Latent Tree Copulas

Sergey Kirshner
Department of Statistics
Purdue University, USA
skirshne@purdue.edu

Abstract

We propose a new approach for estimation of joint densities for continuous observations using latent tree models for copulas, joint distributions with uniform $\mathcal{U}(0,1)$ marginals. Latent tree copulas combine the advantages of the parametrization of the joint density using only bivariate distributions with the ability to approximate complex dependencies with the help of latent variables. The proposed model can also be used to organize the variables in a tree hierarchy. We describe algorithms for estimating binary latent tree copulas from data for both Gaussian and non-Gaussian copulas.

1 Introduction

Many domains generate multivariate real-valued data (e.g., biology, finance, hydrology). Among possible approaches, non-parametric representations of the joint densities suffer from the curse of dimensionality, limiting their use to low-dimensional settings. On the other hand, while there are many parametric forms for one-dimensional densities, the choices for multivariate densities with desirable properties (computationally convenient functional forms, easy estimation of marginal, conditional, and posterior densities) are limited, with the best known, understood, and used tool, multivariate normal distributions, not always applicable for the domain. The latter issue has been one of the reasons why probabilistic graphical models (PGMs) have been slow to extend to non-Gaussian cases.

Copulas, multivariate distributions with uniform $\mathcal{U}(0,1)$ marginals (e.g., Joe, 1997, Nelsen, 2006), provide a convenient framework for modeling of multivariate distributions as this task can be split into two parts: (1) modeling of the univariate marginal distributions, and (2) modeling of the copula which would "bind" the univariate marginals together to make up a joint distribution. This approach provides flexibility of separate specification of the functional form for the copula and the marginals. Copulas can be viewed as a canonical representation for the dependence between the random variables as they preserve the conditional independence relations between the variables while fixing their marginals. This motivates the use of PGMs for the copulas rather than the joint densities, and recent efforts have been extending PGM techniques to copulas (e.g., Elidan, 2010a,b, Kirshner, 2007).

In this paper, we propose *latent tree copulas* (LTCs), tree-structured copulas with some of the variables hidden (not observed), for modeling of multivariate densities. This approach combines two frameworks, tree-structured copulas (Kirshner, 2007) and latent tree models (Zhang, 2004), borrowing strengths from both. Tree-structured copulas decompose the joint copula density into a product of bivariate copula densities (corresponding to the edges in the acyclic conditional independence graph) allowing to use the existing bivariate copula machinery for construction of high-dimensional copulas. The requirement of only bivariate copulas is important because while the choices for bivariate copulas have been thoroughly investigated, only relatively few bivariate functional families have useful multivariate generalizations. However, the conditional independence assumptions imposed by the tree-structure are often unreal-

istic. Borrowing an idea from latent tree models and introducing latent variables allows modeling of more complex dependence relations.

This paper contributes a new parsimonious model for multivariate data and algorithms for its inference and parameter estimation. We also propose a learning algorithm (in the vein of Harmeling and Williams, 2011) to estimate binary latent tree copulas from data. The learned structure can be used for hierarchical clustering of the variables or features.[1]

First, we place the contributions of this paper in the context of related work (Section 1.1). We then introduce copulas, and a particular tree-structured subclass of them (Section 2). This provides enough background to introduce the latent copula trees (Section 3). Estimation of latent copula trees from data is considered in Section 4. Empirical illustration of the above approaches is carried out in Section 5. We conclude in Section 6.

1.1 Related Work

So far, the approaches for learning of latent tree models focus mostly on the discrete random variables (Harmeling and Williams, 2011, Zhang, 2004, Zhang and Kočka, 2004) although recent work of Choi et al. (2011) considered the jointly normal case in addition to discrete variables. Copulas (e.g., Joe, 1997, Nelsen, 2006) are becoming an important tool for dealing with non-Gaussian data with applications in many areas, e.g., finance (Cherubini et al., 2004), Hydrology (Genest and Favre, 2007), and they are attracting the attention of the machine learning community.[2] Several graphical models have been considered for copulas including trees (Kirshner, 2007) which decompose the joint copula density into a a product of bivariate copula densities, and several of their generalizations, among them, vines which incorporate additional conditional dependence with nested bivariate copulas (Aas et al., 2009, Bedford and Cooke, 2002, Kurowicka and Cooke, 2006), and copula Bayesian networks (Elidan, 2010a) which decompose the joint copula density into a product of conditional copula densities and can handle missing data using variational approach (Elidan, 2010b) (although the case of *latent* variables was not explored). A related (albeit not a copula) model is a cumulative distribution network (Huang and Frey, 2008) which models the distribution function as a product of distribution function factors. Its recent extension, a mixed cumulative distribution network (Silva et al., 2011) uses implicit hidden variables.

[1] An implementation of the algorithms described in the paper is avaiable for download from `http://www.stat.purdue.edu/~skirshne/LTC/`.

[2] NIPS 2011 workshop on Copulas in Machine Learning, December 2011, `http://pluto.huji.ac.il/~galelidan/CopulaWorkshop/abstracts.html`.

2 Tree-Structured Copulas

Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a vector of random variables with support $Val(X) \subseteq \mathbb{R}^d$. Let $F(\boldsymbol{x}) = F(\boldsymbol{X} = \boldsymbol{x})$ be a cumulative distribution function (cdf) for $\boldsymbol{X}$, and assume that F is absolutely continuous. Denote by $f(\boldsymbol{x})$ the probability density function (pdf) for $\boldsymbol{X}$. Let $F_u(x_u)$ and $f_u(x_u)$, $u = 1, \ldots, d$, denote the marginal cdfs and pdfs, respectively, of $\boldsymbol{X}$.

2.1 Copulas

Copulas provide a convenient framework for modeling of multivariate distributions by separating the marginals from the multivariate dependence. Let $a_u = F_u(x_u)$ be the marginal distribution function for variable X_u, and let $\boldsymbol{a} = (a_1, \ldots, a_d)$. Denote by $\mathbb{I} = (0, 1)$ the unit interval. A *copula* associated with F is a distribution function (cdf) $C : \mathbb{I}^d \to \mathbb{I}$ satisfying

$$F(\boldsymbol{x}) = C(F_1(x_1), \ldots, F_d(x_d)), \ \boldsymbol{x} \in \mathbb{R}^d,$$

and if F is absolutely continuous on $\mathbb{R}^d$, $C(\boldsymbol{a}) = F\left(F_1^{-1}(a_1), \ldots, F_d^{-1}(a_d)\right)$, and such decomposition of a distribution into its marginals and its copula is unique (Sklar's Theorem, Sklar, 1959).[3] For the absolutely continuous case, the probability density function f can be represented in terms of the marginal densities

[3] If F is not absolutely continuous, C can be obtained using the generalized inverse of the marginal cdfs, and the associated with F copula function is uniquely defined on the absolutely continuous region of the support for $\boldsymbol{X}$.

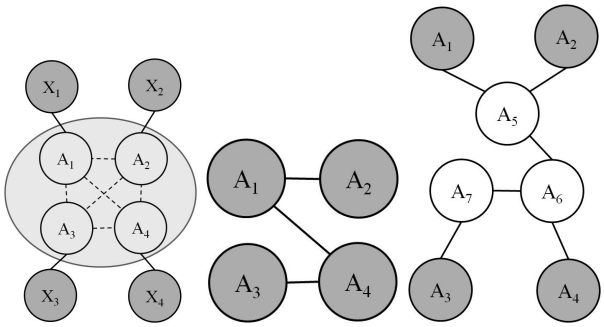

Figure 1: Left: graphical model for copulas. Copula variables are $A_1, \ldots, A_4$ with unknown dependence; original variables are $X_1, \ldots, X_4$ (observed). Middle: tree-structured copula model for $A_1, \ldots, A_4$. Right: latent tree copula model for $A_1, \ldots, A_4$.

$f_1, \ldots, f_d$ and the *copula density function* c:

$$f(\boldsymbol{x}) = c(\boldsymbol{a}) \prod_{u=1}^{d} f_u(x_u), \quad c(\boldsymbol{a}) = \frac{\partial^d C(\boldsymbol{a})}{\partial a_1 \ldots \partial a_d}. \tag{1}$$

Thus a multivariate distribution can be constructed by choosing univariate marginals $F_1, \ldots, F_d$, and then coupling or "gluing" them together with a *separately* chosen multivariate distribution, copula C. Further, the product decomposition in (1) suggests a convenient parameter estimation approach for a multivariate distribution by first estimating the marginals, transforming individual components independently according to the estimated marginals, and then estimating the parameters of the copula based on the transformed values. For a graphical model of a copula see Figure 1 (left).

For in-depth treatment of copulas, please see e.g., Joe (1997), Nelsen (2006). However, we will introduce the Gaussian copula family as it has a number of useful properties discussed in this paper. Suppose $(X_1, X_2)^T$ is a vector of two jointly Gaussian random variables (r.v.s) with mean $(\mu_1, \mu_2)^T$ and covariance matrix $\left(\begin{smallmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{smallmatrix}\right)$. The copula associated with the distribution over (X_1, X_2) is

$$C(a_1, a_2) = \Phi_\rho\left(\Phi^{-1}\left(\tfrac{x_1-\mu_1}{\sigma_1}\right), \Phi^{-1}\left(\tfrac{x_2-\mu_2}{\sigma_2}\right)\right)$$

where Φ is the standard normal cdf, and Φ_ρ is the bivariate normal cdf for the pair of standard normal r.v.s with correlation $\rho \in [-1, 1]$. Gaussian copula easily generalizes to $d > 2$ by replacing ρ with a correlation matrix R. Gaussian copula family (bivariate or multivariate) is perhaps the most commonly employed as it shares many properties with Gaussian distributions.

2.2 Copulas with a Markov Tree Model

In this paper, we are focusing on the *tree-structured* PGMs. Suppose a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an undirected tree with the set of nodes $\mathcal{V} = \{1, \ldots, d\}$, and assumes that $\mathcal{G}$ contains no cycles. Assuming $X_1, \ldots, X_d$ satisfy the Markov assumptions encoded by $\mathcal{G}$, the joint pdf f can be written as

$$f(\boldsymbol{x}) = \left[\prod_{u=1}^{d} f_u(x_u)\right]\left[\prod_{\{u,v\}\in\mathcal{E}} \frac{f_{uv}(x_u, x_v)}{f_u(x_u) f_v(x_v)}\right] \tag{2}$$

where $f_{uv}(x_u, x_v)$ denotes the bivariate marginal density for (X_u, X_v). The same product decomposition holds for the case of discrete random variables $\boldsymbol{X}$ with probability mass functions (pmfs) replacing pdfs.

Kirshner (2007) proposed using tree-structured copulas and their variants for multidimensional density estimation (Figure 1, middle). By combining (2) and (1), the copula density for a tree-structured distribution can be expressed as a product of bivariate copulas on the edges:

$$c_T(\boldsymbol{a}) = \prod_{\{u,v\}\in\mathcal{E}} c_{uv}(a_u, a_v). \tag{3}$$

The converse also holds; if a pdf $c_T(\boldsymbol{a})$ is constructed as a product of bivariate copulas as in (3), then it is a valid copula density. This property permits building high-dimensional tree-structured copulas by separately specifying the Markov tree-structure and a bivariate copula (or its densities) for each edge.

3 Latent Tree Copulas

Copula decomposition suggests that the integral part of modeling the multivariate densities

is modeling their copulas. While computationally convenient, it is unreasonable to expect the multivariate dependence to be tree-structured. Mixtures or ensembles of trees can model more complex dependencies (Kirshner, 2007, Meilă and Jordan, 2000), but they may lack interpretability. Our approach extends the latent tree model of Zhang (2004) to copulas and thus to real-valued data by introducing latent variables (LVs) for copula models while preserving the appealing properties of trees.

Definition 1. Let $\boldsymbol{A} = (A_1, \ldots, A_d)^T$ be a vector of $\mathcal{U}(0,1)$ random variables. $\boldsymbol{A}$ is *tree-decomposable* if there exists a tree (forest) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, \ldots, t\}$, $t \geq d$, and bivariate copula densities $c_{uv}(a_u, a_v)$, $\{u, v\} \in \mathcal{E}$ such that

$$c_{LT}(\boldsymbol{a}) = \idotsint_{\mathbb{I}^{t-d}} \prod_{\{u,v\} \in \mathcal{E}} c_{uv}(a_u, a_v) \, \mathrm{d}a_{d+1} \ldots \mathrm{d}a_t. \tag{4}$$

We call the copula C_{LT} in (4) a *latent tree copula* (LTC).

LTCs generalize the tree-structured copulas by introducing LVs $A_{d+1}, \ldots, A_t$. Unlike the categorical variable setting where the support for each variable is finite, and therefore, bivariate marginals can be naturally represented with a multinomial distribution, modeling of $c_{uv}(a_u, a_v)$ requires additional assumptions. We assume the density for each c_{uv} is parametric with a pair $(\mathcal{M}_{uv}, \theta_{uv})$ denoting the functional form and a vector of parameters, respectively. To define a LTC model for $(A_1, \ldots, A_d)$, one needs to specify a 4-tuple $(t, \mathcal{E}, \mathcal{N}, \boldsymbol{\theta})$: t, the total number of variables, $\mathcal{E}$, the set of $t-1$ index pairs, $\mathcal{M} = (\mathcal{M}_{uv})_{\{u,v\} \in \mathcal{E}}$, the set of $t-1$ copula functional forms, and $\boldsymbol{\theta} = (\theta_{uv})_{\{u,v\} \in \mathcal{E}}$, the set of $t-1$ vectors of parameters for the bivariate copulas.

Using LVs within the trees permits approximating densities, possibly with complex dependencies, using only bivariate copulas. As is the case with tree copulas, this opens up a significant body of existing work on bivariate copulas for construction of multivariate densities, in contrast to copula Bayesian networks (Elidan, 2010a) which require higher-dimensional copulas as building blocks. On the other hand, in contrast to mixtures or ensembles of trees, the latent tree copulas provide a clear interpretation of the dependence between the variables in the model.

LTCs have a somewhat different representational power than their categorical-valued "siblings", LTMs (Zhang, 2004). LTMs are distributions over a finite set of variable states and are not identifiable as many different LTMs can represent the same distribution over the observed variables (OVs) $(X_1, \ldots, X_d)$, (property known as a marginal equivalence of the models). In order to evaluate LTMs, it is therefore necessary to consider the *parsimony* of the model, with preference given to models with fewer free parameters. However, among parsimonious families of LTMs, there are only a finite number of tree-structures possible to represent all possible distributions over a fixed number of OVs (Zhang, 2004). LTCs, depending on the functional families for copulas, $\mathcal{M}$, could have a very large number of parameters and cannot be represented by a simpler model. For practical purposes, we have to limit the number of latent variables in the LTC model, and we have to consider bivariate copula families which use few parameters.

4 Learning

Since LTC is specified as a 4-tuple, $(t, \mathcal{E}, \mathcal{M}, \boldsymbol{\theta})$, potentially all 4 elements of the 4-tuple need to be estimated. In the categorical case, the number of possible minimal models agreeing with marginals over the OVs is superexponential in d, but is finite (Zhang, 2004). For the continuous case, however, there is a potentially infinite number of models which can approximate the data (as for most copula families for the edges, adding additional edges simply increases the flexibility of the model).

To make learning tractable, we therefore have to restrict the class of desired solutions. We make two assumptions: (1) we assume that each bivariate copula comes from the same functional family, $\mathcal{M}_{uv} = \mathcal{M}'$ for all $\{u, v\} \in \mathcal{E}$ with $\mathcal{M}'$ selected apriori, and (2) we restrict the struc-

tures $\mathcal{G}$ to be binary latent trees as in Harmeling and Williams (2011). Binary latent tree with $d > 1$ observed (manifest) variables has $d - 1$ LVs ($t = 2d - 1$); each observed variable corresponds to a leaf of the tree, and each latent variable has exactly 3 neighbor except for one latent variable, the root of the tree, see Figure 1, right ($d = 4$).

Theorem 1. *Any Gaussian LTC can be represented as a Gaussian binary LTC.*[4]

Thus it is justifiable to consider only binary LTCs for the Gaussian case even though Gaussian binary LTCs will have redundant latent nodes, and potentially could have edges joining perfectly dependent variables (i.e., $\theta_{uv} \in \{-1, 1\}$). For the non-Gaussian case, binary latent trees may not necessarily represent all possible tree-decomposable distributions within the family. Still, the families of binary latent trees are flexible, and the resulting trees may provide an intuitive hierarchical interpretation for the components of $\boldsymbol{A}$.

4.1 Structure and Parameter Estimation

For estimation of binary latent trees and their parameters, we propose a greedy algorithm `Greedy-BLTC`, Algorithm 1, which repeatedly merges latent binary trees over subsets of variables into larger subtrees by introducing new latent variables as a root. The algorithm is an adaptation of (`Bin-G`) from Harmeling and Williams (2011) with some modifications to LTCs. The algorithm starts out with each variable $A_1, \ldots, A_d$ as the root of its own tree, with the joint distribution with the *working set* denoting the set of current roots. The variables in different trees are independent; the joint probability distribution over $\boldsymbol{A}$ is the product of probability distributions for each tree. At each step, the two trees with the highest estimated mutual information (based on the posterior probabilities for the latent variables) between their roots variables are merged into a larger tree by creating a new latent variable as the root of the new tree and joining the new variable to the roots of the two subtrees. The parameters of the newly created tree are re-estimated using the EM algorithm (Section 4.2). The process is repeated until all of the nodes belong to the same tree.

[4]Proof is omitted due to space limitations and will appear in the full version.

Algorithm 1 `Greedy-BLTC`

1: **input:** a working set $V = \{1, \ldots, d\}$ of indices for variables $A_1, \ldots, A_d$
2: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = V$, $\mathcal{E} = \emptyset$
3: estimate pairwise MI for each pair of variables
4: **for** $r = d + 1, \ldots, 2d - 1$ **do**
5: /* loop over the index of the new LV (new root) */
6: $\{u, v\} \leftarrow$ pair from V with highest MI for (A_u, A_v)
7: remove u and v from V
8: add new root r to V /* variable A_r is latent */
9: add r to $\mathcal{V}$ and edges $\{r, u\}$ and $\{r, v\}$ to $\mathcal{E}$
10: $\boldsymbol{\theta} \leftarrow$ `EstimateParameters`(subtree with root r) using the EM algorithm (see Section 4.2)
11: estimate pairwise MIs between r and the rest of working set V
12: **end for**
13: **output:** the graph $\mathcal{G}$, with $\mathcal{V} = \{1, \ldots, 2d - 1\}$, parameters $\Theta = \{\theta_{uv}\}_{\{u,v\} \in \mathcal{E}}$

4.2 Parameter Estimation Given Structure

To estimate the parameters of a subtree from line 10 of Algorithm 1, we will employ a variant of the EM algorithm (Dempster et al., 1977). Suppose there are d observed variables labeled $\boldsymbol{A}_O = (A_1, \ldots, A_d)$ and $t - d$ latent variables labeled $\boldsymbol{A}_H = (A_{d+1}, \ldots, A_t)$, $\mathcal{M}'$ is given, and suppose $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a tree. (The tree does not have to be binary.) Suppose we are given a set of i.i.d. observation vectors $\mathcal{D} = \{\boldsymbol{a}_O^1, \ldots, \boldsymbol{a}_O^N\}$, $\boldsymbol{a}_O^n = (a_1^n, \ldots, a_d^n)$ with none of a_u^n ($u = 1, \ldots, d$) missing. Our goal is to find the maximum likelihood estimate (MLE) of the parameters $\boldsymbol{\theta} = \{\theta_{uv}\}_{\{u,v\} \in \mathcal{E}}$:

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\theta} \sum_{n=1}^{N} \ln c_{LT}\left(\boldsymbol{a}_O^n | \theta\right). \tag{5}$$

Each iteration of the standard EM optimizes

$$\sum_{n=1}^{N} E_{c_{LT}\left(A_u^n, A_v^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)} \ln c_{uv}\left(a_u^n, a_v^n | \theta_{uv}\right) \tag{6}$$

with respect to θ_{uv} for each $\{u, v\} \in \mathcal{E}$, where $\boldsymbol{\theta}'$ is the set of parameter values at the start of

the iteration. For the $\mathcal{M}' = \texttt{Gaussian}$ family of copulas, the update can be computed in closed form by finding a root $\theta_{uv} \in [-1, 1]$ of the equation

$$\alpha_3 \theta_{uv}^3 + \alpha_2 \theta_{uv}^2 + \alpha_1 \theta_{uv} + \alpha_0 = 0$$

where

$$\begin{aligned}
&\alpha_3 = N, \\
&\alpha_2 = \alpha_0 = -\sum_{n=1}^{N} E_{\mathcal{N}_T(Z_u^n, Z_v^n | \boldsymbol{z}^n, \boldsymbol{\theta}')} z_u^n z_v^n, \\
&\alpha_1 = \sum_{n=1}^{N} E_{\mathcal{N}_T(Z_u^n, Z_v^n | \boldsymbol{z}^n, \boldsymbol{\theta}')} \left[(z_u^n)^2 + (z_v^n)^2 - 1 \right].
\end{aligned}$$

$\mathcal{N}_T\left(\boldsymbol{Z}^n | \boldsymbol{\theta}'\right)$ is a tree-structured t-variate Gaussian distribution obtained from $C_{LTN}\left(\boldsymbol{A}^n | \boldsymbol{\theta}'\right)$ via independent transformation of the marginals $Z_u^n = \Phi^{-1}(A_u^n)$. The expectations above can be efficiently computed using the Belief Propagation algorithm of Pearl (1988) applied to a multivariate normal $\mathcal{N}_T$, $\mathcal{O}(Nt)$ computational complexity.

4.2.1 Non-Gaussian Case

For cases other than $\mathcal{M}' = \texttt{Gaussian}$, the posterior copula density $c_{LT}\left(Z_u^n, Z_v^n | \boldsymbol{z}^n, \boldsymbol{\theta}'\right)$ may not be computable in closed form, and for these families a direct application of EM is therefore impossible. Instead, we propose using a variational approach. For any density $q^n(\boldsymbol{A}_H^n)$ over $\mathbb{I}^{t-d}$,

$$\begin{aligned}
&\ln c_{LT}\left(\boldsymbol{a}_O^n | \boldsymbol{\theta}'\right) \\
&= \int_{\mathbb{I}^{t-d}} q^n(\boldsymbol{a}_H^n) \ln \frac{c_{LT}\left(\boldsymbol{a}_O^n, \boldsymbol{a}_H^n | \boldsymbol{\theta}'\right)}{q^n\left(\boldsymbol{a}_H^n\right)} \mathrm{d}\boldsymbol{a}_H^n \\
&\quad + D\left(q^n(\boldsymbol{a}_H^n) \parallel c_{LT}\left(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right).
\end{aligned} \tag{7}$$

For variational EM (e.g., Wainwright and Jordan, 2008), the family $\mathcal{Q}$ of distributions $q^n \in \mathcal{Q}$ is chosen in a way so that $D\left(q^n \parallel c_{LT}\left(\cdot | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right)$ can be easily minimized. Notice that unlike the standard setting for variational inference on graphical models, the approximation is not needed to simplify the dependence structure of latent variables conditioned on the observations; the dependence structure imposed on $c_{LT}(\boldsymbol{a}_H^n | \boldsymbol{a}_O^n)$ by $\mathcal{E}$ is already a tree or a forest. Instead for LTCs, the approximating family $\mathcal{Q}$ needs to be chosen so that inference with it does not require closed form marginalization of copula functions or their products. The modified EM algorithm iterates between choosing $q^n(\boldsymbol{a}_H^n)$ minimizing $D\left(q^n \parallel c_{LT}\left(\cdot | \boldsymbol{a}_O^n, \boldsymbol{\theta}'\right)\right)$, $n = 1, \ldots, N$, (E-step) and updating parameters $\boldsymbol{\theta} = \{\theta_{ij}\}_{\{i,j\} \in \mathcal{E}}$ (M-step):

$$\hat{\theta}_{uv} = \operatorname*{argmax}_{\theta_{uv}} \sum_{n=1}^{N} E_{q_{uv}^n(\boldsymbol{A}_H^n)} \ln c_{uv}(a_u^n, a_v^n | \theta_{uv}) \tag{8}$$

Let $\mathcal{E}_H = \{\{u, v\} \in \mathcal{E} : u, v \in H\}$ be the subset of edges joining only the latent variables, and let $\mathcal{E}_{H^+} = \{\{u, v\} \in \mathcal{E} : u \in H, v \in O\}$ be the subset of edges joining one latent and one observed variable. We consider a family of tree structured distributions $\mathcal{Q}$ with Markov graph $\mathcal{G}_H = (H, \mathcal{E}_H)$ so that $\forall q \in \mathcal{Q}$

$$q(\boldsymbol{a}_H) = \prod_{u \in H} q_u(a_u) \left[\prod_{\{u,v\} \in \mathcal{E}_H} \frac{q_{uv}(a_u, a_v)}{q_u(a_u) q_v(a_v)} \right]$$

where q_{uv}, q_u, and q_v are marginals of q for the variables (A_u, A_v), A_u, and A_v, respectively. We represent a pdf $q_{uv}(a_u, a_v)$ on $\mathbb{I}^2$ as a 2-d step function partitioning $\mathbb{I}^2$ into K^2 $\frac{1}{K} \times \frac{1}{K}$ squares (for some preselected integer K), with each square having uniform density $p_{uv}(i, j)$:

$$\begin{aligned}
&q_{uv}(a_u, a_v) = p_{uv}(i, j) \geq 0 \text{ for } a_u \in \mathbb{I}_i,\ a_v \in \mathbb{I}_j, \\
&q_u(a_u) = p_u(i) \geq 0 \text{ for } a_u \in \mathbb{I}_i, \\
&\text{where } \mathbb{I}_i = \left(\frac{i-1}{K}, \frac{i}{K} \right], \text{ and} \\
&\sum_{i=1}^{K} p_u(i) = K,\ u \in H,\ i = 1, \ldots, K, \\
&\sum_{i=1}^{K} p_{uv}(i, j) = K p_v(j), \text{ and} \\
&\sum_{j=1}^{K} p_{uv}(i, j) = K p_u(i), \ \forall \{u, v\} \in \mathcal{E}_H
\end{aligned} \tag{9}$$

with (9) ensuring q has proper marginals, and that the bivariate densities agree on the marginals.

Minimizing $D\left(q^n \parallel c_{LT}\left(\cdot | \boldsymbol{a}_O^n\right)\right)$ is equivalent to maximizing $\int_{\mathbb{I}^{t-d}} q^n(\boldsymbol{a}_H^n) \ln \frac{c_{LT}\left(\boldsymbol{a}_O^n, \boldsymbol{a}_H^n | \boldsymbol{\theta}'\right)}{q^n\left(\boldsymbol{a}_H^n\right)} \mathrm{d}\boldsymbol{a}_H^n$

(Eqn 7), and is equivalent to minimizing

$$
\begin{aligned}
f\left(q^n\right) &= \frac{1}{K}\sum_{u\in H}\sum_{i=1}^{K} p_u^n\left(i\right)\ln p_u^n\left(i\right) \\
&+ \frac{1}{K^2}\sum_{\{u,v\}\in\mathcal{E}_H}\sum_{i=1}^{K}\sum_{j=1}^{K} p_{uv}^n\left(i,j\right)\ln\frac{p_{uv}^n\left(i,j\right)}{p_u^n\left(i\right)p_v^n\left(j\right)} \\
&- \sum_{\{u,v\}\in\mathcal{E}_H^+}\sum_{i=1}^{K} w_u^n\left(i\right)p_u^n\left(i\right) \\
&- \sum_{\{u,v\}\in\mathcal{E}_H}\sum_{i=1}^{K}\sum_{j=1}^{K} w_{uv}^n\left(i,j\right)p_{uv}^n\left(i,j\right), \text{ where}
\end{aligned}
$$

$$
\begin{aligned}
w_u^n\left(i\right) &= \int_{\mathbb{I}_i}\ln c_{uv}\left(a_u^n,a_v^n\right)\mathrm{d}a_u^n, \\
w_{uv}^n\left(i,j\right) &= \int_{\mathbb{I}_i}\int_{\mathbb{I}_j}\ln c_{uv}\left(a_u^n,a_v^n\right)\mathrm{d}a_u^n\mathrm{d}a_v^n.
\end{aligned}
\tag{10}
$$

Whether mean field approximation is used (i.e., assuming $p_{uv}^n\left(i,j\right)=p_u^n\left(i\right)p_v^n\left(j\right)$) or structured mean field, it is straightforward to derive a set of fixed point equations to minimize $f\left(q^n\right)$ subject to the constraints in (9). The integrals of log-copula densities (10) do not have analytic expressions for most bivariate copula families. We employ quadrature methods for estimation of these integrals as they are low-dimensional and have bounded range of integration. While the log-likelihood cannot be evaluated directly, it can be lower-bounded (from Eqn 7), by $-\sum_{n=1}^{N} f\left(q^n\right)$. In our experimentations, choosing $K\geq 50$ led to good fits of q^n to $c_{uv}\left(\boldsymbol{a}_H^n|\boldsymbol{a}_O^n,\theta'_{uv}\right)$ and thus to a good approximation of the log-likelihood in the equation above.

The computational complexity for the proposed approach depends on the number of iterations until mean field equations converge. Per update, the complexity is $\mathcal{O}\left(dK^2\right)$ per data point. This does not include the complexity of computing the integrals in (10), Nd univariate and $d-1$ bivariate such integrals for latent binary trees.

5 Experimental Illustration

For an illustration, we model the S&P 100 monthly stock returns data set described in Choi et al. (2011) with a Gaussian LTC.[5] The data set consists of 216 monthly stock returns of 84 companies in the S&P 100 stock index (and the index itself) for the years 1990–2007. The goal is to approximate the high-dimensional distribution between the returns and to discover useful hierarchies among the variables in question. First, we transformed the data into the copula domain: the marginal densities f_u, $u=1,\ldots,85$ were estimated by Gaussian KDEs with bandwidths determined using the Rule of Thumb (Silverman, 1986), and the data was mapped into $\mathbb{I}^d$ ($d=85$) by applying F_u to each component u of the data vector.[6] We then fit a Gaussian LTC (LTC-G) trained using `Greedy-BLTC` (Algorithm 1) to the transformed data using 10 random restarts for parameter estimation (EM) within each subtree. The results are listed in Table 1; the likelihoods and the number of hidden variables are computed for the original (not transformed) data; `xval8` refers to out-of-sample per-example log-likelihood obtained by 8-fold cross-validation. CL, NJ, and CLRJ procedures are described in Choi et al. (2011); the goal of all of these approaches is to learn a latent tree Gaussian model. Copula- CL, NJ, and CLRJ differ in

Table 1: Comparison of the log-likelihood, BIC, number of created latent variables, number of free parameters, and running time for the S&P 100 monthly stock returns data.

	ll	#latent	xval8
LTC-G	25381	84	108.38
CL	23970	0	105.99
NJ	24408	45	108.45
CLRJ	24361	26	108.50
Copula-CL	24787	0	107.96
Copula-NJ	25350	41	110.08
Copula-CLRJ	25284	30	109.68

[5] The data set is available as a part of the software toolbox for Choi et al. (2011). `http://people.csail.mit.edu/myungjin/latentTree.html`

[6] Non-parametric estimation of marginals distributions is a common approach in copula modeling (e.g., Joe and Xu, 1996).

that their marginals are first mapped into a copula domain, and are then modified by the inverse normal CDF transform. However, the marginals in this case are close to normal, and the improvement of the copula- versions of the algorithms is not significant. LTC-G appears to provide a similar fit as suggested by the log-likelihood.[7]

The graph displaying the hierarchy of the variables is omitted due to limited space. However, similar to the reports in Choi et al. (2011), our approach generates interpretable substructures. For example, there is a subtree populated entirely by the natural gas and oil production and exploration companies (Schlumberger, Baker Hughes, Halliburton, Occidental Petroleum, Exxon, Chevron, ConocoPhillips), and another by the telecommunication companies (Spring, Verizon, and AT&T).

6 Conclusion

We proposed a new model for multivariate continuous data based on a latent tree hierarchy for the copula of the its joint distribution. This model can be used both to model high-dimensional densities without the jointly Gaussian assumption or to group variables into subgroups. We described an algorithm for estimation of the model's binary tree structure together with its parameters from data. In the future, we plan to improve the estimation procedure for the non-Gaussian copula case. We are planning to use the above model for problems in hydrology, in particular, as an approach to regionalization of watersheds.

Acknowledgments

This work has been supported by the US National Science Foundation award AGS-1025430.

References

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

T. Bedford and R. M. Cooke. Vines – a new graphical model for dependent random variables. *Ann. Stat.*, 30 (4):1031–1068, 2002.

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

M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky. Learning latent tree graphical models. *JMLR*, 12:1771–1812, May 2011.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977.

G. Elidan. Copula bayesian networks. In *NIPS*, pages 559–567, 2010a.

G. Elidan. Inference-less density estimation using copula bayesian networks. In *UAI*, pages 151–159, 2010b.

C. Genest and A.-C. Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrological Engineering*, 12(4):347–368, July 2007.

S. Harmeling and C. K. I. Williams. Greedy learning of binary latent trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6):1087–1097, June 2011.

J. C. Huang and B. J. Frey. Cumulative distribution networks and the derivative-sum-product algorithm. In *UAI*, pages 290–297, 2008.

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

H. Joe and J. J. Xu. The estimation method of inference functions for margins for multivariate models. Technical report, Department of Statistics, University of British Columbia, 1996.

S. Kirshner. Learning with tree-averaged densities and distributions. In *NIPS*, pages 761–768, 2007.

D. Kurowicka and R. M. Cooke. *Uncertainty Analysis with High Dimensional Dependence Modelling*. Wiley, 2006.

M. Meilă and M. I. Jordan. Learning with mixtures of trees. *JMLR*, 1(1):1–48, October 2000.

R. B. Nelsen. *An Introduction to Copulas*. Springer, 2nd edition, 2006.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

R. Silva, C. Blundell, and Y. W. Teh. Mixed cumulative distribution networks. *JMLR - Proceedings Track (AISTATS-2010)*, 15:670–678, 2011.

B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

A. Sklar. Fonctions de repartition a n dimensions et leures marges. *Publications de l'Institut de Statistique de L'Universite de Paris*, 8:229–231, 1959.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

N. L. Zhang. Hierarchical latent class models for cluster analysis. *JMLR*, 5:697–723, June 2004.

N. L. Zhang and T. Kočka. Efficient learning of hierarchical latent class models. In *ICTAI*, pages 585–593, 2004.

[7] RG and CLNG from Choi et al. (2011) and their copula versions performed worse than NJ and CLRJ.