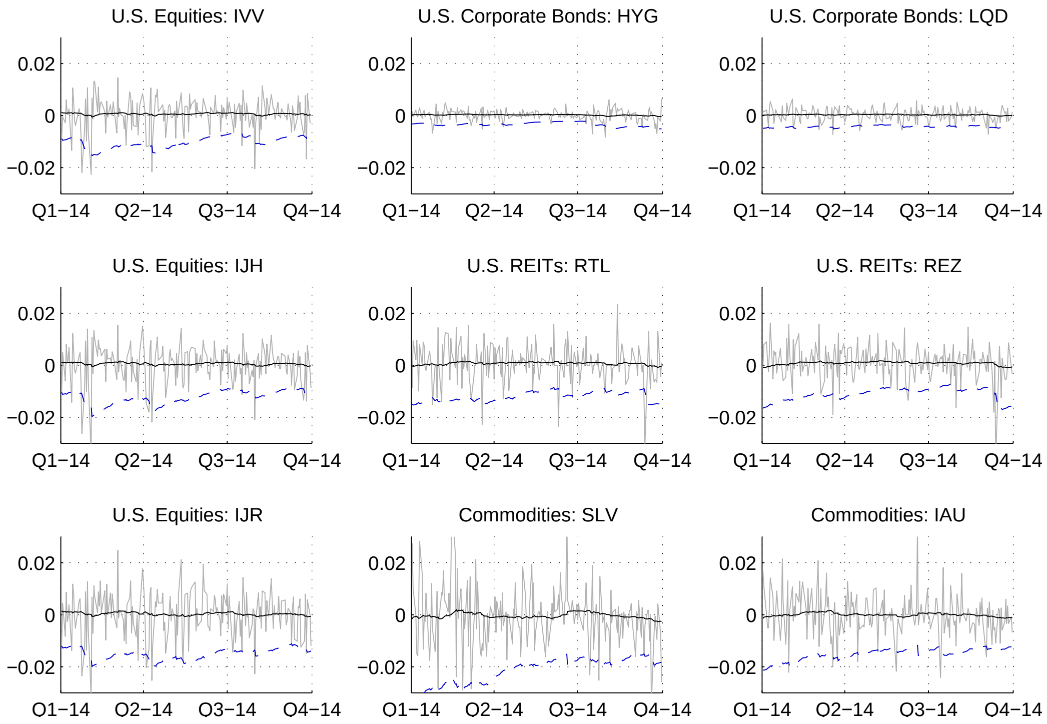


Figure 3.2: Realized log-returns (gray, solid), forecast means (black, solid) and forecast 90% value at risk (blue, dashed) of the nine ETFs.

3.4.3 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 the 2013 data ($t = 1{:}252$) is used to learn the copula dependence model. 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, $t = 253{:}440$, and our forecasting process can be summarized as follows.

```
Select the copula using the $u$-data from 2013 ($t = 1{:}252$).
for each day $t = 253{:}440$ of 2014 do
    Update the univariate DLMs using observation $\mathbf{y}_t$;
    Apply the time evolution step to get the step-ahead priors;
    Sample $k = 1{:}N$ multivariate $\mathbf{u}^k = (u_1^k, \ldots, u_9^k)$-vectors from the posterior-weighted mixture of
    vine copulas;
    Estimate the forecast distribution of $\mathbf{y}_{t+1}$ from the Monte Carlo sample
        $\hat{\mathbf{y}}_{t+1}^k = (T_{\text{non std}}^{-1}(u_1^k; r_{1,t+1}, a_{1,t+1}, R_{1,t+1} + c_{1,t+1}), \ldots, T_{\text{non std}}^{-1}(u_9^k; r_{9,t+1}, a_{9,t+1}, R_{9,t+1} + c_{9,t+1})). \quad (3.18)$
end for
```

Selected copulas. We used Dißmann et al. [2013]'s sequential frequentist selection method, Gruber and Czado [2015b]'s sequential Bayesian method and our three fully Bayesian methods to select the dependence models. The two sequential methods and our to fully Bayesian methods from Section 3.3 select regular vine copulas autonomously; our Bayesian family selection method from Section 3.2 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 D-vine whose first tree T_1 is a path from nodes 1 to 9; this is a very simple

	# 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 3.8: Summary of selected models for the financial data set.

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 3.8 shows summary statistics of the selected models. The models selected by the sequential methods can be found in Tables 7 and 8 of Gruber and Czado [2015b]; Tables 3.22-3.24 of Appendix 3.B 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 [2015b]'s sequential Bayesian method has the most independence pair copulas, and Dißmann et al. [2013]'s sequential frequentist model estimate has the most pair copulas with tail-dependence as well as the most asymmetric pair copulas.

3.4.4 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,

$$\mathbf{w}_t = \underset{\mathbf{w}=(w_1,\ldots,w_9)}{\arg\max} \frac{\mathbf{w}'\widehat{\boldsymbol{\mu}}_t}{\sqrt{\mathbf{w}'\widehat{\boldsymbol{\Sigma}}_t\mathbf{w}_t}} \text{ subject to } \sum_{j=1:9} w_j = 1 \text{ and } 0 \leq w_j \leq 0.25, \tag{3.19}$$

where $\widehat{\boldsymbol{\mu}}_t = (a_{1,t}, \ldots, a_{9,t})$ denotes the forecast returns of the individual ETFs, and $\widehat{\boldsymbol{\Sigma}}_t = \text{diag}((R_{1,t} + c_{1,t}) \cdot r_{1,t}/(r_{1,t} - 2), \ldots, (R_{9,t} + c_{9,t}) \cdot r_{9,t}/(r_{9,t} - 2))$ 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 3.9 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 sequential frequentist and Bayesian methods. Table 3.10 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 sequentially selected models. The forecast errors of all models are very similar and show a mean of about -0.10%. Lastly, Figure 3.3 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 DLMs with a regular vine copula is a strong couple to forecast financial asset returns. The models selected by our fully Bayesian methods yielded superior forecasts than the sequentially 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 these from the sequentially learned models. Our findings confirm that the quantification of model uncertainty through fully Bayesian selection contributes to more reliable risk forecasts.

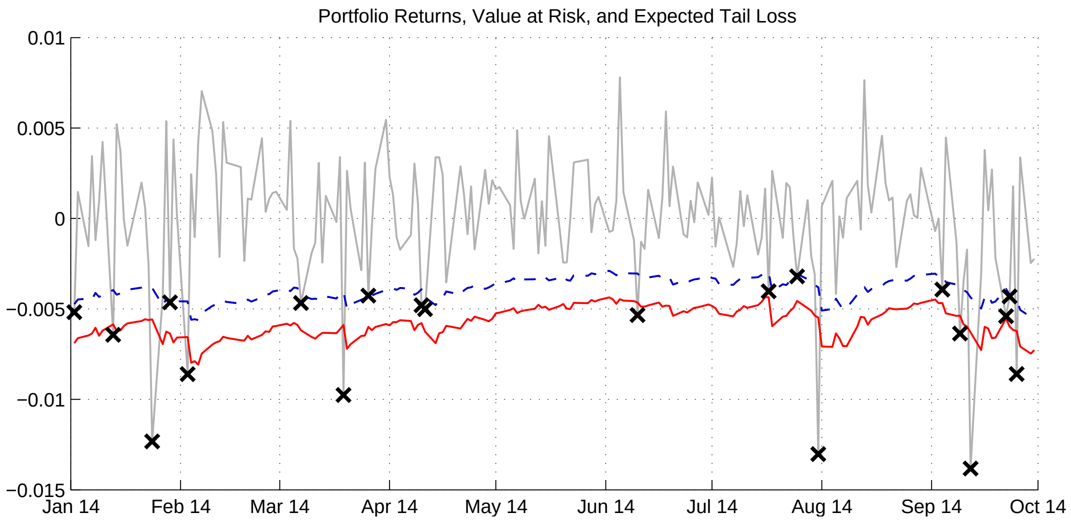


Figure 3.3: Observed portfolio returns (gray, solid), forecast 90% value at risk (blue, dashed) and expected tail loss (red, solid), and observed value at risk hits (black crosses). Forecasts are from our fully Bayesian model (selection method II).

	# 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 3.9: 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 3.10: Forecast expected tail loss (ETL) and average observed losses (OL) under different dependence models. The forecasts with the smallest forecast error are in bold.

3.5 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.

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 a topic of future research.

3.A Selected Models for the Simulation Study

Pair	N	T	G	C	I
1,2	**98.8%**	1.2%	0.0%	0.0%	0.0%
2,3	0.0%	0.0%	0.0%	**100.0%**	0.0%
3,4	0.0%	0.0%	0.0%	**100.0%**	0.0%
3,5	**98.9%**	1.1%	0.0%	0.0%	0.0%
3,6	0.0%	**99.4%**	0.6%	0.0%	0.0%
1,3; 2	0.0%	0.0%	**100.0%**	0.0%	0.0%
2,4; 3	**95.0%**	5.0%	0.0%	0.0%	0.0%
2,5; 3	0.0%	0.0%	0.0%	**100.0%**	0.0%
2,6; 3	**93.7%**	5.3%	0.0%	0.1%	1.0%
1,4; 2,3	0.7%	**90.7%**	7.3%	0.0%	1.3%
1,5; 2,3	**88.4%**	5.6%	3.9%	0.8%	1.3%
1,6; 2,3	0.0%	1.9%	0.2%	**95.7%**	2.2%
4,6; 1,2,3	**49.6%**	4.1%	11.5%	11.6%	23.2%
5,6; 1,2,3	0.7%	5.0%	23.2%	**70.0%**	1.1%
4,5; 1,2,3,6	0.3%	3.8%	**95.5%**	0.1%	0.3%

Table 3.11: Aggregated empirical posterior probabilities of Scenario 1. The original pair copula families are in bold.

Pair	N	T	G	C	I
1,2	0.0%	**100.0%**	0.0%	0.0%	0.0%
1,3	0.0%	0.0%	0.0%	**100.0%**	0.0%
1,4	0.0%	0.0%	0.0%	**100.0%**	0.0%
1,5	**98.0%**	2.0%	0.0%	0.0%	0.0%
1,6	0.0%	**100.0%**	0.0%	0.0%	0.0%
2,3; 1	0.0%	0.3%	**99.7%**	0.0%	0.0%
2,4; 1	0.0%	10.8%	**89.2%**	0.0%	0.0%
2,5; 1	0.0%	0.0%	0.0%	**100.0%**	0.0%
2,6; 1	**92.7%**	7.3%	0.0%	0.0%	0.0%
3,4; 1,2	6.1%	**80.8%**	8.2%	0.3%	4.5%
3,5; 1,2	**88.2%**	7.4%	2.2%	0.4%	1.8%
3,6; 1,2	0.0%	0.2%	0.9%	**98.9%**	0.0%
4,5; 1,2,3	**67.7%**	8.2%	6.5%	2.7%	15.0%
4,6; 1,2,3	0.0%	0.9%	1.2%	**97.9%**	0.0%
5,6; 1,2,3,4	0.0%	2.0%	**96.6%**	1.4%	0.0%

Table 3.12: Aggregated empirical posterior probabilities of Scenario 2. The original pair copula families are in bold.

Pair	N	T	G	C	I
1,2	**89.3%**	8.5%	2.2%	0.0%	0.0%
2,3	0.0%	0.0%	0.2%	**99.8%**	0.0%
3,4	0.0%	0.0%	0.8%	**99.2%**	0.0%
3,5	**90.4%**	7.6%	2.0%	0.0%	0.0%
3,6	7.3%	**86.3%**	6.4%	0.0%	0.0%
1,3; 2	3.1%	0.5%	5.7%	5.5%	**85.2%**
2,4; 3	2.6%	0.0%	4.7%	4.1%	**88.5%**
2,5; 3	1.6%	0.1%	3.5%	3.6%	**91.1%**
2,6; 3	3.0%	0.3%	4.6%	5.6%	**86.6%**
1,4; 2,3	2.1%	0.6%	5.0%	4.2%	**88.2%**
1,5; 2,3	2.7%	0.2%	3.7%	4.5%	**88.9%**
1,6; 2,3	3.2%	0.1%	4.0%	3.8%	**88.9%**
4,6; 1,2,3	2.3%	0.7%	5.2%	4.5%	**87.3%**
5,6; 1,2,3	2.6%	0.2%	3.7%	3.6%	**89.8%**
4,5; 1,2,3,6	2.1%	0.2%	3.3%	3.8%	**90.6%**

Table 3.13: Aggregated empirical posterior probabilities of Scenario 3. The original pair copula families are in bold.

Pair	N	T	G	C	I
1,2	**96.0%**	3.6%	0.0%	0.0%	0.4%
2,3	**96.1%**	3.4%	0.0%	0.0%	0.4%
2,4	**92.6%**	7.2%	0.0%	0.0%	0.2%
3,5	**93.1%**	6.1%	0.7%	0.0%	0.0%
3,6	**81.6%**	5.2%	2.4%	1.5%	9.3%
1,3; 2	**93.8%**	6.2%	0.0%	0.0%	0.0%
2,5; 3	**71.4%**	2.9%	10.0%	3.9%	11.8%
3,4; 2	**91.8%**	7.0%	0.1%	0.0%	1.0%
5,6; 3	**93.8%**	5.7%	0.0%	0.0%	0.5%
1,5; 2,3	**89.6%**	6.4%	2.2%	0.1%	1.7%
2,6; 3,5	**90.4%**	9.0%	0.0%	0.0%	0.6%
4,5; 2,3	**80.1%**	6.8%	8.8%	1.7%	2.6%
1,6; 2,3,5	**92.6%**	7.2%	0.1%	0.0%	0.0%
4,6; 2,3,5	**93.8%**	4.5%	1.6%	0.0%	0.0%
1,4; 2,3,5,6	**86.9%**	5.4%	5.2%	0.2%	2.3%

Table 3.14: Aggregated empirical posterior probabilities of Scenario 4. The original pair copula families are in bold.

Pair	True τ	N	T	G	C	I
0,8	0.67	0.0%	1.5%	**98.5%**	0.0%	0.0%
1,4	0.50	0.1%	0.2%	**90.7%**	9.0%	0.0%
1,5	0.13	0.0%	**16.3%**	20.2%	63.1%	0.3%
2,6	-0.19	2.9%	**39.4%**	57.4%	0.2%	0.0%
3,6	0.40	**91.2%**	4.2%	4.6%	0.0%	0.0%
5,8	-0.50	54.0%	33.9%	**12.2%**	0.0%	0.0%
6,7	0.30	**95.5%**	1.3%	3.1%	0.1%	0.1%
7,8	0.67	0.0%	0.0%	0.0%	**100.0%**	0.0%
7,9	0.50	0.0%	0.0%	8.4%	**89.6%**	2.0%
0,5; 8	0.41	**90.0%**	4.1%	5.1%	0.1%	0.9%
0,7; 8	0.20	9.6%	0.8%	41.5%	**47.2%**	1.0%
1,8; 5	0.49	63.8%	**23.1%**	13.1%	0.0%	0.0%
2,7; 6	-0.50	**61.0%**	1.7%	23.2%	14.1%	0.0%
3,7; 6	-0.50	0.0%	1.0%	2.7%	**96.3%**	0.0%
4,5; 1	0.00	4.4%	3.1%	24.8%	31.9%	**35.9%**
6,8; 7	0.67	3.0%	8.9%	**88.1%**	0.0%	0.0%
6,9; 7	0.33	1.1%	0.7%	6.3%	**91.8%**	0.0%
0,1; 5,8	0.50	0.0%	1.0%	16.4%	**82.6%**	0.0%
0,6; 7,8	-0.41	**20.8%**	5.4%	18.9%	18.0%	36.9%
2,8; 6,7	0.41	42.7%	**9.5%**	45.4%	2.4%	0.0%
2,9; 6,7	0.33	0.1%	0.0%	13.9%	**85.9%**	0.0%
3,8; 6,7	-0.50	0.7%	15.9%	49.3%	**27.3%**	6.8%
4,8; 1,5	-0.50	7.0%	7.3%	**85.1%**	0.6%	0.0%
5,7; 0,8	0.19	46.2%	**3.3%**	35.8%	14.1%	0.6%
0,2; 6,7,8	-0.20	19.6%	0.7%	24.6%	**20.3%**	34.7%
0,3; 6,7,8	0.09	9.4%	0.7%	**15.7%**	14.0%	60.2%
0,4; 1,5,8	0.56	9.2%	7.9%	**80.4%**	2.5%	0.0%
1,7; 0,5,8	0.49	**71.0%**	10.8%	18.1%	0.1%	0.0%
5,6; 0,7,8	0.19	**50.0%**	5.3%	23.9%	16.7%	4.1%
8,9; 2,6,7	-0.23	7.1%	4.5%	**22.4%**	17.8%	48.2%
0,9; 2,6,7,8	-0.23	10.8%	2.7%	**14.8%**	14.8%	56.9%
1,6; 0,5,7,8	0.00	10.7%	2.4%	15.1%	15.2%	**56.7%**
2,5; 0,6,7,8	0.50	14.1%	7.2%	**66.8%**	10.8%	1.0%
3,5; 0,6,7,8	0.00	8.4%	0.8%	13.8%	15.8%	**61.2%**
4,7; 0,1,5,8	-0.33	**59.1%**	9.2%	24.2%	4.6%	3.0%
1,2; 0,5,6,7,8	-0.13	22.4%	**8.1%**	35.3%	22.8%	11.4%
1,3; 0,5,6,7,8	0.00	6.9%	1.2%	10.5%	12.8%	**68.7%**
4,6; 0,1,5,7,8	0.17	21.0%	5.4%	**39.7%**	27.8%	6.1%
5,9; 0,2,6,7,8	0.00	8.7%	2.9%	14.7%	15.7%	**57.9%**
1,9; 0,2,5,6,7,8	0.00	5.2%	1.4%	9.2%	8.9%	**75.3%**
2,3; 0,1,5,6,7,8	0.00	5.8%	0.2%	7.5%	8.2%	**78.3%**
2,4; 0,1,5,6,7,8	0.00	6.0%	2.4%	9.8%	9.2%	**72.6%**
3,9; 0,1,2,5,6,7,8	0.00	6.1%	1.2%	10.9%	10.4%	**71.3%**
4,9; 0,1,2,5,6,7,8	0.00	5.7%	2.7%	9.1%	9.6%	**72.8%**
3,4; 0,1,2,5,6,7,8,9	0.00	7.4%	1.0%	10.1%	8.8%	**72.7%**

Table 3.15: Aggregated empirical posterior probabilities of Scenario X1-T6. The original pair copula families are in bold; incorrect posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	0.67	0.1%	0.1%	**99.7%**	0.1%	0.0%
1,4	0.50	2.4%	1.1%	**88.0%**	8.5%	0.0%
1,5	0.13	0.5%	**5.8%**	30.8%	61.1%	1.9%
2,6	-0.19	6.6%	**36.1%**	48.1%	7.6%	1.5%
3,6	0.40	**76.4%**	10.0%	11.6%	2.0%	0.0%
5,8	-0.50	22.4%	3.6%	**74.0%**	0.0%	0.0%
6,7	0.30	**89.3%**	4.9%	3.6%	2.3%	0.0%
7,8	0.67	0.0%	0.0%	0.0%	**100.0%**	0.0%
7,9	0.50	0.0%	0.0%	5.1%	**94.9%**	0.0%
0,5; 8	0.41	**74.2%**	7.3%	17.7%	0.8%	0.0%
0,7; 8	0.20	6.8%	1.9%	37.4%	**51.2%**	2.7%
1,8; 5	0.49	40.4%	**13.1%**	44.7%	1.3%	0.5%
2,7; 6	-0.50	**71.4%**	7.3%	18.6%	2.7%	0.0%
3,7; 6	-0.50	0.0%	0.0%	8.4%	**91.6%**	0.0%
4,5; 1	0.00	9.0%	1.0%	13.1%	11.6%	**65.4%**
6,8; 7	0.67	0.5%	0.6%	**98.2%**	0.7%	0.0%
6,9; 7	0.33	0.3%	0.0%	20.9%	**78.8%**	0.0%
0,1; 5,8	0.00	6.3%	1.3%	11.4%	10.5%	**70.5%**
0,6; 7,8	0.00	5.9%	0.9%	10.9%	10.1%	**72.2%**
2,8; 6,7	0.00	5.7%	1.0%	11.4%	11.8%	**70.1%**
2,9; 6,7	0.00	5.0%	2.2%	11.2%	9.1%	**72.5%**
3,8; 6,7	0.00	5.3%	0.3%	10.3%	9.7%	**74.4%**
4,8; 1,5	0.00	6.7%	1.9%	11.8%	12.2%	**67.5%**
5,7; 0,8	0.00	5.4%	0.7%	10.5%	9.2%	**74.2%**
0,2; 6,7,8	0.00	5.3%	0.9%	10.8%	11.3%	**71.6%**
0,3; 6,7,8	0.00	7.2%	0.7%	11.9%	12.6%	**67.6%**
0,4; 1,5,8	0.00	6.7%	1.3%	10.1%	10.2%	**71.8%**
1,7; 0,5,8	0.00	4.9%	2.7%	8.5%	8.8%	**75.2%**
5,6; 0,7,8	0.00	4.7%	0.8%	9.5%	9.4%	**75.6%**
8,9; 2,6,7	0.00	4.4%	0.4%	7.2%	7.8%	**80.1%**
0,9; 2,6,7,8	0.00	7.3%	1.6%	13.0%	11.3%	**66.8%**
1,6; 0,5,7,8	0.00	6.4%	0.4%	9.6%	9.4%	**74.3%**
2,5; 0,6,7,8	0.00	7.2%	1.9%	10.5%	10.7%	**69.6%**
3,5; 0,6,7,8	0.00	4.7%	0.8%	8.2%	9.9%	**76.4%**
4,7; 0,1,5,8	0.00	5.9%	3.1%	10.1%	9.7%	**71.3%**
1,2; 0,5,6,7,8	0.00	6.5%	1.6%	11.4%	11.6%	**68.9%**
1,3; 0,5,6,7,8	0.00	6.9%	1.8%	11.6%	10.0%	**69.7%**
4,6; 0,1,5,7,8	0.00	6.8%	1.3%	10.9%	9.3%	**71.7%**
5,9; 0,2,6,7,8	0.00	8.0%	2.7%	10.9%	11.1%	**67.3%**
1,9; 0,2,5,6,7,8	0.00	4.1%	0.0%	4.9%	5.7%	**85.2%**
2,3; 0,1,5,6,7,8	0.00	4.9%	0.8%	9.7%	8.2%	**76.3%**
2,4; 0,1,5,6,7,8	0.00	4.6%	0.8%	7.6%	8.0%	**79.0%**
3,9; 0,1,2,5,6,7,8	0.00	5.9%	0.4%	8.2%	7.9%	**77.6%**
4,9; 0,1,2,5,6,7,8	0.00	4.8%	0.5%	8.3%	8.8%	**77.6%**
3,4; 0,1,2,5,6,7,8,9	0.00	5.1%	1.2%	8.7%	8.7%	**76.4%**

Table 3.16: Aggregated empirical posterior probabilities of Scenario X1-T2. The original pair copula families are in bold; different posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	0.57	59.1%	**29.0%**	12.0%	0.0%	0.0%
1,3	0.48	2.2%	**97.8%**	0.0%	0.0%	0.0%
1,9	0.12	0.0%	1.0%	16.3%	**81.5%**	1.2%
2,4	0.00	5.4%	13.0%	21.6%	60.1%	**0.0%**
2,8	0.00	10.7%	1.8%	10.7%	20.1%	**56.8%**
3,8	0.00	18.6%	0.0%	15.3%	6.8%	**59.3%**
5,9	0.04	**31.7%**	1.3%	7.9%	24.6%	34.5%
6,8	0.28	0.0%	1.0%	3.0%	**96.0%**	0.0%
7,9	-0.02	**13.4%**	1.2%	6.8%	35.7%	42.9%
0,3; 8	-0.30	**66.3%**	12.1%	18.2%	3.3%	0.0%
1,5; 9	-0.26	**99.2%**	0.8%	0.0%	0.0%	0.0%
1,7; 9	-0.15	**94.3%**	1.9%	1.0%	2.8%	0.0%
1,8; 3	-0.03	31.1%	8.8%	24.0%	**14.6%**	21.5%
2,3; 8	0.46	0.5%	0.0%	11.1%	**88.4%**	0.0%
3,6; 8	-0.53	**92.1%**	6.9%	1.0%	0.0%	0.0%
3,9; 1	0.60	**88.3%**	11.7%	0.0%	0.0%	0.0%
4,8; 2	0.25	1.0%	0.0%	4.4%	**94.5%**	0.0%
0,2; 3,8	0.00	14.9%	6.0%	19.3%	22.0%	**37.8%**
1,2; 3,8	0.00	16.9%	5.7%	27.4%	20.4%	**29.5%**
2,6; 3,8	0.74	41.3%	**49.4%**	9.2%	0.0%	0.0%
3,4; 2,8	0.22	12.7%	87.3%	**0.0%**	0.0%	0.0%
3,7; 1,9	0.00	27.0%	0.1%	22.1%	20.8%	**30.1%**
5,7; 1,9	-0.62	**81.5%**	18.5%	0.0%	0.0%	0.0%
8,9; 1,3	-0.35	5.4%	15.5%	**79.1%**	0.0%	0.0%
0,1; 2,3,8	0.00	16.8%	4.1%	16.5%	20.8%	**41.8%**
0,6; 2,3,8	0.00	20.3%	2.5%	23.5%	20.7%	**33.0%**
2,9; 1,3,8	0.11	5.3%	**43.7%**	38.8%	10.2%	2.0%
3,5; 1,7,9	0.79	**80.9%**	18.1%	1.0%	0.0%	0.0%
4,6; 2,3,8	0.33	4.8%	4.5%	35.1%	**55.6%**	0.0%
7,8; 1,3,9	0.45	23.3%	39.2%	**37.5%**	0.0%	0.0%
0,4; 2,3,6,8	-0.38	13.1%	1.5%	22.0%	**20.4%**	43.1%
0,9; 1,2,3,8	-0.43	**15.4%**	4.5%	18.4%	18.6%	43.1%
1,6; 0,2,3,8	-0.14	29.5%	6.3%	**43.1%**	19.1%	2.0%
2,7; 1,3,8,9	0.00	16.9%	7.9%	24.5%	20.6%	**30.1%**
5,8; 1,3,7,9	0.31	1.7%	3.0%	41.8%	**53.5%**	0.0%
0,7; 1,2,3,8,9	0.05	**20.5%**	4.9%	22.5%	13.1%	39.0%
1,4; 0,2,3,6,8	0.63	**67.7%**	27.5%	4.8%	0.0%	0.0%
2,5; 1,3,7,8,9	0.72	32.0%	**38.9%**	29.2%	0.0%	0.0%
6,9; 0,1,2,3,8	-0.23	12.5%	13.0%	**42.6%**	23.5%	8.4%
0,5; 1,2,3,7,8,9	0.00	15.2%	5.8%	18.7%	22.0%	**38.2%**
4,9; 0,1,2,3,6,8	-0.66	20.7%	26.4%	50.3%	**2.6%**	0.0%
6,7; 0,1,2,3,8,9	0.77	**34.8%**	23.7%	38.6%	2.9%	0.0%
4,7; 0,1,2,3,6,8,9	0.73	24.1%	25.6%	45.8%	**4.5%**	0.0%
5,6; 0,1,2,3,7,8,9	-0.37	35.8%	8.7%	**41.2%**	14.3%	0.0%
4,5; 0,1,2,3,6,7,8,9	0.00	20.0%	9.6%	22.9%	20.9%	**26.6%**

Table 3.17: Aggregated empirical posterior probabilities of Scenario X2-T8. The original pair copula families are in bold; different posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	0.57	56.2%	**28.7%**	15.1%	0.0%	0.0%
1,3	0.48	51.5%	**40.0%**	8.5%	0.0%	0.0%
1,9	0.12	6.4%	0.7%	27.4%	**62.6%**	2.9%
2,4	0.00	4.6%	0.3%	8.5%	8.5%	**78.1%**
2,8	0.00	3.5%	0.8%	4.8%	6.2%	**84.8%**
3,8	0.00	2.2%	0.0%	4.5%	2.6%	**90.7%**
5,9	0.04	**9.8%**	0.8%	12.4%	11.4%	65.8%
6,8	0.28	0.0%	0.0%	3.1%	**96.9%**	0.0%
7,9	-0.02	**6.4%**	0.5%	8.1%	10.6%	74.4%
0,3; 8	-0.30	**70.5%**	7.5%	18.4%	3.7%	0.0%
1,5; 9	-0.26	**75.5%**	10.2%	11.7%	2.6%	0.0%
1,7; 9	-0.15	**55.7%**	6.9%	23.4%	12.5%	1.6%
1,8; 3	-0.03	9.2%	2.0%	15.8%	**12.8%**	60.4%
2,3; 8	0.46	0.0%	0.0%	0.0%	**100.0%**	0.0%
3,6; 8	-0.53	**93.0%**	7.0%	0.0%	0.0%	0.0%
3,9; 1	0.60	**91.5%**	7.0%	1.5%	0.0%	0.0%
4,8; 2	0.25	2.9%	0.7%	29.6%	**66.9%**	0.0%
0,2; 3,8	0.00	4.0%	0.7%	8.5%	8.3%	**78.4%**
1,2; 3,8	0.00	6.2%	0.6%	10.6%	11.7%	**70.8%**
2,6; 3,8	0.74	51.9%	**35.6%**	12.4%	0.0%	0.0%
3,4; 2,8	0.22	13.0%	5.1%	**50.1%**	30.8%	1.0%
3,7; 1,9	0.00	6.1%	0.9%	9.6%	10.3%	**73.0%**
5,7; 1,9	-0.62	**83.7%**	12.2%	4.1%	0.0%	0.0%
8,9; 1,3	-0.35	8.3%	3.6%	**63.4%**	24.7%	0.0%
0,1; 2,3,8	0.00	8.1%	1.0%	13.2%	13.9%	**63.8%**
0,6; 2,3,8	0.00	7.2%	0.5%	10.9%	12.2%	**69.2%**
2,9; 1,3,8	0.00	5.8%	0.1%	9.4%	10.8%	**74.0%**
3,5; 1,7,9	0.00	6.5%	1.1%	11.2%	11.7%	**69.4%**
4,6; 2,3,8	0.00	7.1%	1.6%	12.6%	11.8%	**67.0%**
7,8; 1,3,9	0.00	5.8%	0.8%	9.9%	13.0%	**70.5%**
0,4; 2,3,6,8	0.00	6.8%	1.9%	12.8%	10.7%	**67.8%**
0,9; 1,2,3,8	0.00	6.0%	0.9%	11.2%	11.8%	**70.1%**
1,6; 0,2,3,8	0.00	5.7%	0.7%	11.9%	10.5%	**71.3%**
2,7; 1,3,8,9	0.00	6.6%	0.6%	10.7%	9.8%	**72.4%**
5,8; 1,3,7,9	0.00	5.7%	1.0%	9.6%	10.7%	**73.1%**
0,7; 1,2,3,8,9	0.00	5.8%	0.7%	10.2%	9.8%	**73.5%**
1,4; 0,2,3,6,8	0.00	5.1%	0.3%	9.5%	10.4%	**74.8%**
2,5; 1,3,7,8,9	0.00	5.1%	1.4%	8.8%	8.8%	**75.9%**
6,9; 0,1,2,3,8	0.00	6.0%	0.3%	12.2%	11.6%	**69.9%**
0,5; 1,2,3,7,8,9	0.00	4.9%	1.3%	8.5%	9.0%	**76.3%**
4,9; 0,1,2,3,6,8	0.00	5.6%	0.5%	10.3%	10.4%	**73.2%**
6,7; 0,1,2,3,8,9	0.00	8.1%	0.9%	13.8%	12.3%	**64.9%**
4,7; 0,1,2,3,6,8,9	0.00	5.6%	1.0%	10.6%	11.4%	**71.4%**
5,6; 0,1,2,3,7,8,9	0.00	4.7%	1.3%	7.5%	7.7%	**78.8%**
4,5; 0,1,2,3,6,7,8,9	0.00	6.0%	1.2%	9.9%	10.4%	**72.4%**

Table 3.18: Aggregated empirical posterior probabilities of Scenario X2-T3. The original pair copula families are in bold; different posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	-0.41	0.2%	0.0%	**93.9%**	5.9%	0.0%
0,9	-0.67	0.0%	2.0%	**98.0%**	0.0%	0.0%
1,4	0.64	**92.8%**	7.2%	0.0%	0.0%	0.0%
2,7	-0.23	7.4%	**85.5%**	7.1%	0.0%	0.0%
2,9	-0.41	13.9%	**57.7%**	28.4%	0.0%	0.0%
3,5	0.42	34.2%	**56.6%**	9.2%	0.0%	0.0%
4,9	0.64	0.0%	0.0%	**100.0%**	0.0%	0.0%
5,6	0.60	0.0%	1.0%	**99.0%**	0.0%	0.0%
5,9	-0.36	**92.0%**	6.5%	1.5%	0.0%	0.0%
0,5; 9	0.43	14.9%	**83.1%**	2.0%	0.0%	0.0%
1,9; 4	-0.27	**72.0%**	24.3%	3.6%	0.0%	0.0%
2,4; 9	0.58	**84.2%**	15.8%	0.0%	0.0%	0.0%
2,5; 9	0.00	70.6%	1.1%	8.8%	12.4%	**7.1%**
3,6; 5	0.32	25.0%	4.1%	**14.1%**	56.8%	0.0%
6,9; 5	-0.17	1.5%	1.2%	9.9%	**86.6%**	0.8%
7,9; 2	0.30	18.0%	**72.6%**	8.0%	1.4%	0.0%
8,9; 0	0.30	**77.7%**	7.7%	12.4%	2.1%	0.0%
0,6; 5,9	0.40	0.0%	1.0%	24.1%	**74.9%**	0.0%
1,2; 4,9	-0.64	13.6%	22.2%	**64.3%**	0.0%	0.0%
2,6; 5,9	0.00	54.9%	9.0%	10.6%	20.4%	**5.2%**
3,9; 5,6	0.17	14.0%	8.2%	28.6%	**49.2%**	0.0%
4,7; 2,9	0.28	**65.6%**	5.0%	23.0%	6.4%	0.0%
5,7; 2,9	0.02	8.6%	1.2%	19.1%	**22.6%**	48.5%
5,8; 0,9	-0.64	53.6%	**30.9%**	15.5%	0.0%	0.0%
0,3; 5,6,9	-0.57	16.6%	5.6%	**30.1%**	47.7%	0.0%
1,7; 2,4,9	-0.53	**66.6%**	16.9%	15.8%	0.8%	0.0%
2,3; 5,6,9	0.60	14.1%	13.2%	**71.9%**	0.8%	0.0%
4,5; 2,7,9	0.24	13.1%	0.1%	40.1%	**45.7%**	1.0%
6,7; 2,5,9	-0.72	66.8%	**28.4%**	4.8%	0.0%	0.0%
6,8; 0,5,9	0.00	9.3%	1.2%	14.6%	12.2%	**62.7%**
0,2; 3,5,6,9	-0.48	**6.0%**	1.1%	60.8%	32.1%	0.0%
1,5; 2,4,7,9	0.21	15.8%	8.6%	**46.4%**	28.2%	1.0%
3,7; 2,5,6,9	0.00	5.6%	8.9%	30.4%	39.3%	**15.8%**
3,8; 0,5,6,9	-0.54	12.2%	2.0%	17.6%	**18.2%**	50.0%
4,6; 2,5,7,9	0.00	15.2%	1.5%	19.0%	19.6%	**44.7%**
0,7; 2,3,5,6,9	-0.42	3.6%	8.7%	**44.9%**	41.8%	1.0%
1,6; 2,4,5,7,9	0.00	9.2%	2.4%	20.0%	15.8%	**52.7%**
2,8; 0,3,5,6,9	0.78	**8.1%**	0.7%	12.7%	11.6%	66.9%
3,4; 2,5,6,7,9	0.00	10.7%	1.2%	17.5%	21.2%	**49.4%**
0,4; 2,3,5,6,7,9	0.00	9.5%	0.4%	13.2%	15.3%	**61.6%**
1,3; 2,4,5,6,7,9	0.00	26.1%	3.6%	38.1%	23.1%	**9.1%**
7,8; 0,2,3,5,6,9	0.72	7.7%	**2.0%**	13.9%	12.9%	63.5%
0,1; 2,3,4,5,6,7,9	0.32	12.9%	1.6%	34.9%	**29.8%**	20.8%
4,8; 0,2,3,5,6,7,9	0.00	8.9%	2.8%	16.9%	15.8%	**55.6%**
1,8; 0,2,3,4,5,6,7,9	-0.49	7.9%	1.7%	15.6%	**14.1%**	60.6%

Table 3.19: Aggregated empirical posterior probabilities of Scenario X3-T9. The original pair copula families are in bold; different posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	-0.41	0.3%	1.6%	**92.9%**	5.2%	0.0%
0,9	-0.67	0.0%	1.0%	**98.0%**	1.0%	0.0%
1,4	0.64	**92.6%**	6.4%	1.0%	0.0%	0.0%
2,7	-0.23	34.7%	**8.9%**	47.6%	8.7%	0.0%
2,9	-0.41	28.7%	**69.4%**	1.8%	0.0%	0.0%
3,5	0.42	46.1%	**27.9%**	25.9%	0.1%	0.0%
4,9	0.64	0.0%	0.0%	**100.0%**	0.0%	0.0%
5,6	0.60	0.2%	0.6%	**97.3%**	1.9%	0.0%
5,9	-0.36	**93.8%**	5.1%	0.8%	0.2%	0.0%
0,5; 9	0.43	50.5%	**42.5%**	7.0%	0.0%	0.0%
1,9; 4	-0.27	**79.5%**	12.2%	8.0%	0.3%	0.0%
2,4; 9	0.58	**91.1%**	6.9%	2.0%	0.0%	0.0%
2,5; 9	0.00	7.4%	0.9%	13.5%	14.4%	**63.8%**
3,6; 5	0.32	7.6%	4.2%	**48.0%**	40.2%	0.0%
6,9; 5	-0.17	6.2%	3.0%	39.8%	**50.0%**	1.0%
7,9; 2	0.30	43.7%	**12.5%**	38.7%	5.0%	0.0%
8,9; 0	0.30	**84.6%**	8.6%	5.5%	1.2%	0.0%
0,6; 5,9	0.40	0.2%	0.2%	17.0%	**82.6%**	0.0%
1,2; 4,9	-0.64	2.4%	2.5%	**92.2%**	2.9%	0.0%
2,6; 5,9	0.00	6.9%	1.6%	13.5%	13.1%	**64.9%**
3,9; 5,6	0.17	7.1%	1.2%	36.7%	**54.5%**	0.5%
4,7; 2,9	0.28	**60.9%**	6.4%	23.2%	9.5%	0.0%
5,7; 2,9	0.02	8.4%	0.1%	12.4%	**13.5%**	65.6%
5,8; 0,9	-0.64	56.0%	**25.3%**	18.8%	0.0%	0.0%
0,3; 5,6,9	0.00	4.8%	0.8%	8.2%	7.4%	**78.8%**
1,7; 2,4,9	0.00	7.6%	0.8%	11.1%	11.0%	**69.5%**
2,3; 5,6,9	0.00	6.3%	0.3%	12.6%	13.0%	**67.8%**
4,5; 2,7,9	0.00	8.9%	0.5%	15.5%	12.4%	**62.8%**
6,7; 2,5,9	0.00	5.9%	0.6%	10.1%	10.6%	**72.8%**
6,8; 0,5,9	0.00	6.8%	0.8%	10.8%	12.1%	**69.5%**
0,2; 3,5,6,9	0.00	5.7%	0.4%	10.2%	9.8%	**74.0%**
1,5; 2,4,7,9	0.00	6.7%	0.2%	14.9%	13.2%	**65.0%**
3,7; 2,5,6,9	0.00	5.4%	3.0%	9.8%	10.2%	**71.5%**
3,8; 0,5,6,9	0.00	6.9%	0.6%	12.9%	14.3%	**65.3%**
4,6; 2,5,7,9	0.00	8.4%	2.2%	14.3%	13.2%	**61.9%**
0,7; 2,3,5,6,9	0.00	5.2%	0.6%	9.1%	10.4%	**74.7%**
1,6; 2,4,5,7,9	0.00	6.3%	0.9%	11.0%	11.7%	**70.2%**
2,8; 0,3,5,6,9	0.00	8.0%	0.4%	12.5%	14.1%	**65.0%**
3,4; 2,5,6,7,9	0.00	6.9%	0.3%	12.3%	11.4%	**69.0%**
0,4; 2,3,5,6,7,9	0.00	6.9%	1.6%	13.1%	12.6%	**65.8%**
1,3; 2,4,5,6,7,9	0.00	7.9%	0.6%	11.8%	10.9%	**68.8%**
7,8; 0,2,3,5,6,9	0.00	6.9%	0.5%	10.7%	11.5%	**70.4%**
0,1; 2,3,4,5,6,7,9	0.00	7.3%	0.5%	13.3%	13.7%	**65.2%**
4,8; 0,2,3,5,6,7,9	0.00	7.9%	0.9%	11.8%	11.8%	**67.6%**
1,8; 0,2,3,4,5,6,7,9	0.00	5.6%	0.6%	12.2%	11.7%	**69.9%**

Table 3.20: Aggregated empirical posterior probabilities of Scenario X3-T3. The original pair copula families are in bold; different posterior modes are in red.

Pair	True τ	N	T	G	C	I
0,8	-0.41	2.7%	2.8%	**76.7%**	17.8%	0.0%
0,9	-0.67	0.1%	0.0%	**99.1%**	0.8%	0.0%
1,4	0.64	**91.3%**	8.4%	0.3%	0.0%	0.0%
2,7	-0.23	31.7%	**9.1%**	44.7%	14.5%	0.0%
2,9	-0.41	47.9%	**36.7%**	15.3%	0.0%	0.0%
3,5	0.42	41.6%	**28.0%**	28.7%	1.7%	0.0%
4,9	0.64	0.1%	0.9%	**97.9%**	1.0%	0.0%
5,6	0.60	1.8%	2.3%	**91.0%**	4.9%	0.0%
5,9	-0.36	**81.9%**	6.5%	10.5%	1.2%	0.0%
0,5; 9	0.43	46.4%	**19.9%**	32.9%	0.9%	0.0%
1,9; 4	-0.27	**60.9%**	4.6%	25.5%	9.0%	0.0%
2,4; 9	0.58	**84.1%**	9.4%	6.5%	0.0%	0.0%
2,5; 9	0.00	4.9%	0.2%	9.5%	8.6%	**76.7%**
3,6; 5	0.32	10.5%	3.4%	**61.7%**	24.4%	0.0%
6,9; 5	-0.17	8.3%	3.8%	35.6%	**51.4%**	1.0%
7,9; 2	0.30	37.3%	**14.4%**	42.9%	5.3%	0.0%
8,9; 0	0.30	**66.5%**	6.0%	21.4%	6.1%	0.0%
0,6; 5,9	0.00	7.6%	0.4%	13.6%	11.3%	**67.1%**
1,2; 4,9	0.00	6.9%	1.9%	11.3%	11.4%	**68.5%**
2,6; 5,9	0.00	6.1%	0.1%	8.1%	9.6%	**76.0%**
3,9; 5,6	0.00	7.5%	2.0%	13.2%	12.3%	**64.9%**
4,7; 2,9	0.00	6.1%	0.6%	8.8%	8.7%	**75.7%**
5,7; 2,9	0.00	7.1%	0.6%	11.1%	11.7%	**69.5%**
5,8; 0,9	0.00	4.5%	0.2%	8.9%	8.5%	**77.9%**
0,3; 5,6,9	0.00	5.6%	1.9%	11.2%	11.2%	**70.1%**
1,7; 2,4,9	0.00	7.3%	1.1%	11.0%	11.3%	**69.4%**
2,3; 5,6,9	0.00	7.8%	0.6%	10.5%	10.1%	**71.0%**
4,5; 2,7,9	0.00	6.9%	0.1%	10.9%	10.2%	**71.9%**
6,7; 2,5,9	0.00	7.9%	1.7%	12.8%	12.8%	**64.8%**
6,8; 0,5,9	0.00	5.0%	0.4%	7.7%	7.4%	**79.5%**
0,2; 3,5,6,9	0.00	6.1%	0.5%	11.8%	10.7%	**70.9%**
1,5; 2,4,7,9	0.00	6.0%	2.5%	9.9%	10.2%	**71.4%**
3,7; 2,5,6,9	0.00	6.3%	1.5%	10.7%	9.3%	**72.2%**
3,8; 0,5,6,9	0.00	7.4%	1.2%	10.0%	11.0%	**70.3%**
4,6; 2,5,7,9	0.00	6.6%	1.0%	12.9%	12.0%	**67.6%**
0,7; 2,3,5,6,9	0.00	7.9%	0.8%	10.9%	10.8%	**69.6%**
1,6; 2,4,5,7,9	0.00	6.0%	1.2%	10.0%	11.8%	**70.9%**
2,8; 0,3,5,6,9	0.00	5.5%	0.7%	10.8%	10.1%	**72.9%**
3,4; 2,5,6,7,9	0.00	6.1%	1.2%	10.1%	9.8%	**72.8%**
0,4; 2,3,5,6,7,9	0.00	5.3%	0.7%	9.6%	9.1%	**75.2%**
1,3; 2,4,5,6,7,9	0.00	5.5%	0.7%	12.7%	10.8%	**70.4%**
7,8; 0,2,3,5,6,9	0.00	5.9%	0.3%	10.9%	9.2%	**73.8%**
0,1; 2,3,4,5,6,7,9	0.00	6.3%	1.6%	12.5%	13.1%	**66.5%**
4,8; 0,2,3,5,6,7,9	0.00	5.7%	1.9%	14.0%	14.8%	**63.6%**
1,8; 0,2,3,4,5,6,7,9	0.00	5.3%	1.4%	11.5%	10.8%	**71.0%**

Table 3.21: Aggregated empirical posterior probabilities of Scenario X3-T2. The original pair copula families are in bold; different posterior modes are in red.

3.B Selected Models for the Real-Data Example

Pair	N	T	G	C	I
1,2	**79.9%**	20.1%	0.0%	0.0%	0.0%
2,3	21.1%	**78.9%**	0.0%	0.0%	0.0%
3,4	11.3%	**88.7%**	0.0%	0.0%	0.0%
4,5	0.0%	**82.1%**	17.9%	0.0%	0.0%
5,6	0.0%	**100.0%**	0.0%	0.0%	0.0%
6,7	0.0%	**100.0%**	0.0%	0.0%	0.0%
7,8	29.4%	0.0%	**46.9%**	23.7%	0.0%
8,9	**78.2%**	21.9%	0.0%	0.0%	0.0%
1,3; 2	**38.1%**	9.1%	20.5%	32.4%	0.0%
2,4; 3	35.8%	0.0%	**45.3%**	18.9%	0.0%
3,5; 4	10.7%	0.0%	**67.2%**	22.1%	0.0%
4,6; 5	**86.9%**	0.0%	12.0%	1.1%	0.0%
5,7; 6	10.5%	**57.4%**	32.1%	0.0%	0.0%
6,8; 7	**49.9%**	0.0%	13.3%	36.7%	0.0%
7,9; 8	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,4; 2,3	**74.1%**	0.0%	23.7%	2.2%	0.0%
2,5; 3,4	0.0%	0.0%	0.0%	0.0%	**100.0%**
3,6; 4,5	0.5%	0.0%	**52.8%**	46.7%	0.0%
4,7; 5,6	2.0%	0.0%	1.2%	5.0%	**91.8%**
5,8; 6,7	1.2%	0.0%	**51.9%**	46.9%	0.0%
6,9; 7,8	10.4%	0.0%	8.1%	**42.3%**	39.2%
1,5; 2,3,4	8.3%	0.0%	6.7%	6.5%	**78.6%**
2,6; 3,4,5	23.7%	0.0%	**39.4%**	36.9%	0.0%
3,7; 4,5,6	4.3%	0.0%	36.5%	**59.2%**	0.0%
4,8; 5,6,7	**65.0%**	0.0%	34.3%	0.7%	0.0%
5,9; 6,7,8	**51.5%**	0.0%	43.5%	5.0%	0.0%
1,6; 2,3,4,5	**57.4%**	2.3%	39.0%	1.3%	0.0%
2,7; 3,4,5,6	0.0%	0.0%	5.0%	2.1%	**92.9%**
3,8; 4,5,6,7	**84.3%**	13.1%	0.7%	1.9%	0.0%
4,9; 5,6,7,8	18.1%	17.5%	**63.4%**	1.0%	0.0%
1,7; 2,3,4,5,6	21.1%	0.0%	28.7%	**50.2%**	0.0%
2,8; 3,4,5,6,7	33.0%	0.0%	**39.9%**	27.1%	0.0%
3,9; 4,5,6,7,8	**68.1%**	7.9%	18.1%	5.9%	0.0%
1,8; 2,3,4,5,6,7	6.9%	0.0%	**64.4%**	27.6%	1.2%
2,9; 3,4,5,6,7,8	4.5%	0.0%	28.7%	18.6%	**48.1%**
1,9; 2,3,4,5,6,7,8	16.5%	0.0%	**55.2%**	28.4%	0.0%

Table 3.22: Posterior distribution of pair copulas, given a drawable vine tree structure. The biggest posterior probabilities are in bold.

Pair	N	T	G	C	I
0,1	**91.0%**	9.1%	0.0%	0.0%	0.0%
0,2	**100.0%**	0.0%	0.0%	0.0%	0.0%
0,8	**100.0%**	0.0%	0.0%	0.0%	0.0%
1,3	0.0%	**100.0%**	0.0%	0.0%	0.0%
2,4	0.0%	**100.0%**	0.0%	0.0%	0.0%
2,5	0.0%	0.0%	**83.5%**	16.6%	0.0%
5,6	0.0%	**100.0%**	0.0%	0.0%	0.0%
7,8	**95.0%**	2.7%	2.3%	0.0%	0.0%
0,3; 1	**62.3%**	37.7%	0.0%	0.0%	0.0%
0,4; 2	0.0%	0.0%	0.0%	0.0%	**100.0%**
0,5; 2	**52.1%**	0.0%	43.8%	4.1%	0.0%
0,7; 8	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,2; 0	5.7%	**94.3%**	0.0%	0.0%	0.0%
1,8; 0	0.0%	44.7%	**55.3%**	0.0%	0.0%
2,6; 5	0.0%	0.0%	0.0%	0.0%	**100.0%**
0,6; 2,5	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,5; 0,2	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,7; 0,8	22.1%	0.0%	**62.7%**	15.2%	0.0%
2,3; 0,1	0.0%	0.0%	0.0%	0.0%	**100.0%**
2,8; 0,1	0.0%	0.0%	22.2%	0.0%	**77.8%**
4,5; 0,2	**97.0%**	3.0%	0.0%	0.0%	0.0%
1,4; 0,2,5	30.1%	0.0%	**37.8%**	32.1%	0.0%
2,7; 0,1,8	0.0%	0.0%	0.0%	0.0%	**100.0%**
3,5; 0,1,2	2.8%	0.0%	33.9%	**63.2%**	0.0%
3,8; 0,1,2	2.4%	0.0%	**72.5%**	25.1%	0.0%
4,6; 0,2,5	**100.0%**	0.0%	0.0%	0.0%	0.0%
1,6; 0,2,4,5	22.3%	0.0%	**48.3%**	29.4%	0.0%
3,4; 0,1,2,5	10.4%	0.0%	**54.1%**	35.5%	0.0%
3,7; 0,1,2,8	0.0%	0.0%	0.0%	0.0%	**100.0%**
5,8; 0,1,2,3	**72.2%**	0.0%	5.9%	21.9%	0.0%
3,6; 0,1,2,4,5	17.1%	0.0%	**44.9%**	38.0%	0.0%
4,8; 0,1,2,3,5	**45.6%**	0.0%	15.8%	38.6%	0.0%
5,7; 0,1,2,3,8	1.5%	4.4%	0.0%	9.7%	**84.4%**
4,7; 0,1,2,3,5,8	**55.7%**	0.0%	14.5%	29.8%	0.0%
6,8; 0,1,2,3,4,5	23.8%	0.0%	**43.8%**	32.3%	0.0%
6,7; 0,1,2,3,4,5,8	0.0%	0.0%	0.0%	0.0%	**100.0%**

Table 3.23: Posterior distribution of pair copulas, given a drawable vine tree structure. The biggest posterior probabilities are in bold.

Pair	N	T	G	C	I
0,4	0.0%	**63.6%**	33.6%	2.8%	0.0%
0,6	0.0%	0.0%	**70.8%**	29.2%	0.0%
0,8	**82.3%**	17.7%	0.0%	0.0%	0.0%
1,2	15.5%	**84.5%**	0.0%	0.0%	0.0%
1,8	0.0%	**100.0%**	0.0%	0.0%	0.0%
3,6	2.9%	0.0%	**64.9%**	32.1%	0.0%
5,6	0.0%	**100.0%**	0.0%	0.0%	0.0%
7,8	31.8%	**38.0%**	30.2%	0.0%	0.0%
0,1; 8	**98.1%**	1.9%	0.0%	0.0%	0.0%
0,3; 6	**86.3%**	13.7%	0.0%	0.0%	0.0%
0,5; 6	0.0%	12.0%	**85.3%**	2.8%	0.0%
0,7; 8	**54.8%**	0.0%	23.5%	21.7%	0.0%
2,8; 1	4.4%	0.0%	17.4%	16.7%	**61.5%**
4,8; 0	**100.0%**	0.0%	0.0%	0.0%	0.0%
6,8; 0	**100.0%**	0.0%	0.0%	0.0%	0.0%
0,2; 1,8	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,6; 0,8	14.7%	0.0%	7.5%	21.1%	**56.7%**
3,8; 0,6	**45.1%**	0.0%	1.1%	19.2%	34.6%
4,6; 0,8	**80.9%**	19.1%	0.0%	0.0%	0.0%
5,8; 0,6	0.3%	0.0%	8.8%	8.0%	**82.9%**
6,7; 0,8	34.0%	0.0%	**41.6%**	24.4%	0.0%
1,3; 0,6,8	0.0%	0.0%	0.0%	0.0%	**100.0%**
1,4; 0,6,8	8.4%	0.0%	12.5%	25.7%	**53.4%**
1,5; 0,6,8	0.0%	0.0%	0.0%	6.7%	**93.3%**
2,6; 0,1,8	0.0%	0.0%	0.2%	14.8%	**85.1%**
4,7; 0,6,8	16.5%	14.1%	26.3%	**43.1%**	0.0%
1,7; 0,4,6,8	**44.6%**	1.0%	29.2%	25.2%	0.0%
2,4; 0,1,6,8	0.0%	0.0%	2.7%	1.7%	**95.7%**
3,5; 0,1,6,8	**51.6%**	0.0%	28.2%	20.2%	0.0%
4,5; 0,1,6,8	34.3%	0.0%	20.2%	**45.5%**	0.0%
2,7; 0,1,4,6,8	0.0%	0.0%	0.0%	0.0%	**100.0%**
3,4; 0,1,5,6,8	34.6%	9.7%	20.8%	**35.0%**	0.0%
5,7; 0,1,4,6,8	18.4%	0.0%	**38.8%**	22.7%	20.2%
2,5; 0,1,4,6,7,8	11.8%	**46.6%**	10.0%	31.6%	0.0%
3,7; 0,1,4,5,6,8	0.0%	0.0%	0.0%	5.8%	**94.2%**
2,3; 0,1,4,5,6,7,8	10.3%	26.9%	22.9%	**40.0%**	0.0%

Table 3.24: Posterior distribution of pair copulas, given a drawable vine tree structure. The biggest posterior probabilities are in bold.

Part II

Bayesian Modeling and Forecasting of High-Dimensional Time Series

Chapter 4

Introduction to Dynamic Linear Models

The contents of Section 4.4 are a reproduction of the submitted content of [Gruber and West, 2015a, Section 2.1]. Furthermore, the contents of Section 4.5 are a lightly edited reproduction of the contents of [Gruber and West, 2015b,a, Section 1].

4.1 Introduction

Dynamic models are generally defined as "sequences of sets of models" [West and Harrison, 1997, Chapter 1, Section 1.4]; in the context of time series analysis, the set of models varies over time t. The dynamic linear model (DLM) is a Bayesian, dynamic version of the traditional linear model: an observation equation describes a linear relationship between the time t response y_t, predictors $\mathbf{F}_t$ and states $\boldsymbol{\theta}_t$, where the states take the role of the regression coefficients in the normal linear model; an evolution equation specifies the evolution of the states from time t to $t+1$. This very general form of the DLM contains many well-known models as special cases, for example, polynomial trend models, seasonal models, and ARMA models [West and Harrison, 1997, Chapters 7–9].

The DLM can be learned sequentially, for example, for application to on-line decision problems, and retrospective analysis is also available [West and Harrison, 1997, Prado and West, 2010, Chapter 4]. We focus on sequential learning in this dissertation, given that our main interest lies in application to decision problems. The standard DLM, both univariate and multivariate, allows for conjugate sequential learning, and analytic updating equations exist. This makes forward filtering extremely fast, and there are numerous application examples that use the DLM in financial analyses [e.g. Gruber and Czado, 2015b,a, Zhao and West, 2014, Zhou et al., 2014, Gruber and West, 2015a], and in various other fields [e.g. Koop, 2012, Nakajima and West, 2013a, Hosking et al., 2013, Agarwal et al., 2010, Queen, 1994, Anacleto et al., 2013, Trejo et al., 2007, Prado, 2010].

Section 4.2 defines notation on various probability distributions used in connection with DLMs. Section 4.3 introduces the univariate DLM and the steps for conjugate forward filtering; Section 4.4 does the same for the multivariate Wishart DLM. Section 4.5 serves as a primer for Bayesian analysis of high-dimensional time series using the simultaneous graphical DLM, which is the core methodological contribution of this part of the dissertation, and discussed in Chapters 5 and 6, and Section 4.6 provides a brief summary of the main methods used in Chapters 5's derivation of an emulated conjugate sequential learning strategy for the simultaneous graphical DLM.

4.2 Distributions

This section defines our notation; for the most part, we follow standard notation as in [Prado and West, 2010, Chapter 1, Section 6; Chapter 2, Section 7; Chapter 10, Section 6]. The properties of the distributions below are taken from the same reference.

Beta distribution. A random variable η follows a beta distribution with parameters $\alpha > 0$ and $\beta > 0$,

$$\eta \sim Be(\alpha, \beta),$$

if its density function is

$$p(\eta) = \frac{1}{B(\alpha, \beta)} \eta^{\alpha-1}(1-\eta)^{\beta-1}.$$

The beta function $B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$, the expectation $E[\eta] = \frac{\alpha}{\alpha+\beta}$ and the variance $V[\eta] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

Univariate T distribution. A random variable y follows a univariate T distribution with mode μ, scale σ^2, and degrees of freedom ν,

$$y \sim T_\nu(\mu, \sigma^2),$$

if its density function is

$$p(y) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi\sigma^2}} \left(1 + \frac{1}{\nu}\frac{(y-\mu)^2}{\sigma^2}\right)^{-(\nu+1)/2}.$$

If $\nu > 1$, the expectation $E[y] = \mu$; if $\nu > 2$, the variance $V[y] = \frac{\nu}{\nu-2}\sigma^2$.

Multivariate T distribution. A random p-vector $\boldsymbol{\theta}$ follows a multivariate T distribution with mode $\boldsymbol{\mu} \in \mathbb{R}^p$, scale matrix or *covariance matrix factor* $\boldsymbol{\Sigma}$, and degrees of freedom ν,

$$\boldsymbol{\theta} \sim T_\nu(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

if its density function is

$$p(\boldsymbol{\theta}) = \frac{\Gamma(\frac{\nu+p}{2})}{\Gamma(\frac{\nu}{2})(\nu\pi)^{p/2}} |\boldsymbol{\Sigma}|^{-1/2} \left(1 + \frac{1}{\nu}(\boldsymbol{\theta}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})\right)^{-(\nu+p)/2}.$$

If $\nu > 1$, the expectation $E[\boldsymbol{\theta}] = \boldsymbol{\mu}$; if $\nu > 2$, the covariance $V[\boldsymbol{\theta}] = \frac{\nu}{\nu-2}\boldsymbol{\Sigma}$.

Univariate normal distribution. A random variable y follows a univariate normal distribution with mean μ and variance $\sigma^2 > 0$,

$$x \sim N(\mu, \sigma^2),$$

if its density function is

$$p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2}\right).$$

The inverse of the variance, $\lambda = 1/\sigma^2$, is called the precision.

Multivariate normal distribution. A random p-vector $\boldsymbol{\theta}$ follows a multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$,

$$\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

if its density function is

$$p(\boldsymbol{\theta}) = (2\pi)^{-p/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})\right).$$

The inverse of the covariance, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, is called the precision matrix.

Matrix normal distribution. A $p \times m$ random matrix $\mathbf{\Theta}$ follows a matrix normal distribution with mean $\mathbf{M} = (m_{i,j})_{i,j} \in \mathbb{R}^{p\times m}$, column variance matrix $\mathbf{C} = (c_{i,j}) \in \mathbb{R}^{p\times p}$ and row variance matrix $\mathbf{\Sigma} = (\sigma_{i,j}) \in \mathbb{R}^{m\times m}$,

$$\mathbf{\Theta} \sim N(\mathbf{M}, \mathbf{C}, \mathbf{\Sigma}),$$

if its density function is

$$p(\mathbf{\Theta}) = (2\pi)^{-pm/2}|\mathbf{C}|^{-p/2}|\mathbf{\Sigma}|^{-m/2} \times \exp\left(-\frac{1}{2}\,\mathrm{trace}\{(\mathbf{\Theta}-\mathbf{M})'\mathbf{C}^{-1}(\mathbf{\Theta}-\mathbf{M})\mathbf{\Sigma}^{-1}\}\right).$$

The marginal and conditional distributions of elements of $\mathbf{\Theta}$ are normal. The distribution of the i-th row of $\mathbf{\Theta}$, here written as $\mathbf{\Theta}_{i,:}$, is multivariate normal with mean $\mathbf{M}_{i,:}$ and covariance matrix $c_{i,i}\mathbf{\Sigma}$. The distribution of the j-th column of $\mathbf{\Theta}$, written $\mathbf{\Theta}_{:,j}$, is multivariate normal with mean $\mathbf{M}_{:,j}$ and covariance matrix $\sigma_{j,j}\mathbf{C}$. Furthermore, the covariance of two rows $\mathbf{\Theta}_{i,:}$ and $\mathbf{\Theta}_{j,:}$ of $\mathbf{\Theta}$, is $Cov(\mathbf{\Theta}_{i,:}, \mathbf{\Theta}_{j,:}) = c_{i,j}\mathbf{\Sigma}$, and the covariance of two columns $\mathbf{\Theta}_{:,i}$ and $\mathbf{\Theta}_{:,j}$ of $\mathbf{\Theta}$, is $Cov(\mathbf{\Theta}_{:,i}, \mathbf{\Theta}_{:,j}) = \sigma_{i,j}\mathbf{C}$.

Gamma distribution. A random scalar $\lambda > 0$ follows a gamma distribution with shape $\alpha > 0$ and rate $\beta > 0$,

$$\lambda \sim G(\alpha, \beta),$$

if its density function is

$$p(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}.$$

The expectation $E[\lambda] = \frac{\alpha}{\beta}$, and the variance $V[\lambda] = \frac{\alpha}{\beta^2}$.

Normal gamma distribution. A random p-vector $\boldsymbol{\theta}$ and scalar $\lambda > 0$ follow a normal gamma distribution

$$(\boldsymbol{\theta}, \lambda) \sim NG(\mathbf{a}, \mathbf{R}, r, c)$$

with mode $\mathbf{a}$, covariance matrix factor $\mathbf{R}$, degrees of freedom $r > 0$, and parameter $c > 0$ if

$$\begin{aligned}\boldsymbol{\theta}|\lambda &\sim N(\mathbf{a}, \mathbf{R}/(c\lambda)),\\ \lambda &\sim G(r/2, rc/2),\end{aligned}$$

where $N(\boldsymbol{\mu}, \mathbf{\Sigma})$ is multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$, and $G(\alpha, \beta)$ is gamma with shape $\alpha > 0$ and rate $\beta > 0$. Unconditional on λ, the distribution of $\boldsymbol{\theta}$ is multivariate T with mode $\boldsymbol{\mu}$, scale $\mathbf{R}$, and degrees of freedom r; hence the terminology for r.

Inverse Wishart distribution. A random $m\times m$ positive definite and symmetric matrix $\mathbf{\Sigma}$ has an inverse Wishart distribution with $n > 0$ degrees of freedom and a positive definite and symmetric sum-of-squares parameter matrix $\mathbf{D} \in \mathbb{R}^{m\times m}$

$$\mathbf{\Sigma} \sim IW(n, \mathbf{D}),$$

if its density function is

$$p(\mathbf{\Sigma}) = c|\mathbf{\Sigma}|^{-(m+n)/2}\exp\left(-\frac{1}{2}\,\mathrm{trace}\{\mathbf{\Sigma}\mathbf{D}\}\right),$$

with a constant $c = |\mathbf{D}|^{-(n+m-1)/2}2^{(n+m-1)m/2}\pi^{m(m-1)/4}\prod_{i=1:m}\Gamma((n+m-1)/2)$. If $n > 2$, the expectation is $E[\mathbf{\Sigma}] = \mathbf{D}/(n-2)$.

Wishart distribution. A random $m \times m$ positive definite and symmetric matrix $\mathbf{\Omega}$ has a Wishart distribution with parameters $h = n+m-1 \geq m$ and positive definite, symmetric parameter matrix $\mathbf{A} = \mathbf{D}^{-1}$, if its inverse $\mathbf{\Sigma} = \mathbf{\Omega}^{-1} \sim IW(n, \mathbf{D})$ follows an inverse Wishart distribution with n degrees of freedom and sum-of-squares parameter matrix $\mathbf{D}$.

Normal inverse Wishart distribution. A random $p \times m$ matrix $\mathbf{\Theta}$ and a $m \times m$ positive definite and symmetric matrix $\mathbf{\Sigma}$ have a normal inverse Wishart distribution

$$(\mathbf{\Theta}, \mathbf{\Sigma}) \sim NIW(\mathbf{M}, \mathbf{C}, n, \mathbf{D}),$$

when

$$\begin{aligned}\mathbf{\Theta}|\mathbf{\Sigma} &\sim N(\mathbf{M}, \mathbf{C}, \mathbf{\Sigma}),\\ \mathbf{\Sigma} &\sim IW(n, \mathbf{D}),\end{aligned}$$

where $N(\mathbf{M}, \mathbf{C}, \mathbf{\Sigma})$ is a matrix normal distribution with mean $\mathbf{M} \in \mathbb{R}^{p\times m}$, column variance matrix $\mathbf{C} \in \mathbb{R}^{p\times p}$ and row variance matrix $\mathbf{\Sigma}$, and $IW(n, \mathbf{D})$ is an inverse Wishart distribution with $n > 0$ degrees of freedom and sum-of-squares parameter matrix $\mathbf{D} \in \mathbb{R}^{m\times m}$.

4.3 The Univariate DLM

The dynamic linear model (DLM) is a conditionally Gaussian state-space model. This section introduces the univariate DLM with beta-gamma evolutions for stochastic variances and discount learning [West and Harrison, 1997, Prado and West, 2010].

4.3.1 Model Structure

The univariate time series y_t is modeled via

$$y_t' = \mathbf{F}_t'\boldsymbol{\theta}_t + \nu_t', \qquad \nu_t \sim N(0, 1/\lambda_t), \tag{4.1}$$

$$\boldsymbol{\theta}_t = \mathbf{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t, \qquad \boldsymbol{\omega}_t \sim N\left(\mathbf{0}, \frac{\mathbf{W}_t}{\lambda_t/E[\lambda_t]}\right), \tag{4.2}$$

where ν_t is the observation error and $\boldsymbol{\omega}_t$ is the state evolution noise, which are mutually independent and independent of all past such terms. The p-dimensional predictor vector $\mathbf{F}_t$, the $p \times p$ state evolution matrix $\mathbf{G}_t$, and the $p \times p$ evolution covariance matrix $\mathbf{W}_t$ are known at time t. The p-dimensional state vector $\boldsymbol{\theta}_t$ evolves according to (4.2) and the precisions λ_t follow a stochastic volatility variance discount model. Equation (4.1) is called the *observation equation* and (4.2) the *evolution equation*.

4.3.2 Forward Filtering

Sequential learning of the time-varying states $\boldsymbol{\theta}_t$ and stochastic precisions λ_t is facilitated by conjugate filtering of the time series y_t. The following describes standard forward filtering with conjugate normal-gamma priors as described in [West and Harrison, 1997, Chapter 4]. For each time t, denote by $\mathcal{D}_t$ all available information, here assumed to comprise just the past data with $\mathcal{D}_t = \{\mathcal{D}_{t-1}, y_t\}$.

Prior at time t. The conjugate prior for $\boldsymbol{\theta}_t$ and λ_t given information set $\mathcal{D}_{t-1}$ is normal-gamma

$$(\boldsymbol{\theta}_t, \lambda_t|\mathcal{D}_{t-1}) \sim NG(\mathbf{a}_t, \mathbf{R}_t, r_t, c_t). \tag{4.3}$$

Here $\mathbf{a}_t$ is the prior mode of $\boldsymbol{\theta}_t$, $\mathbf{R}_t$ is the $p \times p$ covariance matrix factor, degrees of freedom $r_t > 0$, and $c_t > 0$; note that c_t is the harmonic mean estimate of the residual variance $1/\lambda_t$ in (4.1) via $1/E[\lambda_t] = c_t$.

Forecasts at t. The forecast distribution $y_t|\mathcal{D}_{t-1} \sim T_{r_t}(f_t, q_t)$ is obtained by integration of the observation equation (4.1) with respect to the prior distribution (4.3). This yields the univariate T distribution with r_t degrees of freedom, mode $f_t = \mathbf{F}_t'\mathbf{a}_t$ and scale $q_t = \mathbf{F}_t'\mathbf{R}_t\mathbf{F}_t + c_t$.

Posterior at t. The posterior distribution of $\boldsymbol{\theta}_t$ and λ_t given information $\mathcal{D}_t$ follows after a conjugate posterior update as a normal-gamma distribution

$$(\boldsymbol{\theta}_t, \lambda_t | \mathcal{D}_t) \sim NG(\mathbf{m}_t, \mathbf{C}_t, n_t, s_t), \tag{4.4}$$

with mode $\mathbf{m}_t = \mathbf{a}_t + \mathbf{A}_t e_t$, covariance matrix factor $\mathbf{C}_t = (\mathbf{R}_t - \mathbf{A}_t \mathbf{A}_t' q_t) z_t$, degrees of freedom $n_t = r_t + 1$, and residual variance estimate $s_t = z_t c_t$. These parameters can be calculated after calculating the forecast error $e_t = y_t - \mathbf{F}_t \mathbf{a}_t$, volatility update factor $z_t = (r_t + e_t^2/q_t)/(r_t + 1)$, and adaptive coefficient vector $\mathbf{A}_t = \mathbf{R}_t \mathbf{F}_t / q_t$.

Evolution to time $t+1$. State $\boldsymbol{\theta}_t$ and precision λ_t are evolved in two steps; here we use beta-gamma evolutions for stochastic variances and discount learning [Prado and West, 2010, Chapter 4, Section 3.7]. Precision λ_t is evolved to time $t+1$ by applying a beta shock $\eta_{t+1} \sim Be(\beta n_t/2, (1-\beta) n_t/2)$ that increases the step-ahead prior variance of λ_{t+1} to prevent the prior (and posterior) variances from converging to zero as more observations are observed:

$$\lambda_{t+1} = \lambda_t \eta_{t+1} / \beta, \tag{4.5}$$

$$\Rightarrow \lambda_{t+1} | \mathcal{D}_t \sim G(r_{t+1}/2, r_{t+1} c_{t+1}/2). \tag{4.6}$$

Here $\beta > 0$ is a discount factor, and the degrees of freedom are $r_{t+1} = \beta n_t$, and $c_{t+1} = s_t$. This shock leaves expectation of λ_{t+1} unchanged at $E[\lambda_{t+1}|\mathcal{D}_t] = E[\lambda_t|\mathcal{D}_t] = 1/s_t$, but increases the variance by the factor $1/\beta$: $V[\lambda_{t+1}|\mathcal{D}_t] = 1/(\beta n_t s_t^2) = \frac{1}{\beta} V[\lambda_t|\mathcal{D}_t]$. The discount factor β governs the rate at which past information is discounted: small values of β discount earlier observations more quickly and give a lot of weight to recent observations, values closer to 1 make the model less responsive to new observations, and for $\beta = 1$, the model becomes a constant variance model. Conditional on λ_{t+1}, we have $\boldsymbol{\theta}_t | \lambda_{t+1}, \mathcal{D}_t \sim N(\mathbf{m}_t, \mathbf{C}_t/(c_{t+1}\lambda_{t+1}))$, and the evolution of $\boldsymbol{\theta}_t$ to time $t+1$ follows state evolution equation (4.2). The evolved prior distribution of $\boldsymbol{\theta}_{t+1} | \lambda_{t+1}, \mathcal{D}_t$ results as

$$\boldsymbol{\theta}_{t+1} | \lambda_{t+1}, \mathcal{D}_t \sim N(\mathbf{G}_{t+1} \mathbf{m}_t, \mathbf{G}_{t+1} \frac{\mathbf{C}_t}{c_{t+1}\lambda_{t+1}} \mathbf{G}_{t+1}' + \frac{\mathbf{W}_{t+1}}{c_{t+1}\lambda_{t+1}}) \tag{4.7}$$

$$=: N(\mathbf{a}_{t+1}, \mathbf{R}_{t+1}/(c_{t+1}\lambda_{t+1})), \tag{4.8}$$

where $\mathbf{a}_{t+1} = \mathbf{G}_{t+1} \mathbf{m}_t$ and $\mathbf{R}_{t+1} = \mathbf{G}_{t+1} \mathbf{C}_t \mathbf{G}_{t+1}' + \mathbf{W}_{t+1}$. The joint step-ahead prior for $\boldsymbol{\theta}_{t+1}$ and λ_{t+1} given $\mathcal{D}_t$ is then as in (4.3), with t updated to $t+1$:

$$(\boldsymbol{\theta}_{t+1}, \lambda_{t+1} | \mathcal{D}_t) \sim NG(\mathbf{a}_{t+1}, \mathbf{R}_{t+1}, r_{t+1}, c_{t+1}). \tag{4.9}$$

4.4 The Wishart Matrix Discounting Model

This section describes the Wishart Matrix Discounting Model (WDLM) for multivariate dynamic linear models. We focus on the standard variant of the traditional Beta-Bartlett Wishart discounting model [Quintana et al., 2003] for stochastic volatility; full details are in Prado and West [2010, Chapter 10, Section 4.8] whose notation we adopt here. This summary is a lightly edited reproduction of [Gruber and West, 2015a, Section 2.1].

4.4.1 Model Structure

The m-dimensional time series $\mathbf{y}_t := (y_{1t}, \dots, y_{mt})'$ is modeled via *observation equation* (4.10)

$$\mathbf{y}_t' = \mathbf{F}_t' \boldsymbol{\Theta}_t + \boldsymbol{\nu}_t', \qquad \boldsymbol{\nu}_t \sim N(\mathbf{0}, \boldsymbol{\Sigma}_t), \tag{4.10}$$

$$\boldsymbol{\Theta}_t = \mathbf{G}_t \boldsymbol{\Theta}_{t-1} + \boldsymbol{\Omega}_t, \qquad \boldsymbol{\Omega}_t \sim N(\mathbf{0}, \mathbf{W}_t, \boldsymbol{\Sigma}_t), \tag{4.11}$$

where $\mathbf{F}_t$ is a known p-dimensional predictor vector; the $p \times m$ state matrix $\boldsymbol{\Theta}_t$ evolves according to *evolution equation* (4.11) with known $p \times p$ state transition matrix $\mathbf{G}_t$ and (known) innovation covariance matrix $\mathbf{W}_t$; the $m \times m$ matrix $\mathbf{\Sigma}_t$ is a time-varying volatility matrix; and $\mathbf{\Omega}_t$ is a matrix normal innovation. Each univariate element y_{jt} of $\mathbf{y}_t$ then follows the model

$$y_{jt} = \mathbf{F}_t' \boldsymbol{\theta}_{jt} + \nu_{jt}, \qquad \nu_{jt} \sim N(0, \sigma_{jjt}), \tag{4.12}$$

$$\boldsymbol{\theta}_{jt} = \mathbf{G}_t \boldsymbol{\theta}_{j,t-1} + \boldsymbol{\omega}_{jt}, \qquad \boldsymbol{\omega}_{jt} \sim N(\mathbf{0}, \sigma_{jjt}\mathbf{W}_t), \tag{4.13}$$

where, for each $j = 1{:}m$, the state evolution vectors $\boldsymbol{\omega}_{jt}$ are the columns of $\mathbf{\Omega}_t$; the state vectors $\boldsymbol{\theta}_{jt}$ are the columns of $\mathbf{\Theta}_t$; and σ_{jjt} is the $j - th$ diagonal element of $\mathbf{\Sigma}_t$. Non-zero covariances in $\mathbf{\Sigma}_t$ induce cross-series dependencies via $Cov(\nu_{it}, \nu_{jt}) = \sigma_{ijt}$ and $Cov(\boldsymbol{\omega}_{it}, \boldsymbol{\omega}_{jt}) = \sigma_{ijt}\mathbf{W}_t$ for $i \neq j$.

4.4.2 Forward Filtering

Key analysis components involve one-step evolution, forecasting and updating, as follows. Again, denote by $\mathcal{D}_t$ all available information at time t. The step-ahead prior evolution of the volatility matrix $\mathbf{\Sigma}_t$ follows a variant of the Beta-Bartlett Wishart volatility model [Prado and West, 2010, Chapter 10, Section 4.8].

Prior at time t. The conjugate prior for $\mathbf{\Theta}_t, \mathbf{\Sigma}_t$ given $\mathcal{D}_{t-1}$ is a matrix normal, inverse Wishart distribution

$$(\mathbf{\Theta}_t, \mathbf{\Sigma}_t | \mathcal{D}_{t-1}) \sim NIW(\mathbf{a}_t, \mathbf{R}_t, r_t, \mathbf{B}_t). \tag{4.14}$$

Here $\mathbf{a}_t$ is the $p \times m$ prior mode of $\mathbf{\Theta}_t$ and $\mathbf{R}_t$ the $p \times p$ column covariance matrix; the conditional prior of $\mathbf{\Theta}_t | \mathbf{\Sigma}_t$ is matrix normal $N(\mathbf{a}_t, \mathbf{R}_t, \mathbf{\Sigma}_t)$. Parameter $r_t > 0$ is the prior degrees-of-freedom, and $\mathbf{B}_t$ is the $m \times m$ prior sum-of-squares matrix of the marginal inverse Wishart prior $\mathbf{\Sigma}_t \sim IW(r_t, \mathbf{B}_t)$; the prior mean of $\mathbf{\Sigma}_t$ is $\mathbf{B}_t/(r_t - 2)$, if $r_t > 2$.

Forecasts at t. Integration of (4.10) with respect to $p(\mathbf{\Theta}_t, \mathbf{\Sigma}_t | \mathcal{D}_{t-1})$ in (4.14) yields the multivariate T, one-step forecast distribution $\mathbf{y}_t | \mathcal{D}_{t-1} \sim T_{r_t}(\mathbf{f}_t, \mathbf{Q}_t)$ with r_t degrees of freedom, mode $\mathbf{f}_t = \mathbf{F}_t' \mathbf{a}_t$ and scale matrix $\mathbf{Q}_t = q_t \mathbf{B}_t / r_t$ where $q_t = 1 + \mathbf{F}_t' \mathbf{R}_t \mathbf{F}_t$; the forecast variance matrix is $q_t \mathbf{B}_t/(r_t - 2)$ if $r_t > 2$.

Posterior at t. The posterior of $\mathbf{\Theta}_t, \mathbf{\Sigma}_t$ follows conjugate analysis upon observation of $\mathbf{y}_t$,

$$(\mathbf{\Theta}_t, \mathbf{\Sigma}_t | \mathcal{D}_t) \sim NIW(\mathbf{m}_t, \mathbf{C}_t, n_t, \mathbf{D}_t), \tag{4.15}$$

with updated parameters $\mathbf{m}_t = \mathbf{a}_t + \mathbf{A}_t \mathbf{e}_t$, $\mathbf{C}_t = \mathbf{R}_t - \mathbf{A}_t \mathbf{A}_t' q_t$, $n_t = r_t + 1$, and $\mathbf{D}_t = \mathbf{B}_t + \mathbf{e}_t \mathbf{e}_t'/q_t$ based on adaptive coefficient vector $\mathbf{A}_t = \mathbf{R}_t \mathbf{F}_t / q_t$ and forecast error vector $\mathbf{e}_t = \mathbf{y}_t - \mathbf{f}_t$. The distribution of inverse volatility matrix $\mathbf{\Phi}_t = \mathbf{\Sigma}_t^{-1}$ follows as a Wishart distribution $W(h_t, \mathbf{D}_t^{-1})$, where $h_t = n_t + p - 1$.

Evolution to time $t + 1$. In moving ahead to time $t + 1$, the posterior (4.15) evolves to the implied prior of the form of (4.14) but with index $t \to t + 1$. The DLM state matrix $\mathbf{\Theta}_{t+1}$ evolves according to state evolution equation (4.11), and the time $t + 1$ prior parameters follow as $\mathbf{a}_{t+1} = \mathbf{G}_{t+1} \mathbf{m}_t$ and $\mathbf{R}_{t+1} = \mathbf{G}_{t+1} \mathbf{C}_t \mathbf{G}_{t+1}' + \mathbf{W}_{t+1}$. A variant of the Beta-Bartlett Wishart volatility model [Prado and West, 2010, Chapter 10, Section 4.8] is a multivariate generalization of the beta-gamma discount evolutions for the univariate DLM; here the evolved prior parameters follow as $r_{t+1} = \beta n_t$ and $\mathbf{B}_{t+1} = b_{t+1} \mathbf{D}_t$, where $b_{t+1} = (r_{t+1} + m - 1)/(n_t + m - 1)$, and $\beta \in (0, 1)$ is the discount factor governing the extent of stochastic changes in the evolution $\mathbf{\Sigma}_t \to \mathbf{\Sigma}_{t+1}$. Specifically, the evolved parameters are obtained as follows. Consider the Bartlett decomposition of the inverse volatility matrix $\mathbf{\Phi}_t$,

$$\mathbf{\Phi}_t = \mathbf{P}_t' \mathbf{U}_t' \mathbf{U}_t \mathbf{P}_t, \tag{4.16}$$

where $\mathbf{P}_t$ is the upper triangular Cholesky component of $\mathbf{D}_t^{-1}$ and the $m \times m$ matrix $\mathbf{U}_t$ is upper triangular. Under the Wishart posterior distribution of $\mathbf{\Phi}_t$, the squared diagonal entries $u_{t,i,i}^2 \sim G((h_t - i + 1)/2, 1/2)$ of $\mathbf{U}_t$ are gamma-distributed, the above-diagonal entries $u_{t,i,j} \sim N(0, 1)$ (for $j = (i+1){:}m$ and $i = 1{:}m$) are standard normally distributed, and all random entries of $\mathbf{U}_t$ are mutually independent. In evolution to $t+1$, apply beta-distributed shocks to the diagonal entries of $\mathbf{U}_t$ to obtain $\widetilde{\mathbf{U}}_{t+1} = (\widetilde{u}_{t+1,i,j})$: here the off-diagonal entries $\widetilde{u}_{t+1,i,j} = u_{t,i,j}$, $i \neq j$, are those of $\mathbf{U}_t$, and $\widetilde{u}_{t+1,i,i} = u_{t,i,i}\sqrt{\eta_{t+1,i}}$ with beta shocks $\eta_{t+1,i} \sim Be(\beta_{t+1,i}(h_t-i+1)/2, (1-\beta_{t+1,i})(h_{t,i}-i+1)/2)$ and $\beta_{t+1,i} = (\beta n_t+m-i)/(n_t+m-i)$. Taking

$$\mathbf{\Phi}_{t+1} := \frac{1}{b_{t+1}}\mathbf{P}_t'\widetilde{\mathbf{U}}_{t+1}'\widetilde{\mathbf{U}}_{t+1}\mathbf{P}_t, \tag{4.17}$$

the evolved prior distribution of the precision matrix is $\mathbf{\Phi}_{t+1}|\mathcal{D}_t \sim W(\beta n_t+m-1, (b_{t+1}\mathbf{D}_t)^{-1})$, and the evolved prior of the volatility matrix is $\mathbf{\Sigma}_{t+1}|\mathcal{D}_t \sim IW(\beta n_t, b_{t+1}\mathbf{D}_t) =: IW(r_{t+1}, \mathbf{B}_{t+1})$.

4.5 A Primer on Simultaneous Graphical DLMs

Scaling on-line Bayesian analyses of multivariate dynamic models to increasingly high-dimensions demands new modeling and computational strategies. A key need is for sequential analysis methods that are computationally accessible while applying to models of interest in many application areas, for example, macroeconomics [Koop, 2012, Nakajima and West, 2013a], financial portfolio studies [Aguilar and West, 2000, Zhou et al., 2014], commercial and governmental forecasting [Agarwal et al., 2010, Queen, 1994, Anacleto et al., 2013], large scale-networks in energy demand forecasting [Hosking et al., 2013], and neuroscience [Trejo et al., 2007, Prado, 2010].

Sparse models and efficient computation are critical to successfully scaling analyses to several hundred dimensions, as the number of covariance parameters grows quadratically. Chapter 5 introduces *Simultaneous Graphical Dynamic Linear Models (SGDLMs)* and develops their Bayesian analyses utilizing GPU-based computation. SGDLMs consist of flexible state-space models of individual series; separately, each is amenable to efficient, closed form sequential filtering and forecasting [Prado and West, 2010, West and Harrison, 1997]. Cross-series contemporaneous dependencies are captured via a sparse and dynamic simultaneous equations formulation. Section 5.2 presents a variational Bayes strategy for emulated conjugate sequential learning of SGDLMs (see Figure 4.1): we start at time t with *decoupled*, independent conjugate normal-gamma priors for each series $j = 1{:}m$; then each series is updated individually, and the independent product of the within-series posteriors is a naive approximation of the exact posterior; these parallel univariate analyses are *recoupled* using importance sampling-based reweighting of sets of direct simulations from the univariate models; the exact, joint multivariate posterior is then *decoupled* into an independent product of normal-gamma distributions using a variational Bayes strategy [Jordan et al., 1999, Jaakkola and Jordan, 2000, Jaakkola, 2000] (see summary discussions in Section 4.6); lastly, moving from time t to $t+1$, within-series prior evolutions are applied to these emulating independent normal-gamma posteriors. To scale-up the number of time series, we exploit (C++/CUDA) GPU computation [Suchard et al., 2010, Lee et al., 2010] that is ideally suited to the analysis. Univariate time series model updates and simulations are performed in parallel at each time point, and then recoupled for coherent inference and forecasting before decoupling again to move to the next time point and re-parallelization. The overall modeling and computational strategy is scalable with time series dimension as a result. Unlike Cholesky-style and factor models [e.g. Aguilar and West, 2000, Lopes et al., 2010, Pitt and Shephard, 1999, Lopes and Polson, 2010, Nakajima and West, 2013b], SGDLMs do not require an ordering of the series, which can be an obstacle to model specification. Compared to dynamic graphical models of precision matrices [e.g. Quintana and West, 1987, Carvalho and West, 2007, Wang and West, 2009], SGDLMs are scalable computationally, and flexible in allowing for patterns of change in volatility matrices that are specified implicitly via state-space models for sets of simultaneous regression parameters. Compared to the standard multivariate WDLM (Section 4.4), SGDLMs inherently define sparse representations underlying time-varying variance matrices, and allow for differing sets of predictors in each univariate series.

The SGDLM requires specification of a set of *parental* time series to use as contemporaneous predictors of each univariate series in a simultaneous equations formulation; to address this, Chapter 6 introduces a novel and practicable selection strategy for the parental sets. Furthermore, that chapter discusses several quantitative investment rules based on various portfolio utility functions of practical interest, and presents a case study in managing a 400-asset portfolio using the SGDLM combined with these rules to drive investment decisions.

$$\begin{tikzcd}
\text{Prior} & \text{Naive Posterior} & \text{Exact Posterior} & \text{VB Posterior} & \text{Prior} \\
p_{1t}(\boldsymbol{\theta}_{1,t}, \lambda_{1,t} | \mathcal{I}_t, \mathcal{D}_{t-1}) \arrow[r, "\text{observe } y_{1,t}"] & \widetilde{p}_{1t}(\boldsymbol{\theta}_{1,t}, \lambda_{1,t} | \mathcal{D}_t) \arrow[rdd, "\text{recouple}"] & & q_{1t}(\boldsymbol{\theta}_{1,t}, \lambda_{1,t} | \mathcal{D}_t) \arrow[r, "\text{evolve}"] & p_{1,t+1}(\boldsymbol{\theta}_{1,t+1}, \lambda_{1,t+1} | \mathcal{I}_{t+1}, \mathcal{D}_t) \\
p_{2t}(\boldsymbol{\theta}_{2,t}, \lambda_{2,t} | \mathcal{I}_t, \mathcal{D}_{t-1}) \arrow[r, "\text{observe } y_{2,t}"] & \widetilde{p}_{2t}(\boldsymbol{\theta}_{2,t}, \lambda_{2,t} | \mathcal{D}_t) \arrow[rd, "\text{recouple}"'] & & q_{2t}(\boldsymbol{\theta}_{2,t}, \lambda_{2,t} | \mathcal{D}_t) \arrow[r, "\text{evolve}"] & p_{2,t+1}(\boldsymbol{\theta}_{2,t+1}, \lambda_{2,t+1} | \mathcal{I}_{t+1}, \mathcal{D}_t) \\
\vdots & \vdots & p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{D}_t) \arrow[ruu, "\text{decouple}"] \arrow[ru, "\text{decouple}"'] \arrow[rd, "\text{decouple}"'] \arrow[ddd, dashed] & \vdots & \vdots \\
p_{mt}(\boldsymbol{\theta}_{m,t}, \lambda_{m,t} | \mathcal{I}_t, \mathcal{D}_{t-1}) \arrow[r, "\text{observe } y_{m,t}"] & \widetilde{p}_{mt}(\boldsymbol{\theta}_{m,t}, \lambda_{m,t} | \mathcal{D}_t) \arrow[ru, "\text{recouple}"'] & & q_{mt}(\boldsymbol{\theta}_{m,t}, \lambda_{m,t} | \mathcal{D}_t) \arrow[r, "\text{evolve}"] & p_{m,t+1}(\boldsymbol{\theta}_{m,t+1}, \lambda_{m,t+1} | \mathcal{I}_{t+1}, \mathcal{D}_t) \\
p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{I}_t, \mathcal{D}_{t-1}) & \widetilde{p}(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{D}_t) & & q(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{D}_t) & p(\boldsymbol{\Theta}_{t+1}, \boldsymbol{\Lambda}_{t+1} | \mathcal{I}_{t+1}, \mathcal{D}_t) \\
= \prod_j p_{jt|\mathcal{I}_t, \mathcal{D}_{t-1}} & = \prod_j \widetilde{p}_{jt|\mathcal{D}_t} & \propto |\mathbf{I} - \boldsymbol{\Gamma}_t| \prod_j \widetilde{p}_{jt|\mathcal{D}_t} & = \prod_j q_{jt|\mathcal{D}_t} & = \prod_j p_{j,t+1|\mathcal{I}_{t+1}, \mathcal{D}_t}
\end{tikzcd}$$

Figure 4.1: Decoupling and recoupling scheme for forward filtering in SGDLMs.

4.6 Methods for Simultaneous Graphical DLMs

This section provides a brief summary of the main basic concepts we will use in our emulated conjugate sequential learning scheme for the SGDLM in Section 5.2, *variational mean field approximation* and *importance sampling*.

4.6.1 Variational Mean Field Approximation

Variational strategies transform a given problem into a related optimization problem. Such strategies are commonly used when the problem of interest can not be solved directly; a typical example is to approximate posterior probabilities, or bounds thereof, of graphical statistical models [Jordan et al., 1999, Jaakkola and Jordan, 2000, Jaakkola, 2000].

We will focus on the variational mean field method as described in [Jordan et al., 1999, Section 6] and [Jaakkola, 2000, Section 4]. Here one approximates the target distribution with density p by a more tractable class of probability distributions with density $q_{\boldsymbol{\lambda}}$ and *variational parameters* $\boldsymbol{\lambda}$. The variational parameters are set to minimize the Kullback-Leibler divergence $KL_{p|q_{\boldsymbol{\lambda}}}$ of p from $q_{\boldsymbol{\lambda}}$,

$$\boldsymbol{\lambda} = \arg\min_{\boldsymbol{\lambda}} KL_{p|q_{\boldsymbol{\lambda}}} = \arg\min_{\boldsymbol{\lambda}} E_{q_{\boldsymbol{\lambda}}}[\log(q_{\boldsymbol{\lambda}}(\mathbf{z})/p(\mathbf{z}))], \tag{4.18}$$

where $E_{q_{\boldsymbol{\lambda}}}[\cdot]$ denotes the expectation under the approximating distribution $q_{\boldsymbol{\lambda}}$, and $\mathbf{z}$ is a random variable; the densities p and $q_{\boldsymbol{\lambda}}$ are assumed to have common support. If the class of approximating distributions $q_{\boldsymbol{\lambda}}$ contains the target distribution p, optimization rule (4.18) will select select $\boldsymbol{\lambda}$ such that $q_{\boldsymbol{\lambda}} = p$, given that the Kullback-Leibler divergence is always non-negative, and is zero, if and only if the distributions are equal [Kullback and Leibler, 1951].

4.6.2 Importance Sampling for Estimation of $E_p[\cdot]$

Importance sampling is a sampling method that can be used to estimate characteristics of a probability distribution that cannot be sampled from directly [e.g. Dunn and Shultis, 2011, Chapter 5, Section 2]. Let the target distribution have density function $p(\cdot)$, and let $\widetilde{p}(\cdot)$ denote the density function of an alternative distribution that can be sampled from. Suppose that we want to estimate the expectation $E_p[f(\boldsymbol{\Theta}, \boldsymbol{\Lambda})]$ of a functional transformation $f(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$ of a pair of random variables $(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \sim p(\cdot)$ from the target distribution. This can be done by estimating the expectation of a different functional transformation of $(\boldsymbol{\Theta}, \boldsymbol{\Lambda})$ under $\widetilde{p}$:

$$\begin{aligned} E_p[f(\boldsymbol{\Theta}, \boldsymbol{\Lambda})] &= \int f(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) p(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \, d(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \\ &= \int f(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \frac{p(\boldsymbol{\Theta}, \boldsymbol{\Lambda})}{\widetilde{p}(\boldsymbol{\Theta}, \boldsymbol{\Lambda})} \widetilde{p}(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \, d(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \\ &= E_{\widetilde{p}}[f(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) \alpha(\boldsymbol{\Theta}, \boldsymbol{\Lambda})], \end{aligned}$$

with *importance function* $\alpha(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) := \frac{p(\boldsymbol{\Theta}, \boldsymbol{\Lambda})}{\widetilde{p}(\boldsymbol{\Theta}, \boldsymbol{\Lambda})}$.

To implement, generate $i = 1{:}N$ samples $(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i) \sim \widetilde{p}(\cdot)$ from the alternative distribution $\widetilde{p}(\cdot)$ for some large number N. Assign each sample $(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i)$ *importance weight* $\alpha_i \propto \alpha(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i) = \frac{p(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i)}{\widetilde{p}(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i)}$, scaled such that $\sum_{i=1:N} \alpha_i = 1$. The importance sampling-based estimate of expectation $E_p[f(\boldsymbol{\Theta}, \boldsymbol{\Lambda})]$ under the target distribution $p(\cdot)$ is then

$$\widehat{E}_p[f(\boldsymbol{\Theta}, \boldsymbol{\Lambda})] = \sum_{i=1:N} \alpha_i f(\boldsymbol{\Theta}^i, \boldsymbol{\Lambda}^i). \tag{4.19}$$

Chapter 5

GPU-Accelerated Bayesian Learning and Forecasting in Simultaneous Graphical Dynamic Linear Models

The contents of this chapter have already been published in Gruber and West [2015b]. This chapter is a lightly edited reproduction of selected contents of the accepted manuscript. Appendix 5.B contains unpublished proofs and details underlying our variational Bayes decoupling strategy.

This chapter develops the simultaneous graphical dynamic linear model (simultaneous graphical DLM or SGDLM) as a sparse Bayesian model of multivariate stochastic volatility. Section 5.1 introduces the model structure of the SGDLM; a novel variational Bayes strategy for sequential learning is developed in Section 5.2. Section 5.3 evaluates the computational cost of the proposed strategy and elaborates on the GPU-accelerated implementation. Section 5.4 demonstrates how the implementation enables real-time Bayesian analysis of a 400-dimensional daily stock return time series and Section 5.5 finishes this chapter with additional comments.

5.1 Simultaneous Graphical DLMs

5.1.1 Model Structure: Definitions and Notation

Series-specific form. Consider the $m-$dimensional time series $\mathbf{y}_t = (y_{1t}, \ldots, y_{mt})'$, $t = 1, 2, \ldots$. Each univariate series y_{jt} is represented via a linear, normal state-space model with a traditional variance discount model of stochastic volatility [West and Harrison, 1997, Prado and West, 2010]. With the convention that all vectors are columns, the basic model form is:

$$\begin{aligned} y_{jt} &= \mathbf{F}_{jt}'\boldsymbol{\theta}_{jt} + \nu_{jt} \;=\; \mathbf{x}_{jt}'\boldsymbol{\phi}_{jt} + \mathbf{y}_{sp(j),t}'\boldsymbol{\gamma}_{jt} + \nu_{jt}, & (5.1)\\ \boldsymbol{\theta}_{jt} &= \mathbf{G}_{jt}\boldsymbol{\theta}_{j,t-1} + \boldsymbol{\omega}_{jt}, & (5.2)\end{aligned}$$

where the observation error $\nu_{jt} \sim N(0, \lambda_{jt}^{-1})$ and state evolution error $\boldsymbol{\omega}_{jt} \sim N(\mathbf{0}, \mathbf{W}_{jt}/(c_{jt}\lambda_{jt}))$ are independent, zero mean normals and are independent of all past such terms and across all series $i \neq j$. Predictor vectors $\mathbf{x}_{jt}$ and $\mathbf{y}_{sp(j),t}$ are catenated to define $\mathbf{F}_{jt}' = (\mathbf{x}_{jt}',\ \mathbf{y}_{sp(j),t}')$; corresponding state vectors $\boldsymbol{\phi}_{jt}$ and $\boldsymbol{\gamma}_{jt}$ similarly define $\boldsymbol{\theta}_{jt}' = (\boldsymbol{\phi}_{jt}',\ \boldsymbol{\gamma}_{jt}')$. Here $\mathbf{x}_{jt}$ is a series-specific vector of exogenous predictors, while $\mathbf{y}_{sp(j),t}$ is a vector of contemporaneous values of some of the other series, indexed by $sp(j) \subseteq \{1{:}m\} \setminus \{j\}$ and called the *simultaneous parental set* for series j. State vector $\boldsymbol{\phi}_{jt}$ has dimension $p_{j\phi}$, state vector $\boldsymbol{\gamma}_{jt}$ has dimension $p_{j\gamma} = |sp(j)|$, so that $\boldsymbol{\theta}_{jt}$ has dimension $p_j = p_{j\phi} + p_{j\gamma}$. The model context assumes the parental sets, and hence their dimensions, are fixed over time (at least, fixed over the period of time chosen for analysis) as part of the model specification. Conditional on the state vector and exogenous predictors, write $\mu_{jt} = \mathbf{x}_{jt}'\boldsymbol{\phi}_{jt}$. The state vector evolves according to the linear evolution with state matrix $\mathbf{G}_{jt}$. The states $\boldsymbol{\theta}_{jt}$ and λ_{jt} are learned sequentially, while all other quantities are specified through prior or model choices.

Equations (5.1, 5.2) define a set of coupled, dynamic simultaneous equations that cohere across $j = 1{:}m$, representing a set of dynamic structural equations for the multivariate model of $\mathbf{y}_t$ [e.g. Palomo et al., 2007, and references therein].

Across-series form. Define $\boldsymbol{\Theta}_t = \{\boldsymbol{\theta}_{1t}, \ldots, \boldsymbol{\theta}_{mt}\}$ and $\boldsymbol{\Lambda}_t = \{\lambda_{1t}, \ldots, \lambda_{mt}\}$, the sets of m state vectors and precisions, and

$$\boldsymbol{\mu}_t = \begin{pmatrix} \mu_{1t} \\ \mu_{2t} \\ \vdots \\ \mu_{m-1,t} \\ \mu_{mt} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Gamma}_t = \begin{pmatrix} 0 & \gamma_{1,2,t} & \gamma_{1,3,t} & \cdots & \gamma_{1,m,t} \\ \gamma_{2,1,t} & 0 & \gamma_{2,3,t} & \cdots & \gamma_{2,m,t} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_{m-1,1,t} & \cdots & \gamma_{m-1,m-2,t} & 0 & \gamma_{m-1,m,t} \\ \gamma_{m,1,t} & \gamma_{m,2,t} & \cdots & \gamma_{m,m-1,t} & 0 \end{pmatrix}.$$

where we extend the γ_* notation so that $\gamma_{jht} = 0$ for each $h \notin sp(j)$, $j = 1{:}m$.

It follows that

$$\mathbf{y}_t \sim N(\mathbf{A}_t \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{5.3}$$

where

$$\mathbf{A}_t = (\mathbf{I} - \boldsymbol{\Gamma}_t)^{-1} \quad \text{and} \quad \boldsymbol{\Omega}_t \equiv \boldsymbol{\Sigma}_t^{-1} = (\mathbf{I} - \boldsymbol{\Gamma}_t)' \boldsymbol{\Lambda}_t (\mathbf{I} - \boldsymbol{\Gamma}_t). \tag{5.4}$$

Practical models will typically have small parental sets $sp(j)$ so the resulting $\boldsymbol{\Gamma}_t$ matrix will be sparse. That, coupled with the state evolution models for the $\boldsymbol{\gamma}_{jt}$, defines a flexible class of multivariate volatility models for the implied variance matrix $\boldsymbol{\Sigma}_t$ and its inverse– the precision matrix– $\boldsymbol{\Omega}_t$. Very sparse $\boldsymbol{\Gamma}_t$ can imply (albeit less) sparse precision matrices; the correspondence of zeros in $\boldsymbol{\Omega}_t$ with conditional independencies in the resulting Gaussian graphical models [Carvalho and West, 2007] underlie the designation of this class of models as *simultaneous graphical dynamic linear models.* With even modest m, practical models will have relatively small parental sets. If the maximum parental set size is k, the model has mk non-zero elements in $\boldsymbol{\Gamma}_t$ so that $k < (m-1)/2$ means $\boldsymbol{\Omega}_t$ is not over-parametrized. Our motivation for these models and their expected utility is in problems with increasingly large m; our financial time series example in Section 5.4 has $m = 400, k = 10$, so represents 79,800 precision parameters in terms of just 4,000 simultaneous parental parameters.

We note connections with the use of simultaneous/structural specifications in spatial analysis on lattice data. Simultaneous autoregressive (SAR) models define joint distributions of outcomes on a spatial lattice via univariate conditional models based on sparse simultaneous parental (or neighboring) sets in a form similar to that adopted here [e.g. Anselin, 1988, De Oliveira and Song, 2008, Whittle, 1954]. Part of the inspiration for the work here derives from the utility of such models in spatial studies, especially with regard to scalability to larger problems [Mukherjee et al., 2014].

5.1.2 Sequential Learning: Structure and Challenges

Sequential analysis moves over time t and updates summary posterior distributions for model state vectors and precisions as new data is observed. At time $t-1$, denote historical data and information by $\mathcal{D}_{t-1}$. Evolving to time t, this information set updates to $\mathcal{D}_t = \{\mathbf{y}_t,\ \mathcal{I}_t,\ \mathcal{D}_{t-1}\}$ where $\mathcal{I}_t$ denotes any additional information or model changes used between times $t-1$ and t, for example, the specification of the state evolution matrices $\mathbf{G}_{jt}$, evolution variances $\mathbf{W}_{jt}$, or changes in the simultaneous parental sets $sp(j)$ [West and Harrison, 1989, 1997, Chapter 11]. Then, analysis over times $t-1$ to t involves: (i) using the time $t-1$ posterior $p(\boldsymbol{\Theta}_{t-1}, \boldsymbol{\Lambda}_{t-1} | \mathcal{D}_{t-1})$ to infer the prior for time t, namely $p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{I}_t, \mathcal{D}_{t-1})$; (ii) using this prior to compute forecast distributions for $\mathbf{y}_t$ and future outcomes beyond time t, as desired; (iii) on moving to time t, updating to the current posterior $p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t | \mathcal{D}_t)$.

The full multivariate model raises computational challenges due to the non-linearities in state vectors in (5.3, 5.4). The time t likelihood function for $\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t$ is directly derived from the $m-$variate normal density of (5.3), as

$$p(\mathbf{y}_t | \boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t) \propto |\mathbf{I} - \boldsymbol{\Gamma}_t| \prod_{j=1:m} p(y_{jt} | \boldsymbol{\theta}_{jt}, \lambda_{jt}) \tag{5.5}$$

where the product is of normal densities from the set of univariate models, namely $y_{jt} \sim N(\mathbf{F}'_{jt}\boldsymbol{\theta}_{jt}, \lambda_{jt}^{-1})$ for $j = 1{:}m$. As a result, the time t updated posterior is

$$p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t|\mathcal{D}_t) \propto |\mathbf{I} - \boldsymbol{\Gamma}_t| \; p(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t|\mathcal{I}_t, \mathcal{D}_{t-1}) \prod_{j=1:m} p(y_{jt}|\boldsymbol{\theta}_{jt}, \lambda_{jt}). \tag{5.6}$$

The determinant factor here induces the computational challenges. Apart from the special cases of *compositional models* [Nakajima and West, 2013a, Zhao and West, 2014] in which $\boldsymbol{\Gamma}_t$ is triangular with a diagonal of zeros and so $|\mathbf{I} - \boldsymbol{\Gamma}_t| = 1$, this determinant term contributes to the likelihood for the $\boldsymbol{\gamma}_{jt}$ vectors. In very sparse models, the determinant term can tend to be quite diffuse as a function of $\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t$ when compared to the product of individual likelihood terms. However, it matters generally; it arises theoretically to ensure positive definiteness and symmetry of $\boldsymbol{\Omega}_t$. In our financial time series example in Section 5.4 we demonstrate some of the negative practical consequences of ignoring this term.

5.2 Model Decoupling/Recoupling Strategy

5.2.1 Motivation and Summary

The forms of (5.5, 5.6) suggest opportunity to exploit separate, parallel analyses of each univariate series in order to define a sequential computational strategy for the full multivariate model. Each univariate DLM of (5.1, 5.2) is a linear, normal state-space model for $\boldsymbol{\theta}_{jt}$ that, when coupled with the traditional variance discount model for stochastic volatilities $\sqrt{\lambda_{jt}^{-1}}$, is amenable to standard forward filtering and forecasting analysis in closed form [West and Harrison, 1997]. Directly implemented for series j without regard to the other univariate models, this analysis involves sequentially updated normal/gamma priors $p(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_{t-1})$, and posteriors $p(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)$, with simple, closed-form updates. Were we to simply use these separate univariate DLMs in parallel and assume independence across series j, then the implied joint priors and posteriors for the $\{\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t\}$ would factorize. Thus, the naive approximation of ignoring the determinant factor in (5.5, 5.6) is akin to running a univariate DLM on each series individually. We use this idea to define a computational strategy that improves on this naive approach while retaining the analytical tractability of the time evolution and update steps.

The steps involved in our *decoupling/recoupling* strategy are fully detailed in Section 5.2.2. In summary here, standing at time t before observation of $\mathbf{y}_t$, the analysis proceeds as follows:

A. At time t, adopt *decoupled* priors $p(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{I}_t, \mathcal{D}_{t-1})$ assumed independent over $j = 1{:}m$.

B. To predict $\mathbf{y}_{t+k}$ into the future $k = 0, 1, \ldots,$ simulate these independent priors and use sampled values to evaluate aspects of full multivariate forecast distributions.

C. At time t on observing $\mathbf{y}_t$, perform parallel, independent updates to posteriors in the m DLMs and take their product to yield a *naive* posterior approximation $\tilde{p}(\boldsymbol{\Theta}_{jt}, \boldsymbol{\Lambda}_{jt}|\mathcal{D}_t)$.

D. *Recouple* the analyses by evaluating the exact posterior using importance sampling.

E. *Decouple* the series by emulating the exact posterior by a product of margins over $j = 1{:}m$ using variational Bayes.

F. Apply state evolutions independently over $j = 1{:}m$ to move to time $t + 1$.

Note that, in special cases when $\boldsymbol{\Gamma}_t$ is chosen to be– or just happens to be– diagonal, we have a compositional (directed) graphical model specification, with $|\mathbf{I} - \boldsymbol{\Gamma}_t| = 1$. The analysis is then closed-form: the naive posterior from C equals the exact posterior, and steps D and E can be omitted. In the kinds of practical problems of focus– with more than a few series– the compositional approach is typically a non-starter since it requires that the modeler can define a strict ordering of the series. While this can be done based on substantive reasoning with relatively few series, choosing an ordering is otherwise challenging and arbitrary, and results heavily dependent on its choice.

5.2.2 Model Emulation: Decoupling for Forward Filtering

A. Time t prior. The prior at time t for model states and volatilities is a product of a decoupled set of m conjugate normal/gamma priors (see Section 4.3.2), namely

$$p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t | \mathcal{I}_t, \mathcal{D}_{t-1}) = \prod_{j=1:m} p_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt} | \mathcal{I}_t, \mathcal{D}_{t-1}) \tag{5.7}$$

where $p_{jt}(\cdot|\cdot)$ denotes the density function of

$$(\boldsymbol{\theta}_{jt}, \lambda_{jt} | \mathcal{I}_t, \mathcal{D}_{t-1}) \sim NG(\mathbf{a}_{jt}, \mathbf{R}_{jt}, r_{jt}, c_{jt}). \tag{5.8}$$

This uses standard notation [Prado and West, 2010] for the normal/gamma

$$\begin{aligned}(\boldsymbol{\theta}_{jt} | \lambda_{jt}, \mathcal{I}_t, \mathcal{D}_{t-1}) &\sim N(\mathbf{a}_{jt}, \mathbf{R}_{jt}/(c_{jt}\lambda_{jt})),\\ (\lambda_{jt} | \mathcal{I}_t, \mathcal{D}_{t-1}) &\sim G(r_{jt}/2, r_{jt}c_{jt}/2),\end{aligned}$$

in which $N(\mathbf{a}, \mathbf{A})$ is multivariate normal with mean $\mathbf{a}$ and variance matrix $\mathbf{A}$, and $G(r, rc)$ is gamma with shape r, rate rc, and mean $r/(rc) = 1/c$. The implied $\boldsymbol{\theta}_{jt}$ margin of (5.8) is multivariate T with r_{jt} degrees of freedom, mode $\mathbf{a}_{jt}$ and scale matrix $\mathbf{R}_{jt}$; the marginal variance matrix is $\mathbf{R}_{jt}r_{jt}/(r_{jt} - 2)$ in usual cases that $r_{jt} > 2$.

B. Time t predictions. The one-step ahead predictive distribution is efficiently simulated by drawing from the set of m independent normal/gamma priors above, so defining a simulation sample $\{\mathbf{\Theta}_t^r, \mathbf{\Lambda}_t^r\}$ from this emulating prior, where the superscript r indexes Monte Carlo samples for prediction, with $r = 1 : R$ for some (large) sample size R. Each sampled value then defines Monte Carlo values of one-step forecast moments $\mathbf{A}_t^r \boldsymbol{\mu}_t^r, \mathbf{\Sigma}_t^r$ in (5.3, 5.4). Predictions more than one-step ahead follow similarly. Conditional on sampled moments, the resulting conditionally normal predictive distributions can be summarized or simulated for predictive inferences.

C. Naive time t posterior updates. Standard updating equations applied independently and in parallel lead to

$$(\boldsymbol{\theta}_{jt}, \lambda_{jt} | \mathcal{D}_t) \sim NG(\widetilde{\mathbf{m}}_{jt}, \widetilde{\mathbf{C}}_{jt}, \widetilde{n}_{jt}, \widetilde{s}_{jt}) \tag{5.9}$$

with density functions denoted by $\widetilde{p}_{jt}(\cdot|\cdot)$. See Section 4.3.2 for details and the explicit updating formulæ. The resulting naive posterior approximation to $p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t | \mathcal{D}_t)$ is then

$$\widetilde{p}(\mathbf{\Theta}_t, \mathbf{\Lambda}_t | \mathcal{D}_t) = \prod_{j=1:m} \widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt} | \mathcal{D}_t), \tag{5.10}$$

which ignores the determinant term in (5.6).

D. Recoupling to exact time t posterior. We know that the exact posterior is

$$p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t | \mathcal{D}_t) \propto |\mathbf{I} - \mathbf{\Gamma}_t| \prod_{j=1:m} \widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt} | \mathcal{D}_t). \tag{5.11}$$

We use importance sampling (Section 4.6.2) to estimate features of the exact joint posterior distribution in (5.11). Draw N independent samples from the m-variate naive posterior approximation in (5.10); the product form of the density can be exploited for independent and parallelized sampling from the m within-series posteriors in (5.9); these are then combined to define the full sample $\{\mathbf{\Theta}_t^i, \mathbf{\Lambda}_t^i\}$. Compute and normalize importance weights

$$\alpha_{ti} \propto \frac{p(\mathbf{\Theta}_t^i, \mathbf{\Lambda}_t^i | \mathcal{D}_t)}{\widetilde{p}(\mathbf{\Theta}_t^i, \mathbf{\Lambda}_t^i | \mathcal{D}_t)} \propto |\mathbf{I} - \mathbf{\Gamma}_t^i|, \quad i = 1 : N. \tag{5.12}$$

These samples and weights

$$\{\mathbf{\Theta}_t^i,\ \mathbf{\Lambda}_t^i,\ \alpha_{ti}\}, \quad i = 1 : N, \tag{5.13}$$

define an importance sample that can be used, for example, to estimate expectations under exact joint posterior distribution $p(\cdot|\cdot)$.

Note that the importance sample weights do not depend on the values of the simulated $\phi^i_{jt}, \lambda^i_{jt}$; they depend only on the simultaneous coefficients γ_{jt}. This reflects the view that the parallel conjugate models will effectively emulate the full multivariate model in inferring series-specific states and volatilities, while some "corrections" will be needed for inferences on the parental states that explicitly define cross-series structure. Importance sampling is the natural and elegant approach to making these corrections. The univariate series/models are *decoupled* for updates and direct simulation, and then *recoupled* for importance sampling targeting the exact posterior.

E. Decoupling of time t posterior. To move ahead to the next time point, *decouple* the posterior from part D into a product of conjugate forms across series $j = 1{:}m$. A standard variational Bayes (VB) approach, or mean-field approximation (Section 4.6.1) emulates the exact posterior by an independent product of normal/gammas

$$q(\boldsymbol{\Theta}_t, \boldsymbol{\Lambda}_t|\mathcal{D}_t) \propto \prod_{j=1:m} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t) \tag{5.14}$$

with components

$$(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t) \sim NG(\mathbf{m}_{jt}, \mathbf{C}_{jt}, n_{jt}, s_{jt}). \tag{5.15}$$

The variational parameters $\mathbf{m}_{jt}$, $\mathbf{C}_{jt}$, n_{jt}, and s_{jt} are chosen to minimize the Kullback-Leibler divergence[1] $KL_{q|p}$ of the emulating posterior distribution $q(\cdot|\cdot)$ from the exact posterior distribution $p(\cdot|\cdot)$. Using $E_p[\cdot]$ to denote expectations under $p(\cdot|\cdot)$, standard theory [e.g. West and Harrison, 1997, Chapter 12, Section 3] implies that:

- $\mathbf{m}_{jt} = E_p[\lambda_{jt}\boldsymbol{\theta}_{jt}]/E_p[\lambda_{jt}]$,
- $\mathbf{V}_{jt} = E_p[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})']$,
- $d_{jt} = E_p[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'\mathbf{V}_{jt}^{-1}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})]$,
- n_{jt} is the unique value that satisfies
  $\log(n_{jt} + p_j - d_{jt}) - \psi(n_{jt}/2) - (p_j - d_{jt})/n_{jt} - \log(2E_p[\lambda_{jt}]) + E_p[\log \lambda_{jt}] = 0$,
- $s_{jt} = (n_{jt} + p_j - d_{jt})/(n_{jt}E_p[\lambda_{jt}])$, and
- $\mathbf{C}_{jt} = s_{jt}\mathbf{V}_{jt}$.

Appendix 5.B provides mathematical proofs and a detailed derivation of the variational parameters. The expectations $E_p[\cdot]$ with respect to the exact posterior distribution can be easily evaluated using the importance sample $\{\boldsymbol{\Theta}^i_t, \boldsymbol{\Lambda}^i_t, \alpha_{ti}\}$, and the variational parameters are easily computed, with n_{jt} requiring a (trivial) iterative numerical approach; see Appendix 5.A.4. Both the conceptual basis and technical aspects of mapping to sets of conjugate forms has a long history in the Bayesian dynamic modeling and forecasting literature [e.g. Harrison and Stevens, 1971, Alspach and Sorenson, 1972, Harrison and Stevens, 1976, Smith and West, 1983, West and Harrison, 1997, Chapter 12, Section 3.4].

F. Evolution to time $t+1$. Moving ahead one time point, the states evolve via independent models of (5.2). This results in evolved priors as given in (5.7) in part A above, but with time index t updated to $t+1$. Formulæ are given in Section 4.3.2, simply following standard DLM theory and notation [West and Harrison, 1997].

[1] For any random quantity $\mathbf{z}$, the KL divergence of a distribution with density $q(\mathbf{z})$ from one with density $p(\mathbf{z})$ is $KL_{q|p} = E_p\big[\log\{p(\mathbf{z})/q(\mathbf{z})\}\big]$. Here densities are continuous, discrete or mixed and have common support, and $E_p[\cdot]$ is the expectation under $p(\cdot)$.

5.2.3 KL Divergence and IS-VB Strategy

The VB-based posterior of (5.14) represents an improvement over the initial $\widetilde{p}(\cdot|\cdot)$ of (5.10) as it minimizes $KL_{q|p}$ over all $q(\mathbf{\Theta}_t, \mathbf{\Lambda}_t)$ that are products of m normal/gamma forms for the $(\boldsymbol{\theta}_{jt}, \lambda_{jt}),\ j = 1{:}m.$ It turns out that the importance sampling weights α_{ti} in (5.13) provide a direct assessment of the value of the divergence $KL_{\widetilde{p}|p}$. This general but, apparently, not well-known result relating KL divergences to importance sampling weights is of utility here as well as more broadly. Specifically, write $H_N = \sum_{i=1:N} \alpha_{ti} \log(N\alpha_{ti})$, the entropy of the importance sampling weights α_{ti} relative to a set of N uniform weights– one measure of efficacy of importance samplers [e.g. West, 1993]. It is easily shown that, as $N \to \infty$, $H_N \to KL_{\widetilde{p}|p}$; hence, the relative entropy gives a direct estimate of the KL divergence. It can also be shown that $H_N \leq N \sum_{i=1:N} \alpha_{ti}^2 - 1 = N/S_N - 1$ where S_N is effective sample size; for large N, the limiting value of S_N/N is bounded above by $1/(1 + KL_{\widetilde{p}|p})$.

Hence H_N gives an estimate of the upper bound of the minimized divergence; if H_N is already small– based on calibrating to effective sample size as above– then we are assured of closeness of the revised VB-based posterior approximation. Furthermore, the KL divergence of any subset of parameters $(\mathbf{\Theta}_t, \mathbf{\Lambda}_t)$ cannot exceed the divergence on the full set. This makes the latter an operational upper bound on divergences of the approximating marginal posteriors in any of the m individual models. If the overall approximation is good, we do not have to monitor the margins on models $j = 1{:}m.$

A further positive theoretical feature relates to evolution from time t to $t+1$. The KL-optimized product of normal/gamma posteriors for $(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t)$ evolves to a similar analytic form for the time $t+1$ prior $p(\mathbf{\Theta}_{t+1}, \mathbf{\Lambda}_{t+1}|\mathcal{I}_{t+1}, \mathcal{D}_t)$. Now, we know that KL divergence decreases through convolutions; hence, the divergence of this normal/gamma product for $(\mathbf{\Theta}_{t+1}, \mathbf{\Lambda}_{t+1})$ is a better approximation of the exact time $t+1$ prior than was the case for the time t posterior. If we have a high-quality posterior approximation at time t, then the situation only improves following evolution.

5.3 GPU-Accelerated Implementation

5.3.1 General comments

The analysis strategy of Section 5.2 is ideally suited to distributed implementation on computers with graphics processing units (GPUs). GPUs feature hundreds or thousands of compute cores that can be used to execute single instruction, multiple data (SIMD) operations in a massively parallel mode (currently common multi-core desktop processors typically have no more than eight cores). Whenever the same set of numerical operations has to be performed many times on different data, GPU-accelerated implementations offer the potential to vastly outperform CPU equivalents, based on distributing these computations to cores in parallel. Our model and computational development, and accompanying code, contribute to the growing body of literature linked to Bayesian statistical computations that are inherently enabled via GPU implementations [e.g. Suchard et al., 2010, Lee et al., 2010] due to being simply ideally suited to the GPU hardware/software model; these references also discuss the relative speed-up that can be achieved over CPU computation. We note some specifics related to each of the steps in the analysis of Section 5.2.

5.3.2 Predictive Computations

Computations for predictive distributions (Section 5.2-B) are immediately parallelizable, exploiting the decoupling/recoupling strategy. For one-step ahead predictions, Monte Carlo draws $\{\boldsymbol{\theta}_t^r, \lambda_t^r\}$, $r = 1 : R$, from the independent priors at time t in (5.7) are simulated in parallel. The sampled states and volatilities are then sent to the CPU to combine and compute the implied Monte Carlo values of one-step forecast moments $\mathbf{A}_t^r \boldsymbol{\mu}_t^r, \mathbf{\Sigma}_t^r$ in (5.3, 5.4). It is then trivial to numerically summarize and/or simulate the one-step ahead predictive distribution $p(\mathbf{y}_t|\mathcal{I}_t, \mathcal{D}_{t-1})$ from this Monte Carlo sample of means and variance matrices. Predictions more than one-step ahead follow similarly, and again exploit distributed computation as the series-specific states and volatilities evolve independently in state (5.4).

5.3.3 Posterior Update Computations

Updates and simulations of naive posterior approximations in Section 5.2-C,D are each immediately parallelizable. The analytic update computations of parameters of the naive posterior approximations are trivially computed in parallel, and then fed into parallel cores for simulation of the m posteriors in parallel; this is distributable over both m and the Monte Carlo sample size N to maximally utilize the capacity of available GPU cores. Following simulation, the Monte Carlo samples are recoupled, i.e., returned to the CPU for evaluation and normalization of the importance sampling weights α_{ti}.

Based on the multivariate importance samples $\{\mathbf{\Theta}_t^i, \mathbf{\Lambda}_t^i, \alpha_{ti}\}$, we can then trivially compute summaries of the weights to monitor Monte Carlo accuracy, including the effective sample size and entropy measures $S_N,\ H_N$ discussed in Section 5.2.3.

5.3.4 Computational Costs

Let M represent Monte Carlo sample size, whether $M = R$ for forward sampling for prediction, or $M = N$ in importance sampling for posterior updates (often, we will simply take $R = N$ so that M is the common value). Scaling computations in M is then a critical interest.

The lead complexity of sampling $\{\mathbf{\Theta}_t, \mathbf{\Lambda}_t\}$ is $\mathcal{O}(Mmp_{\max}^2)$ where $p_{\max} = \max_{j=1:m} p_j$. Computing a determinant $|\mathbf{I} - \mathbf{\Gamma}_*|$ is the most expensive operation, with a cost of $\mathcal{O}(Mm^3)$. In $k-$step forecasting for $k \geq 1$, simulation of states $\{\mathbf{\Theta}_{t+k}, \mathbf{\Lambda}_{t+k}\}$ uses previously sampled states $\{\mathbf{\Theta}_{t+k-1}, \mathbf{\Lambda}_{t+k-1}\}$ and incurs additional costs of $\mathcal{O}(Mmp_{\max}^3)$ for each k. Then inverting $\mathbf{I} - \mathbf{\Gamma}_{t+k}$ costs $\mathcal{O}(Mm^3)$. So the overall computational cost of forecasting at each of $k-$steps ahead is $\mathcal{O}(kM(m^3 + mp_{\max}^2))$ as $M \to \infty$. The order of computational costs in M, m is the same for posterior decoupling/recoupling updates and for forecasting, while the actually incurred costs will increase linearly with the numbers of steps k that we choose to forecast.

5.4 Evaluation: Stock Return Study

5.4.1 Data and Study Set-Up

We analyze daily log returns of $m = 400$ S&P stocks. In a simple class of SGDLMs, we evaluate $1-$step ahead forecasts for both selected individual series and across all series. We study of the effects of ignoring the coupling of the set of simultaneous model equations, to bear out the utility of the decoupling/recoupling strategy. We also compare the results with those from a benchmark analysis using the standard Wishart discount model of multivariate stochastic volatility [WDLM; West and Harrison, 1997, Chapter 16, Section 4].

Our data represent $m = 400$ current members of the S&P 500 index, restricting to those 400 that were continuously listed from October 2000 to October 2013, our study period. The data are daily log returns, which are differences in daily log prices. We use the first 845 daily observations, $T_1 = 1 : 845$ (up to December 2003) as training data for an initial exploratory analysis to define the simultaneous parental sets for each of the $m = 400$ series. Based on these chosen parental sets, we then use the following 522 daily observations, $T_2 = 846{:}1{,}367$ (from January 2004 through December 2005) as further training data to evaluate and select suitable discount factors for the dynamic models, and to provide priors for analysis of the following test data. The test data are the remaining 2,044 observations, $T_{\text{test}} = 1{,}368{:}3{,}411$ (from January 2006 through October 2013). This provides an honest sequential forecasting analysis and evaluation on this substantial series of hold-out/test data. In both training and test data analyses, the Monte Carlo sample sizes R, for forecasting, and N, for IS-based posterior updates, were set at $R = N = 10{,}000$.

5.4.2 Model Structure and Specification via Training Data Analysis

SGDLM form. For each of the 400 univariate series, we use the local-level (random walk) for a time-varying trend DLM [West and Harrison, 1997, Chapter 2] common in studies of financial returns [e.g. Aguilar and West, 2000, Nakajima and West, 2013b]. This is coupled with a specified set of 10 simultaneous parents, whose coefficients also evolve via random walks. In (5.1), we then have $p_{j\phi} = 1$, $\mathbf{x}_{jt} = 1$ and $p_{j\gamma} = 10$. The state evolution models of (5.2) have $\mathbf{G}_{jt} = \mathbf{I}$, the 11-dimensional identity matrix. We use the standard block discounting approach to specify the variance matrices $\mathbf{W}_{jt}$ [West and Harrison, 1997, Chapter 6, Section 3.2]. Specifically, each $\mathbf{W}_{jt}$ is defined using two discount factors $\delta_\phi, \delta_\gamma$, each in $(0, 1)$, that determine evolution variance matrix block components for level and simultaneous parental coefficients, respectively.

Simultaneous parental sets. We used the initial 3 years of data, $T_1 = 1 : 845$ over 2003, for exploratory analysis to select the $sp(j)$ for each $j = 1 : 400$. On this initial training data set, we simply ran separate, univariate DLMs with a local level and with *all* the remaining 399 series as simultaneous parents. We then chose $p_j = 10$ series to define each $sp(j)$ by selecting those with the largest estimated effect sizes over the later part of the training data.

The set of simultaneous parents obviously plays a key role in model fit and forecasting. Here, we emphasize that more formal model selection is not a theme in the current paper. In practice, we operate with a chosen set of parental sets over given periods of time, refreshing/modifying the parental sets periodically via off-line analysis, and/or using multiple such sets– a restricted number– and engaging in model averaging. A future paper will address this question of model uncertainty. The current paper takes the parental sets as given– based on detailed exploratory analysis of training data– the focus and contributions involving the SGDLM modeling innovation and the decoupling/recoupling computational strategy. We show in this example that the enhancements under this strategy are not merely theoretical considerations, but significantly improve forecasting in this $m = 400-$dimensional setting.

WDLM form. The benchmark model for comparison is a local-level DLM with Wishart discount-based multivariate volatility [West and Harrison, 1997, Chapter 16, Section 4] for the full $400-$dimensional time series. This uses a discount factor δ to define the evolution variances of the local levels as $\mathbf{W}_t = \frac{1-\delta}{\delta}\mathbf{C}_{t-1}$, and a discount factor β for the Wishart discount-based evolution of the full 400×400 observation variance matrix. Initial priors are taken as $\mathbf{m}_0 = 0$, $\mathbf{C}_0 = 0.001$, $\mathbf{S}_0 = 0.1\mathbf{I}$ and $n_0 = 5$ in the notation of the above reference, and the resulting forward filtering and forecasting equations are as detailed in Theorem 16.4 of West and Harrison [1997].

Discount factor specification. We ran the SGDLM analysis on the second training set of data, $T_2 =$ 846:1,367. This initialized at $t = 846$ with priors of (5.8) is based on: $r_{j,846} = 5$ and $s_{j,845} = 0.001$, $\mathbf{a}_{j,846} = \mathbf{0}$ and $\mathbf{R}_{j,846} = \text{diag}(0.0001, 0.01, \cdots, 0.01)$. This analysis was used to explore the impact of varying the state discount factors $\delta_\phi, \delta_\gamma$ and the volatility discount $\beta_j = \beta$, all assumed the same across series j. This settled on chosen values $\beta_j = 0.98$, $\delta_\phi = 0.98, \delta_\gamma = 0.99$ based on standard evaluation of one-step ahead forecasting accuracy.

We then reran the analysis but without the IS-VB step of the computations; i.e., simply using the parallel, independently updated set of models. This analysis led to choices $\beta_j = 0.98$ and $\delta_\phi = 0.98$ again, but we found improved $1-$step ahead forecasting with an appropriately higher discount factor $\delta_\gamma = 0.999$ on the parental predictors. Below we evaluate forecasting on the hold-out/test data using this incoherent/approximate set of independent model as well using the full decoupled/recoupled analysis. This differential choice of discount factors makes this an honest comparison as we begin the analysis of test data with optimized parameter specification for each strategy.

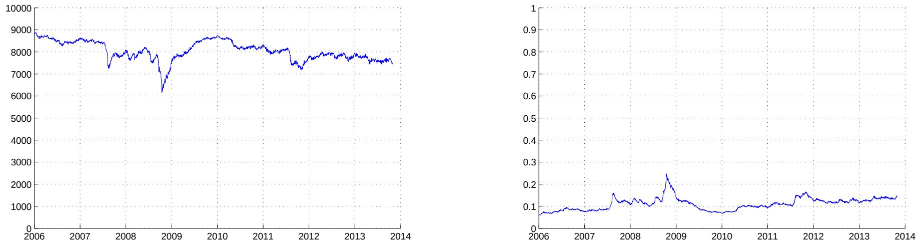
A parallel analysis of the benchmark WDLM was similarly evaluated, and led to optimal discount factors of $\delta = 0.98$ and $\beta = 0.9975$.

5.4.3 Forecasts of Stock Returns: Test Data Analysis

We review sequential learning and forecasting in the test data period, $T_{\text{test}} = 1{,}368{:}3{,}411$, with a number of summaries. This includes aspects of Monte Carlo accuracy, and forecast assessments for individual time series as well as in the aggregate. Part of this includes analyzing coverage rates of $1-$step ahead forecast intervals and validation against observed values. The initial priors at time $t = 1{,}368$ are, in each analysis, simply those evolved from the corresponding posteriors at time $t-1$, the last day of the training data period.

Analysis of the IS-VB Strategy

Figure 5.1 shows the effective sample size S_N of the IS-VB step at each time point t. Based on the $N = 10{,}000$ samples, S_N exceeds 7,000 during approximately 98% of the test time period T_{test}. There is a short-lived drop to about 6,100 during a period of extreme market stress following the collapse of Bear Sterns in September 2008. In the range from 7,000 to 9,000, the typical effective sample size indicates excellent performance of the importance sampler. Figure 5.1 also shows the corresponding IS-based estimate H_N of the KL divergence $KL_{\widetilde{p}|p}$ of $\widetilde{p}(\cdot)$ from the $p(\cdot)$ at each time. Bounding the KL of the variational Bayes posterior and the exact posterior from above implies that the variational Bayes posterior will be very close to the exact posterior.



Effective sample size S_N

KL divergence $KL_{\widetilde{p}|p}$

Figure 5.1: S_N and $KL_{\widetilde{p}|p}$ during the test data time period T_{test}.

Aggregate Analysis

Table 5.1 shows the average coverage rates of forecasts across all series and the entire test data time period $T_{\text{test}} = 1{,}368{:}3{,}411$. Forecast intervals from the benchmark WDLM are broadly similar to those from the SGDLM without IS-VB decoupling/recoupling (steps D and E of Section 5.2; indicated as “no IS-VB” in the table). The full SGDLM analysis using the importance sampling/variational Bayes strategy (indicated simply as “IS-VB” in the table) is more accurate–across all coverage levels– than the analysis that does not use the IS-VB strategy.

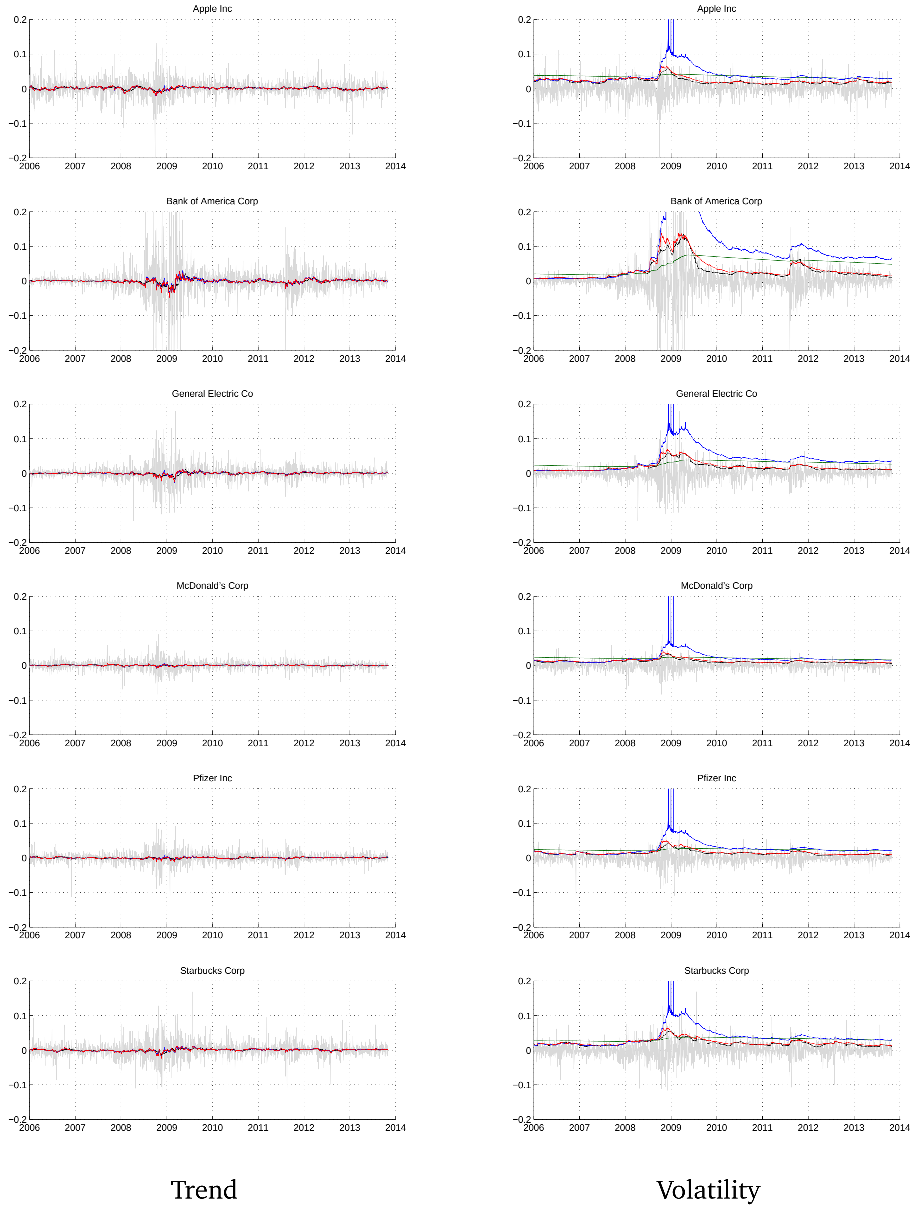
Forecast interval	99.0%	95.0%	90.0%	80.0%	50.0%	20.0%	10.0%
IS-VB	98.4%	95.6%	92.4%	85.5%	59.7%	27.2%	14.4%
no IS-VB	99.1%	97.6%	96.0%	92.4%	76.3%	41.8%	23.5%
WDLM	99.2%	98.2%	97.0%	94.4%	79.7%	43.4%	24.4%

Table 5.1: Coverage of centered forecast intervals across all series $j = 1{:}m$ and the entire test data time period T_{test}.

Close-Up Analysis of Individual Stocks

We now focus on series-specific forecasting performance for stock returns of six well-known companies: Apple, Bank of America, General Electric, McDonald’s, Pfizer and Starbucks.

Trend. Figure 5.2 shows the 60-day tracking moving average of the daily log returns, comparing empirical trends with those forecast under the SGDLM with IS-VB, the SGDLM without IS-VB, and under the WDLM. The three models perform very similarly. A more complex model with series-specific predictors $\mathbf{x}_{jt}$ would lead to noticeable performance improvement of the SGDLMs relative to the WDLM, as the latter requires the same predictors across all series $j = 1{:}m$. We see no obvious advantage of using the posterior decoupling/recoupling in this aspect of the analysis. This is to be expected: the importance sampling weights are defined by $|\mathbf{I} - \mathbf{\Gamma}_t|$, and these values are not affected by values of the trend parameters; as a result, the marginal posteriors will be adequately estimated without the IS-VB steps.



Trend

Volatility

Figure 5.2: Comparison of the realized 60 day moving average/volatility and forecast returns/volatilities. The observed returns are in gray, the observed trend is in black, the results from the WDLM analysis are in green, these from the full SGDLM analysis are in red, and those from analysis without VB are in blue.

Volatility. We represent volatility via standard deviations of returns. Figure 5.2 overlays the observed 60-day tracking moving average of volatility for the six selected stocks, with volatilities as estimated from our analyses. The volatility forecasts from SGDLM analysis with and without IS-VB track each other closely until the onset of the financial crisis in the fall of 2008. Thereafter, without IS-VB– i.e., ignoring the cross-series constraints– the volatility forecasts skyrocket and never return to historically normal levels. In stark contrast to the positive, the full SGDLM analysis generates forecast volatilities in extremely good agreement with the empirically evaluated levels, before, throughout and after the financial crisis and recessionary years. Volatility forecasts from the 400-dimensional WDLM are comparatively unresponsive to changes over time, and just do not reflect key aspects of the empirically estimated volatilities. The WDLM consistently and significantly over-estimates the empirical volatility both before and after the financial crisis; and, during the period of heightened market stress, the volatility forecasts do not adequately reflect higher realized volatilities.

Coverage of forecast intervals. Coverage rates of forecast intervals for the six selected companies are reported in Table 5.2. The results confirm at these individual levels the findings from the aggregate analysis. While forecast intervals up to 95% tend to be over-estimated by either SGDLM, the IS-VB decoupling/recoupling analysis generally improves forecasting performance, while also yielding more precise forecast intervals. Averaged across the test data period T_{test}, the forecast intervals from the WDLM analysis are similar to those from the analysis of the SGDLM without IS-VB decoupling/recoupling, again reflecting the aggregate results.

Forecast interval	99.0%	95.0%	90.0%	80.0%	50.0%	20.0%	10.0%
				Apple Inc			
IS-VB	98.4%	95.3%	92.4%	86.0%	59.4%	26.0%	13.6%
no IS-VB	99.0%	97.9%	95.8%	92.4%	74.3%	40.3%	20.4%
WDLM	99.6%	99.1%	98.1%	95.6%	79.4%	43.7%	23.1%
				Bank of America Corp			
IS-VB	98.2%	95.1%	91.9%	85.9%	60.9%	27.2%	14.6%
no IS-VB	98.2%	96.1%	94.5%	91.3%	79.1%	51.3%	30.9%
WDLM	97.8%	96.1%	94.6%	91.0%	80.0%	47.0%	26.3%
				General Electric Co			
IS-VB	98.2%	94.7%	91.1%	83.9%	58.6%	25.8%	12.9%
no IS-VB	98.9%	97.5%	95.2%	92.2%	77.4%	44.9%	25.6%
WDLM	98.7%	98.0%	96.8%	94.0%	80.9%	45.5%	24.9%
				McDonald’s Corp			
IS-VB	98.5%	96.1%	92.8%	86.4%	59.3%	26.5%	13.3%
no IS-VB	99.1%	98.4%	96.6%	92.9%	73.7%	38.6%	20.3%
WDLM	99.7%	99.2%	98.5%	97.1%	82.3%	44.2%	24.7%
				Pfizer Inc			
IS-VB	98.5%	95.5%	92.3%	85.4%	60.5%	27.0%	14.1%
no IS-VB	99.6%	98.0%	96.6%	93.0%	77.5%	40.7%	21.0%
WDLM	99.5%	98.8%	98.1%	95.4%	80.6%	43.5%	23.9%
				Starbucks Corp			
IS-VB	98.2%	95.5%	92.7%	86.0%	60.3%	27.4%	13.9%
no IS-VB	98.8%	97.0%	95.7%	92.0%	74.2%	39.7%	21.8%
WDLM	99.2%	98.0%	96.5%	94.0%	76.9%	40.4%	22.3%

Table 5.2: Coverage of centered forecast intervals of individual stock returns averaged over the test data time period $T_{\text{test}} = 1{,}368{:}3{,}411$.

5.4.4 Realized Computation Time

Section 5.3.4 noted theoretical considerations concerning computational loads. That is complemented here by an empirical assessment based on rerunning multiple analyses using varying numbers of series m and sizes of the parental sets. Table 5.3 summarizes the empirical run times across several values, in each case generating $N = M = 10{,}000$ posterior and $1-$step forecast samples at each time point. The scaling of the run times is roughly in line with our theoretical estimates, considering some fixed time for memory transfers and communication overheads for coordinating four GPUs. The times were measured on a 2012 computer with four NVidia Tesla C2050 GPUs with 448 CUDA cores each.

	$p_j = 5$		$p_j = 10$		$p_j = 20$	
	Posterior	Forecast	Posterior	Forecast	Posterior	Forecast
$m = 50$	0.07	0.07	0.11	0.08	0.20	0.10
$m = 100$	0.16	0.25	0.21	0.26	0.43	0.32
$m = 200$	0.50	1.17	0.61	1.20	1.04	1.31
$m = 400$	2.13	6.75	2.32	6.78	3.22	7.01

Table 5.3: Realized run times (seconds) of SGDLM analysis and forecasting with $R = N = 10{,}000$.

5.5 Additional Comments

The multivariate dynamic model formulation via SGDLMs is persuasive in terms of the theoretical ability to flexibly represent– in state-space forms– the dynamics of individual series coupled with contemporaneous, cross-series multivariate dependencies. The framework conceptually allows for scalability to higher dimensional series and completely frees the modeler from conceptual and technical constraints encountered in existing models. With that outlook, while also recognizing the inherent opportunities for model decoupling as part of an overall analysis and with the insight that individual, univariate model analyses can often come close to representing core aspects of the full multivariate model, we defined the decoupling/recoupling strategy that builds on importance sampling and variational Bayes to define computationally efficient and practically effective analyses. The computational efficacy arises from massively distributed computation, for which GPU hardware is ideally suited; we have presented the ideas and key details of GPU-enhanced computations, and provide freely available software for interested researchers to follow-up. The practical effectiveness is demonstrated in the $400-$dimensional time series study, and underpinned by the discussion of theoretical questions of importance sampling and variational Bayes/KL divergence-based approximation of unknowable exact posteriors.

Our analyses show that the SGDLM– with importance sampling and variational Bayes or without– can significantly improve model adequacy and forecast accuracy relative to the standard matrix-normal WDLM. Then, we also show substantial improvements in forecasting performance using SGDLMs with the overlay of the IS-VB strategy. Importantly, this latter benefit is realized at reasonable computational costs: the run-time is only increased by about a third in this case study. On a rather standard 2012 desktop (with four NVidia Tesla C2050 GPUs having 448 CUDA cores each), posterior updating and forecasting with $N = M = 10{,}000$ Monte Carlo samples at each time step for our $400-$dimensional time series complete in less than 10 seconds per step. This easily allows for real-time analysis in intervals less than one minute and will scale to several hundreds and low thousands of series on current and emerging commodity desktop/laptop machines.

Current and future work includes the refinement of our existing Matlab interface, and development of an R interface to our GPU-accelerated implementation of Bayesian learning and forecasting of SGDLMs. Current applied and methodological questions under study are questions of parental set selection– which again must involve a focus on practicalities and move away from a purely theoretical but practically unworkable “global model averaging” perspective. Here again forecasting performance coupled with innovations in computational strategies will likely be key to practical progress. A further potential direction is to consider the development of methods of sequential Monte Carlo (SMC), based on particle filtering and learning concepts, as mooted by a referee. This seems a propitious direction for methodology development, especially in view of the potential to alleviate some of the inherent degeneracy issues faced by SMC methods through the key and central decoupling/recoupling approach we have introduced.

5.A C++/CUDA Implementation Details

We provide an overview of the GPU implementation of the complete filtering/forecasting analysis, noting a number of technical aspects and requirements as well as giving some flavor of the structure of C++/CUDA programming[2].

[2] Our implementation and code is based on a computer with an NVidia CUDA-enabled GPU with a compute capability of at least 2.0. Interested users may run the code, so long as CUDA runtime, CUBLAS and CURAND libraries of Version 5.5 or newer are installed. We also have a user-friendly Matlab interface compatible with Matlab versions as early as R2010a; though the C++/CUDA is free-standing, some researchers may be interested in accessing GPU facilities via Matlab.

5.A.1 Key Features of CUDA Implementation

Wherever possible, our implementation uses batched functions from the CUBLAS library to perform the linear algebra operations for forward filtering and forecasting. Batched functions are a recent innovation introduced into the CUBLAS library. They accelerate operations on a large number of small matrices by bundling them into a single function call. This reduces the overhead of initiating many small operations on the GPU individually. In particular, we use:

- `cublasDgemmBatched(...)` for batched matrix-matrix multiplications,
- `cublasDgetrfBatched(...)` for batched LU factorization,
- `cublasDgetriBatched(...)` for batched matrix inversion.

We developed customized kernels to achieve maximum performance for operations that fall outside the scope of current batched CUBLAS functions; see below for details.

5.A.2 Multiple Device Parallelization

This presents the key points regarding GPU parallelization using multiple GPU devices. On top of massively parallel execution of SIMD operations in kernels, CUDA streams can be used to organize sequences of operations. The operations within each stream are executed sequentially, but different streams are executed concurrently on the GPU. Furthermore, different streams can be assigned to different GPU devices, allowing multiple GPU computing. Concurrent execution and data transfers allow the most efficient usage of GPU resources.

A call to the function `cudaSetDevice(k)` activates the k-th GPU device. This means all future GPU calls will be executed on this device until the current device is changed by another call to `cudaSetDevice(...)`. Our implementation sets up one CUDA stream per GPU. In our importance sampling steps to generate a posterior sample of size N and then forecast simulation of M samples, each of K GPUs will compute N/K importance samples and then N/M direct samples; this is parallelization by particles.

The key functions from the CUDA API that organize asynchronous GPU execution are:

- `cudaStreamCreate(...)` to create a stream,
- `cudaMemcpyAsync(...)` to asynchronously move data between computer and GPU,
- `cudaStreamSynchronize(...)` to wait until a stream's execution is completed,
- `cudaStreamDestroy(...)` to destroy a stream.

Calls to CUDA kernels, most API function calls, and asynchronous memory copy operations return control to the host immediately. This allows us to send commands to different GPUs in a `for` loop and still have them executed in parallel. However, it is necessary to wait until the operations are completed before returning results; this is where the function `cudaStreamSynchronize(...)` comes in. This operates after the GPU operations that are to be executed in parallel are sent to their respective devices.

Below is an outline of a multiple device parallelized program.

Pseudo Code

```
for GPU device $k = 1 : K$ do
    Call cudaSetDevice(k) to activate the $k$-th GPU.
    Call cudaStreamCreate(stream_k) to create a stream on the $k$-th GPU.
    Call cudaMemcpyAsync(..., stream_k) to asynchronously copy input arguments onto device memory.
end for
for GPU device $k = 1 : K$ do
    Call cudaSetDevice(k) to activate the $k$-th GPU.
    Call the Variational Bayes posterior estimation subroutine to generate $N/K$ importance samples, and then
    the forecasting simulation subroutine to generate $M$ forecast samples.
end for
for GPU device $k = 1 : K$ do
    Call cudaSetDevice(k) to activate the $k$-th GPU.
    Call cudaMemcpyAsync(..., stream_k) to asynchronously copy results to host memory.
```

```
end for
for GPU device $k = 1 : K$ do
    Call cudaSetDevice(k) to activate the $k$-th GPU.
    Call cudaStreamSynchronize(stream_k) to wait until the computations and memory transfers are completed.
    Call cudaStreamDestroy(stream_k) to destroy the stream on the $k$-th GPU.
end for
Combine and post-process the results on the CPU, if applicable.
```

5.A.3 Generation of Gamma Variates

While the CUDA toolkit provides basic random number generators for the uniform and normal distributions, it is up to the developer to implement other distributions. Our implementation of a gamma random number generator uses a standard rejection sampling algorithm [Marsaglia and Tsang, 2000], requiring one uniform and one normal random number at each step. With expected acceptance probabilities over 90%, running the scheme $2n$ times to generate n Gamma random numbers will almost certainly suffice.

We generate $2n$ uniforms and normals with just one call to `curandGenerateUniformDouble` and `curandGenerateNormalDouble`, respectively. Bundled calls to these CUDA toolkit functions are the most efficient way to generate random numbers on the GPU. Generating enough random numbers for $2n$ proposals leads to highly efficient GPU processing without unnecessary synchronization with the host device.

Specifically, a block of CUDA threads is tasked with generating a batch of $k < n$ gamma random numbers. The batch size is typically chosen as the maximum number of parallel threads the GPU can evaluate, that is, $k = 512$ or $k = 1024$. That batch size is large enough so that there is going to be at least one rejected proposal. This means that at least one thread will have to try a second attempt for acceptance. On the other hand, k is large enough that $2k$ proposals will yield at least k acceptances.

The architecture of a GPU is such that no thread of a block is available when at least one thread is busy. The second attempt of at least one thread makes the entire block unavailable for other operations. Parallel processing of the threads means that it does not cost additional time to let all threads of the same block generate a second proposal. Furthermore, synchronization within a block is very fast. We will count the number of acceptances and rejections of the first attempt and fill up the k-size vector of gamma random variables in the second attempt.

5.A.4 Computing n_{jt} in Variational Bayes

In the VB optimization of Section 5.2.2, the degrees-of-freedom parameter n_{jt} is defined implicitly by

$$\log(n_{jt} + p_j - d_{jt}) - \psi(n_{jt}/2) - (p_j - d_{jt})/n_{jt} - \log(2E[\lambda_{jt}]) + E[\log \lambda_{jt}] = 0. \tag{5.16}$$

Our C++/CUDA program implements a CUDA kernel for the Newton-Raphson method [Atkinson, 1989] to solve for n_{jt}, initialized at $\widetilde{n}_{jt}$. This uses standard numerical approximations for the digamma and trigamma functions [Abramowitz and Stegun, 1972], which we implemented for this problem as the CUDA toolkit does not provide implementations of these functions. These computations are parallelized over series $j = 1{:}m$.

5.A.5 Memory Management

C++/CUDA programming requires proper memory management to avoid segmentation faults as well as memory leaks. Memory management on GPUs is very similar to memory management on the host device/CPUs.

Memory is allocated using the `malloc` or `cudaMalloc` commands and freed using `free` or `cudaFree`. This is a conceptually trivial task. However, keeping track of all allocated memory poses a real challenge when more involved software projects branch into different paths. To address this, we developed a universal and lightweight helper class, `memory_manager`, that automates memory management. This class keeps pointers to allocated memory in a C++/boost list and provides methods to allocate, and keep track of, memory as well as a `clear` method to free all memory it tracks.

Each function starts its own instance of our `memory_manager` to manage memory that is not to live beyond the lifetime of that function. At every branch that ends a function, we simply invoke the `clear` method of the respective function's `memory_manager` to free that memory.

Furthermore, there is a global instance of `memory_manager` to manage the large memory blocks needed for simulation of forecasting and variational Bayes approximation. Allocating and freeing that memory at every time step t would be prohibitively time-consuming.

5.A.6 Matlab Interface

MathWorks provides a C library, `mex.h`, to allow the development of a Matlab interface to C, C++ and CUDA software. This library provides functionality to read Matlab input, write Matlab output and write Matlab messages.

Recent versions of the Matlab Parallel Computing toolbox have another library for Matlab GPU input/output, `mxGPUArray.h`, that also allows the allocation and freeing of GPU memory. However, in contrast to our memory manager, Matlab's memory tracking only avoids memory leaks when Matlab ends, not while the function is active.

We decided against the use of the Matlab GPU library for three reasons. First, up to Matlab R2013b, Matlab ships with an outdated version of the CUDA toolkit library that does not yet provide the latest batch parallelism functionality that we use. Second, by not relying on the Matlab GPU library, our functions can be compiled and run by users that do not license the parallel computing toolbox. Third, we add backwards compatibility to earlier versions of Matlab that do not come with GPU capabilities by programming the GPU interface ourselves. This has been tested with Matlab versions as early as R2010a without any problems. We will of course revisit the questions as new versions of Matlab come along.

5.B Details and Proofs on Our Variational Bayes Strategy for Posterior Decoupling

This section provides an extensive derivation of our variational Bayes posterior decoupling strategy that underlies the posterior decoupling scheme used in Section 5.2.

Unlike the univariate DLM and the multivariate Wishart DLM, the simultaneous graphical DLM does not have conjugate priors, which are needed for analytical forward filtering. Our fast forward filtering scheme (Section 5.2) emulates conjugacy by approximating the exact time t posterior distribution of $\mathbf{\Theta}_t$ and $\mathbf{\Lambda}_t$ by an independent product of normal-gamma distributions using a slight adaptation of the variational mean field method of Section 4.6.1.

Priors and Posteriors

Recall from Section 5.2.2 that the time t prior density $p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{I}_t, \mathcal{D}_{t-1})$ of $\mathbf{\Theta}_t$ and $\mathbf{\Lambda}_t$ in the simultaneous graphical DLM is

$$p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{I}_t, \mathcal{D}_{t-1}) = \prod_{j=1:m} p_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{I}_t, \mathcal{D}_{t-1}), \tag{5.17}$$

where, for each $j = 1{:}m$, $p_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{I}_t, \mathcal{D}_{t-1})$ is the density of the within-series normal-gamma prior

$$(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{I}_t, \mathcal{D}_{t-1}) \sim NG(\mathbf{a}_{jt}, \mathbf{R}_{jt}, r_{jt}, c_{jt}). \tag{5.18}$$

The (exact) time t posterior density of $\mathbf{\Theta}_t$ and $\mathbf{\Lambda}_t$ results as

$$p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t) \propto |\mathbf{I} - \mathbf{\Gamma}_t| \prod_{j=1:m} \widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t), \tag{5.19}$$

where $\widetilde{p}_{jt}$ is the density function of a normal-gamma distribution $NG(\widetilde{\mathbf{m}}_{jt}, \widetilde{\mathbf{C}}_{jt}, \widetilde{n}_{jt}, \widetilde{s}_{jt})$ and $\mathbf{\Gamma}_t$ is a function of $\mathbf{\Theta}_t$; the posterior parameters $\widetilde{\mathbf{m}}_{jt}, \widetilde{\mathbf{C}}_{jt}, \widetilde{n}_{jt}$, and $\widetilde{s}_{jt}$ follow from the updating equations in Section 4.3.2.

Emulating Distribution and Kullback-Leibler Divergence

We choose the class of emulating posterior distributions q to be of the same form as the prior distribution,

$$q(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t) = \prod_{j=1:m} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t) \tag{5.20}$$

where, for each $j = 1{:}m$, q_{jt} denotes the density function of the j-th component

$$(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t) \sim NG(\mathbf{m}_{jt}, \mathbf{V}_{jt}, n_{jt}, s_{jt}), \tag{5.21}$$

and $\mathbf{m}_{jt}$, $\mathbf{V}_{jt}$, n_{jt}, and s_{jt} are the variational parameters.

The Kullback-Leibler divergence $KL_{p|q}$ of the exact posterior distribution p from the emulating posterior distribution q follows as

$$\begin{aligned} KL_{p|q} &= E_q[\log(q(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t)/p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t))] \\ &= c + E_q[\log|\mathbf{I} - \mathbf{\Gamma}_t|] + \sum_{j=1:m} E_q[\log(q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)/\widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t))], \end{aligned} \tag{5.22}$$

while the Kullback-Leibler divergence $KL_{q|p}$ of the emulating posterior distribution q from the exact posterior distribution p is

$$\begin{aligned} KL_{q|p} &= E_p[\log(p(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t)/q(\mathbf{\Theta}_t, \mathbf{\Lambda}_t|\mathcal{D}_t))] \\ &= c + E_p[\log|\mathbf{I} - \mathbf{\Gamma}_t|] + \sum_{j=1:m} E_p[\log(\widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)/q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t))] \\ &= \tilde{c} + \sum_{j=1:m} E_p[\log(\widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)/q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t))]. \end{aligned} \tag{5.23}$$

Optimization of the Variational Parameters

We observe that the problem of minimizing (5.23) reduces to minimizing each summand

$$E_p[\log(\widetilde{p}_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)/q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t))] \tag{5.24}$$

individually, given that each component set of variational parameters $\mathbf{m}_{jt}$, $\mathbf{V}_{jt}$, n_{jt}, and s_{jt} affects exactly one such summand; the expectation of the determinant $|\mathbf{I} - \mathbf{\Gamma}_t|$ is under p, and so independent of the variational parameters. On the other hand, minimizing (5.22) requires minimization of the expectation of the determinant $|\mathbf{I} - \mathbf{\Gamma}_t|$ under q, which is substantially more complex. For this reason, we minimize $KL_{q|p}$ in (5.23) instead of $KL_{p|q}$ in (5.22). Note that the optimization rule in (5.23) also guarantees that the exact posterior distribution p is selected, if it is contained in the class of emulating distributions q.

Using Leibniz's rule for differentiation under the integral, the partial derivative of $KL_{q|p}$ with respect to any variational parameter $\mathbf{x}_{jt} \in \{\mathbf{m}_{jt}, \mathbf{V}_{jt}, n_{jt}, s_{jt}\}$ follows as

$$\frac{\partial}{\partial \mathbf{x}_{jt}} KL_{q|p} = -E_p\left[\frac{\frac{\partial}{\partial \mathbf{x}_{jt}} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}{q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}\right], \tag{5.25}$$

where the density function q_{jt} is

$$\begin{aligned} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t) =& f_{N(\mathbf{m}_{jt}, \mathbf{C}_{jt}/(s_{jt}\lambda_{jt}))}(\boldsymbol{\theta}_{jt}|\lambda_{jt}) \times f_{G(n_{jt}/2, n_{jt}s_{jt}/2)}(\lambda_{jt}) \\ =& (2\pi)^{-\frac{p_j}{2}} \left|\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})' \left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right) \\ & \times \frac{\left(\frac{n_{jt}s_{jt}}{2}\right)^{n_{jt}/2}}{\Gamma(\frac{n_{jt}}{2})} \lambda_{jt}^{n_{jt}/2-1} \exp\left(-\frac{n_{jt}s_{jt}}{2}\lambda_{jt}\right). \end{aligned}$$

Derivation of $\mathbf{m}_{jt}$.

$$\begin{aligned}
0 &= -\frac{\partial}{\partial \mathbf{m}_{jt}} KL_{q|p} \\
&= E_p\left[\frac{\frac{\partial}{\partial \mathbf{m}_{jt}} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}{q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}\right] \\
&= E_p\left[\frac{\frac{\partial}{\partial \mathbf{m}_{jt}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})' \left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right)}{\exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})' \left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right)}\right] \\
&= E_p\left[\left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right] \\
&= s_{jt}\mathbf{C}_{jt}\left(E_p[\lambda_{jt}\boldsymbol{\theta}_{jt}] - E_p[\lambda_{jt}]\mathbf{m}_{jt}\right)
\end{aligned}$$

$$\Leftrightarrow \mathbf{m}_{jt} = \frac{E_p[\lambda_{jt}\boldsymbol{\theta}_{jt}]}{E_p[\lambda_{jt}]}.$$

Derivation of $\mathbf{C}_{jt}$. Using rules of matrix calculus from Petersen and Pedersen [2008],

$$\begin{aligned}
0 &= -\frac{\partial}{\partial \mathbf{C}_{jt}} KL_{q|p} \\
&= E_p\left[\frac{\frac{\partial}{\partial \mathbf{C}_{jt}} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}{q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}\right] \\
&= E_p\left[\frac{\frac{\partial}{\partial \mathbf{C}_{jt}} |\mathbf{C}_{jt}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})' \left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right)}{|\mathbf{C}_{jt}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})' \left(\frac{\mathbf{C}_{jt}}{s_{jt}\lambda_{jt}}\right)^{-1} (\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})\right)}\right] \\
&= E_p\left[-\frac{1}{2}\left(\mathbf{I} - s_{jt}\lambda_{jt}\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'\right)\mathbf{C}_{jt}^{-1}\right] \\
&= -\frac{1}{2}\left(\mathbf{I} - s_{jt}\mathbf{C}_{jt}^{-1}E_p\left[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'\right]\right)\mathbf{C}_{jt}^{-1}
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow \mathbf{I} &= s_{jt}\mathbf{C}_{jt}^{-1}E_p\left[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'\right] \\
\Leftrightarrow \mathbf{C}_{jt} &= s_{jt}E_p[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'] \\
&=: s_{jt}\mathbf{V}_{jt}.
\end{aligned}$$

Derivation of s_{jt}.

$$\begin{aligned}
0 &= -\frac{\partial}{\partial s_{jt}} KL_{q|p} \\
&= E_p\left[\frac{\frac{\partial}{\partial s_{jt}} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}{q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}\right] \\
&= E_p\left[\frac{\frac{\partial}{\partial s_{jt}} s^{\frac{p_j}{2}} \exp\left(-\frac{1}{2}s_{jt}\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})\right) s_{jt}^{\frac{n_{jt}}{2}} \exp\left(-\frac{n_{jt}s_{jt}}{2}\lambda_{jt}\right)}{s^{\frac{p_j}{2}} \exp\left(-\frac{1}{2}s_{jt}\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})\right) s_{jt}^{\frac{n_{jt}}{2}} \exp\left(-\frac{n_{jt}s_{jt}}{2}\lambda_{jt}\right)}\right] \\
&= E_p\left[\frac{n_{jt}+p_j}{2}\frac{1}{s_{jt}} - \frac{\lambda_{jt}}{2}\left((\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt}) + n_{jt}\right)\right] \\
&= \frac{1}{2}\left(\frac{n_{jt}+p_j}{s_{jt}} - E_p\left[\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})\right] - n_{jt}E_p\left[\lambda_{jt}\right]\right)
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow s_{jt} &= \frac{n_{jt}+p_j}{E_p\left[\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{C}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})\right] + n_{jt}E_p\left[\lambda_{jt}\right]} \\
&= \frac{n_{jt}+p_j}{\frac{1}{s_{jt}}E_p\left[\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})'\mathbf{V}_{jt}^{-1}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})\right] + n_{jt}E_p\left[\lambda_{jt}\right]} \\
&=: \frac{n_{jt}+p_j}{\frac{1}{s_{jt}}d_{jt} + n_{jt}E_p\left[\lambda_{jt}\right]}
\end{aligned}$$

$$\Leftrightarrow s_{jt} = \frac{n_{jt}+p_j-d_{jt}}{n_{jt}E_p\left[\lambda_{jt}\right]}.$$

Derivation of n_{jt}.

$$\begin{aligned}
0 &= -\frac{\partial}{\partial n_{jt}} KL_{q|p} \\
&= E_p\left[\frac{\frac{\partial}{\partial n_{jt}} q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}{q_{jt}(\boldsymbol{\theta}_{jt}, \lambda_{jt}|\mathcal{D}_t)}\right] \\
&= E_p\left[\frac{\frac{\partial}{\partial n_{jt}}\left(\Gamma\left(\frac{n_{jt}}{2}\right)\right)^{-1}\lambda_{jt}^{\frac{n_{jt}}{2}-1}\exp\left(-\lambda_{jt}\frac{n_{jt}s_{jt}}{2}\right)\left(\frac{n_{jt}s_{jt}}{2}\right)^{\frac{n_{jt}}{2}}}{\left(\Gamma\left(\frac{n_{jt}}{2}\right)\right)^{-1}\lambda_{jt}^{\frac{n_{jt}}{2}-1}\exp\left(-\lambda_{jt}\frac{n_{jt}s_{jt}}{2}\right)\left(\frac{n_{jt}s_{jt}}{2}\right)^{\frac{n_{jt}}{2}}}\right] \\
&= E_p\left[\frac{1}{2}\left(\frac{\Gamma'\left(\frac{n_{jt}}{2}\right)}{\Gamma\left(\frac{n_{jt}}{2}\right)} - \log\lambda_{jt} + \lambda_{jt}s_{jt} - \log\left(\frac{n_{jt}s_{jt}}{2}\right) - 1\right)\right] \\
&= \frac{1}{2}\left(\psi\left(\frac{n_{jt}}{2}\right) - E_p\left[\log\lambda_{jt}\right] + s_{jt}E_p\left[\lambda_{jt}\right] - \log\left(\frac{n_{jt}s_{jt}}{2}\right) - 1\right)
\end{aligned}$$

$$\begin{aligned}
\Leftrightarrow 0 &= \psi\left(\frac{n_{jt}}{2}\right) - E_p\left[\log\lambda_{jt}\right] + s_{jt}E_p\left[\lambda_{jt}\right] - \log\left(\frac{n_{jt}s_{jt}}{2}\right) - 1 \\
&= \psi\left(\frac{n_{jt}}{2}\right) - E_p\left[\log\lambda_{jt}\right] + \frac{n_{jt}+p_j-d_{jt}}{n_{jt}E_p\left[\lambda_{jt}\right]}E_p\left[\lambda_{jt}\right] - \log\left(\frac{n_{jt}\frac{n_{jt}+p_j-d_{jt}}{n_{jt}E_p[\lambda_{jt}]}}{2}\right) - 1 \\
&= \psi\left(\frac{n_{jt}}{2}\right) - E_p\left[\log\lambda_{jt}\right] + \frac{p_j-d_{jt}}{n_{jt}} - \log(n_{jt}+p_j-d_{jt}) + \log(2E_p[\lambda_{jt}]).
\end{aligned}$$

Summary. The optimized variational parameters can be calculated in the following order.

- $\mathbf{m}_{jt} = E_p[\lambda_{jt}\boldsymbol{\theta}_{jt}]/E_p[\lambda_{jt}]$,
- $\mathbf{V}_{jt} = E_p[\lambda_{jt}(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})(\boldsymbol{\theta}_{jt}-\mathbf{m}_{jt})']$,

- $d_{jt} = E_p[\lambda_{jt}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})'\mathbf{V}_{jt}^{-1}(\boldsymbol{\theta}_{jt} - \mathbf{m}_{jt})]$,
- n_{jt} is the unique value that satisfies
  $\log(n_{jt} + p_j - d_{jt}) - \psi(n_{jt}/2) - (p_j - d_{jt})/n_{jt} - \log(2E_p[\lambda_{jt}]) + E_p[\log \lambda_{jt}] = 0$,
- $s_{jt} = (n_{jt} + p_j - d_{jt})/(n_{jt}E_p[\lambda_{jt}])$, and
- $\mathbf{C}_{jt} = s_{jt}\mathbf{V}_{jt}$.

Here all expectations $E_p[\cdot]$ are with respect to the exact posterior distribution and can be easily evaluated, for example, using importance sampling (Section 4.6.2).

Chapter 6

Bayesian Forecasting and Portfolio Decisions Using Simultaneous Graphical Dynamic Linear Models

The contents of this chapter have been submitted for publication in Gruber and West [2015a]. This chapter is a lightly edited reproduction of selected contents of the submitted manuscript.

This chapter develops a customized SGDLM focused on a case study in financial forecasting and portfolio optimization with 400 daily S&P 500 stock prices over 2007–2014. It includes new methodological contributions in model selection for SGDLMs and Bayesian decision analysis/portfolio decisions. The case study includes benchmarks of forecast performance as well as portfolio return and risk metrics against the standard multivariate Wishart DLM (WDLM; Section 4.4). This is the appropriate benchmark as it has been a standard model in Bayesian financial time series and portfolio analysis—in industry and academic research—for years, being quite flexible and trivially implemented, the latter key to scaling. Section 6.1 introduces a novel and practicable selection strategy for the parental sets. Section 6.2 discusses several quantitative investment rules based on various portfolio utility functions of practical interest. Section 6.3 presents a portfolio manager's view of managing a 400-asset portfolio using the SGDLM combined with such rules to drive investment decisions. Section 6.4 concludes with final remarks.

6.1 Selection of Simultaneous Parental Sets: The Bayesian Hotspot

As introduced in Gruber and West [2015b], the SGDLM framework assumes knowledge of the simultaneous parental sets $sp(j)$. We need a systematic approach to selecting these sets and address this here with a novel, natural strategy.

We will typically have $|sp(j)|$ much smaller than m in problems where m is at all large. With $m = 400$ in our S&P case study (Section 6.3), there are many patterns of time-varying dependencies among stocks, but it is inappropriate to expect real practical value in estimating co-volatilities from models with more than, say, 20 or so simultaneous predictors. That is, the implied dynamic graphical model—represented by zeros/non-zeros in $\mathbf{\Gamma}_t$ and $\mathbf{\Omega}_t$—will typically be quite sparse. Collinearities among potential simultaneous parental series will typically mean that many possible choices of a (smallish) parental set for any one series will yield similar predictions, so working with one set of selected $sp(j)$ is desirable. Any specific choice can be updated periodically over time, and the models adapted and refitted, as desired.

A useful strategy is to select the $sp(j)$ based on analysis of prior training data using a standard, trivially computed analysis via a WDLM. An extension is to run a WDLM in parallel to the SGDLM, and therefore have an opportunity to potentially change the parental set choices from time to time, based on relationships indicated by the flexible and adaptive but, for forecasting, likely inferior WDLM analysis. This discussion underlies our use of what we term the "Bayesian Hotspot," the name being suggested by visualization of the proposed automatic strategy for parental set selection.

The Bayesian Hotspot selection mechanism operates on an initial set of training data, prior to the full analysis of the test data. A multivariate WDLM is applied to the training data; this matrix discount learning DLM can be chosen to have the same covariates used in the SGDLM, or use just a local-level model ($\mathbf{F}_t = 1$ for all t) for simplicity. The WGDLM estimates the full, $m \times m$ time-varying precision matrix $\mathbf{\Omega}_t$ without any constraints; there are no zeros in $\mathbf{\Omega}_t$, implying that each series is conditionally, contemporaneously dependent on all the other $m-1$ series. Off-diagonal elements in row j are, up to a constant given by the reciprocal of the (j,j) diagonal element, implied conditional regression coefficients of these $m-1$ series in predicting y_{jt} at time t; hence, larger values are candidates for inclusion in a selected, small parental set. At any time t, the implied Wishart prior $(\mathbf{\Omega}_t|\mathcal{D}_{t-1}) \sim W(r_t, \mathbf{B}_t)$ in the WDLM is trivially simulated, and a large Monte Carlo sample can then be used to compute the implied prior *ranking* of the absolute values of the $m-1$ off-diagonal entries in row j. Choosing a target parental set size $|sp(j)|$, we then select this number of the most highly ranked series. A heat-map of the prior rankings shows the simultaneous parents as the "hottest spots" when visualized, hence the terminology.

6.2 Bayesian Portfolio Analysis

Part of our comparison of SGDLM and WDLM approaches in the S&P case study in Section 6.3 involves assessments of a range of dynamically optimized and updated portfolios, considering several practically relevant variants of fundamental mean-variance portfolio rules [Markowitz, 1952, Aguilar and West, 2000, Carvalho and West, 2007, Quintana et al., 2003, 2010, Prado and West, 2010, Chapter 10, Section 4.7]. The analysis models daily log-returns on stocks and sequentially updates the portfolio allocation across these stocks via Bayesian decision analysis using chosen portfolio utility functions. Mean-variance optimization aims to control risk while aiming for positive returns, and modified utilities overlay additional, practically relevant constraints. In addition to specific *target return* portfolios, we consider utility functions that incorporate a benchmark index and require that optimized portfolios be, in expectation, uncorrelated with the benchmark in addition to target return and risk components.

Our models are applied to the vector of daily returns $\mathbf{y}_t$. In all models, the mean and variance matrix of the one-step ahead forecast distribution $p(\mathbf{y}_t|\mathcal{D}_{t-1})$ are key ingredients. Denote these by $\mathbf{p}_t = E(\mathbf{y}_t|\mathcal{D}_{t-1})$ and $\mathbf{P}_t = V(\mathbf{y}_t|\mathcal{D}_{t-1})$. As noted in Section 4.4, these moments are trivially computed in a WDLM, as the forecast is a multivariate T distribution; in the SGDLM, they are computed via Monte Carlo simulation based on the recoupled importance sampling analysis at each time t. A portfolio weight vector $\mathbf{w}_t = (w_{1,t}, \ldots, w_{m,t})'$ defines the allocation of capital across the m assets. The decision is to choose $\mathbf{w}_t$ at market close on day $t-1$, and then act on that reallocation; on day t, the new closing prices are realized and the process repeats on the following day. Based on the forecast distribution of returns, the implied one-step ahead forecast mean and variance of the portfolio for any specific weight vector $\mathbf{w}_t$ are $\mathbf{w}_t'\mathbf{p}_t$ and $\mathbf{w}_t'\mathbf{P}_t\mathbf{w}_t$, respectively.

Minimum variance portfolio. The standard or baseline minimum variance portfolio chooses $\mathbf{w}_t$ as that vector minimizing the expected portfolio variance $\mathbf{w}_t\mathbf{P}_t\mathbf{w}_t$ subject $\mathbf{1}'\mathbf{w}_t = 1$. The optimal weight vector is trivially computed. More practically relevant portfolio strategies overlay additional constraints, as follows.

Target return mean-variance portfolio. The original [Markowitz, 1952] mean-variance portfolio rule minimizes the risk—again in terms of portfolio variance—for a given, desired target return τ_t. The relevant decision analysis simply modifies the minimum variance portfolio optimization by adding the constraint $\mathbf{w}_t'\mathbf{p}_t \geq \tau_t$, or its practical equivalent $\mathbf{w}_t'\mathbf{p}_t = \tau_t$. Note that the targets τ_t can vary over time, and be chosen adaptively by either direct specification or an automated rule.

Benchmark-neutral portfolio. This refinement mandates that the portfolio be uncorrelated, in expectation, at each step with a selected benchmark time series. To implement this, joint one-step ahead forecast distributions are required for the assets of interest together with the benchmark series. With no loss of generality, we do this by taking the selected benchmark series as $j = 1$. The relevant decision analysis then simply modifies the minimum variance and/or target return portfolio optimization by adding the constraints $w_{1t} = 0$ and $\mathbf{w}_t'\mathbf{P}_{\cdot 1t} = 0$ where $\mathbf{P}_{\cdot 1t}$ is the first column of $\mathbf{P}_t$ containing the covariances of all series with the benchmark.

6.3 Case Study: S&P 500 Stocks

6.3.1 Context and Data

We use data on the S&P 500 stock market index (SPX) and 400 S&P 500 member stocks that have been continuously listed since 2002. We are interested in—among other things—comparisons using benchmark neutral portfolios, and take SPX as the benchmark; our models are thus for the $m = 401$-dimensional vector of returns comprising SPX as the first entry, followed by the 400 stocks. We use additional series as predictors: the VIX volatility index (VIX) and the ten year treasury yields (TNX). The SPX data and stock data are quoted in daily log-returns $y_t = \log(\text{price}_t/\text{price}_{t-1})$. The VIX and TNX data are the levels of the indices: annualized implied 30-day volatility and annualized interest rate, respectively. The data covers the years 2002 through Q3-2013; the years 2002–2006 are used as training data, and the remaining years are used as test data.

Our comparative analyses assume that there are no bid-ask spreads and that the cost of all trades is 20 basis points of the traded volume. We operate under the assumption that all trades can be executed at the daily closing price and that short-selling is possible. Our calculations of annualized returns and volatilities assume that a year consists of 252 trading days.

6.3.2 Forecast Model Specifications

We use the WDLM and SGDLM from Sections 4.4 and 5.1 in combination with different external predictors and discount factors. Table 6.1 provides a full summary of the models used.

External predictors. For clarity, denote the S&P 500 index log-returns by $y_{\text{SPX},t} \equiv y_{1t}$. The simplest DLM form is the local-level model, specified via

$$\mathbf{F}_{jt} = \mathbf{F}_t = 1 \tag{6.1}$$

for all $j = 1{:}m$ and t. We use this as a base model, and study several extensions hoping to achieve better forecasting performance. The first extension of the base model is a two-factor approach using the time $t-1$ log-return $y_{\text{SPX},t-1}$ of the S&P 500 index as a momentum factor, and the time $t-1$ 10-year treasury yield TNX_{t-1} as an interest rate factor. That is,

$$\mathbf{F}_{jt} = \mathbf{F}_t = (1, y_{\text{SPX},t-1}, \text{TNX}_{t-1})'. \tag{6.2}$$

A second extension is a three-factor model using short-term momentum changes of the S&P 500 index $\Delta y_{\text{SPX},t-1} := y_{\text{SPX},t-1} - y_{\text{SPX},t-2}$, one-day changes of the 10-year treasury yield $\Delta\text{TNX}_{t-1} := \text{TNX}_{t-1} - \text{TNX}_{t-2}$, and two-day changes of the VIX index level ($\Delta^2\text{VIX}_{t-1} := \text{VIX}_{t-1} - \text{VIX}_{t-3}$). Thus

$$\mathbf{F}_{jt} = \mathbf{F}_t = (1, \Delta y_{\text{SPX},t-1}, \Delta\text{TNX}_{t-1}, \Delta^2\text{VIX}_{t-1})'. \tag{6.3}$$

The third extension uses smoothed momentum changes of the S&P 500 index in combination with two-day changes of the VIX index levels, namely $\Delta^2\text{VIX}_{t-1}$. In this case,

$$\mathbf{F}_{jt} = \mathbf{F}_t = (1, 0.5\sum_{k=1:2} y_{\text{SPX},t-k} - 0.2\sum_{k=1:5} y_{\text{SPX},t-k}, \Delta^2\text{VIX}_{t-1})'. \tag{6.4}$$

These predictors $\mathbf{F}_t$ can be used with the WDLM as well as the SGDLM, given that they are not series-specific. For use only with the SGDLM, we modify the last variant of (6.4) to use the smoothed momentum changes of each series $j = 1{:}m$ instead of the changes of the S&P 500 index. That is,

$$\mathbf{F}_{jt} = (1, 0.5\sum_{k=1:2} y_{j,t-k} - 0.2\sum_{k=1:5} y_{j,t-k}, \Delta^2\text{VIX}_{t-1})'. \tag{6.5}$$

Model	Predictors	Discount Factors
WDLM 1	(6.1)	$\beta = 0.980$, $\delta = 0.990$
SGDLM 1	(6.1)	$\beta_j = 0.975$, $\delta_{j\phi} = 0.980$, $\delta_{j\gamma} = 0.990$
WDLM 2	(6.2)	$\beta = 0.980$, $\delta = 0.998$
SGDLM 2	(6.2)	$\beta_j = 0.975$, $\delta_{j\phi} = 0.985$, $\delta_{j\gamma} = 0.990$
WDLM 3	(6.3)	$\beta = 0.980$, $\delta = 0.998$
SGDLM 3	(6.3)	$\beta_j = 0.975$, $\delta_{j\phi} = 0.985$, $\delta_{j\gamma} = 0.990$
WDLM 4	(6.4)	$\beta = 0.980$, $\delta = 0.998$
SGDLM 4	(6.4)	$\beta_j = 0.975$, $\delta_{j\phi} = 0.985$, $\delta_{j\gamma} = 0.990$
SGDLM 5	(6.5)	$\beta_j = 0.975$, $\delta_{j\phi} = 0.985$, $\delta_{j\gamma} = 0.990$

Table 6.1: Dynamic models and discount factors compared for S&P study.

6.3.3 Training Data Analysis

An initial period of data from 2002 through 2006 was used for training analysis to guide choice of discount factors and initial parental sets. Analysis of WDLMs and SGDLMs with varying specifications allows for selection of those demonstrating the best predictive performance—in terms of responsiveness to changing market levels and resilience to individual outliers—during that period. In some cases where varying the discount factors and parental sets impacts on forecasting performance only marginally, we selected those generating highest risk-adjusted return for portfolio rule \$1 (see Table 6.4) during the training data period.

For the WDLM, we investigated $\beta \in \{0.980, 0.998\}$ and $\delta \in \{0.990, 0.998\}$; for the SGDLM we investigated combinations of $\beta_j \in \{0.975, 0.990\}$, $\delta_{j\phi} \in \{0.980, 0.985, 0.990\}$ and $\delta_{j\gamma} \in \{0.990\}$. The selection of $\delta_{j\gamma} = 0.990$ follows previous analyses [e.g. Section 5 in Gruber and West, 2015b]. The Bayesian Hotspot strategy was used to identify initial sets of 20 simultaneous parents for each series, and the parental sets and starting priors are subsequently updated every year as detailed in the next section.

From the training data analysis with WDLMs, we selected the Wishart volatility discount factor $\beta = 0.98$, and WDLM state discount factor $\delta \in \{0.990,\ 0.998\}$ depending on the choice of predictors $\mathbf{F}_t$. For the SGDLM, we selected univariate volatility discount factors $\beta_j = 0.975$, external predictor state discount factors $\delta_{j\phi} \in \{0.980,\ 0.985\}$ depending on the choice of predictors $\mathbf{F}_{jt}$, and parental predictor state discount factors $\delta_{j\gamma} = 0.990$ for all series $j = 1{:}m$. The external predictors used in (6.2)–(6.5) contain time-varying effects, so that high discount factors δ and $\delta_{j\phi}$ were appropriate choices for models WDLM 2–4 and SGDLM 2–5. The local-level models, WDLM 1 and SGDLM 1, required smaller discount factors δ and $\delta_{j\phi}$ to capture time-varying trends by allowing for more stochastic variation in the states $\mathbf{\Theta}_t$ of the WDLM and $\boldsymbol{\phi}_{jt}$ of the SGDLM. Models and parameters are summarized in Table 6.1.

6.3.4 Sequential Analysis of Time Series

The test data period is from the start of 2007 through Q3-2013. During this period, the SGDLMs are set to autonomously update their simultaneous parents and starting priors on the first trading day of each year. Denote by $t_{1,y}$ the first trading day of the new year, and write $t_{1,y-1}$ and $t_{\text{end},y-1}$ for the first and last trading days of the previous year, respectively. The simultaneous parents for each new year are selected by the Bayesian Hotspot (Section 6.1) using the step-ahead prior for $t_{1,y}$ of the local-level WDLM on the last trading day of the previous year, $t_{\text{end},y-1}$. We use the same local-level WDLM to inform the selection of the parental sets of all SGDLMs to keep the effects of different covariance/dependence models separate from the effect of different mean-level predictors.

At the start of each year, priors for the revised parental set models are specified by re-filtering the previous year's observations $\mathbf{y}_t$ for $t = t_{1,y-1}{:}t_{\text{end},y-1}$ using the new simultaneous parents; here the analyses are started on the first trading day of the previous year, $t_{1,y-1}$, with relatively diffuse initial priors informed by the posterior parameters of the SGDLM of the previous year.

2007	2008	2009	2010	2011	2012	2013
SPX	SPX	SPX	SPX	SPX	SPX	SPX
C UN	**ABT UN**	**AAPL UW**	AAPL UW	*AAPL UW*	***ABC UN***	BCR UN
EBAY UW	***AEE UN***	*ABT UN*	***AEE UN***	**BCR UN**	***AXP UN***	**BF/B UN**
EQR UN	***AFL UN***	*APD UN*	*BEN UN*	**CPB UN**	*BCR UN*	**BMY UN**
EXC UN	**APD UN**	**BEN UN**	*BLL UN*	DOV UN	***BK UN***	CPB UN
HES UN	***BBT UN***	**BLL UN**	**DOV UN**	EBAY UW	CPB UN	EBAY UW
HRS UN	***BHI UN***	EBAY UW	EBAY UW	*EIX UN*	***D UN***	INTC UW
IFF UN	***COP UN***	**EIX UN**	EIX UN	***HCN UN***	***DNB UN***	**JNJ UN**
MKC UN	***DOV UN***	***HES UN***	***ETR UN***	INTC UW	*DOV UN*	MCK UN
MRO UN	EBAY UW	***JNJ UN***	***HD UN***	***JPM UN***	EBAY UW	MSFT UW
MTB UN	***GIS UN***	***MAS UN***	**INTC UW**	**KO UN**	INTC UW	NI UN
NSC UN	***LMT UN***	***MKC UN***	**L UN**	*L UN*	*KO UN*	**OMC UN**
PCP UN	*NSC UN*	***MTB UN***	**MSFT UW**	MSFT UW	**MCK UN**	**PCLN UW**
PEP UN	*PX UN*	**NEE UN**	*NEE UN*	**NI UN**	MSFT UW	PG UN
PX UN	SIAL UW	***PG UN***	***NSC UN***	**PG UN**	NI UN	**RSG UN**
SIAL UW	SO UN	*SIAL UW*	**PKI UN**	*PKI UN*	PG UN	SO UN
SO UN	**TEG UN**	*SO UN*	***T UN***	TMO UN	**SO UN**	**TROW UW**
STI UN	***TMO UN***	*TEG UN*	**TMO UN**	***TROW UW***	*TMO UN*	**TRV UN**
T UN	**UPS UN**	***UNP UN***	***VZ UN***	***VTR UN***	XEL UN	XEL UN
YHOO UW	***XL UN***	*UPS UN*	**XEL UN**	XEL UN	**XOM UN**	XOM UN
	14 new	12 new	13 new	9 new	8 new	8 new

Table 6.2: Simultaneous parental sets of Amazon from 2007 through 2013 in model SGDLM 1. New parents are in bold and parents that exit in the following year are italicized.

6.3.5 Aspects of Selected Simultaneous Parents

Table 6.2 shows the simultaneous parental sets for Amazon stock from 2007 through 2013. During this time, on average 10.7 out of 20 parents were replaced every year; furthermore, there were three technology stocks—Apple, ebay, Intel and Microsoft—that were more persistent than the other simultaneous parents, which is a result consistent with the nature of the companies and the dynamics of their stock valuations over this period. Across all 400 stocks, on average 11.7 out of the 20 simultaneous parents were replaced each year. In the heart of the financial crisis—2008 through 2010—average turnover of parents across the 400 stocks was 13.0 of the 20, markedly higher than the average turnover of 10.5 during the next three years, 2011 through 2013.

6.3.6 Trends, Volatilities and Co-Volatilities

Point forecasts from the local-level models WDLM 1 and SGDLM 1 are relatively smooth, and similar to the 60-day moving average line, as should be expected; see Figure 6.1. The daily, time-varying predictors $\mathbf{F}_{jt}$ and $\mathbf{F}_t$ of the more structured models induce increased volatility in predictions. A visual analysis of the figures does not reveal obvious differences among forecasts means, though very small differences—that may be hard to discern visually—can, of course, be practically relevant in terms of their impact on portfolio outcomes. Table 6.3 reports mean-squared errors and correlations with observed values as metrics of forecast quality; the WDLM shows slightly smaller aggregate mean-squared errors than the SGDLM, but no general statement can be made about the correlations of the forecasts and realized returns.

Estimated volatilities of all WDLMs substantially over-estimate the realized volatilities and adapt only slowly to changing realized volatilities. In contrast, volatilities inferred from all SGDLMs are very close to the realized volatilities and are much more responsive to changing volatility levels. More importantly, and justifying the sparse simultaneous parental structure of the SGDLM, the forecasts of across-series co-volatilities are improved very substantially; see Figure 6.1, for example.

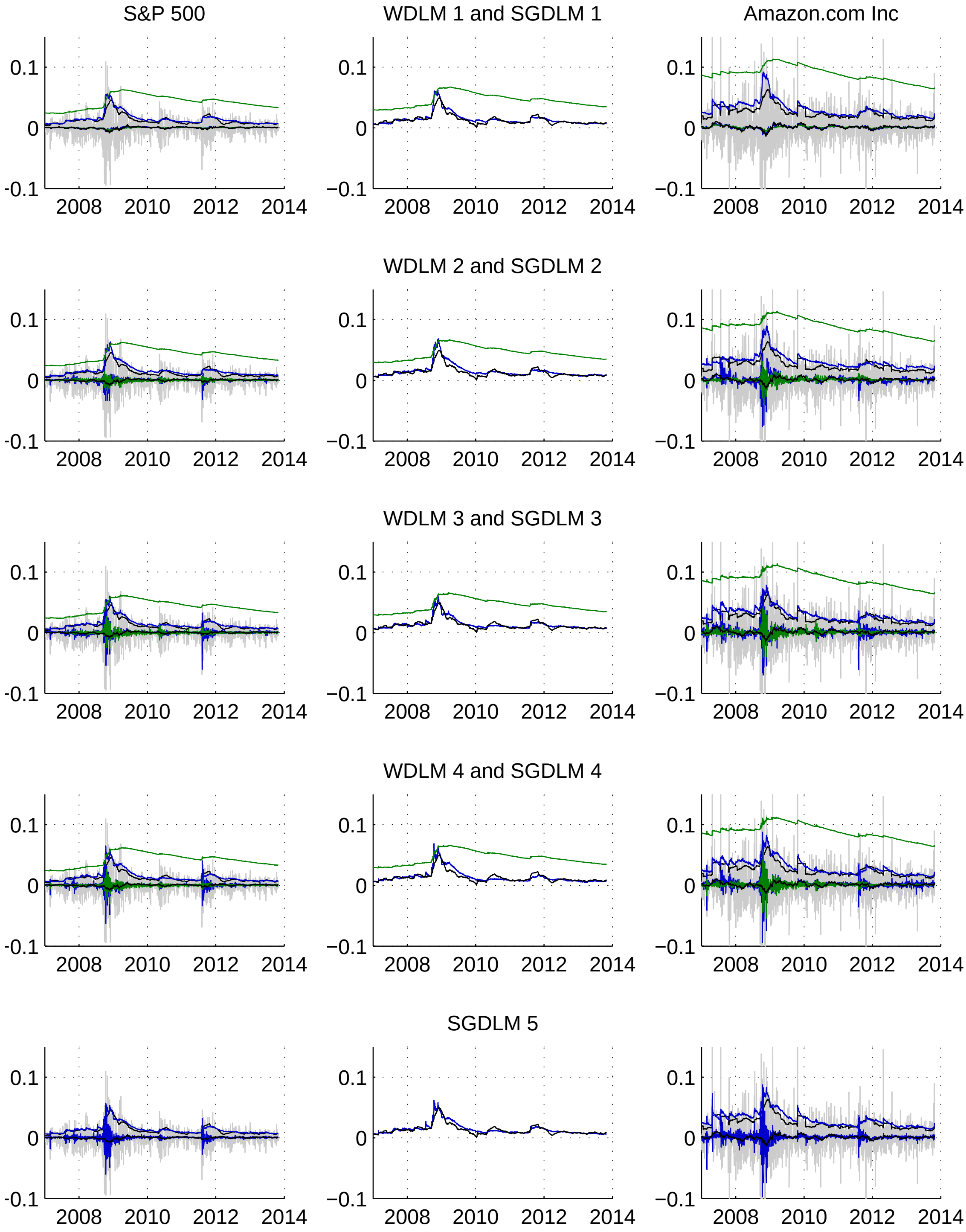


Figure 6.1: The left and right columns show the log-returns and volatilities of the S&P 500 and Amazon, respectively; the central column shows the co-volatilities. Observed returns are in gray, the 60-day moving averages, volatilities and co-volatilities are in black, and forecasts from the WDLM and SGDLM are in green and blue, respectively.

		WDLM 1	WDLM 2	WDLM 3	WDLM 4	SGDLM 1	SGDLM 2	SGDLM 3	SGDLM 4	SGDLM 5
2002–2006	Corr. SPX	-1.1%	0.8%	-3.4%	**4.8%**	-2.1%	-3.6%	-3.9%	4.1%	3.8%
	Corr. all	-3.6%	-0.8%	-0.7%	**1.4%**	-3.6%	-1.2%	-0.1%	0.8%	0.7%
	MSE SPX	1.00e-4	1.00e-4	1.00e-4	**0.99e-4**	1.00e-4	1.02e-4	1.03e-4	1.01e-4	1.01e-4
	MSE all	0.44e-3	**0.43e-3**	0.44e-3	**0.43e-3**	0.44e-3	0.45e-3	0.45e-3	0.44e-3	0.45e-3
2007–2013	Corr. SPX	-1.4%	8.6%	11.4%	8.4%	-0.2%	4.1%	**14.4%**	11.8%	12.7%
	Corr. all	-2.8%	2.5%	4.9%	3.5%	-2.2%	0.9%	**6.8%**	5.9%	7.2%
	MSE SPX	2.16e-4	**2.14e-4**	**2.14e-4**	2.15e-4	2.17e-4	2.21e-4	2.17e-4	2.18e-4	2.19e-4
	MSE all	**0.63e-3**	**0.63e-3**	**0.63e-3**	**0.63e-3**	**0.63e-3**	0.65e-3	0.65e-3	0.64e-3	0.65e-3

Table 6.3: Summary of forecast metrics: correlation of step-ahead forecasts with realized values and mean-squared errors of forecasts; "SPX" refers to the S&P 500 index and "all" refers to the 400 individual stocks.

Strategy	Description
\$1	target return $\tau_t = 10\%/252$
\$2	target return $\tau_t = 15\%/252$
\$3	target return $\tau_t = \hat{\mu}_{\text{SPX},t} + 5\%/252$
\$4	SPX neutral, target return $\tau_t = 10\%/252$
\$5	SPX neutral, target return $\tau_t = 15\%/252$
\$6	SPX neutral, target return $\tau_t = \hat{\mu}_{\text{SPX},t} + 5\%/252$

Table 6.4: Portfolio investment strategies compared in S&P study.

6.3.7 Portfolio Comparisons

We analyze the optimization-based investment strategies summarized in Table 6.4. We paired each of the nine models in Table 6.1 with the quantitative investment strategies in Table 6.4. The investment analysis proceeded as follows for each model-strategy pair: (i) at market close on day $t-1$, update the model distributions based on the observation $\mathbf{y}_{t-1}$; (ii) compute or simulate the one-step ahead forecast distribution for $\mathbf{y}_t$; solve the portfolio optimization and adjust the portfolio investment weight vector $\mathbf{w}_t$ to the new, optimized value; (iii) move to time t, observe and record the realized returns, and continue with $t \to t+1$.

Realized portfolio returns and estimated volatilities of the investment strategies in Table 6.4 are summarized in Table 6.5. Here we add a n_{miss}, the number of years during which the target return was not achieved (on an annualized basis). Across all investment strategies, superior portfolio performance in nominal as well as risk-adjusted terms is achieved by using forecasts from the SGDLM to inform investment decisions. Furthermore, adding the zero-correlation constraint to the portfolio optimization effectively reduces the realized correlations of the portfolios to the S&P 500, and increases the nominal and risk-adjusted returns. An interesting finding is that the best-performing models in terms of MSE or correlation of forecasts and realized values (Table 6.3) are not the best models to use for investment decisions. This latter point is consistent with experience in other areas of statistical and decision analytic work, where utility-guided selection of models can lead to different model structures than those favored on purely statistical metrics [e.g. Jones et al., 2005, Carvalho and West, 2007].

Figure 6.2 visualizes the portfolio value processes, in terms of cumulative returns, of the best-performing WDLM and SGDLM models using portfolio investment strategies \$3 and \$6. Under strategy \$3 using forecasts from model SGDLM 4, \$1,000 invested at the beginning of 2007 would have grown to \$2,300 after accounting for trading costs. Based on forecasts from model WDLM 2, the same investment strategy would have generated only \$1,070. Adding the SPX benchmark neutrality constraints further improved the portfolio performance, realizing growth of \$1,000 to \$2,575 and \$1,089 under these two models, respectively. To put these numbers in perspective: a passive investment in the S&P 500 would have grown into \$1,238 during the same time period. This example shows that the SGDLM is a superior model for use in decision analysis, and its adoption can lead to significant monetary gains.

		WDLM 1	WDLM 2	WDLM 3	WDLM 4	SGDLM 1	SGDLM 2	SGDLM 3	SGDLM 4	SGDLM 5
						Investment Strategy $1				
2007–2013	μ	*-0.96%*	*-0.34%*	*0.43%*	*-0.12%*	10.09%	*8.63%*	*7.11%*	10.91%	**11.25%**
	σ	11.94%	11.81%	**11.72%**	11.79%	14.08%	13.10%	12.81%	13.12%	13.41%
	μ/σ	-0.08	-0.03	0.04	-0.01	0.72	0.66	0.56	0.83	**0.84**
	ρ_{SPX}	37.62%	37.41%	39.06%	38.20%	38.27%	35.27%	34.09%	35.81%	**33.55%**
	n_{miss}	5/7	4/7	5/7	5/7	**3/7**	**3/7**	**3/7**	**3/7**	**3/7**
						Investment Strategy $2				
2007–2013	μ	*-0.46%*	*-0.15%*	*0.36%*	*-0.39%*	*9.94%*	*8.98%*	*8.16%*	*11.43%*	***12.19%***
	σ	11.99%	11.92%	**11.76%**	11.89%	13.96%	13.52%	13.29%	13.09%	13.55%
	μ/σ	-0.04	-0.01	0.03	-0.03	0.71	0.66	0.61	0.87	**0.90**
	ρ_{SPX}	36.82%	36.63%	38.29%	37.10%	36.36%	35.67%	32.92%	36.21%	**32.50%**
	n_{miss}	7/7	7/7	6/7	7/7	**3/7**	**3/7**	**3/7**	**3/7**	**3/7**
						Investment Strategy $3				
2007–2013	μ	*0.05%*	*0.96%*	*-2.08%*	*-0.21%*	9.47%	*6.74%*	*6.22%*	**11.76%**	*6.31%*
	σ	11.94%	**11.86%**	12.03%	11.96%	14.58%	13.59%	14.66%	13.75%	15.39%
	μ/σ	0.00	0.08	-0.17	-0.02	0.65	0.50	0.42	**0.86**	0.41
	ρ_{SPX}	40.84%	35.30%	35.87%	36.36%	51.12%	40.41%	37.95%	38.92%	**34.82%**
	n_{miss}	4/7	4/7	4/7	4/7	**3/7**	4/7	**3/7**	**3/7**	4/7
						Investment Strategy $4				
2007–2013	μ	*-1.13%*	*-0.71%*	*-0.38%*	*-0.11%*	10.18%	*9.14%*	*7.83%*	**12.44%**	12.13%
	σ	12.02%	12.04%	11.97%	**11.94%**	13.57%	12.88%	13.01%	13.19%	13.73%
	μ/σ	-0.09	-0.06	-0.03	-0.01	0.75	0.71	0.60	**0.94**	0.88
	ρ_{SPX}	22.08%	21.07%	22.01%	22.66%	18.12%	22.00%	17.53%	19.19%	**17.09%**
	n_{miss}	5/7	5/7	5/7	5/7	**3/7**	**3/7**	**3/7**	**3/7**	**3/7**
						Investment Strategy $5				
2007–2013	μ	*-1.04%*	*-0.22%*	*-1.08%*	*-0.30%*	*11.65%*	*8.64%*	*7.07%*	***12.64%***	*11.77%*
	σ	12.18%	12.15%	**12.08%**	12.09%	14.18%	13.23%	12.78%	13.48%	14.00%
	μ/σ	-0.09	-0.02	-0.09	-0.02	0.82	0.65	0.55	**0.94**	0.84
	ρ_{SPX}	21.76%	20.64%	22.04%	21.47%	17.38%	21.08%	18.25%	18.42%	**16.32%**
	n_{miss}	7/7	5/7	6/7	6/7	**3/7**	4/7	5/7	**3/7**	**3/7**
						Investment Strategy $6				
2007–2013	μ	*-1.28%*	*1.20%*	*-3.46%*	*-1.14%*	**13.45%**	*6.60%*	9.33%	13.36%	11.26%
	σ	**12.11%**	12.17%	12.42%	12.32%	15.13%	14.10%	15.23%	14.45%	16.02%
	μ/σ	-0.11	0.10	-0.28	-0.09	0.89	0.47	0.61	**0.92**	0.70
	ρ_{SPX}	24.69%	19.04%	19.71%	21.28%	22.53%	20.21%	**15.29%**	17.49%	16.60%
	n_{miss}	4/7	4/7	4/7	4/7	**3/7**	**3/7**	4/7	**3/7**	4/7

Table 6.5: Results of investment strategies $1–$6. μ denotes the average annualized realized returns (italicized figures indicate failure to achieve the return target); σ denotes the average annualized realized volatility; ρ_{SPX} denotes the realized correlations of daily returns with the S&P 500 benchmark; and n_{miss} denotes the number of years during which the target return was not achieved.

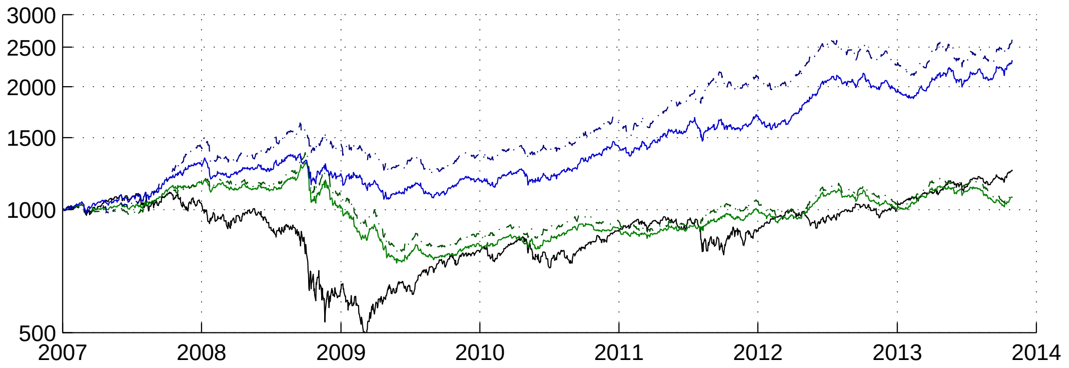


Figure 6.2: Portfolios $3 (solid lines) and $6 (dashed-dotted lines) and S&P 500 (black); the forecasts from the WDLM and SGDLM are in green and blue, respectively.

6.4 Summary Comments

Our S&P study investigated the multivariate forecast performance of the SGDLM for use in short-term forecasting and Bayesian portfolio decision analysis. The sparse, dynamic graphical model structure induced by the dynamic simultaneous parental predictor construct defines a parsimonious and potentially effective approach to structuring the contemporaneous relations in dynamic models. That is, the number of time-varying parameters to describe the structure of multivariate volatility is substantially reduced relative to standard models, which include the class of WDLMs. Data-respected and informed sparsity patterns, and adaptivity in representing such patterns as they may change over time, has the potential to improve forecasting accuracy and decisions based on such forecasts. The S&P study results bear out this potential. The SGDLM modeling approach delivers substantially improved characterizations of volatility and co-volatility, in terms of forecast accuracy as well as usability in decision processes. The latter point is clearly highlighted in our portfolio investment evaluations. Portfolios reliant on model-based forecast information yield consistently higher nominal and risk-adjusted returns relative to standard approaches, and desired optimization constraints are more reliably achieved. As just one take-home summary to add to the more detailed results discussed in the study above, we note that the empirical performance reported here shows average annualized investment returns of SGDLM-driven quantitative investment rules as high as 13.5% over a seven year period starting close to the pre-financial crisis peak in 2007 through Q3-2013; that is a period during which the annual gains of the S&P 500 averaged only 3.0% (log-returns normalized to a 252 day year).

Bibliography

Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics 44*(2), 182–198.

Abramowitz, M. and I. A. Stegun (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications.

Acerbi, C. and D. Tasche (2003). Expected Shortfall: A Natural Coherent Alternative to Value at Risk. *Economic Notes 31*, 379–388.

Agarwal, D., D. Chen, L. Lin, J. Shanmugasundaram, and E. Vee (2010). Forecasting high-dimensional data. In A. K. Elmagarmid and D. Agarwal (Eds.), *Proceedings of the ACM SIGMOD International Conference on Management of Data, Indianapolis, USA*, pp. 1003–1012. ACM.

Aguilar, O. and M. West (2000). Bayesian dynamic factor models and portfolio allocation. *Journal of Business & Economic Statistics 18*(3), 338–357.

Alspach, D. L. and H. W. Sorenson (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control 17*, 439–448.

Anacleto, O., C. Queen, and C. J. Albers (2013). Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 62*, 251–270.

Anselin, L. (1988). *Spatial econometrics: Methods and models*. Dordrecht: Kluwer Academic Publishers.

Atkinson, K. (1989). *An Introduction to Numerical Analysis*. New York: Wiley.

Ausin, M. C. and H. F. Lopes (2010). Time-varying joint distribution through copulas. *Computational Statistics & Data Analysis 54*(11), 2383–2399.

Bartram, S. M., S. J. Taylor, and Y.-H. Wang (2007). The Euro and European financial market dependence. *Journal of Banking & Finance 31*(5), 1461–1481.

Bedford, T. and R. M. Cooke (2001). Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines. *Annals of Mathematics and Artificial Intelligence 32*, 245–268.

Boost Community (2014). Boost C++ Libraries. `http://boost.org`. Accessed: 2014-03-20.

Boubaker, H. and N. Sghaier (2013). Portfolio optimization in the presence of dependent financial returns with long memory: A copula based approach. *Journal of Banking & Finance 37*(2), 361–377.

Brechmann, E. C., C. Czado, and K. Aas (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics 40*(1), 68–85.

Brechmann, E. C. and U. Schepsmeier (2013). Modeling dependence with c- and d-vine copulas: The r package cdvine. *Journal of Statistical Software 52*(3), 1–27.

Carvalho, C. M. and M. West (2007). Dynamic matrix-variate graphical models. *Bayesian Analysis 2*(1), 69–98.

Cayley, A. (1889). A theorem on trees. *The Quarterly Journal of Mathematics 23*, 376–378.

Chen, L., V. Singh, S. Guo, A. Mishra, and J. Guo (2013). Drought Analysis Using Copulas. *Journal of Hydrologic Engineering 18*(7), 797–808.

Chen, X. and Y. Fan (2006a). Estimation and model selection of semiparametric copula-based multivariate dynamic models under copula misspecification. *Journal of Econometrics 135*(1-2), 125–154.

Chen, X. and Y. Fan (2006b). Estimation of copula-based semiparametric time series models. *Journal of Econometrics 130*(2), 307–335.

Cherubini, U., E. Luciano, and W. Vecchiato (2004). *Copula Methods in Finance*. Wiley.

Cherubini, U., S. Mulinacci, and F. Gobbi (2011). *Dynamic Copula Methods in Finance*. Wiley.

COIN-OR Foundation (2014). A Package for Differentiation of C++ Algorithms. http://www.coin-or.org/CppAD/. Accessed: 2014-03-20.

Cong, R.-G. and M. Brady (2012). The Interdependence between Rainfall and Temperature: Copula Analyses. *The Scientific World Journal 2012*.

Czado, C., E. C. Brechmann, and L. Gruber (2013). Selection of vine copulas. In P. Jaworski, F. Durante, and W. K. Härdle (Eds.), *Copulae in Mathematical and Quantitative Finance*, Lecture Notes in Statistics, pp. 17–37. Springer Berlin Heidelberg.

Czado, C., S. Jeske, and M. Hofmann (2013). Selection strategies for regular vine copulae. *Journal de la Societe Francaise de Statistique 154*(1), 174–191.

Danaher, P. J. and M. S. Smith (2011). Modeling Multivariate Distributions Using Copulas: Applications in Marketing. *Marketing Science 30*, 4–21.

De Oliveira, V. and J. J. Song (2008). Bayesian analysis of simultaneous autoregressive models. *The Indian Journal of Statistics 70-B*, 323–350.

Dißmann, J. F., E. C. Brechmann, C. Czado, and D. Kurowicka (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis 59*(1), 52–69.

Dunn, W. and J. K. Shultis (2011). *Exploring Monte Carlo Methods*. Wiley.

Erhardt, T., C. Czado, and U. Schepsmeier (2015a). R-vine models for spatial time series with an application to daily mean temperature. *Biometrics*.

Erhardt, T., C. Czado, and U. Schepsmeier (2015b). Spatial composite likelihood inference using local C-vines. *Journal of Multivariate Analysis*.

Frees, E. W. and P. Wang (2006). Copula credibility for aggregate loss models. *Insurance: Mathematics and Economics 38*(2), 360–373.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*, 711–732.

Gregoire, V., C. Genest, and M. Gendron (2008). Using copulas to model price dependence in energy markets. *Energy risk 5*, 58–64.

Grimaldi, S. and F. Serinaldi (2006). Asymmetric copula in multivariate flood frequency analysis. *Advances in Water Resources 29*(8), 1155–1167.

Gruber, L. F. (2011). Bayesian Analysis of R-Vine Copulas. Master's thesis, Technische Universität München.

Gruber, L. F. and C. Czado (2015a). Bayesian Model Selection of Regular Vine Copulas. Submitted.

Gruber, L. F. and C. Czado (2015b). Sequential Bayesian Model Selection of Regular Vine Copulas. *Bayesian Analysis*. Advance Publication, 4 February 2015.

Gruber, L. F. and M. West (2015a). Bayesian Forecasting and Portfolio Decisions Using Simultaneous Graphical Dynamic Linear Models. Submitted.

Gruber, L. F. and M. West (2015b). GPU-Accelerated Bayesian Learning and Forecasting in Simultaneous Graphical Dynamic Linear Models. *Bayesian Analysis*. Advance Publication, 2 March 2015.

Harrison, P. J. and C. F. Stevens (1971). A bayesian approach to short-term forecasting. *Operations Research Quarterly 22*, 341–362.

Harrison, P. J. and C. F. Stevens (1976). Bayesian forecasting (with discussion). *Journal of the Royal Statistical Society (Series B, Methodological) 38*, 205–247.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*, 97–109.

Heinen, A. and E. Rengifo (2007). Multivariate autoregressive modeling of time series count data using copulas. *Journal of Empirical Finance 14*(4), 564–583.

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

Hosking, J. R. M., R. Natarajan, S. Ghosh, S. Subramanian, and X. Zhang (2013). Short-term forecasting of the daily load curve for residential electricity usage in the Smart Grid. *Applied Stochastic Models in Business and Industry 29*(6), 604–620.

Hu, L. (2006). Dependence patterns across financial markets: A mixed copula approach. *Applied Financial Economics 16*(10), 717–729.

Jaakkola, T. S. (2000). Tutorial on variational approximation methods. In M. Opper and D. Saad (Eds.), *Advanced Mean Field Methods: Theory and Practice*, pp. 129–159. MIT Press.

Jaakkola, T. S. and M. I. Jordan (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing 10*, 25–27.

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.

Joe, H. (2001). *Multivariate Models and Dependence Concepts*. Chapman & Hall.

Joe, H. (2014). *Dependence Modeling with Copulas*. Chapman and Hall/CRC.

Jondeau, E. and M. Rockinger (2006). The Copula-GARCH model of conditional dependencies: An international stock market application. *Journal of International Money and Finance 25*(5), 827–853.

Jones, B., A. Dobra, C. M. Carvalho, C. Hans, C. Carter, and M. West (2005). Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science 20*, 388–400.

Jordan, M. I., Z. Ghahramani, T. S. Jaakkola, and L. K. Saul (1999). An Introduction to Variational Methods for Graphical Models. *Machine Learning 37*, 183–233.

Kim, D., J.-M. Kim, S.-M. Liao, and Y.-S. Jung (2013). Mixture of D-vine copulas for modeling dependence. *Computational Statistics & Data Analysis 64*, 1–19.

Kirchhoff, G. (1847). Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. *Ann. Phys. Chem. 72*, 497–508.

Koop, G. (2012). Using VARs and TVP-VARs with many macroeconomic variables. *Central European Journal of Economic Modelling and Econometrics 4*, 143–167.

Krämer, N., E. C. Brechmann, D. Silvestrini, and C. Czado (2013). Total loss estimation using copula-based regression models. *Insurance: Mathematics and Economics 53*(3), 829–839.

Kullback, S. and R. A. Leibler (1951). On Information and Sufficiency. *Ann. Math. Statist. 22*(1), 79–86.

Kurowicka, D. and R. M. Cooke (2006). *Uncertainty Analysis with High Dimensional Dependence Modelling*. John Wiley & Sons Ltd.

Kurowicka, D. and H. Joe (Eds.) (2010). *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co. Pte. Ltd.

Lee, A., C. Yau, M. B. Giles, A. Doucet, and C. C. Holmes (2010). On the utility of graphics cards to perform massively parallel simulation with advanced Monte Carlo methods. *Journal of Computational & Graphical Statistics 19*(4), 769–789.

Lee, T.-H. and X. Long (2009). Copula-based multivariate GARCH model with uncorrelated dependent errors. *Journal of Econometrics 150*(2), 207–218.

Lopes, H. F., R. E. McCulloch, and R. Tsay (2010). Cholesky stochastic volatility. Technical report, University of Chicago, Booth Business School.

Lopes, H. F. and N. G. Polson (2010). Bayesian inference for stochastic volatility. In K. Böcker (Ed.), *Rethinking Risk Measurement and Reporting: Uncertainty, Bayesian Analysis and Expert Judgement*, pp. 515–551. Risk Books/Incisive Financial Publishing Ltd: London.

Madadgar, S. and H. Moradkhani (2013). Drought Analysis under Climate Change Using Copula. *Journal of Hydrologic Engineering 18*(7), 746–759.

Mai, J.-F. and M. Scherer (2012). *Simulating Copulas*. Number 4 in Series in Quantitative Finance. Singapore: World Scientific.

Mai, J.-F. and M. Scherer (2014). *Financial Engineering with Copulas Explained*. Financial Engineering Explained. Palgrave Macmillan.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance 7*(1), 77–91.

Marsaglia, G. and W. W. Tsang (2000). A simple method for generating gamma variables. *ACM Transactions on Mathematical Software 26*(3), 363–372.

McNeil, A. J., R. Frey, and P. Embrechts (2005). *Quantitative Risk Management: Concepts, Techniques, and Tools*. Princeton University Press.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics 21*, 1087–1092.

Min, A. and C. Czado (2010). Bayesian Inference for Multivariate Copulas Using Pair-Copula Constructions. *Journal of Financial Econometrics 8*(4), 511–546.

Min, A. and C. Czado (2011). Bayesian model selection for D-vine pair-copula constructions. *Canadian Journal of Statistics 39*(2), 239–258.

Min, A. and C. Czado (2014). SCOMDY models based on pair-copula constructions with application to exchange rates. *Computational Statistics & Data Analysis 76*, 523–535.

Morales-Napoles, O. (2011). Counting vines. In D. Kurowicka and H. Joe (Eds.), *Dependence Modeling: Vine Copula Handbook*. World Scientific Publishing Co.

Mukherjee, C., P. S. Kasibhatla, and M. West (2014). Spatially-varying SAR models and Bayesian inference for high-resolution lattice data. *Annals of the Institute of Statistical Mathematics 66*, 473–494.

Nakajima, J. and M. West (2013a). Bayesian analysis of latent threshold dynamic models. *Journal of Business & Economic Statistics 31*, 151–164.

Nakajima, J. and M. West (2013b). Bayesian dynamic factor models: Latent threshold approach. *Journal of Financial Econometrics 11*, 116–153.

Nelsen, R. B. (2006). *An Introduction to Copulas*. Springer.

Palomo, J., D. B. Dunson, and K. Bollen (2007). Bayesian structural equation modeling. In S. Lee (Ed.), *Handbook of Latent Variables and Related Models*, Volume 1 of *Handbook of Computing and Statistics with Applications*, pp. 163–188. North Holland.

Patton, A. J. (2006). Modelling asymmetric exchange rate dependence. *International Economic Review 47*(2), 527–556.

Patton, A. J. (2009). Copula-Based Models for Financial Time Series. In *Handbook of Financial Time Series*, pp. 767–785. Springer.

Petersen, K. B. and M. S. Pedersen (2008). *The Matrix Cookbook*.

Pitt, M. and N. Shephard (1999). Time varying covariances: A factor stochastic volatility approach (with discussion). In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (Eds.), *Bayesian Statistics VI*, pp. 547–570. Oxford University Press.

Prado, R. (2010). Multi-state models for mental fatigue. In A. O'Hagan and M. West (Eds.), *The Handbook of Applied Bayesian Analysis*, pp. 845–874. Oxford University Press.

Prado, R. and M. West (2010). *Time Series: Modelling, Computation & Inference*. Chapman & Hall/CRC Press.

Queen, C. M. (1994). Using the multiregression dynamic model to forecast brand sales in a competitive product market. *Journal of the Royal Statistical Society, Series D (The Statistician) 43*, 87–98.

Quintana, J. M., C. M. Carvalho, J. Scott, and T. Costigliola (2010). Futures markets, Bayesian forecasting and risk modeling. In A. O'Hagan and M. West (Eds.), *The Handbook of Applied Bayesian Analysis*, pp. 343–365. Oxford University Press.

Quintana, J. M., V. Lourdes, O. Aguilar, and J. Liu (2003). Global gambling. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (Eds.), *Bayesian Statistics 7*, pp. 349–368. Oxford University Press.

Quintana, J. M. and M. West (1987). An analysis of international exchange rates using multivariate DLMs. *The Statistician 36*, 275–281.

Renard, B. and M. Lang (2007). Use of a Gaussian copula for multivariate extreme value analysis: Some case studies in hydrology. *Advances in Water Resources 30*(4), 897–912.

Rodriguez, J. C. (2007). Measuring financial contagion: A Copula approach. *Journal of Empirical Finance 14*(3), 401–423.

Sharpe, W. F. (1966). Mutual fund performance. *The Journal of Business 39*(1), 119–138. Part 2: Supplement on Security Prices.

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.

Smith, A. F. M. and M. West (1983). Monitoring renal transplants: An application of the multi-process Kalman filter. *Biometrics 39*, 867–878.

Smith, M., A. Min, C. Almeida, and C. Czado (2010). Modeling Longitudinal Data Using a Pair-Copula Decomposition of Serial Dependence. *Journal of the American Statistical Association 105*(492), 1467–1479.

Stöber, J., H. Joe, and C. Czado (2013). Simplified Pair Copula Constructions-Limitations and Extensions. *Journal of Multivariate Analysis 119*(0), 101–118.

Suchard, M. A., Q. Wang, C. Chan, J. Frelinger, A. J. Cron, and M. West (2010). Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *Journal of Computational and Graphical Statistics 19*(2), 419–438.

Sun, J., E. W. Frees, and M. A. Rosenberg (2008). Heavy-tailed longitudinal data modeling using copulas. *Insurance: Mathematics and Economics 42*(2), 817–830.

Trejo, L. J., K. Knuth, R. Prado, R. Rosipal, K. Kubitz, R. Kochavi, B. Matthews, and Y. Zhang. (2007). EEG-based estimation of mental fatigue: Convergent evidence for a three-state model. In D. D. Schmorrow and L. M. Reeves (Eds.), *Augmented Cognition, HCII 2007, LNAI 4565*, pp. 201–211. Berlin Heidelberg: Springer-Verlag.

van den Goorbergh, R. W. J., C. Genest, and B. J. M. Werker (2005). Bivariate option pricing using dynamic copula models. *Insurance: Mathematics and Economics 37*, 101–114.

Wang, H. and M. West (2009). Bayesian analysis of matrix normal graphical models. *Biometrika 96*, 821–834.

West, M. (1993). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society (Ser. B) 54*, 553–568.

West, M. and J. Harrison (1989). Subjective intervention in formal models. *Journal of Forecasting 8*, 33–53.

West, M. and J. Harrison (1997). *Bayesian Forecasting & Dynamic Models* (2nd ed.). Springer Verlag.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika 41*, 434—449.

Yang, X., E. W. Frees, and Z. Zhang (2011). A generalized beta copula with applications in modeling multivariate long-tailed data. *Insurance: Mathematics and Economics 49*(2), 265–284.

Zhang, J. and D. Guegan (2008). Pricing bivariate option under GARCH processes with time-varying copula. *Insurance: Mathematics and Economics 42*(3), 1095–1103.

Zhang, L. and V. Singh (2006). Bivariate Flood Frequency Analysis Using the Copula Method. *Journal of Hydrologic Engineering 11*(2), 150–164.

Zhang, L. and V. Singh (2007). Bivariate rainfall frequency distributions using Archimedean copulas. *Journal of Hydrology 332*(1-2), 93–109.

Zhao, Z. Y. and M. West (2014). Dynamic compositional regression modelling: Application in financial time series forecasting and portfolio decisions. Technical report, Department of Statistical Science, Duke University.

Zhou, X., J. Nakajima, and M. West (2014). Bayesian forecasting and portfolio decisions using dynamic dependent sparse factor models. *International Journal of Forecasting 30*(4), 963–980.