MCMC Methods for Bayesian Mixtures of Copulas

Ricardo Silva
Department of Statistical Science
University College London
ricardo@stats.ucl.ac.uk

Robert B. Gramacy
Statistical Laboratory
University of Cambridge
bobby@statslab.cam.ac.uk

Abstract

Applications of copula models have been increasing in number in recent years. This class of models provides a modular parameterization of joint distributions: the specification of the marginal distributions is parameterized separately from the dependence structure of the joint, a convenient way of encoding a model for domains such as finance. Some recent advances on how to specify copulas for arbitrary dimensions have been proposed, by means of mixtures of decomposable graphical models. This paper introduces a Bayesian approach for dealing with mixtures of copulas which, due to the lack of prior conjugacy, raise computational challenges. We motivate and present families of Markov chain Monte Carlo (MCMC) proposals that exploit the particular structure of mixtures of copulas. Different algorithms are evaluated according to their mixing properties, and an application in financial forecasting with missing data illustrates the usefulness of the methodology.

1 CONTRIBUTION

We present and evaluate new approaches for computing posterior distributions of mixtures of copula models. The goal is to provide new tools for density estimation in multivariate analysis where copula models are deemed appropriate. This requires efficient Markov chain Monte Carlo (MCMC) methods that exploit the particularities of copula models, and such algorithms are evaluated according to their mixing properties.

The copula approach for multivariate analysis provides a *modular* parameterization of distributions and densities (Nelsen, 2007): the choice of families for the univariate marginals is arbitrary, and a *copula function* defines the full joint for a fixed choice of marginals. This is particularly attractive in domains where good models for individual objects are known and well-motivated, but a *dependency model* for such objects less so. A popular example is financial modeling: while models such as heavy-tailed t-distributions provide a good fit to individual stocks, it is less clear how to model the joint distribution of stock prices. The copula parameterization allows for different models while conveniently preserving a choice of univariate marginals that are well-suited to the problem. Nelsen (2007), Joe (1997), Nicoloutsopoulos (2005) and Kirshner (2007) describe the theory and applications.

Defining a copula function for distributions with three or more variables is difficult. However, through mixtures of tree-structured distributions, one can extend bivariate copulas to problems of arbitrary dimension. Kirshner (2007) describes the tree-copula formulation and a particular setup for finite mixtures, as well as a maximum likelihood approach for learning. In this paper, the challenge is how to perform Bayesian inference. Copula models are not in the exponential family and we will have to deal with non-conjugate prior distributions. In order to provide a self-contained presentation, Section 2 contains a brief description of copula models. Our mixture model is described in Section 3. The bulk of our contribution is contained in Section 4, where we detail different MCMC proposals suited for this class of models. Experiments are described in Section 5. The final application concerns an illustration of stock market predictions under missing data.

2 REVIEW OF COPULA MODELS

A bivariate *copula function* $C(u, v)$ is a cumulative distribution function (CDF) over the interval $[0, 1] \times [0, 1]$ with uniform marginals (Nelsen, 2007). If the density function exists, we denote it by $c(u, v)$. This concept is

Appearing in Proceedings of the 12^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5.

particularly useful for parameterizing bivariate distributions. If $F_i(\cdot)$ and $F_j(\cdot)$ are the marginal CDFs for Y_i and Y_j, the joint CDF $F(Y_i, Y_j)$ is fully determined by the triplet

$$\{F_i(\cdot), F_j(\cdot), C(a_i(\cdot), a_j(\cdot))\},$$

where $a_i(\cdot) \equiv F_i^{-1}(\cdot)$. Changing the copula function will define a different joint while keeping the same marginals. Changing the marginals but keeping the copula fixed will result in a different joint: in this case, however, any measure of association $\rho(Y_i, Y_j)$ that is invariant with respect to strictly monotonic transformations (e.g., taking logarithms) will remain the same. Conversely, any continuous CDF will imply an unique copula function (a fundamental result in multivariate analysis derived in *Sklar's theorem*; see, e.g., page 18 of Nelsen, 2007). An example is the density of the Gaussian copula function with parameter ρ:

$$\Phi_\rho(u, v) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u^2 + v^2 - 2\rho uv}{2(1-\rho^2)}\right) \quad (1)$$

This copula is the one implied by the Gaussian CDF, and can be used to construct non-Gaussian distributions if applied with non-Gaussian marginal CDFs. Pitt et al. (2006) describe a Bayesian approach for Gaussian copula models with applications in finance and health care. Nicoloutsopoulos (2005) discusses a variety of problems that have natural representations in terms of copulas and univariate marginals.

2.1 Tree-Copulas and Mixtures

It is, however, very hard to construct multivariate copulas for three or more random variables (Nelsen, 2007), the Gaussian being an exception. Kirshner (2007) uses a simple but clever trick to define copulas of arbitrary dimensionality d by assuming the distribution is Markov with respect to a tree $\mathcal{T}$ with edge set $\mathcal{E}(\mathcal{T})$. Assume that the data is continuous and the joint density $p(\mathbf{Y})$ exists. The copula-parameterized density function is given by

$$p(\mathbf{Y}|\mathcal{T}, \Theta) = \left[\prod_{v=1}^{d} f_v(Y_v \mid \Lambda_v)\right] c_{\mathcal{T}}(\mathbf{a}) \quad (2)$$

$$c_{\mathcal{T}}(\mathbf{a}) \equiv \prod_{\{u,v\}\in\mathcal{E}(\mathcal{T})} c_{uv}(a_u(Y_u), a_v(Y_v) \mid \Theta_{uv}),$$

where Λ_v is a set of parameters for the marginal $f_v(\cdot)$ and Θ_{uv} is a set of parameters for the copula density function $c_{uv}(\cdot,\cdot)$. The result is interesting due to the fact that the set $\{c_{uv}(\cdot,\cdot)\}$ can be an *arbitrary* set of bivariate copula densities, and $c_{\mathcal{T}}(\mathbf{a})$ is still guaranteed to be a valid multivariate copula density. The drawback is that the tree imposes many conditional independence constraints that might be undesirable.

Much more flexibility can be achieved by using mixtures of trees. Let $\mathbf{Y}^{(i)}$ denote a particular data point. The model of Kirshner is defined by

$$\mathbf{Y}^{(i)} | \mathcal{T}^{(i)}, \boldsymbol{\Theta} \sim p(\cdot \mid \mathcal{T}^{(i)}, \Theta) \quad (3)$$
$$\mathcal{T}^{(i)} \sim p_T(\cdot \mid \Theta_T)$$

where Θ consists of parameters for all marginals and copula parameters for *each pair* of variables. As in Equation (2), only parameters associated with a particular edge are used in the definition of $p(\cdot|\,\mathcal{T}^{(i)}, \Theta)$. Trees are hidden variables, with a distribution parameterized by Θ_T. Without loss of generality, the trees can always be connected graphs. The absence of an edge $Y_u - Y_v$ is equivalent to choosing the *independence copula* $C(u, v) = uv$ as the respective copula function.

In what follows, we describe a Bayesian approach and show how efficient MCMC proposals can be constructed. To the best of our knowledge, this is the first Bayesian approach to this problem.

3 MODEL AND PRIORS

Kirshner (2007) also introduced a maximum likelihood estimator (MLE) for mixtures of tree-copulas. In his model, the set of all tree models is parameterized by a single matrix of $O(d^2)$ parameters: each pair of variables is associated with a particular copula function independently of the trees. As such, different trees will share the same parameters corresponding to the common edges. Both the matrix and the mixture proportions are learned by finding their MLE. This choice of parameterization is motivated by the fact one can obtain simple expressions for updating the parameter estimates in a iterative scheme (at a cost of $O(d^3)$ per iteration), which is an adaptation of the setup described by Meilă and Jaakkola (2006). Although one can also motivate the above parameterization of the tree mixture by claiming it to be conjugate to the density of $\mathbf{Y}$ given a tree, the approach of Kirshner (2007) is not Bayesian. It also assumes there is no missing data.

In the Bayesian case, there is little motivation to use the constraints of Meilă and Jaakkola (2006), which requires massive parameter sharing across trees. The reason is that the parameters cannot be integrated out analytically, and so we will need to resort to MCMC methods anyways. Each tree-copula in our proposed mixture has a different set of parameters, allowing the potential number of mixture components to be infi-

nite.[1]

Our proposed model is related to—but different from—the mixture of trees approach of Kirshner and Smyth (2007), which is learned through a Bayesian approach but which has a different set of *marginal parameters* per mixture component. This is not suitable for problems motivated by the modular nature of copula parameterizations. Moreover, the models of Kirshner and Smyth (2007) are restricted to discrete distributions and conjugate priors.

3.1 Model Details

The basic prior for our model follows the standard Dirichlet process (DP) mixture formulation (Neal, 2000), except that only trees and copula parameters vary between different mixture components. Each tree will have the same univariate marginals, so that all univariate marginals are fully defined independently of the mixture of trees. If non-parametric marginals or copulas are desirable, they can be parameterized accordingly (Nicoloutsopoulos, 2005), but once again disentangled from the prior over trees.

Define the finite mixture model with K components as follows:

$$\begin{aligned} Y^{(i)} \mid z^{(i)}, [\mathcal{T}], \Lambda, [\Theta] &\sim p(\,\cdot \mid \mathcal{T}_{z_i}, \Lambda, \Theta_{z_i}) \\ \Lambda &\sim p_\Lambda(\,\cdot\,) \\ \Theta_z &\sim p_\Theta(\,\cdot\,) \\ z^{(i)} \mid \pi &\sim \text{Discrete}(\,\cdot \mid \pi_1, \ldots, \pi_K) \\ \mathcal{T}_z &\sim T_0(\cdot) \\ \pi &\sim \text{Dirichlet}(\,\cdot \mid \alpha/K, \ldots, \alpha/K) \end{aligned} \tag{4}$$

where $[\mathcal{T}]$ is the set of all possible trees indexed by z; $[\Theta]$ is the set of all copula parameters, a copula for each pair of variables; Λ is the set of univariate marginal parameters. The functions on the right-hand side are respective density functions or mass functions defined in the next sections.

We will assume that the joint prior for the trees, marginal and copula parameters is fully factorized, and denote the marginal priors for Λ_u and $\Theta_z \equiv \{\Theta_{uv;z}\}$ by $p_\Lambda(\Lambda_u)$ and $p_\Theta(\Theta_{uv;z})$, respectively.

The Dirichlet process mixture over trees arises in the limit, as $K \to \infty$. A diagram for the model, in the plate notation, is depicted in Figure 3.1.

3.2 Remarks on Transdimensional Methods

From the model specification, it is clear that there will be copula parameters $\Theta_{uv;z^{(i)}}$ that are independent of $\mathbf{Y}^{(i)}$ given the latent tree $\mathcal{T}^{(i)}$. Unlike Meilă and Jaakkola (2006), parameters are not re-used across trees. Unlike Kirshner and Smyth (2007), parameters are not a priori exchangeable between pairs of variables (copula families might differ for different pairs) and hence can not be interpreted as coming from the same array of dimensionality $O(d)$.

[1] In the Bayesian setup, copula parameters for a given pair of variables still naturally share a common prior. This provides a softer version of parameter sharing.

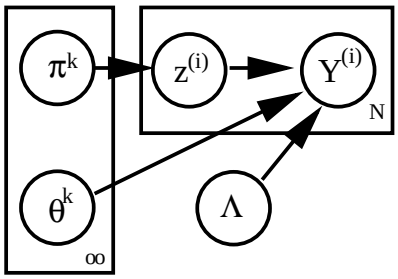

Figure 1: A plate graphical model for the generation of N data points $\mathbf{Y}$. Indicators z are hidden. The figure stresses that the mixture model is over the copula parameter Θ, which does not include the marginal parameterization Λ (which could be itself another Dirichlet process mixture).

It is sensible to question the need for such parameters. In transdimensional Monte Carlo approaches (also known as reversible jump MCMC, Green, 1995), parameters are created and destroyed as needed by moves through model space. However, the Dirichlet process mixture requires a common base measure for all component mixtures, and as such the parameter space for two different tree components has to be the same. Given the flexibility and the many successful applications of DP mixtures, we favor this approach over the finite mixture models approach.

As a matter of fact, the likelihood function for a particular tree component can be written so that it is a (trivial) function of all $O(d^2)$ copula parameters. Let $\mathbf{E}$ be the symmetric adjacency matrix representation of an undirected tree $\mathcal{T}$, and let $\mathcal{E}$ be the space of matrices encoding spanning trees. The likelihood of point $\mathbf{Y}$ is therefore described by Equation (2) with $c_\mathcal{T}(\mathbf{a})$ given by

$$c_\mathcal{T}(\mathbf{a}) = \prod_{u,v;1\le u<v\le d} c_{uv}(a_u(Y_u), a_v(Y_v))^{e_{uv}} \tag{5}$$

where $e_{uv} \in \{0,1\}$ is the respective element of $\mathbf{E}$. The base measure of the DP mixture is defined over the distribution of matrices in $\mathcal{E}$ and the common distribution of copula parameters. We have to guarantee that changes in $\mathbf{E}$ result in a valid model, and as such the (fixed) space of parameters must account for all possible combinations of edges.

The fact that, for a fixed tree, some parameters play a trivial role in the likelihood function will have implications in the MCMC approaches, as we will see.

4 FAMILIES OF PROPOSALS

We need to sample parameters $\Theta_z \cup \Lambda \cup \pi$, trees $\mathcal{T}_z$ and latent variables $\mathbf{z}$. Sampling $\mathbf{z}$ given the other elements can be done using Algorithm 8 of Neal (2000) and we will not discuss it further. In what follows we describe the proposals to be used in a Metropolis-Hastings (MH) scheme when sampling parameters and trees. The non-standard step consists of sampling trees given all other elements, which we now describe in full detail.

4.1 Tree Proposals

We cannot sample from the posterior marginal of $\mathcal{T}$ directly, but sampling becomes doable after conditioning on the copula parameters, i.e., by sampling from the distribution $T_{\Theta_z}(\mathcal{T}_z) \equiv P(\mathcal{T}_z \mid \Theta_z)$. As a matter of fact, this distribution assumes the form

$$T_\Theta(\mathcal{T}) = \frac{1}{Z_{T_\Theta}} \prod_{\{u,v\}\in\mathcal{E}(\mathcal{T})} \beta_{uv}. \tag{6}$$

It is well-known that there are exact and randomized algorithms for sampling from such a distribution, and this has been successfully applied in other MCMC contexts (Kirshner and Smyth, 2007). However, in our case we face the challenge that the weights β_{uv} are not obtained after marginalizing parameters, as in Kirshner and Smyth (2007), but conditioned on parameters sampled from a MCMC iteration. Importantly, most of such parameters were not associated with any data point in the previous iteration. Sampling from (6) has unwanted consequences: the previously detached parameters were sampled from the prior (as implied by (5)), and therefore are likely to be bad choices for modeling the data points in that particular cluster.

As a matter of fact, a pilot implementation revealed that the randomized algorithms for sampling from (6) are useless for all practical purposes: most copula density functions are essentially zero, which implies the algorithm will not converge in any reasonable amount of time. The exact algorithm is also problematic: not only might the mixing of the chain be bad due to a dependence on a large uninformative set of parameters, but also due to the extreme variability of the copula densities which translate into numerical instabilities. For problems with conjugate priors or small dimensional ones, the exact algorithm is helpful. However, in general we will need a different approach.

We now define three different tree proposals that can be used in a MH scheme.

The SIMPLE proposal: the simplest tree proposal consists of choosing an edge $Y_u - Y_v$ uniformly at random and moving it uniformly at random to any legal place $Y_m - Y_n$ that will result in a new spanning tree. Although we are still keeping the same sampled parameters, this local change makes only a small modification to the tree model. Parameters associated with the unchanged edges remain in the likelihood function. However, it is fast and easy to implement.

The TREEANGLE proposal: the SIMPLE proposal has two shortcomings. In the choice of edge, the resulting path connecting Y_u and Y_v can be very large, and their association will typically be much smaller than the one in the current tree model. The chances of rejection for this case can be higher than in moves that do not propose a great excursion from the current associations implied by the tree. Moreover, it will be sensible to choose a new set of parameter values in tandem with the new edge. An arbitrary edge move might complicate the choice of new sensible values for the parameters of the added edge $Y_m - Y_n$.

To increase acceptance, we favor an approach that makes more localized changes and proposes new parameter values simultaneously. For a fixed tree $\mathcal{T}$, define a proposal $q(T^\star \mid \mathcal{T})$ by the following algorithm:

1. for each $Y_i - Y_j$ in $\mathcal{E}(\mathcal{T})$, let $w_{ij} = (\#n_i - 1) + (\#n_j - 1)$, where $\#n_i$ is the number of neighbors of Y_i in $\mathcal{T}$. Let $W_\mathcal{T} = \sum_{\{i,j\}\in\mathcal{E}(\mathcal{T})} w_{ij}$
2. choose an edge $Y_u - Y_v$ from $\mathcal{T}$ with probability $w_{uv}/W_\mathcal{T}$
3. choose a neighbor Y_t of $\{Y_u, Y_v\}$ in $\mathcal{T}\backslash\{Y_u, Y_v\}$ uniformly at random
4. return a tree $\mathcal{T}^\star$ that results from removing from $\mathcal{T}$ the edge $Y_u - Y_v$ and adding the edge $Y_\star - Y_t$, where $Y_\star = Y_u$ if Y_v and Y_t are adjacent in $\mathcal{T}$, and otherwise $Y_\star = Y_v$.

We call such a tree modification a *treeangular move*, because it changes a "treeangle" $Y_i - Y_j - Y_k$ in $\mathcal{T}$ into a new subpath $Y_j - Y_i - Y_k$. The choice of treeangle is uniform, since $q(\mathcal{T}^\star|\mathcal{T}) \propto 1$ for all $\mathcal{T}^\star$. This move always results in a spanning tree, as illustrated by Figure 2(a)-(b), and can be computed at a cost of $O(d)$. It is possible to traverse the whole space of spanning trees with sequences of treeangular moves.

Proposition 1 *Let $\mathcal{T}$ and $\mathcal{T}^\star$ be two spanning trees. Then there is a sequence of treeangular moves that transforms $\mathcal{T}$ into $\mathcal{T}^\star$.*

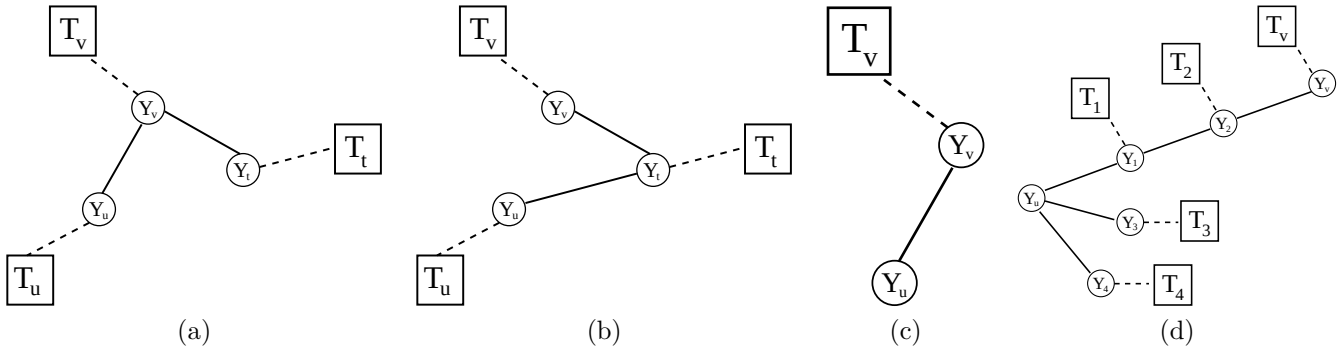

Figure 2: In the pictures above, circles represent vertices and squares represent subtrees. Dashed edges represent the possibility that a circled vertex is adjacent to several vertices in the subtree. The treeangular move that replaces the edge $Y_u - Y_v$ in (a) with edge $Y_u - Y_t$ always results in another spanning tree, as illustrated in (b). To obtain an tree in which Y_u has a single neighbor Y_v, as illustrated by (c), one can start from an arbitrary tree such as (d), and sequentially "move" Y_u towards Y_v in the (unique) path between them using treeangular moves. After connecting Y_u with Y_v, other neighbors of Y_u can be eliminated also by treeangular moves.

Proof: Every spanning tree has at least one vertex with exactly one neighbor. Let Y_u be such a vertex in $\mathcal{T}^\star$ and Y_v its respective neighbor, as in Figure 2(c). If Y_u and Y_v are not neighbors in $\mathcal{T}$, there is an unique path $Y_u - Y_{p(1)} - Y_{p(2)} - \cdots - Y_v$ in $\mathcal{T}$ (where it might be possible that $Y_{p(2)} = Y_v$). Since Y_u and $Y_{p(2)}$ are not adjacent, exchange edge $Y_u - Y_{p(1)}$ with $Y_u - Y_{p(2)}$ using the local triangular move. This can be propagated through the path until Y_u and Y_v are adjacent, as illustrated by Figure 2(d).

If Y_u has other neighbors, they can be passed to Y_v again with the same move until Y_u has Y_v as its single neighbor. The edge $Y_u - Y_v$ exists in both trees, and it is not part of any path but the one-edge path containing $Y_u - Y_v$. It is then possible to ignore the existence of this edge in both trees and repeat the process until $\mathcal{T} = \mathcal{T}^\star$. □

Given a proposed tree $\mathcal{T}^\star$ with new edge $Y_u - Y_t$, we also sample a new parameter vector $\Theta^\star_{ut}$ for the corresponding tree. Ideally, the new value should be based on the current *implied copula* that results from the distribution based on $\mathcal{T}$, i.e., the copula that corresponds to the joint of $\{Y_u, Y_t\}$ as encoded by the current $\{\Theta, \mathcal{T}\}$.

Even though the dimensionality of our model is fixed, we can follow the common practice in the literature of transdimensional Monte Carlo methods (Green, 1995): parameterize the proposal distribution of the new parameter set $\Theta^\star_{ut}$ according to functionals of the implied copula. In particular, in copula functions with *one parameter* only ($\theta^\star_{ut}$), there is usually a one-to-one correspondence between $\theta^\star_{ut}$ and a canonical measure of association between Y_u and Y_v. This includes, for instance, Kendall's tau and Spearman's rank correlation ρ (e.g., see the tables in Chapter 5 of Joe, 1997).

For the rest of the paper, consider the case where all bivariate copulas that define the tree-copulas are one-parameter functions. This includes a large number of families of copulas, including the Gaussian. The following is a template for proposals for a new $\theta^\star_{ut}$ to be sampled along a treeangle move $Y_u - Y_v - Y_t \rightarrow Y_u - Y_t - Y_v$:

1. calculate the rank correlation ρ_{uv} corresponding to θ_{uv}
2. calculate the rank correlation ρ_{ut} corresponding to the copula function implied by $Y_u - Y_v - Y_t$
3. find a ρ^0_{ut} that trades-off the following:
   (a) ρ^0_{ut} is "close" to ρ_{ut}
   (b) ρ_{uv} is "close" to the implied rank correlation of Y_u and Y_v as given by the path $Y_u - Y_t - Y_v$ and parameters θ_{tv} and $\theta(\rho^0_{ut})$
4. calculate θ^0_{ut} from ρ^0_{ut}
5. propose $\theta^\star_{ut}$ from a distribution parameterized by θ^0_{ut}

A possible measure of distance between ranks is the Euclidean distance, and the sum of the individual distances in 3.(a) and 3.(b) above defines a simple trade-off to be minimized. In general, the transformation into rank correlations defines a copula-agnostic common scale to ease the choice of distance measure. In practice, mapping between parameters and rank correlations, and finding the corresponding ρ^0_{ut}, will require a numerical procedure. However, since we are dealing with a density over three variables only, pre-computed tabulated solutions can be feasibly provided as reasonable approximations. For instance, Joe (1997) provides several tables for approximately mapping parameters to rank correlations (the approximation does not

affect the correctness of the sampler, since the MH procedure will provide detailed balance as long as we provide the proposal probabilities according to the approximation). Furthermore, one important case can be solved analytically: when all bivariate copulas are Gaussian. An example in the Gaussian case will help to clarify the procedure.

Example (Gaussian copulas): If each copula in the path $Y_u - Y_v - Y_t$ is Gaussian with parameters $\{\theta_{uv}, \theta_{vt}\}$, respectively, it is known that the implied copula for $\{Y_u, Y_t\}$ is also Gaussian with parameter $\theta_{uv}\theta_{vt}$ (for simplicity, in this example we are bypassing the rank correlation transformation and working directly in the parameter space). We can then choose our θ^0_{ut} as the one that minimizes

$$(\theta^0_{ut} - \theta_{uv}\theta_{vt})^2 + (\theta_{uv} - \theta^0_{ut}\theta_{vt})^2 \tag{7}$$

where the first term in the sum corresponds to the distance in 3.(a) and, the second term, to the distance in 3.(b) (with the sum of both defining the trade-off). The solution is given by $\theta^0_{ut} = 2\theta_{uv}\theta_{vt}/(1+\theta^2_{vt})$. We then sample $\theta^\star_{ut}$ from a uniform distribution in $(\max(-1, \theta^0_{ut} - w), \min(1, \theta^0_{ut} + w))$ for a user-specified parameter w (recall that the Gaussian copula parameter lies in $[-1, 1]$). □

Let $q_{uvt}(\theta^\star_{ut;z} \mid \mathcal{T}_z)$ represent the parameter proposal within cluster z as determined by the choice of treeangle move $Y_u - Y_v - Y_t$ (where, of course, we are conditioning on the other parameters, data, and latent variables). We also need to define a proposal for $\theta^\star_{uv}$ in order for the move to be reversible. One possibility is to define a proposal based on the sampled value of $\theta^\star_{ut}$ using an analog mechanism. However, notice that this parameter will not affect the likelihood of the new tree model. As such, we suggest taking the proposal simply be the prior, i.e., $q_{uvt}(\theta^\star_{uv;z} \mid \theta^\star_{ut;z}, \mathcal{T}_z) \equiv p_\Theta(\theta^\star_{uv})$.

Let $\mathcal{D}_z$ be the subset of the data currently assigned to cluster z. To summarize, the acceptance probability of the coupled tree-parameter proposal is given by the $\min\{1, \mathcal{R}\}$, where

$$\mathcal{R} = \left[\prod_{\mathbf{Y}^{(i)} \in \mathcal{D}_z} \frac{c_{ut}(a(Y^{(i)}_u), a(Y^{(i)}_t) | \theta^\star_{ut;z})}{c_{uv}(a(Y^{(i)}_u), a(Y^{(i)}_v) | \theta_{uv;z})}\right] \frac{q(\mathcal{T}_z \mid \mathcal{T}^\star_z)}{q(\mathcal{T}^\star_z \mid \mathcal{T}_z)} \tag{8}$$
$$\times \frac{p_\Theta(\theta^\star_{ut;z}) p_\Theta(\theta^\star_{uv;z})}{p_\Theta(\theta_{ut;z}) p_\Theta(\theta_{uv;z})} \frac{T_0(\mathcal{T}^\star)}{T_0(\mathcal{T})} \frac{q_{uvt}(\theta_{ut;z}, \theta_{uv;z} \mid \mathcal{T}^\star_z)}{q_{uvt}(\theta^\star_{ut;z}, \theta^\star_{uv;z} \mid \mathcal{T}_z)}$$

The Hybrid proposal: finally, we suggest a more computer-intensive variation of the Treeangle proposal. Given a treeangle $Y_u - Y_v - Y_t$, calculate the marginal likelihood of the copula densities of $Y_u - Y_t - Y_v$ and $Y_v - Y_u - Y_t$ and choose between one of the two subtree structures with weights proportional to the evaluated marginal likelihoods. One-parameter copulas with fully factorized priors require only the (numerical) computation of three unidimensional integrals: those corresponding to the marginal likelihoods of $Y_u - Y_t$, $Y_u - Y_v$ and $Y_t - Y_v$. This can be efficiently done by quadrature methods. Once the new subtree structure is chosen, we propose new parameters using the same proposal of Treeangle and evaluate the joint (tree, parameters) proposal using the same MH update (8), but with a different tree proposal $q'(\mathcal{T}^\star_z \mid \mathcal{T}_z)$. Since we are combining a quadrature method with MCMC updates, we call this the Hybrid method.

The overhead of solving integrals numerically is strongly amortized by increasing sample sizes and dimensionality. Since each point belongs to a single cluster at each MCMC iteration, a single pass through the data is performed for all trees in all clusters in any given iteration. Moreover, the cost does not increase with the dimensionality of the data. Meanwhile, several passes will be necessary in order to update the marginal parameters at a $O(d)$ cost.

4.2 Sampling Parameters

Sampling given marginal or copula parameters by fixing all other parameters can be done by several standard approaches, such as MH or slice sampling (Neal, 2003). It is relevant to discuss the required factors used in evaluating a MH proposal for the marginal parameters Λ_u of a given variable Y_u. Let $\mathcal{D}_{uv}$ denote the set of datapoints associated with trees where the edge $Y_u - Y_v$ exists. When proposing new marginal parameters $\Lambda^\star_u$, let the respective ratio $\mathcal{R}_u$ (proposals omitted) be:

$$\mathcal{R}_u \equiv \left[\prod_{i=1}^{d} \frac{f(Y^{(i)}_u \mid \Lambda^\star_u)}{f(Y^{(i)}_u \mid \Lambda_u)}\right] \times \frac{p_\Lambda(\Lambda^\star_u)}{p_\Lambda(\Lambda_u)} \tag{9}$$
$$\times \left\{\prod_{v \neq u} \left[\prod_{\mathbf{Y}^{(i)} \in \mathcal{D}_{uv}} \frac{c_{uv}(a^\star(Y^{(i)}_u), a(Y^{(i)}_v))}{c_{uv}(a(Y^{(i)}_u), a(Y^{(i)}_v))}\right]\right\}$$

For notational simplicity, we also omitted the dependency of $c_{uv}(\cdot, \cdot)$ on $\Theta_{uv;z}$. This clarifies the comment made at the end of the previous section: when a new $\Lambda^\star_u$ is accepted, we will still need to make a new (possibly partial) pass through the dataset when proposing $\Lambda^\star_v$ for any Y_v connected to Y_u in some tree. This follows from the fact that all $a(Y_u(i))$ have been modified, since $a(Y^{(i)}_u) \equiv F^{-1}(Y^{(i)}_u)$, a function of Λ_u.

Other remarks: Strictly speaking, it seems we need to sample all copula parameters for a particular cluster z at each iteration of parameter sampling. However, only $O(d)$ parameters affect the likelihood function in any given cluster, as discussed. It seems wasteful to

sample the remaining parameters for applications that do not need them (e.g., prediction problems). It is indeed the case that we never need to explicitly sample $O(d^2)$ parameters. Since the parameters not associated with any tree are by definition sampled from the prior, no history of such parameters is necessary when proposing a new MH move. Instead, we can sample the "old" parameter Θ_{ut} on-demand, by sampling it from the prior when a new edge $Y_u - Y_t$ has to be evaluated, as if it was sampled in the previous iteration. Notice that an exact tree sampler, which requires the evaluation of (6), cannot take advantage of this shortcut and will require sampling all parameters.

5 EXPERIMENTS

We evaluate how the different proposals compare in Section 5.1 and illustrate a simple application of the MCMC methodology in 5.2. For simplicity, we define T_0 to be the uniform distribution over spanning trees so that we can efficiently sample from this distribution, as required by Algorithm 8 of Neal (2000). A stepping-out slice sampler (Neal, 2003) is used to sample copula parameters. MH with Gaussian or uniform proposals is used to sample marginals.

5.1 Algorithm Comparison

We use Gaussian bivariate copulas as the base copula, with uniform priors in [-1, 1] for the copula parameters. In the first experiment, we compare SIMPLE, TREEANGLE, HYBRID and an exact algorithm (EXACT) for sampling from (6) without changing parameters[2]. Marginals are set to the empirical CDF and fixed throughout the whole procedure to better evaluate the differences between the copula samplers.

We chose nine datasets from the UCI Repository (Blake and Merz, 1998). Discrete variables were removed (defined to be variables that take three or fewer distinct values in the training set). A burn-in period of 1,000 points is followed by 50,000 samples which we used in the comparison. We summarize the results in Table 1. We choose the copula correlation matrix implicitly encoded by the tree-copulas, averaged over training points, as a cluster label-independent statistic to be traced. We calculate the effective sample size (ESS) (Kass et al., 1998) of each independent entry in the average matrix (a total of $d(d-1)/2$ entries), and *adjust* it by dividing by the total sampling time of the algorithm. For each entry, we then calculate the ratios of the adjusted ESS for HYBRID (H) with respect to each of the other algorithms ((S)imple, (T)reengle and (E)xact) and report the average over the matrix entries in Table 1. We also report the non-adjusted ratios (i.e., without correcting for the computing time) to give a better idea of the improvements of the HYBRID algorithm, since the time difference between the algorithms tends to zero for larger datasets and non-fixed marginals. These are reported as the H/*n columns. We also report the acceptance rate for the tree moves (Acc*) (we omit the EXACT algorithm, since it is not comparable to the others).

For most of the experiments, there is a clear advantage of TREEANGLE and HYBRID over the others, and an advantage of HYBRID over TREEANGLE[3]. This is reflected both by measuring the ESS of the average latent copula correlation matrix, and through the acceptance rate of different trees. It is also clear that with the default sampling parameters, some difficulties arise for all samplers with CLOUD as the acceptance rate for the trees is overall low. This can be partially explained by plotting the data: it can be seen that there are strongly non-linear pairwise trends that might create difficulties for the mixture of Gaussian copulas. Nevertheless, it is clear that the proposed samplers show consistent improvement over standard approaches.

5.2 Missing Data Example

In order to show a simple application that cannot be performed by analytically integrating trees even under conjugate conditions (Meilă and Jaakkola, 2006), we apply the Bayesian copula mixture to a problem with missing data. In stock market data analysis, it is common to have missing measurements at any particular time point − partially because some of the stocks did not exist at that time. We used the monthly returns data from NYSE and AMEX from 1968–1998 described by Gramacy et al. (2008). As a test set, we removed the monthly returns for the stocks in the final year. Tree-copulas with Gaussian components are used along with t-marginals under Jeffrey's prior. We ran 10 different trials by randomly selecting 10 stocks with no missing data (from a total of over 1200), plus another 20 stocks with 10–100 missing entries. We compare the full MCMC methodology where missing data is sampled using a standard MH procedure, against an

[2] In each step, if we could not perform the required matrix inversions due to numerical instabilities, we would apply the SIMPLE procedure. The proposal for the copula parameters is the one described in the Example of Section 4.1 with $w = 0.15$. We initialize the clusters with 10 partitions by uniform sampling, and then do k-means as follows: fit the maximum likelihood tree for each cluster and compute the similarity of each point to each cluster using its log-likelihood; points are reassigned to the most similar cluster and the process iterated. Parameters and trees are initialized by their MLEs. The DP hyperparameter α is given a Gamma$(0.1, 1)$ prior.

[3] The exceptions are the datasets GLASS and YEAST. The difficulty in GLASS might be due to several variables having the majority of their values at zero.

Table 1: Comparison of the ratio of the average effective sample sizes for the different algorithms (H, S, T, E) in 9 different datasets, as explained in the text. In the table, d refers to the number of variables and N to the sample size. The respective proportion of accepted trees is given as the last three columns.

Dataset	d	N	H/S	H/T	H/E	T/S	T/E	H/Sn	H/Tn	H/En	AccS	AccT	AccH
CLOUD	10	1024	2.35	1.06	4.27	4.30	8.01	3.17	1.36	5.50	0.010	0.036	0.031
CONCRETE	9	1030	3.23	1.98	1.20	1.83	0.74	4.25	2.58	1.78	0.066	0.102	0.137
ECOLI	5	336	1.62	1.43	1.03	1.20	0.82	3.03	2.70	1.84	0.168	0.204	0.245
FIRE	7	517	1.34	1.15	2.17	1.14	2.15	1.79	1.58	3.07	0.147	0.178	0.218
GLASS	8	214	0.81	0.84	0.89	1.03	1.19	1.09	1.14	1.14	0.112	0.172	0.214
SEG	16	205	9.16	1.13	8.02	9.17	8.55	10.5	1.36	8.23	0.047	0.088	0.141
VOWEL	10	990	1.17	1.84	1.23	1.03	0.95	1.71	2.42	1.98	0.070	0.091	0.141
WDBC	30	569	6.38	3.04	4.86	5.52	6.96	7.00	3.22	4.02	0.028	0.088	0.110
YEAST	6	1484	0.94	0.67	1.18	1.40	1.77	1.48	1.08	1.80	0.208	0.262	0.315

Table 2: Predictive log-likelihood for 10 trials with the financial data with two methods for missing data treatment (SAMPLED and AVG).

Trial	Sampled	Avg	Trial	Sampled	Avg
1	-35.07	-35.47	6	-36.10	-35.57
2	-38.89	-39.49	7	-34.87	-35.38
3	-38.39	-40.15	8	-36.27	-36.55
4	-36.95	-37.27	9	-34.62	-34.61
5	-35.80	-37.00	10	-36.12	-36.75

off-the-shelf estimator that fills in the missing entries with the average of the observed entries. The HYBRID sampler is used in both cases. There is a total of 361 training points and 12 test points. The average log-likelihood of the test set is calculated under both approaches and the result is shown in in Table 5.2. There is a consistent advantage to treating the missing data as part of the inference process that marginalizes latent variables.

6 CONCLUSION

We described how MCMC approaches for a Bayesian mixture of copulas can be efficiently designed. An unusual characteristic of this problem is the fact that there will be parameters that, at any sampling stage, are independent of the data given other parameters (i.e., the tree adjacency matrices). Although this is related to transdimensional MCMC, we are not aware of other problems with this exact characteristic that have been tackled with MCMC methods. As future work, we seek practical ways of imposing hierarchical priors over parameters from different trees. develop specialized approaches for special patterns of missing data, such as monotone missingness patterns, for computational gains in higher dimensional problems. Performing model selection on the types of copulas used in each edge is also an open challenge. In particular, an analogue of the structure learning problem in general Markov random fields corresponds in our formalism to the problem of choosing which pairs should be given the independence copula across trees.

Acknowledgements

The algorithm and code for the EXACT method was kindly provided by Sergey Kirshner. This work was supported by EPSRC Grant #EP/D065704/1.

References

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

R. B. Gramacy, J. Lee, and R. Silva. On estimating covariances between many assets with histories of highly variable length. Technical Report 0710.5837, arXiv, 2008. url: http://arxiv.org/abs/0710.5837.

P. Green. Reversible jump MCMC computation and Bayesian model determination. *Biometrika*, 82, 1995.

H. Joe. *Multivariate Models and Dependencies Concepts*. Chapman-Hall, 1997.

R. Kass, B. Carlin, A. Gelman, and R. Neal. Markov chain Monte Carlo in practice: a roundtable discussion. *The American Statistician*, 52, 1998.

S. Kirshner. Learning with tree-averaged densities and distributions. *NIPS*, 2007.

S. Kirshner and P. Smyth. Infinite mixtures of trees. *ICML*, 2007.

M. Meilă and T. Jaakkola. Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 2006.

R. Neal. Markov chain sampling methods for Dirichlet process mixture models. *J. Comp. Graph. Stats.*, 9, 2000.

R. Neal. Slice sampling. *The Annals of Statistics*, 31, 2003.

R. Nelsen. *An Introduction to Copulas*. Springer, 2007.

D. Nicoloutsopoulos. *Parametric and Bayesian Nonparametric Estimation of Copulas*. PhD Thesis, University College London, 2005.

M. Pitt, D. Chan, and R. Kohn. Efficient Bayes. inf. for Gaussian copula regress. models. *Biometrika*, 93, 2006.

Errata (20/04/2009):

There is a typo in page 2, line 5: the definition of $a_i(\cdot)$ should be $a_i \equiv F_i(\cdot)$ (this aliasing is convenient to make it equal to the conventional notation in the copula literature).

Thanks to Frederik Eaton for pointing this out.