Learning with Tree-Averaged Densities and Distributions

Sergey Kirshner
AICML and Dept of Computing Science
University of Alberta
Edmonton, Alberta, Canada T6G 2E8
sergey@cs.ualberta.ca

Abstract

We utilize the ensemble of trees framework, a tractable mixture over super-exponential number of tree-structured distributions [1], to develop a new model for multivariate density estimation. The model is based on a construction of tree-structured copulas – multivariate distributions with uniform on $[0, 1]$ marginals. By averaging over all possible tree structures, the new model can approximate distributions with complex variable dependencies. We propose an EM algorithm to estimate the parameters for these tree-averaged models for both the real-valued and the categorical case. Based on the tree-averaged framework, we propose a new model for joint precipitation amounts data on networks of rain stations.

1 Introduction

Multivariate real-valued data appears in many real-world data sets, and a lot of research is being focused on the development of multivariate real-valued distributions. One of the challenges in constructing such distributions is that univariate continuous distributions commonly do not have a clear multivariate generalization. The most studied exception is the multivariate Gaussian distribution owing to properties such as closed form density expression with a convenient generalization to higher dimensions and closure over the set of linear projections. However, not all problems can be addressed fairly with Gaussians (e.g., mixtures, multimodal distributions, heavy-tailed distributions), and new approaches are needed for such problems.

While modeling multivariate distributions is in general difficult due to complicated functional forms and the curse of dimensionality, learning models for individual variables (univariate marginals) is often straightforward. Once the univariate marginals are known (or assumed known), the rest can be modeled using *copulas*, multivariate distributions with all univariate marginals equal to uniform distributions on $[0, 1]$ (e.g., [2]). A large portion of copula research concentrated on bivariate copulas as extensions to higher dimensions are often difficult. Thus if one could represent the desired distribution in some decomposition into its univariate marginals (possibly estimated or approximated) and only bivariate distributions, then the machinery of copulas can be fully utilized.

Distributions with undirected tree-structured graphical models (e.g., [3]) have exactly these properties, as probability density functions over the variables with tree-structured conditional independence graphs can be written as a product of the univariate marginals and bivariate marginals corresponding to its edges. While tree-structured dependence is perhaps too restrictive, one can obtain a much richer variable dependence by averaging over a small number of different tree structures [4] or *all* possible tree structures; the latter can be done analytically for categorical-valued distributions with an ensemble-of-trees model [1]. In this paper, we extend this tree-averaged model to continuous variables with the help of copulas and derive a learning algorithm to estimate the parameters within the maximum likelihood framework with EM [5]. Within this framework, the parameter estimation

for tree-structured and tree-averaged models requires optimization over only univariate and bivariate densities potentially avoiding the curse of dimensionality, a property not shared by alternative models that relax the dependence restriction of trees (e.g., vines [6]).

The main contributions of the paper are the new tree-averaged model for multivariate copulas, a parameter estimation algorithm for tree-averaged framework (for both categorical and real-valued complete data), and a new model for multi-site daily precipitation amounts, an important application in hydrology. In the process, we introduce previously unexplored tree-structured copula density and an algorithm for estimation of its structure and parameters. The paper is organized as follows. First, we describe copulas, their densities, and some of their useful properties (Section 2). We then show how to construct multivariate copulas with tree-structured dependence from bivariate copulas (Section 3.1) and show how to estimate the parameters of the bivariate copulas and perform the edge selection. To allow more complex dependencies between the variables, we describe a *tree-averaged copula*, a novel copula object constructed by averaging over all possible spanning trees for tree-structured copulas, and derive a learning algorithm for the estimation of the parameters from data for the tree-averaged copulas (Section 4). We then explore how the tree-averaged framework can be applied to model multi-site precipitation amounts, a problem involving both binary (rain/no rain) and continuous (how much rain) variables (Section 5).

2 Copulas

Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a vector random variable with corresponding probability distribution F (cdf) defined on $\mathbb{R}^d$. We denote by $\mathcal{V}$ the set of d components (variables) of $\boldsymbol{X}$ and refer to individual variables as X_v for $v \in \mathcal{V}$. For simplicity, we will refer to assignments to random variables by lower case letters, e.g., $X_v = x_v$ will be denoted by x_v. Let $F_v(x_v) = F(X_v = x_v, X_u = \infty : u \in \mathcal{V} \setminus \{v\})$ denote a univariate marginal of F over the variable X_v. Let $p_v(x_v)$ denote the probability density function (pdf) of X_v. Let $a_v = F_v(x_v)$, and let $\boldsymbol{a} = (a_1, \ldots, a_d)$, so $\boldsymbol{a}$ is a vector of quantiles of components of $\boldsymbol{x}$ with respect to corresponding univariate marginals. Next, we define copula, a multivariate distribution over vectors of quantiles.

Definition 1. The **copula** associated with F is a distribution function $C : [0,1]^d \rightarrow [0,1]$ that satisfies

$$F(\boldsymbol{x}) = C(F_1(x_1), \ldots, F_d(x_d)), \ \boldsymbol{x} \in \mathbb{R}^d. \tag{1}$$

If F is a continuous distribution on $\mathbb{R}^d$ with univariate marginals $F_1, \ldots, F_d$, then $C(\boldsymbol{a}) = F\left(F_1^{-1}(a_1), \ldots, F_d^{-1}(a_d)\right)$ is the unique choice for (1).

Assuming that F has d-th order partial derivatives, the probability density function (pdf) can be obtained from the distribution function via differentiation:

$$p(\boldsymbol{x}) = \frac{\partial^d F(\boldsymbol{x})}{\partial x_1 \ldots \partial x_d}.$$

Pdfs can also be rewritten in terms of derivatives of copulas:

$$p(\boldsymbol{x}) = \frac{\partial^d F(\boldsymbol{x})}{\partial x_1 \ldots \partial x_d} = \frac{\partial^d C(\boldsymbol{a})}{\partial x_1 \ldots \partial x_d} = \frac{\partial^d C(\boldsymbol{a})}{\partial a_1 \ldots \partial a_d} \prod_{v \in \mathcal{V}} \frac{\partial a_v}{\partial x_v} = c(\boldsymbol{a}) \prod_{v \in \mathcal{V}} p_v(x_v) \tag{2}$$

where

$$c(\boldsymbol{a}) = \frac{\partial^d C(\boldsymbol{a})}{\partial a_1 \ldots \partial a_d}$$

is referred to as *copula density function*.

3 Exploiting Tree-Structured Dependence

Joint probability distributions are often modeled with probabilistic graphical models where the structure of the graph captures the conditional independence relations of the variables. The joint distribution is then represented as a product of functions over subsets of variables. We would like to keep the number of variables for each of the functions small as the number of parameters and the number of points needed for parameter estimation often grows exponentially with the number of variables.

Thus, we examine building of copulas with tree dependence. Trees play an important role in probabilistic graphical models as they allow for efficient exact inference [7] as well as structure and parameter learning [3]. They can also be placed in a fully Bayesian framework with decomposable priors allowing to compute expected values (over all possible spanning trees) of product of functions defined on the edges of the trees [1]. As we will see later in this section, under the tree-structured dependence, a copula density can be computed as products of bivariate copula densities over the edges of the graph. This property allows us to estimate the parameters for the edge copulas independently.

3.1 Tree-Structured Copulas

We consider tree-structured Markov networks, i.e., undirected graphs that do not have loops. For a distribution F admitting tree-structured Markov networks (referred from now on as tree-structured distributions), assuming that $p(\boldsymbol{x}) > 0$ and $p(\boldsymbol{x}) < \infty$ for $\boldsymbol{x} \in \mathcal{R} \subseteq \mathbb{R}^d$, the density (for $\boldsymbol{x} \in \mathcal{R}$) can be rewritten as

$$p(\boldsymbol{x}) = \left[\prod_{v \in \mathcal{V}} p_v(x_v)\right] \prod_{\{u,v\} \in \mathcal{E}} \frac{p_{uv}(x_u, x_v)}{p_u(x_u)\, p_v(x_v)}. \tag{3}$$

This formulation easily follows from the Hammersley-Clifford theorem [8, 9]. Note that for $\{u, v\} \in \mathcal{E}$ a copula density $c_{uv}(a_u, a_v)$ for $F(x_u, x_v)$ can be computed using Equation 2:

$$c_{uv}(a_u, a_v) = \frac{p_{uv}(x_u, x_v)}{p_u(x_u)\, p_v(x_v)}. \tag{4}$$

Using Equations 2, 3, and 4, $c_p(\boldsymbol{a})$ for $F(\boldsymbol{x})$ can be computed as

$$c_p(\boldsymbol{a}) = \frac{p(\boldsymbol{x})}{\prod_{v \in \mathcal{V}} p_v(x_v)} = \prod_{\{u,v\} \in \mathcal{E}} \frac{p_{uv}(x_u, x_v)}{p_u(x_u)\, p_v(x_v)} = \prod_{\{u,v\} \in \mathcal{E}} c_p(a_u, a_v). \tag{5}$$

Equation 5 states that a copula density for a tree-structured distribution decomposes as a product of bivariate copulas over its edges. A logical question is whether the converse is true.

Theorem 1. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ without loops and copula densities $c_{uv}(a_u, a_v)$ for $\{u, v\} \in \mathcal{E}$,

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

is a valid copula density.

A consequence of Theorem 1 is a construction and a computation of a tree-structured d-copula density given bivariate copula densities for the edges. If fewer than $d - 1$ edges are specified (i.e., the edges make up a forest but not a spanning tree), *product* or independence copula $C_\Pi(a_u, a_v) = a_u a_v$ can be used to add edges and connect disconnected components as under C_Π corresponding variables are independent. Assuming univariate marginals and copula densities are in a parametric form, one can use a maximum likelihood or a maximum a posteriori principle to estimate the parameters from data. Suppose we are given a complete data set $\mathcal{D} = \left\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\right\}$ of d-component real-valued vectors $\boldsymbol{x}^n = \left(x_1^n, \ldots, x_1^d\right)$ under i.i.d. assumption. In a tree-structured framework, the log-likelihood of $\mathcal{D}$ can be written as:

$$\ln p(\mathcal{D}) = \sum_{v \in \mathcal{V}} \sum_{n=1}^{N} \ln p_v(x_v^n) + \sum_{\{u,v\} \in \mathcal{E}} \sum_{n=1}^{N} \ln c_{uv}\left(F_u(x_u^n), F_v(x_v^n)\right). \tag{6}$$

The terms of the log-likelihood (6) are not independent as the second term in the sum is defined in terms of the probability expressions in the first summand. To decouple the parameter estimation for the univariate and multivariate terms, we assume that the univariate marginals can be accurately estimated (by either fitting parameters for the parametric form of some appropriate univariate distribution or by applying non-parametric methods) from the first term in (6) alone.[1] This is equivalent to modeling univariate marginals using just the corresponding components from all vectors in the data,

[1] In copula literature, these approaches are referred to as *inference for the margins* (IFM) and *canonical maximum likelihood* (CML), respectively.

the usual modeling choice when one is interested only in univariate marginals. Let $\hat{p}_v(x_v)$ be the estimated pdf for component v and $\hat{F}_v$ be the corresponding cdf. One can then compute $a_v^n = \hat{F}_v(x_v^n)$ for all $v \in \mathcal{V}$ and $n = 1, \ldots, N$ thus creating a set of estimated quantiles $\mathcal{A} = \left\{\boldsymbol{a}^1, \ldots, \boldsymbol{a}^N\right\}$ with $\boldsymbol{a}^n = (a_1^n, \ldots, a_d^n) = \left(\hat{F}(x_1^n), \ldots, \hat{F}(x_d^n)\right)$. For a tree-structured density, the log-likelihood can be rewritten as

$$\ln p(\mathcal{D}) = \sum_{v \in \mathcal{V}} \sum_{n=1}^{N} \ln \hat{p}_v(x_v^n) + l(\mathcal{A}) \quad \text{where} \quad l(\mathcal{A}) = \sum_{\{u,v\} \in \mathcal{E}} \sum_{n=1}^{N} \ln c_{uv}(a_u^n, a_v^n). \tag{7}$$

A maximum of the log-likelihood can be found by maximizing $\sum_{n=1}^{N} \ln c_{uv}(a_u^n, a_v^n)$ independently for different pairs $\{u, v\} \in \mathcal{E}$. The rest of the algorithm is straightforward. (See supplement Section B.2). We note that one does not have to specify the tree-structure, but rather learn it from the data as well, as in the Chow-Liu algorithm [3].

4 Tree-Averaged Copulas

While the framework from Section 3.1 is computationally efficient and convenient for implementation, the imposed tree-structured dependence is too restrictive for real-world problems. Vines [6], for example, deal with this problem by allowing recursive refinements for the bivariate probabilities over variables not connected by the tree edges. However, vines require estimation of additional characteristics of the distribution (e.g., conditional rank correlations) requiring estimation over large sets of variables, which is not advisable when the amount of available data is not large. Our proposed method would only require optimization of parameters of bivariate copulas from the corresponding two components of weighted data vectors. Using the Bayesian framework for spanning trees from [1], it is possible to construct an object constituting a convex combination over *all* possible spanning trees allowing a much richer set of conditional independencies than a single tree.

Meilă and Jaakkola [1] proposed a decomposable prior over all possible spanning tree structures. Let $\boldsymbol{\beta}$ be a symmetric matrix of non-negative weights for all pairs of distinct variables and zeros on the diagonal ($\beta_{uv} = \beta_{vu} \geq 0 \; \forall u, v \in \mathcal{V}, \; u \neq v$ and $\beta_{vv} = 0 \; \forall v \in \mathcal{V}$). Let $\mathfrak{E}$ be a set of all possible spanning trees over $\mathcal{V}$. The probability distribution over all spanning tree structures over $\mathcal{V}$ is defined as

$$P(\mathcal{E} \in \mathfrak{E} | \boldsymbol{\beta}) = \frac{1}{Z} \prod_{\{u,v\} \in \mathcal{E}} \beta_{uv} \text{ where } Z = \sum_{\mathcal{E} \in \mathfrak{E}} \prod_{\{u,v\} \in \mathcal{E}} \beta_{uv}. \tag{8}$$

Z can be computed in closed form even though there are $|\mathfrak{E}| = d^{d-2}$ trees to sum over.

Theorem 2. (Meilă and Jaakkola [1]) Let $P(\mathcal{E})$ be a distribution over spanning tree structures defined by (8). Then the normalization constant Z is equal to the determinant $|\mathbf{L}^\star(\boldsymbol{\beta})|$, with matrix $\mathbf{L}^\star(\boldsymbol{\beta})$ representing the first $(d-1)$ rows and columns of the matrix $\mathbf{L}(\boldsymbol{\beta})$ given by:

$$L_{uv}(\boldsymbol{\beta}) = L_{vu}(\boldsymbol{\beta}) = \begin{cases} -\beta_{uv} & u, v \in \mathcal{V}, \; u \neq v; \\ \sum_{w \in \mathcal{V}} \beta_{vw} & u, v \in \mathcal{V}, \; u = v. \end{cases}$$

Note that $\boldsymbol{\beta}$ is a generalization of an adjacency matrix, $\mathbf{L}(\boldsymbol{\beta})$ is a generalization of the Laplacian matrix, and Theorem 2 is a generalization of the Kirchoff's Matrix Tree Theorem. It is also worth noting that the decomposable prior (Equation 8) over spanning tree structures can be derived using maximum entropy method [10] by setting constraints to conserve the proportion of trees containing a particular edge in the data and the distribution.

The decomposability property of the tree prior (Equation 8) allows us to compute the average of the tree-structured distributions over all d^{d-2} tree structures. An *ensemble-of-trees*, a term coined in [1], describes the object resulting from such an average; in [1], the average was applied to tree-structured distributions over categorical variables. Similarly, we define a *tree-averaged copula* density as a convex combination of copula densities of the form (5):

$$r(\boldsymbol{a}) = \sum_{\mathcal{E} \in \mathfrak{E}} P(\mathcal{E} | \boldsymbol{\beta}) \, c(\boldsymbol{a}) = \frac{1}{Z} \sum_{\mathcal{E} \in \mathfrak{E}} \left[\prod_{\{u,v\} \in \mathcal{E}} \beta_{uv} \right] \left[\prod_{\{u,v\} \in \mathcal{E}} c_{uv}(a_u, a_v) \right] = \frac{|\mathbf{L}^\star(\boldsymbol{\beta}\mathbf{c}(a))|}{|\mathbf{L}^\star(\boldsymbol{\beta})|}$$

where entry (uv) of matrix $\boldsymbol{\beta}\mathbf{c}(\boldsymbol{a})$ denotes $\beta_{uv} c_{uv}(a_u, a_v)$. A finite convex combination of copulas is a copula, so $r(\boldsymbol{a})$ is a copula density.

4.1 Parameter Estimation

We suggest finding suitable parameter values $\boldsymbol{\beta}$ (edge weight matrix) and $\boldsymbol{\theta}$ (parameters for edge copulas) by maximizing the log-likelihood of the set of probability values $\mathcal{A}$ (obtained by replacing each data value with its estimated quantile value, see Section 3.1):

$$l(\boldsymbol{\beta}, \boldsymbol{\theta}) = \ln p(\mathcal{A}|\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \ln r(\boldsymbol{a}^n|\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{n=1}^{N} \ln |L^\star(\boldsymbol{\beta}\boldsymbol{c}(\boldsymbol{a}^n|\boldsymbol{\theta}))| - N \ln |L^\star(\boldsymbol{\beta})|. \quad (9)$$

The parameter optimization of $l(\boldsymbol{\beta}, \boldsymbol{\theta})$ cannot be done analytically. Instead, we notice that we are dealing with a mixture model (granted, one where the number of mixture components is super-exponential), and thus we propose performing our parameter optimization with the EM algorithm [5].[2] While there are d^{d-2} possible mixture components (spanning trees), in the E-step, we only need to compute the posterior probabilities for $d(d-1)/2$ edges. Each step of EM consists of finding parameters $\boldsymbol{\beta}', \boldsymbol{\theta}'$ maximizing the expected joint log-likelihood $M(\boldsymbol{\beta}', \boldsymbol{\theta}'; \boldsymbol{\beta}, \boldsymbol{\theta})$ given current parameter values $\boldsymbol{\beta}, \boldsymbol{\theta}$ where

$$\begin{aligned}
M(\boldsymbol{\beta}', \boldsymbol{\theta}'; \boldsymbol{\beta}, \boldsymbol{\theta}) &= \sum_{n=1}^{N} \sum_{\mathcal{E}_n \in \mathfrak{E}} P(\mathcal{E}_n|\boldsymbol{a}^n, \boldsymbol{\beta}, \boldsymbol{\theta}) \ln \left[P(\mathcal{E}|\boldsymbol{\beta}') c(\boldsymbol{a}^n|\mathcal{E}, \boldsymbol{\theta}') \right] \quad (10)\\
&= \sum_{\{u,v\}} \sum_{n=1}^{N} s_n(\{u,v\}) (\ln \beta'_{uv} + \ln c_{uv}(a_u^n, a_v^n|\theta'_{uv})) - N \ln \left| \mathbf{L}^\star(\boldsymbol{\beta}') \right|;\\
s_n(\{u,v\}) &= \sum_{\substack{\mathcal{E} \in \mathfrak{E} \\ \{u,v\} \in \mathcal{E}}} P(\mathcal{E}_n|\boldsymbol{a}^n, \boldsymbol{\beta}, \boldsymbol{\theta}) \text{ where}\\
P(\mathcal{E}_n|\boldsymbol{a}^n, \boldsymbol{\beta}, \boldsymbol{\theta}) &= \frac{P(\mathcal{E}_n|\boldsymbol{\beta}) c(\boldsymbol{a}^n|\mathcal{E}_n, \boldsymbol{\theta})}{r(\boldsymbol{a}^n|\boldsymbol{\beta}, \boldsymbol{\theta})} = \frac{\prod_{\{u,v\} \in \mathcal{E}} (\beta_{uv} c_{uv}(a_u^n, a_v^n|\boldsymbol{\theta}_{uv}))}{|L^\star(\boldsymbol{\beta}\boldsymbol{c}(\boldsymbol{a}^n))|}. \quad (11)
\end{aligned}$$

The probability distribution in Equation 11 is of the same form as the tree prior, so to compute $s_n(\{u,v\})$ one needs to compute the sum of probabilities of all trees containing edge $\{u,v\}$.

Theorem 3. Let $P(\mathcal{E}|\boldsymbol{\beta})$ be a tree prior defined in Equation 8. Let $\mathbf{Q}(\boldsymbol{\beta}) = (\mathbf{L}^\star(\boldsymbol{\beta}))^{-1}$ where $\mathbf{L}^\star$ is obtained by removing row and column w from $\mathbf{L}$. Then

$$\sum_{\mathcal{E} \in \mathfrak{E}:\ \{u,v\} \in \mathcal{E}} P(\mathcal{E}|\boldsymbol{\beta}) = \begin{cases} \beta_{uv}(Q_{uu}(\boldsymbol{\beta}) + Q_{uv}(\boldsymbol{\beta}) - 2Q_{uv}(\boldsymbol{\beta})) & : \quad u \neq v, u \neq w, v \neq w, \\ \beta_{uw} Q_{uw}(\boldsymbol{\beta}) & : \quad v = w, \\ \beta_{wv} Q_{wv}(\boldsymbol{\beta}) & : \quad u = w, \\ 0 & : \quad u = v. \end{cases}$$

As a consequence of Theorem 3, for each $n = 1, \ldots, N$, $d(d-1)/2$ edge probabilities $s_n(\{u,v\})$ can be computed *simultaneously* with time complexity of a single $(d-1) \times (d-1)$ matrix inversion, $\mathcal{O}(d^3)$. Assuming a candidate bivariate copula c_{uv} has one free parameter[3] θ_{uv}, θ_{uv} can be optimized by setting

$$\frac{\partial M(\boldsymbol{\beta}', \boldsymbol{\theta}'; \boldsymbol{\beta}, \boldsymbol{\theta})}{\partial \theta'_{uv}} = \sum_{n=1}^{N} s_n(\{u,v\}) \frac{\partial \ln c_{uv}(a_u^n, a_v^n; \theta'_{uv})}{\partial \theta'_{uv}}, \quad (12)$$

to 0. (See supplement Section C for copula optimization.) The parameters of the tree prior can be updated by maximizing

$$\sum_{\{u,v\}} \left(\frac{1}{N} \sum_{n=1}^{N} s_n(\{u,v\}) \right) \ln \beta'_{uv} - \ln |\mathbf{L}^\star(\boldsymbol{\beta})|,$$

an expression concave in $\ln \beta_{uv}\ \forall \{u,v\}$. $\boldsymbol{\beta}'$ can be updated using a gradient ascent algorithm on $\ln \beta_{uv}\ \forall \{u,v\}$, with time complexity $\mathcal{O}(d^3)$ per iteration. The outline of the EM algorithm is

[2] A possibility of EM algorithm for ensemble-of-trees with categorical data was mentioned [1], but the idea was abandoned due to the concern about the M-step.

[3] Copulas with multiple parameters can be optimized similarly.

Algorithm TREEAVERAGEDCOPULADENSITY($\mathcal{D}, \boldsymbol{c}$)
Inputs: A complete data set $\mathcal{D}$ of d-component real-valued vectors; a set of of bivariate parametric copula densities $\boldsymbol{c} = \{c_{uv} : u, v \in \mathcal{V}\}$

1. Estimate univariate margins $\hat{F}_v(X_v)$ for all components $v \in \mathcal{V}$ treating all components independently.
2. Replace $\mathcal{D}$ with $\mathcal{A}$ consisting of vectors $\boldsymbol{a}^n = \left(\hat{F}_1(x_1^n), \ldots, \hat{F}_d(x_d^n)\right)$ for each vector $\boldsymbol{x}^n$ in $\mathcal{D}$
3. Initialize $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$
4. Run until convergence (as determined by change in log-likelihood, Equation 9)
   - E-step: For all vectors a^n and pairs $\{u, v\}$, compute $P(\{u, v\} \in \mathcal{E} | \boldsymbol{a}^n, \boldsymbol{\beta}, \boldsymbol{\theta})$
   - M-step:
     - Update $\boldsymbol{\beta}$ with gradient ascent
     - Update $\boldsymbol{\theta}_{uv}$ for all pairs by setting partial derivative with respect to parameters of $\boldsymbol{\theta}_{uv}$ (Equation 12) to zero and solving corresponding equations

Output: Denoting $a_u = \hat{F}(x_u)$ and $a_v = \hat{F}(x_v)$, $\hat{p}(\boldsymbol{x}) = \left[\prod_{v \in \mathcal{V}} \hat{p}_v(x_v)\right] \frac{|\mathbf{L}^\star(\boldsymbol{\beta}\boldsymbol{c}(\boldsymbol{a}))|}{|\mathbf{L}^\star(\boldsymbol{\beta})|}$

Figure 1: Algorithm for estimation of a pdf with tree-averaged copulas.

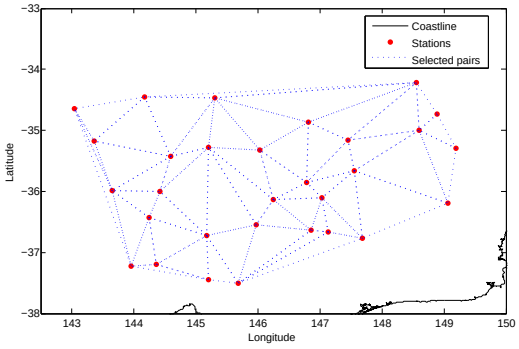

Figure 2: Station map with station locations (red dots), coastline, and the pairs of stations selected according to Delaunay triangulation (dotted lines)

shown in Figure 1. Assuming the complexity of each bivariate copula update is $\mathcal{O}(N)$, the time complexity of each EM iteration is $\mathcal{O}(Nd^3)$.

The EM algorithm can be easily transferred to ensemble of trees for categorical data. The E-step does not change, and in the M-step, the parameters for the univariate marginals are updated ignoring bivariate terms. Then, the parameters from bivariate distributions for each edge are updated constrained on the new values of the parameters for the univariate distributions. While the algorithm does not guarantee a maximization of the expected log-likelihood, it nonetheless worked well in our experiments.

5 Multi-Site Precipitation Modeling

We applied the tree-averaged framework to the problem of modeling daily rainfall amounts for a regional spatial network of stations. The task is to build a generative model capturing the spatial and temporal properties of the data. This model can be used in at least two ways: first, to sample sequences from it and to use them as inputs for other models, e.g., crop models; and second, as a descriptive model of the data. Hidden Markov models (possible with non-homogeneous transitions) are being frequently used for this task (e.g., [11]) with the transition distribution responsible

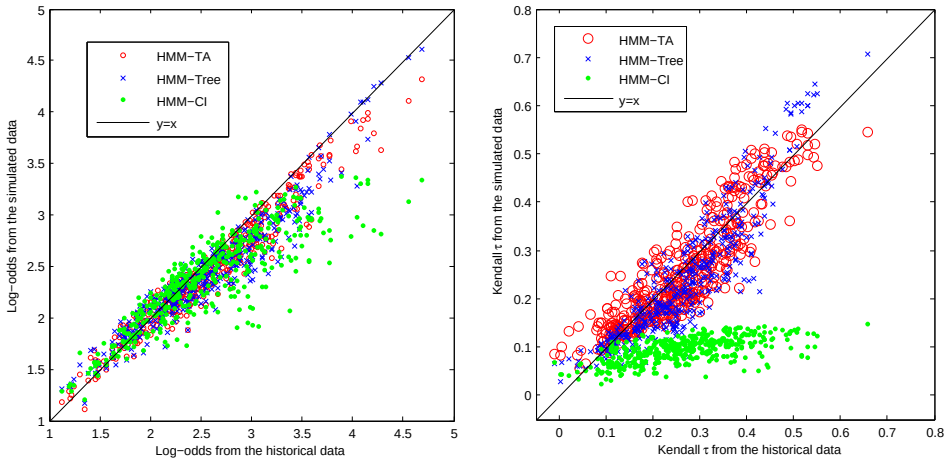

Figure 3: Scatter-plots of log-odds ratios for occurrence (left) and Kendall's τ measure of concordance (right) for all pairs of stations for the historical data vs HMM-TA (red o), HMM-Tree (blue x), and HMM-CI (green $\cdot$).

for modeling of temporal dependence, and the emission distributions capturing most of the spatial dependence. Additionally, HMMs can be viewed as assigning rainfall daily patterns to "weather states" (or corresponding emission components), and both these states (as described by either their parameters or the statistics of the patterns associated with it) and their temporal evolution often offer useful synoptic insight. We will use HMMs as the wrapper model with tree-averaged (and tree-structured) distributions to model the emission components.

The distribution of daily rainfall amounts for any given station can be viewed as a non-overlapping mixture with one component corresponding to zero precipitation, and the other component to positive precipitation. For a station v, let r_v be the precipitation amount, π_v be a probability of positive precipitation, and let $f_v\left(r_v|\boldsymbol{\lambda}_v\right)$ be a probability density function for amounts given positive precipitation:

$$p\left(r_v|\pi_v,\boldsymbol{\lambda}_v\right)=\left\{\begin{array}{lcl}1-\pi_v & : & r_v=0,\\ \pi_v f_v\left(r_v|\boldsymbol{\lambda}_v\right) & : & r_v>0.\end{array}\right.$$

For a pair of stations $\{u,v\}$, let π_{uv} denote the probability of simultaneous positive amounts and $c_{uv}\left(F_u\left(r_u|\boldsymbol{\lambda}_u\right),F_v\left(r_v|\boldsymbol{\lambda}_v\right)|\boldsymbol{\theta}_{uv}\right)$ denote the copula density for simultaneous positive amounts; then

$$p\left(r_u,r_v|\pi_u,\pi_v,\pi_{uv},\boldsymbol{\lambda}_u,\boldsymbol{\lambda}_v\right)=\left\{\begin{array}{lcl}1-\pi_u-\pi_v+\pi_{uv} & : & r_u=0,\ r_v=0,\\ \left(\pi_v-\pi_{uv}\right)f_v\left(r_v|\boldsymbol{\lambda}_v\right) & : & r_u=0,\ r_v>0,\\ \left(\pi_u-\pi_{uv}\right)f_u\left(r_u|\boldsymbol{\lambda}_u\right) & : & r_u>0,\ r_v=0,\\ \pi_{uv}f_u\left(r_u\right)f_v\left(r_v\right)c\left(F_u\left(r_u\right),F_v\left(r_v\right)\right) & : & r_u>0,\ r_v>0.\end{array}\right.$$

We can now define a tree-structured and tree-averaged probability distributions, $p_t\left(\boldsymbol{r}\right)$ and $p_{ta}\left(\boldsymbol{r}\right)$, respectively, over the amounts:

$$\begin{aligned}\omega_{uv}\left(\boldsymbol{r}\right) &= \frac{p\left(r_u,r_v|\pi_u,\pi_v,\pi_{uv},\boldsymbol{\lambda}_u,\boldsymbol{\lambda}_v\right)}{p\left(r_u|\pi_u,\boldsymbol{\lambda}_u\right)p\left(r_v|\pi_v,\boldsymbol{\lambda}_v\right)},\\ p_t\left(\boldsymbol{r}|\boldsymbol{\pi},\boldsymbol{\lambda},\boldsymbol{\theta},\mathcal{E}\right) &= \left[\prod_{v\in\mathcal{V}}p\left(r_v|\pi_v\right)\right]\prod_{\{u,v\}\in\mathcal{E}}\omega_{uv}\left(\boldsymbol{r}\right), \qquad (13)\\ p_{ta}\left(\boldsymbol{r}|\boldsymbol{\pi},\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{\beta}\right) &= \sum_{\mathcal{E}\in\mathfrak{E}}P\left(\mathcal{E}|\boldsymbol{\beta}\right)p_t\left(\boldsymbol{r}|\boldsymbol{\pi},\boldsymbol{\lambda},\boldsymbol{\theta},\mathcal{E}\right)=\left[\prod_{v\in\mathcal{V}}p\left(r_v|\pi_v\right)\right]\frac{\left|\mathbf{L}^{\star}\left(\boldsymbol{\beta}\boldsymbol{\omega}\left(\boldsymbol{r}\right)\right)\right|}{\left|\mathbf{L}^{\star}\left(\boldsymbol{\beta}\right)\right|}. \qquad (14)\end{aligned}$$

We employ univariate exponential distributions $f_v\left(r_v\right)=\lambda_v e^{-\lambda_v r_v}$ and bivariate Gaussian copulas $c_{uv}\left(a_u,a_v\right)=\frac{1}{\sqrt{1-\theta_{uv}^2}}e^{-\frac{\theta_{uv}^2\Phi^{-1}(a_u)^2+\theta_{uv}^2\Phi^{-1}(a_v)^2-2\theta_{uv}\Phi^{-1}(a_u)\Phi^{-1}(a_v)}{2\left(1-\theta_{uv}^2\right)}}$.

We applied the models to a data set collected from 30 stations from a region in Southeastern Australia (see Figure 2) from 1986 to 2005, April-October, (20 sequences 214 30-dimensional vectors each). We used a 5-state HMM with three different types of emission distributions: tree-averaged (Equation 14), tree-structured (Equation 13), and conditionally independent (first term of the RHS of the Equation 13). We will refer to these models HMM-TA, HMM-Tree, and HMM-CI. For HMM-TA, we reduced the number of free parameters by only allowing edges for stations adjacent to each other as determined by the the Delaunay triangulation (see Figure 2). We also did not learn the edge weights (β) setting them to 1 for selected edges and to 0 for the rest. To make sure that the models do not overfit, we computed their out-of-sample log-likelihood with cross-validation, leaving out one year at a time. (5 states were chosen because the leave-one-year out log-likelihood starts to flatten out for HMM-TA at 5 states.) The resulting log-likelihoods divided by the number of days and stations are -0.9392, -0.9522, and -1.0222 for HMM-TA, HMM-Tree, and HMM-CI, respectively. The higher number for HMM-TA suggests that HMM-TA is not overfitting. To see how well the models capture the properties of the data, we trained each model on the whole data set (with 50 restarts of EM), and then simulated 500 sequences of length 214. We are particularly interested in how well they measure pairwise dependence; we concentrate on two measures: log-odds ratio for occurrence and Kendall's τ measure of concordance for pairs when both stations had positive amounts. Both are shown in Figure 3. Both plots suggest that HMM-CI underestimates the pairwise dependence for strongly dependent pairs (as indicated by its trend to predict lower absolute values for log-odds and concordance); HMM-Tree estimating the dependence correctly mostly for strongly dependent pairs (as indicated by good prediction for high values), but underestimating it for moderate dependence; and HMM-TA performing the best for most pairs except for the ones with very strong dependence.

6 Summary

We introduced a new tree-averaged model for multivariate copulas and derived a reasonably efficient EM algorithm to estimate its parameters from complete data. This tree-averaged framework was used to develop a new model for multi-site precipitation amounts data. For the future, we would like to extend this work to handle missing data as most data sets are not complete. Also, an extension for conditional distributions can further enhance its applicability for rainfall modeling.

References

[1] M. Meilă and T. Jaakkola. Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 16(1):77–92, 2006.

[2] H. Joe. *Multivariate Models and Dependence Concepts*, volume 73 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, 1997.

[3] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3):462–467, May 1968.

[4] M. Meilă and M. I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1(1):1–48, October 2000.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via EM algorithm. *Journal of the Royal Statistical Society Series B-Methodological*, 39(1):1–38, 1977.

[6] T. Bedford and R. M. Cooke. Vines – a new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Francisco, California, 1988.

[8] J. M. Hammersley and P. E. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript, 1971.

[9] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society Series B-Methodological*, 36(2):192–236, 1974.

[10] E.T. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9):939–952, September 1982.

[11] E. Bellone. *Nonhomogeneous Hidden Markov Models for Downscaling Synoptic Atmospheric Patterns to Precipitation Amounts*. PhD thesis, Department of Statistics, University of Washington, Seattle, Washington, 2000.

[12] W. F. Darsow, B. Nguyen, and E. T. Olsen. Copulas and Markov processes. *Illinois Journal of Mathematics*, 36(4):600–642, Winter 1992.

[13] R. B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer, 2nd edition, 2006.

[14] T. P. Hutchinsion and C. D. Lai. *Continuous Bivariate Distributions, Emphasising Applications*. Rumsby Scientific Publishing, Adelaide, South Australia, 1990.

[15] D. Morgenstern. Einfache Beispiele zweidimensionaler Verteilungen. *Mitteilungsblatt fur Mathematische Statistik*, 8:234–235, 1956.

[16] D G Clayton. A model for association in bivariate life tables and its application in epidemiological studies of familial tendency in chronic disease incidence. *Biometrika*, 65:141–151, 1978.

[17] M M Ali, N N Mikhail, and M S Haq. A class of bivariate distributions including the bivariate logistic. *Journal of Multivariate Analysis*, 8:405–412, 1978.

[18] E J Gumbel. Bivariate exponential distributions. *Journal of American Statistical Association*, 55:698–707, 1960.

[19] V Barnett. Some bivariate uniform distributions. *Communications in Statistics Series A – Theory Methods*, 9:453–461, 1980.

[20] M. J. Frank. On the simultaneous associativity of $F(x,y)$ and $x+y-F(x,y)$. *Aequationes Math*, 19:194–226, 1979.

A Proofs

Theorem 1. Given a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ without loops and copula densities $c_{uv}(a_u,a_v)$ for $\{u,v\}\in\mathcal{E}$,

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

is a valid copula density.

To prove this theorem, we first introduce a construction of multivariate copulas by combining copulas defined over smaller number of variables. Let $C_1(a_1,\ldots,a_m)$ be an m-variate copula and let $C_2(a_m,\ldots,a_{m+n-1})$ be an n-variate copula. Let $C_1\star C_2:[0,1]^{m+n-1}\to[0,1]$ be defined as

$$C_1\star C_2(a_1,\ldots,a_{m+n-1})=\int_0^{a_m}\frac{\partial C_1}{\partial a_m}(a_1,\ldots,a_{m-1},\xi)\frac{\partial C_2}{\partial a_m}(\xi,a_{m+1},\ldots,a_{m+n-1})\,\mathrm{d}\xi.$$

$C_1\star C_2$ is an $m+n-1$-variate copula. The $\star$ operator for copulas is a way to construct copulas with Markov properties, with variables $A_1,\ldots,A_{m-1}$ conditionally independent from $A_{m+1},\ldots,A_{m+n-1}$ given A_m under $C_1\star C_2$ [12].

Example 1. Let $C_{12}(a_1,a_2)$ and $C_{23}(a_2,a_3)$ be bivariate Gaussian copulas with correlations ρ and ς, respectively. The copula $C_{123}=C_{12}\star C_{23}$ is a 3-variate Gaussian copula with the functional form

$$C_{123}(a_1,a_2,a_3)=\Phi_\Sigma\left(\Phi^{-1}(a_1),\Phi^{-1}(a_2),\Phi^{-1}(a_3)\right)$$

where a correlation matrix Σ is computed as

$$\Sigma=\begin{pmatrix}1&\rho&\rho\varsigma\\ \rho&1&\varsigma\\ \rho\varsigma&\varsigma&1\end{pmatrix}.$$

Proof. We will show that $C_{\mathcal{E}}$ is the $\star$-product over all edges of $\mathcal{E}$, $C_{\mathcal{E}}(\boldsymbol{a})=\star_{\{u,v\}\in\mathcal{E}}C_{uv}(a_u,a_v)$. (Copula constructor $\star$ is associative [12].) We will do so by induction on the number of edges $|\mathcal{E}|$. If $|\mathcal{E}|=1$, assume $\mathcal{E}=\{\{u,v\}\}$; then $c_{\mathcal{E}}=c_{uv}$ is a copula density. Now we assume that the statement of the theorem holds for trees $\mathcal{E}$ with number of edges $|\mathcal{E}|\le n$. We need to show that it would hold for $|\mathcal{E}|=n+1$. Pick a leaf $v\in\mathcal{V}$ and the edge $\{u,v\}\in\mathcal{E}$ connecting v to another node in $\mathcal{V}$.

$$\begin{aligned}C_{\mathcal{E}}(\boldsymbol{a}) &= C_{\mathcal{E}\setminus\{u,v\}}\star C_{uv}(\boldsymbol{a})=\int_0^{a_u}\frac{\partial C_{\mathcal{E}\setminus\{u,v\}}}{\partial a_u}\left(\boldsymbol{a}_{\mathcal{V}\setminus\{u,v\}},\xi_u\right)\frac{\partial C_{uv}}{\partial a_u}(\xi_u,a_v)\,\mathrm{d}\xi_u\\ &= \underbrace{\cdots\int_0^{a_w}\cdots}_{w\in\mathcal{V}\setminus\{u,v\}}\int_0^{a_u}\prod_{\{w,z\}\in\mathcal{E}\setminus\{u,v\}}c_{wz}(\xi_w,\xi_z)\underbrace{\ldots\mathrm{d}\xi_w\ldots}_{w\in\mathcal{V}\setminus\{u,v\}}\int_0^{a_v}c_{uv}(\xi_u,\xi_v)\,\mathrm{d}\xi_u\mathrm{d}\xi_v\\ &= \underbrace{\cdots\int_0^{\xi_v}\cdots}_{v\in\mathcal{V}}\prod_{\{u,v\}\in\mathcal{E}}c_{uv}(\xi_u,\xi_v)\underbrace{\ldots\mathrm{d}\xi_v\ldots}_{v\in\mathcal{V}},\end{aligned}$$

so $\prod_{\{u,v\}\in\mathcal{E}}c_{uv}$ is a copula density for $C_{\mathcal{E}}$. □

Theorem 3. Let $P(\mathcal{E}|\boldsymbol{\beta})$ be a tree prior defined in Equation 8. Let $\mathbf{Q}(\boldsymbol{\beta}) = (\mathbf{L}^\star(\boldsymbol{\beta}))^{-1}$ where $\mathbf{L}^\star$ is obtained by removing row and column w from $\mathbf{L}$. Then

$$\sum_{\mathcal{E}\in\mathfrak{E}:\ \{u,v\}\in\mathcal{E}} P(\mathcal{E}|\boldsymbol{\beta}) = \begin{cases} \beta_{uv}\left(Q_{uu}(\boldsymbol{\beta}) + Q_{uv}(\boldsymbol{\beta}) - 2Q_{uv}(\boldsymbol{\beta})\right) & : \quad u\neq v, u\neq w, v\neq w, \\ \beta_{uw}Q_{uw}(\boldsymbol{\beta}) & : \quad v = w, \\ \beta_{wv}Q_{wv}(\boldsymbol{\beta}) & : \quad u = w, \\ 0 & : \quad u = v. \end{cases}$$

Proof. From Lemma 2 of [1], $\frac{\partial|\mathbf{L}^\star(\boldsymbol{\beta})|}{\partial\beta_{uv}} = M_{uv}|\mathbf{L}^\star(\boldsymbol{\beta})|$ where

$$\mathbf{M} = \begin{cases} Q_{uu} + Q_{uv} - 2Q_{uv} & : \quad u\neq v, u\neq w, v\neq w, \\ Q_{uw} & : \quad v = w, \\ Q_{wv} & : \quad u = w, \\ 0 & : \quad u = v. \end{cases}$$

On the other hand, $|L^\star(\boldsymbol{\beta})|$ can be decomposed into contributions from trees containing edge $\{u,v\}$ and from trees not containing it. Let $\boldsymbol{\beta} : \beta_{uv} = a$ denote a matrix $\boldsymbol{\beta}$ where entries (uv) and (vu) were replaced with a with the rest of the matrix kept unchanged. For a $d\times d$ matrix A with rows an columns indexed by nodes of $\mathcal{V}$, let A^{-w}_{-u} and A^{-wz}_{-uv} denote a $(d-1)\times(d-1)$ and $(d-2)\times(d-2)$ submatrices of A obtained by removal of row u and column w, and rows u and v and columns w and z, respectively. Let the weight of a tree $\mathcal{E}$ (denoted by $w(\mathcal{E}|\boldsymbol{\beta})$) be the product of the contributions from its edges:

$$w(\mathcal{E}|\boldsymbol{\beta}) = \prod_{\{u,v\}\in\mathcal{E}} \beta_{uv}.$$

Then the total weight of all trees is equal to the weight of all trees that include edge $\{u,v\}$ and the weight of all trees that do not:

$$|L^\star(\boldsymbol{\beta})| = \sum_{\mathcal{E}\in\mathfrak{E}} w(\mathcal{E}|\boldsymbol{\beta}) = \sum_{\substack{\{u,v\}\in\mathcal{E}\\ \mathcal{E}\in\mathfrak{E}}} w(\mathcal{E}|\boldsymbol{\beta}) + \sum_{\substack{\{u,v\}\notin\mathcal{E}\\ \mathcal{E}\in\mathfrak{E}}} w(\mathcal{E}|\boldsymbol{\beta}) = \sum_{\substack{\{u,v\}\in\mathcal{E}\\ \mathcal{E}\in\mathfrak{E}}} w(\mathcal{E}|\boldsymbol{\beta}) + |L^\star(\boldsymbol{\beta}:\beta_{uv}=0)|$$

as setting $\beta_{uv} = 0$ removes all probability mass from the trees containing edge $\{u,v\}$ without affecting weights of other trees. Note that setting entry $\beta_{uv} = 0$ affects only four entries of the matrix $L(\boldsymbol{\beta})$: $uu,\ uv,\ vu,\ vv$. The weight contributions from the trees containing $\{u,v\}$ can be computed as

$$\begin{aligned}
\sum_{\substack{\{u,v\}\in\mathcal{E}\\ \mathcal{E}\in\mathfrak{E}}} w(\mathcal{E}|\boldsymbol{\beta}) &= |L^\star(\boldsymbol{\beta})| - |L^\star(\boldsymbol{\beta}:\beta_{uv}=0)| = \sum_{w\in\mathcal{V}\setminus\{u\}} (-1)^{i(w)+i(v)} L_{vw}(\boldsymbol{\beta})\left|L(\boldsymbol{\beta})^{-uw}_{-uv}\right| \\
&\quad - \sum_{w\in\mathcal{V}\setminus\{u\}} (-1)^{i(w)+i(v)} L_{vw}(\boldsymbol{\beta}:\beta_{uv}=0)\left|L(\boldsymbol{\beta}:\beta_{uv}=0)^{-uw}_{-uv}\right| \\
&= L_{vv}(\boldsymbol{\beta})\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right| - L_{vv}(\boldsymbol{\beta}:\beta_{uv}=0)\left|L(\boldsymbol{\beta}:\beta_{uv}=0)^{-uv}_{-uv}\right| \\
&\quad + \sum_{w\in\mathcal{V}\setminus\{u,v\}} (-1)^{i(w)+i(v)}\left(L_{vw}(\boldsymbol{\beta})\left|L(\boldsymbol{\beta})^{-uw}_{-uv}\right| - L_{vw}(\boldsymbol{\beta}:\beta_{uv}=0)\left|L(\boldsymbol{\beta}:\beta_{uv}=0)^{-uw}_{-uv}\right|\right) \\
&= (L_{vv}(\boldsymbol{\beta}) - L_{vv}(\boldsymbol{\beta}:\beta_{uv}=0))\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right| \\
&\quad + \sum_{w\in\mathcal{V}\setminus\{u,v\}} (-1)^{i(w)+i(v)}\left(L_{vw}(\boldsymbol{\beta})\left|L(\boldsymbol{\beta})^{-uw}_{-uv}\right| - L_{vw}(\boldsymbol{\beta})\left|L(\boldsymbol{\beta})^{-uw}_{-uv}\right|\right) \\
&= \left(\sum_{w\in\mathcal{V}}\beta_{wv} - \sum_{w\in\mathcal{V}\setminus\{u\}}\beta_{wv}\right)\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right| = \beta_{uv}\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right|.
\end{aligned}$$

It follows that

$$\sum_{\mathcal{E}\in\mathfrak{E}:\ \{u,v\}\in\mathcal{E}} P(\mathcal{E}|\boldsymbol{\beta}) = \frac{1}{|\mathbf{L}^\star(\boldsymbol{\beta})|}\sum_{\substack{\{u,v\}\in\mathcal{E}\\ \mathcal{E}\in\mathfrak{E}}} w(\mathcal{E}|\boldsymbol{\beta}) = \frac{\beta_{uv}\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right|}{|\mathbf{L}^\star(\boldsymbol{\beta})|}.$$

Also since neither $\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right|$ nor $|L^\star(\boldsymbol{\beta}:\beta_{uv}=0)|$ depend on β_{uv},

$$\frac{\partial Z}{\partial\beta_{uv}} = \left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right|.$$

Thus

$$\sum_{\mathcal{E}\in\mathfrak{E}:\ \{u,v\}\in\mathcal{E}} P(\mathcal{E}|\boldsymbol{\beta}) = \frac{\beta_{uv}\left|L(\boldsymbol{\beta})^{-uv}_{-uv}\right|}{|\mathbf{L}^\star(\boldsymbol{\beta})|} = \frac{\beta_{uv}}{|\mathbf{L}^\star(\boldsymbol{\beta})|}\frac{\partial Z}{\partial\beta_{uv}} = \beta_{uv}M_{uv}.$$

□

B Additional Information

B.1 Copula Examples

Example 2. Let F be a bivariate normal distribution with mean $\boldsymbol{\mu} = (\mu_1, \mu_2)$ and covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{11}^2 & \rho\sigma_{11}\sigma_{22} \\ \rho\sigma_{11}\sigma_{22} & \sigma_{22}^2 \end{pmatrix}$ with correlation ρ. By $\Phi_\rho(x_1, x_2)$ we denote a bivariate normal cdf with 0 mean, unit variances, and correlation ρ; by $\Phi(x)$ we denote a univariate normal cdf with 0 mean and unit variance. (Neither cdf can be computed analytically.) F can be expressed as a linear transformation of Φ_ρ:

$$F(x_1, x_2) = \Phi_\rho\left(\frac{x_1 - \mu_1}{\sigma_{11}}, \frac{x_2 - \mu_2}{\sigma_{22}}\right).$$

Both marginals are univariate normal distributions with $F_1(x_1) = \Phi\left(\frac{x_1-\mu_1}{\sigma_{11}}\right)$ and $F_2(x_2) = \Phi\left(\frac{x_2-\mu_2}{\sigma_{22}}\right)$. Then the copula C of F is equal to

$$C(a_1, a_2) = \Phi_\rho\left(\Phi^{-1}(a_1), \Phi^{-1}(a_2)\right).$$

Note that C does not depend on either means or variances. The copula density c of the bivariate normal also does not have an analytic expression:

$$c(a_1, a_2) = \frac{1}{\sqrt{1-\rho^2}} e^{-\frac{\rho^2\Phi^{-1}(a_1)^2 + \rho^2\Phi^{-1}(a_2)^2 - 2\rho\Phi^{-1}(a_1)\Phi^{-1}(a_2)}{2\left(1-\rho^2\right)}}.$$

Example 3. [4] Let F be a Gumbel's bivariate exponential distribution

$$F(x_1, x_2) = 1 - e^{-x_1} - e^{-x_2} + e^{-(x_1 + x_2 + \theta x_1 x_2)}$$

for $x_1, x_2 \geq 0$. The marginal distributions are

$$a_1 = F_1(x_1) = 1 - e^{-x_1}, \quad a_2 = F_2(x_2) = 1 - e^{-x_2}.$$

Random variables X_1 and X_2 can be represented as inverses of their cdf marginals:

$$x_1 = F_1^{-1}(a_1) = -\ln(1 - a_1), \; x_2 = F_2^{-1}(a_2) = -\ln(1 - a_2).$$

The corresponding copula is then

$$\begin{aligned} C(\boldsymbol{a}) &= a_1 + a_2 - 1 + (1 - a_1)(1 - a_2) e^{-\theta \ln(1-a_1)\ln(1-a_2)}, \\ c(\boldsymbol{a}) &= [(1 - \theta\ln(1 - a_1))(1 - \theta\ln(1 - a_2)) - \theta] \exp(-\theta\ln(1 - a_1)\ln(1 - a_2)). \end{aligned}$$

It is easy to verify that

$$p(\boldsymbol{x}) = [(1 + \theta x_1)(1 + \theta x_2) - \theta] e^{-x_1 - x_2 - \theta x_1 x_2} = c(P_1(x_1), P_2(x_2)) p_1(x_1) p_2(x_2).$$

B.2 Algorithm for Parametric Tree-Structured Copula Learning from Complete Data

Assuming a given copula density has one free parameter[5], $c_{uv}(a_u, a_v; \boldsymbol{\theta}_{uv})$ with one scalar parameter θ_{uv}, the partial derivative of the log-likelihood with respect to θ_{uv} is

$$\frac{\partial l(\mathcal{A})}{\partial \theta_{uv}} = \sum_{n=1}^{N} \frac{\partial \ln c_{uv}(a_u^n, a_v^n; \theta_{uv})}{\partial \theta_{uv}}, \tag{15}$$

and $\hat{\theta}_{uv}$ can be chosen by setting the above derivative to 0. Given a library of bivariate parametric copula densities $\mathcal{C}$ (see supplement Section C) for each edge $\{u, v\} \in \mathcal{E}$, one can select the values of parameters maximizing the log-likelihood $\sum_{n=1}^{N} \ln c_l(a_u^n, a_v^n)$ for each copula density $c_l \in \mathcal{C}$ in the library, and then select the copula with the highest log-likelihood.

$$\left(\hat{c}_{uv}, \hat{\boldsymbol{\theta}}_{uv}\right) = \arg\max_{c \in \mathcal{C}, \boldsymbol{\theta}^c} \sum_{n=1}^{N} \ln c(a_u^n, a_v^n; \boldsymbol{\theta}^c). \tag{16}$$

[4] Borrowed in part from Example 2.9 in [13]

[5] Copulas with multiple parameters can be handled similarly.

Algorithm TREECOPULADENSITY$(\mathcal{D}, \mathcal{C})$
Inputs: A complete data set $\mathcal{D}$ of d-component real-valued vectors; a library of bivariate parametric copula densities $\mathcal{C}$; procedure MWST(weights) that outputs a maximum weight spanning tree

1. Estimate univariate margins $\hat{F}_v\left(X_v\right)$ for all components $v \in \mathcal{V}$ treating all components independently.
2. Replace $\mathcal{D}$ with $\mathcal{A}$ consisting of vectors $\boldsymbol{a}^n = \left(\hat{F}_1\left(x_1^n\right), \ldots, \hat{F}_d\left(x_d^n\right)\right)$ for each vector $\boldsymbol{x}^n$ in $\mathcal{D}$
3. Find pair $\left(\hat{c}_{uv}, \hat{\boldsymbol{\theta}}_{uv}\right)$ (Expression 16) and $I_{uv} = \sum_{n=1}^{N} \ln \hat{c}_{uv}\left(a_u^n, a_v^n; \hat{\boldsymbol{\theta}}_{uv}\right)$ for all pairs $\{u, v\} \subset \mathcal{V}$
4. $\mathcal{E} = \text{MWST}\left(\{I_{uv}\}\right)$

Output: Denoting $a_u = \hat{F}\left(x_u\right)$ and $a_v = \hat{F}\left(x_v\right)$, $\hat{p}\left(\boldsymbol{x}\right) = \left[\prod_{v \in \mathcal{V}} \hat{p}_v\left(x_v\right)\right]\left[\prod_{\{u,v\} \in \mathcal{E}} \hat{c}_{uv}\left(a_u, a_v; \hat{\boldsymbol{\theta}}_{uv}\right)\right]$

Figure 4: Algorithm for estimation of a pdf from data using tree-structured copulas.

We can go a step further and learn the spanning tree structure $\mathcal{E}$ by maximizing the bivariate terms $l\left(\mathcal{A}\right)$ (7) of the log-likelihood not only with respect to the copula densities from $\mathcal{C}$ and corresponding parameters, but also with respect to the spanning tree edges $\mathcal{E}$:

$$\max_{\mathcal{E}, \boldsymbol{c}, \boldsymbol{\theta}} l\left(\mathcal{A}\right) = \max_{\mathcal{E}} \max_{\substack{c_{uv}, \boldsymbol{\theta}_{uv}: \\ \{u,v\} \in \mathcal{E}}} \sum_{n=1}^{N} \ln c_{uv}\left(a_u^n, a_v^n; \boldsymbol{\theta}_{uv}\right) = \max_{\mathcal{E}} \sum_{n=1}^{N} \ln \hat{c}_{uv}\left(a_u^n, a_v^n; \hat{\boldsymbol{\theta}}_{uv}\right)$$

where $\left(\hat{c}_{uv}, \hat{\boldsymbol{\theta}}_{uv}\right)$ are the best copula and its parameters (Equation 16) for pairs of nodes $\{u, v\}$. The algorithm for the spanning tree edge selection is the same as Chow-Liu algorithm for categorical variables [3]: compute mutual information $I_{uv} = I\left(X_u, X_v\right)$ for each pair of variables according to their empirical distribution, and then solve a maximum spanning tree problem on a graph with nodes $\mathcal{V}$ with edge weights defined by I_{uv}. In our case, we do not compute the exact value of $I\left(X_u, X_v\right)$ (we have only a finite number of points to estimate an integral of a continuous function), but instead use the corresponding log-likelihood term $\sum_{n=1}^{N} \ln \hat{c}_{uv}\left(a_u^n, a_v^n; \hat{\boldsymbol{\theta}}_{uv}\right)$.

The algorithm for estimation of $p\left(\boldsymbol{x}\right)$ from $\mathcal{D}$ with tree-structure copulas is shown in Figure 4. The exact complexity of the algorithm depends on the parametric form of the univariate marginals and the bivariate copulas in the library; nonetheless, we attempt to comment on the complexity of each of the steps. The estimation of the univariate parametric marginals can often be done either in closed form or iteratively in the time $\mathcal{O}\left(Nd\right)$ (possibly, per iteration). The computation of $\mathcal{A}$ is proportional to the size of the data set ($\mathcal{O}\left(Nd\right)$). Assuming that finding a maximum likelihood estimate for parameters of a bivariate copula is proportional to N (possibly, per iteration), step 3 of the algorithm can be computed in $\mathcal{O}\left(Nd^2\left|\mathcal{C}\right|\right)$. Finally, finding a maximum weight spanning tree can be obtained in $\mathcal{O}\left(d^2\right)$.

B.3 Derivation of Tree Prior with Maximum Entropy Method

To see that the tree prior in Equation 8 can be derived using maximum entropy method, let $P_e\left(\mathcal{E}\right)$ be an empirical distribution over all spanning trees $\mathfrak{E}$. We wish to find a distribution $P\left(\mathcal{E}\right)$ that has a maximum possible entropy while satisfying constraints of the form

$$\sum_{\mathcal{E} \in \mathfrak{E}} P_e\left(\mathcal{E}\right) f_{uv}\left(\mathcal{E}\right) = \sum_{\substack{\{u,v\} \in \mathcal{E} \\ \mathcal{E} \in \mathfrak{E}}} P_e\left(\mathcal{E}\right) = \sum_{\substack{\{u,v\} \in \mathcal{E} \\ \mathcal{E} \in \mathfrak{E}}} P\left(\mathcal{E}\right) = \sum_{\mathcal{E} \in \mathfrak{E}} P\left(\mathcal{E}\right) f_{uv}\left(\mathcal{E}\right) \tag{17}$$

where

$$f_{uv}\left(\mathcal{E}\right) = \begin{cases} 1 & : \quad \{u, v\} \in \mathcal{E}, \\ 0 & : \quad \{u, v\} \notin \mathcal{E}. \end{cases}$$

Let $\mathcal{F}$ denote the set of edges corresponding to the constraints in (17). Denoting Lagrangian coefficients $\boldsymbol{\Lambda} = \{\lambda_{uv} : \{u, v\} \in \mathcal{E}_c\}$, with λ_{uv} corresponding to constraint for edge $\{u, v\}$, one needs to find $P\left(\mathcal{E}\right)$ to

maximize

$$H(P) = -\sum_{\mathcal{E}\in\mathfrak{E}} P(\mathcal{E}) \ln P(\mathcal{E}) + \sum_{\{u,v\}\in\mathcal{F}} \lambda_{uv} \left(\sum_{\mathcal{E}\in\mathfrak{E}} P_e(\mathcal{E}) f_{uv}(\mathcal{E}) - \sum_{\mathcal{E}\in\mathfrak{E}} P(\mathcal{E}) f_{uv}(\mathcal{E}) \right) + \lambda \left(\sum_{\mathcal{E}\in\mathfrak{E}} P(\mathcal{E}) - 1 \right).$$

By differentiating $H(P)$ with respect to $P(\mathcal{E})$, we get

$$\frac{\partial H(P)}{\partial P(\mathcal{E})} = -\ln P(\mathcal{E}) - 1 + \sum_{\{u,v\}\in\mathcal{F}} \lambda_{uv} f_{uv}(\mathcal{E}) + \lambda.$$

By setting derivatives to 0, we find the functional form for $P(\mathcal{E})$:

$$P(\mathcal{E}) = \exp\left(\lambda + 1 + \sum_{\{u,v\}\in\mathcal{F}} \lambda_{uv} f_{uv}(\mathcal{E}) \right) = \frac{1}{Z} \prod_{\{u,v\}\in\mathcal{E}} \beta_{uv}$$

where $\beta_{uv} = \begin{cases} e^{\lambda_{uv}} & : \{u,v\} \in \mathcal{F} \\ 0 & : \{u,v\} \notin \mathcal{F} \end{cases}$ and $Z = \exp(-\lambda - 1) = \sum_{\mathcal{E}\in\mathfrak{E}} \prod_{\{u,v\}\in\mathcal{E}} \beta_{uv}$, precisely the functional form of Equation 8.

B.3.1 Extension for Categorical Data

The EM algorithm for tree-averaged copulas can be partially transferred to the categorical case. Assuming that $\boldsymbol{X}$ is a multivariate categorical variable. A tree-structured distribution $T(\boldsymbol{x})$ can be formulated as

$$T(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{E}}) = \left(\prod_{v\in\mathcal{V}} \theta_v(x_v) \right) \left(\prod_{\{u,v\}\in\mathcal{E}} \frac{\theta_{uv}(x_u, x_v)}{\theta_u(x_u)\,\theta_v(x_v)} \right)$$

with $\theta_{uv}(i,j) = T(X_u = i, X_v = j)$ and $\theta_v(i) = T(X_v = i)$ specifying a constrained multivariate distributions pairs of edge variables and singleton variables, respectively.[6] Pairwise multinomial distributions can be defined for all pairs of variables with the constraint that if two such distribution model the same variable X_v, their marginals on X_v must agree with $\boldsymbol{\theta}_v$. Then ensemble of trees $R(\boldsymbol{x})$ is defined as an expected probability of a tree-structured distribution over all possible tree structures:

$$R(\boldsymbol{x}) = \sum_{\mathcal{E}\in\mathfrak{E}} P(\mathcal{E}|\boldsymbol{\beta})\, T(\boldsymbol{x}|\boldsymbol{\theta}_{\mathcal{E}}) = \left[\prod_{v\in\mathcal{V}} \theta_v(x_v) \right] \frac{|\mathbf{L}^{\star}(\boldsymbol{\beta}\boldsymbol{\omega}(\boldsymbol{x}))|}{|\mathbf{L}^{\star}(\boldsymbol{\beta})|}$$

where $\omega_{uv}(\boldsymbol{x}) = \frac{\theta_{uv}(x_u,x_v)}{\theta_u(x_u)\theta_v(x_v)}$. Similar to Equation 10, the EM objective function (without Lagrangians for constraints) can be written as

$$M\left(\boldsymbol{\beta}', \boldsymbol{\theta}'; \boldsymbol{\beta}, \boldsymbol{\theta}\right) = \sum_{v\in\mathcal{V}} \sum_{j} \sum_{n=1:\, x_v^n = j}^{N} \ln \theta'_v(j) - N \ln \left|\mathbf{L}^{\star}\left(\boldsymbol{\beta}'\right)\right| + \sum_{\{u,v\}} \sum_{i,j} \sum_{\substack{n=1 \\ x_u^n = i, x_v^n = j}}^{N} s_n(\{u,v\}) \left(\ln \beta'_{uv} + \ln \frac{\theta'_{uv}(i,j)}{\theta'_u(i)\,\theta'_v(j)} \right). \quad (18)$$

While this expression cannot be analytically maximized with respect $\boldsymbol{\theta}$, a reasonable update can be found by using the approach taken with parameter estimation for densities. First, we ignore the multivariate dependence terms (line 18) and estimate the parameters of univariate multinomials, and then update the parameters of the bivariate multinomials given the new values for the univariate terms. This update approach worked well in our experiments.

C Common Bivariate Parametric Copula Densities and Their Parameter Estimation

We list several commonly used bivariate copulas, and compute relevant parameter updates for Algorithm TREECOPULADENSITY (in particular, Equation 15) and Algorithm TREEAVERAGEDCOPULADENSITY (in particular, Equation 12). An expanded list of copulas and their properties can be found elsewhere (e.g., [13, 2, 14]).

[6] This notation is adopted from [1].

Before listing copula families, it is worth first to mention what range of bivariate dependence they can represent. We are interested in modeling data with various bivariate concordances (scale-invariant dependence)[7]; the concordance of a bivariate distribution is captured entirely by its copula. Each parametric family of copulas can only model a range of possible concordances. We will use two commonly used measures of concordance, Kendall's τ and Spearman's ρ to illustrate the applicability of certain copula families. Assume a pair of random variables (X, Y) is distributed according to a bivariate distribution P; assume that P is characterized by a copula $C(u, v)$. Kendall's τ is defined as a difference between probabilities of concordance and discordance between two vectors (X_1, Y_1) and (X_2, Y_2) independently drawn from P:

$$\begin{aligned}\tau &= P\left[(X_1 - X_2)(Y_1 - Y_2) > 0\right] - P\left[(X_1 - X_2)(Y_1 - Y_2) < 0\right] \\ &= 4\int_{\text{Ran}(P_X)}\int_{\text{Ran}(P_Y)} C(u, v)\, c(u, v)\, \mathrm{d}u\mathrm{d}v - 1.\end{aligned}$$

Spearman's ρ is defined as a difference between probabilities of concordance and discordance between a vector (X_1, Y_1) drawn from P and a vector (X_2, Y_3) where the components are drawn from margins P_X and P_Y independently (as a contrast to being drawn from a bivariate distribution for τ):

$$\rho = P\left[(X_1 - X_2)(Y_1 - Y_3) > 0\right] - P\left[(X_1 - X_2)(Y_1 - Y_3) < 0\right] = 12\int_{\text{Ran}(P_X)}\int_{\text{Ran}(P_Y)} C(u, v)\, \mathrm{d}u\mathrm{d}v - 3.$$

For bivariate copulas (and distributions) the largest negative and positive concordance in absolute terms (-1 and 1) is achieved at the Fréchet-Hoeffding bounds, copulas W and M, respectively:

$$W(u, v) = \max(u + v - 1, 0) \leq C(u, v) \leq \min(u, v) = M(u, v)\ \forall u, v \in [0, 1].$$

Product copula $\Pi(u, v) = uv$ corresponds to independence of variables and thus has 0 concordance. Copula $\frac{\Pi}{\Sigma - \Pi}(u, v) = \frac{uv}{u+v-uv}$ commonly appears on the boundaries of copula families; this copula has positive concordance ($\tau = \frac{1}{3}$, $\rho = 4\pi^2 - 39 \approx 0.4784$).

Both TREECOPULADENSITY and TREEAVERAGEDCOPULADENSITY update parameters θ_{uv} for copula density $c_{uv}(a_u, a_v; \theta_{uv})$ by setting the corresponding partial derivative to 0:

$$\frac{\partial l(\mathcal{A})}{\partial \theta_{uv}} = \sum_{n=1}^{N} w_n \frac{\partial \ln c_{uv}(a_u^n, a_v^n; \theta_{uv})}{\partial \theta_{uv}} = 0 \tag{19}$$

where $w_1, \ldots, w_N \geq 0$ are weights assigned to individual data points. (For the case of TREECOPULADENSITY, $w_n = 1,\ n = 1, \ldots, N$; for the case of TREEAVERAGEDCOPULADENSITY, $w_n = s_n(\{u, v\})$). Equation 19 cannot always be solved analytically for θ_{uv}. If no analytical solution exists, we suggest computing second derivative with respect to θ_{uv} and using Newton-Raphson method to estimate θ_{uv}.

To generate samples from a copula $C(u, v)$, one needs to know $c_u^{-1}(t)$, the inverse function of $\frac{\partial C(u,v)}{\partial u}$ the derivative of the copula with respect to one of the variables. Then a sample $(u, v) \sim C$ can be obtained by drawing two samples $u, t \sim U([0, 1])$ and transforming one of the variables, $v = c_u^{-1}(t)$ (e.g., [13]).

1. Farlie-Gumbel-Morgenstern (FGM) copula [15]:

$$\begin{aligned}C(u, v; \theta) &= uv\left[1 + \theta(1 - u)(1 - v)\right],\quad \theta \in [-1, 1]; \\ C(u, v; 0) &= \Pi;\quad \tau = \frac{2}{9}\theta \in \left[-\frac{2}{9}, \frac{2}{9}\right],\quad \rho = \frac{1}{3}\theta \in \left[-\frac{1}{3}, \frac{1}{3}\right]; \\ c_u(v) &= \theta(2u - 1)v^2 + (1 + \theta - 2\theta u)v, \\ c_u^{-1}(t) &= \begin{cases} \frac{1}{2} - \frac{1}{2\theta(2u-1)} + \sqrt{\left(\frac{1}{2} - \frac{1}{2\theta(2u-1)}\right)^2 + \frac{t}{\theta(2u-1)}} & : \quad \theta(2u - 1) > 0, \\ \frac{1}{2} - \frac{1}{2\theta(2u-1)} - \sqrt{\left(\frac{1}{2} - \frac{1}{2\theta(2u-1)}\right)^2 + \frac{t}{\theta(2u-1)}} & : \quad \theta(2u - 1) < 0; \end{cases} \\ c(u, v; \theta) &= 1 + \theta(1 - 2u)(1 - 2v),\quad \ln c(u, v; \theta) = \ln\left(1 + \theta(1 - 2u)(1 - 2v)\right); \\ \frac{\partial \ln c(u, v; \theta)}{\partial \theta} &= \frac{(1 - 2u)(1 - 2v)}{1 + \theta(1 - 2u)(1 - 2v)}; \\ \frac{\partial^2 \ln c(u, v; \theta)}{\partial \theta^2} &= -\left(\frac{(1 - 2u)(1 - 2v)}{1 + \theta(1 - 2u)(1 - 2v)}\right)^2.\end{aligned}$$

[7] Pearson's linear correlation is not scale-invariant as it is not preserved under non-linear monotonic transformations of the variables for a bivariate distribution.

2. Gaussian copula (from Example 2):

$$\begin{aligned}
C(u,v;\theta) &= \Phi_\theta\left(\Phi^{-1}(u),\Phi^{-1}(v)\right), \quad \theta\in(-1,1);\\
C(u,v;-1) &= W, \quad C(u,v;0)=\Pi, \quad C(u,v;1)=M; \quad \tau=\frac{2}{\pi}\sin^{-1}\theta, \rho=\frac{6}{\pi}\sin^{-1}\frac{\theta}{2}\in(-1,1);\\
c_u(v) &= \Phi\left(\frac{\Phi^{-1}(v)-\theta\Phi^{-1}(u)}{\sqrt{1-\theta^2}}\right), \quad c_u^{-1}(t)=\Phi\left(\sqrt{1-\theta^2}\Phi^{-1}(t)+\theta\Phi^{-1}(u)\right);\\
c(u,v;\theta) &= \left(1-\theta^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\theta^2\Phi^{-1}(u)^2+\theta^2\Phi^{-1}(v)^2-2\theta\Phi^{-1}(u)\Phi^{-1}(v)}{2(1-\theta^2)}\right);\\
\ln c(u,v;\theta) &= -\frac{1}{2}\ln\left(1-\theta^2\right)-\frac{\theta^2\Phi^{-1}(u)^2+\theta^2\Phi^{-1}(v)^2-2\theta\Phi^{-1}(u)\Phi^{-1}(v)}{2(1-\theta^2)};\\
\frac{\partial\ln c(u,v;\theta)}{\partial\theta} &= \frac{\theta\left(1-\theta^2\right)-\left(\theta\Phi^{-1}(u)^2+\theta\Phi^{-1}(v)^2-\Phi^{-1}(u)\Phi^{-1}(v)-\theta^2\Phi^{-1}(u)\Phi^{-1}(v)\right)}{\left(1-\theta^2\right)^2}.
\end{aligned}$$

Then the Equation 19 can be solved analytically:

$$\begin{aligned}
\frac{\partial l(\mathcal{A})}{\partial\theta_{uv}}=0 \quad\Leftrightarrow\quad & \frac{1}{(1-\theta_{uv})^2}\sum_{n=1}^{N}w_n\left(\theta_{uv}\left(1-\theta_{uv}^2\right)-\theta_{uv}\Phi^{-1}(a_u^n)^2-\theta_{uv}\Phi^{-1}(a_v^n)^2\right)\\
& +\frac{1}{(1-\theta_{uv})^2}\sum_{n=1}^{N}w_n\left(\Phi^{-1}(a_u^n)\Phi^{-1}(a_v^n)+\theta_{uv}^2\Phi^{-1}(a_u^n)\Phi^{-1}(a_v^n)\right)=0;\\
\Leftrightarrow\quad & \alpha_3\theta_{uv}^3+\alpha_2\theta_{uv}^2+\alpha_1\theta_{uv}+\alpha_0=0
\end{aligned}$$

where

$$\alpha_3=\sum_{n=1}^{n}w_n,\ \alpha_2=\alpha_0=-\sum_{n=1}^{N}w_n\Phi^{-1}(a_u^n)\Phi^{-1}(a_v^n),\ \alpha_1=\sum_{n=1}^{N}w_n\left(\Phi^{-1}(a_u^n)^2+\Phi^{-1}(a_v^n)^2-1\right).$$

3. Clayton copula [16]:

$$\begin{aligned}
C(u,v;\theta) &= \left(u^{-\theta}+v^{-\theta}-1\right)^{-\frac{1}{\theta}}, \quad \theta\in[-1,0)\cup(0,\infty),\\
C(u,v;-1) &= W, \quad C(u,v;0)=\Pi, \quad C(u,v;1)=\frac{\Pi}{\Sigma-\Pi}, \quad C(u,v;\infty)=M,\\
& \tau=\frac{\theta}{\theta+2}, \rho\in(0,1);\\
c_u(v) &= u^{-(\theta+1)}\left(u^{-\theta}+v^{-\theta}-1\right)^{-\frac{\theta+1}{\theta}}, \quad c_u^{-1}(t)=\left(1+u^{-\theta}\left(t^{-\frac{\theta}{\theta+1}}-1\right)\right)^{-\frac{1}{\theta}};\\
c(u,v;\theta) &= (\theta+1)(uv)^{-\theta-1}\left(u^{-\theta}+v^{-\theta}-1\right)^{-\frac{2\theta+1}{\theta}}, \quad \theta>0;\\
\ln c(u,v;\theta) &= \ln(\theta+1)-(\theta+1)\ln(uv)-\frac{2\theta+1}{\theta}\ln\left(u^{-\theta}+v^{-\theta}-1\right);\\
\frac{\partial\ln c(u,v;\theta)}{\partial\theta} &= \frac{1}{\theta+1}-\ln(uv)+\frac{1}{\theta^2}\ln\left(u^{-\theta}+v^{-\theta}-1\right)+\left(2+\frac{1}{\theta}\right)\frac{u^{-\theta}\ln u+v^{-\theta}\ln v}{u^{-\theta}+v^{-\theta}-1};\\
\frac{\partial^2\ln c(u,v;\theta)}{\partial\theta^2} &= -\frac{1}{(\theta+1)^2}-\frac{2}{\theta^2}\left(\frac{u^{-\theta}\ln u+v^{-\theta}\ln v}{u^{-\theta}+v^{-\theta}-1}+\frac{1}{\theta}\ln\left(u^{-\theta}+v^{-\theta}-1\right)\right)\\
& -\left(2+\frac{1}{\theta}\right)\left(\frac{u^{-\theta}\ln^2 u+v^{-\theta}\ln^2 v}{u^{-\theta}+v^{-\theta}-1}-\left(\frac{u^{-\theta}\ln u+v^{-\theta}\ln v}{u^{-\theta}+v^{-\theta}-1}\right)^2\right).
\end{aligned}$$

4. Ali-Mikhail-Haq copula [17]:

$$
\begin{aligned}
C(u,v;\theta) &= \frac{uv}{1-\theta(1-u)(1-v)}, \quad \theta\in[-1,1],\\
C(u,v;0) &= \Pi, \quad C(u,v;1)=\frac{\Pi}{\Sigma-\Pi},\\
&\tau=\frac{3\theta-2}{3\theta}-\frac{2(1-\theta)^2}{3\theta^2}\ln(1-\theta)\in\left[\frac{5-8\ln 2}{3},\frac{1}{3}\right]\approx[-0.1817,0.3333],\\
&\rho=\frac{12(1+\theta)}{\theta^2}\mathrm{di}\log(1-\theta)-\frac{24(1-\theta)}{\theta^2}\ln(1-\theta)-\frac{3(\theta+12)}{\theta} \text{ where}\\
&\mathrm{di}\log(x)=\int_1^x\frac{\ln t}{1-t}\mathrm{d}t; \quad \rho\in\left[33-48\ln 2,4\pi^2-39\right]\approx[-0.2711,0.4784];\\
c_u(v) &= \frac{v-\theta v+\theta v^2}{(1-\theta(1-u)(1-v))^2},\\
c_u^{-1}(t) &= -\frac{b}{2a}+\frac{\sqrt{b^2-4ac}}{2a} \text{ where } a=\theta-\theta^2(1-u)^2t,\ c=-(1-\theta(1-u))^2t, \text{ and}\\
&b=1-\theta-2\theta(1-u)(1-\theta(1-u))t;\\
c(u,v;\theta) &= \frac{1-\theta+2\theta uv-\theta(1-u)(1-v)+(1-u)(1-v)\theta^2}{(1-\theta(1-u)(1-v))^3};\\
\ln c(u,v;\theta) &= \ln\left(1-\theta+2\theta uv-\theta(1-u)(1-v)+(1-u)(1-v)\theta^2\right)-3\ln(1-\theta(1-u)(1-v));\\
\frac{\partial\ln c(u,v;\theta)}{\partial\theta} &= \frac{-1+2uv+(2\theta-1)(1-u)(1-v)}{1-\theta+2\theta uv-\theta(1-u)(1-v)+(1-u)(1-v)\theta^2}+\frac{3(1-u)(1-v)}{1-\theta(1-u)(1-v)};\\
\frac{\partial^2\ln c(u,v;\theta)}{\partial\theta^2} &= \frac{2(1-u)(1-v)}{1-\theta+2\theta uv-\theta(1-u)(1-v)+(1-u)(1-v)\theta^2}\\
&\quad-\left(\frac{-1+2uv+(2\theta-1)(1-u)(1-v)}{1-\theta+2\theta uv-\theta(1-u)(1-v)+(1-u)(1-v)\theta^2}\right)^2\\
&\quad+\frac{3(1-u)^2(1-v)^2}{(1-\theta(1-u)(1-v))^2}.
\end{aligned}
$$

5. Gumbel-Barnett copula [18, 19]:

$$
\begin{aligned}
C(u,v;\theta) &= uv\exp(-\theta\ln u\ln v), \quad \theta\in[0,1],\\
C(u,v;0) &= \Pi, \quad \tau\in[-0.3613,0], \quad \rho\in[-0.5239,0];\\
c_u(v) &= v(1-\theta\ln v)\exp(-\theta\ln u\ln v),\\
c_u^{-1}(t) &= v \text{ s.t. } (1-\theta\ln u)\ln v+\ln(1-\theta\ln v)=\ln t;\\
c(u,v;\theta) &= [(1-\theta\ln u)(1-\theta\ln v)-\theta]\exp(-\theta\ln u\ln v);\\
\ln c(u,v;\theta) &= \ln[(1-\theta\ln u)(1-\theta\ln v)-\theta]-\theta\ln u\ln v;\\
\frac{\partial\ln c(u,v;\theta)}{\partial\theta} &= \frac{2\theta\ln u\ln v-\ln u-\ln v-1}{(1-\theta\ln u)(1-\theta\ln v)-\theta}-\ln u\ln v,\\
\frac{\partial^2\ln c(u,v;\theta)}{\partial\theta^2} &= \frac{2\ln u\ln v}{(1-\theta\ln u)(1-\theta\ln v)-\theta}-\left(\frac{2\theta\ln u\ln v-\ln u-\ln v-1}{(1-\theta\ln u)(1-\theta\ln v)-\theta}\right)^2.
\end{aligned}
$$

6. Frank copula [20]:

$$
\begin{aligned}
C(u,v;\theta) &= -\frac{1}{\theta}\ln\left(1+\frac{\left(e^{-\theta u}-1\right)\left(e^{-\theta v}-1\right)}{e^{-\theta}-1}\right),\quad \theta\in(-\infty,0)\cup(0,\infty),\\
C(u,v;-\infty) &= W,\quad C(u,v;0)=\Pi,\quad C(u,v;\infty)=M,\\
&\quad \tau=1-\frac{4}{\theta}\left(1-D_1(\theta)\right),\rho=1-\frac{12}{\theta}\left(D_1(\theta)-D_2(\theta)\right)\in(-1,0)\cup(0,1),\\
&\quad D_k(x)=\frac{k}{x^k}\int_0^x\frac{t^k}{e^t-1}\mathrm{d}t;\\
c_u(v) &= \frac{e^{-\theta(u+v)}-e^{-\theta u}}{e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}},\quad c_u^{-1}(t)=-\frac{1}{\theta}\ln\left(\frac{te^{-\theta}+(1-t)e^{-\theta u}}{t+(1-t)e^{-\theta u}}\right);\\
c(u,v;\theta) &= -\theta\frac{e^{-\theta(u+v)}\left(e^{-\theta}-1\right)}{\left(e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}\right)^2};\\
\ln c(u,v;\theta) &= \ln\left(\theta\left(1-e^{-\theta}\right)\right)-\theta(u+v)-2\ln\left|e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}\right|;\\
\frac{\partial\ln c(u,v;\theta)}{\partial\theta} &= \frac{1}{\theta}+\frac{e^{-\theta}}{1-e^{-\theta}}-(u+v)+2\frac{e^{-\theta}+(u+v)e^{-\theta(u+v)}-ue^{-\theta u}-ve^{-\theta v}}{e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}};\\
\frac{\partial^2\ln c(u,v;\theta)}{\partial\theta^2} &= -\frac{1}{\theta^2}-\frac{e^{-\theta}}{\left(1-e^{-\theta}\right)^2}-2\frac{e^{-\theta}+(u+v)^2e^{-\theta(u+v)}-u^2e^{-\theta u}-v^2e^{-\theta}}{e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}}\\
&\quad -2\left(\frac{e^{-\theta}+(u+v)e^{-\theta(u+v)}-ue^{-\theta u}-ve^{-\theta v}}{e^{-\theta}+e^{-\theta(u+v)}-e^{-\theta u}-e^{-\theta v}}\right)^2.
\end{aligned}
$$