Dependence structure estimation using Copula Recursive Trees

Oskar Laverny, Esterina Masiello, Véronique Maume-Deschamps, Didier Rullière

Dependence structure estimation using Copula Recursive Trees

Oskar Laverny[a,*], Esterina Masiello[a], Véronique Maume-Deschamps[a], Didier Rullière[b]

[a] *Institut Camille Jordan, UMR 5208, Université Claude Bernard Lyon 1, Lyon, France*

[b] *Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne France*

Abstract

We construct the COpula Recursive Tree (CORT) estimator: a flexible, consistent, piecewise linear estimator of a copula, leveraging the patchwork copula formalization and various piecewise constant density estimators. While the patchwork structure imposes a grid, the CORT estimator is data-driven and constructs the (possibly irregular) grid recursively from the data, minimizing a chosen distance on the copula space. The addition of the copula constraints makes usual density estimators unusable, whereas the CORT estimator is only concerned with dependence and guarantees the uniformity of margins. Refinements such as localized dimension reduction and bagging are developed, analyzed, and tested through simulated data.

Keywords: Bagging, CORT, Density estimation trees, Nonparametric estimation, Patchwork copula, Piecewise linear copula, Quadratic program
2010 MSC: 62E17, 62H10, 62H20, 62G30

1. Introduction

Although the estimation of copula [15, 26, 38] is a wide-treated subject, most efficient estimators available in the literature are based on restricted, parametric estimation. Vine copulas [34–37, 40], although useful in high dimensions, often use parametric models, such as Archimedean copulas, as base building blocks. On the other hand, graphical models [17, 28] assume a Gaussian dependence structure and therefore are fast but under restrictive assumptions. Classical nonparametric density estimators such as kernels [23, 45–47] or wavelets [18, 20, 33] are not suited to satisfy constraints such as the uniformity of margins (one counter-example may be found in [8, 16]). We explore here a specific class of non-parametric copula density estimators with tree-structured piecewise constant densities, and design an estimator that lies in this class, the CORT estimator.

The CORT estimator is based on the density estimation tree from [42], a tree-structured non-parametric density estimator, and on the framework of patchwork copulas from [11, 13, 14]. There already exist several other piecewise constant density estimators: the cascaded histograms of [22], the Dirichlet-based Polya tree [39], the distribution element trees by [31], the adaptative sparse grids of [41], the framework of Gaussian mixtures by [9], the Bayesian sequential partitioning techniques by [27, 29] with their interesting asymptotic consistency results, and the Wasserstein compression techniques provided by [30] are all worth noting in the field of non-parametric piecewise density estimation. But these models are built to estimate densities without taking into account uniformity of margins, and they do not always lead to proper copulas when applied on pseudo-observations or true copula samples.

*Corresponding author. Email address: oskar.laverny@univ-lyon1.fr

The CORT estimator has the particularity of being tree-shaped which ensures on one hand that the computation of the estimated density and the distribution function on new data points is fast, and on the other hand that the storage of the model is efficient. Thus, it could be used for tasks such as re-sampling a dataset outside the already existing points, or for compression purposes, when dealing with big-data dependencies. Finally, under mild conditions, the estimator is a proper copula, where classical non-parametric estimators, such as Deheuvel's empirical copula, are not.

This paper is organized as follows. Section 2 describes the class of piecewise linear copulas and gives some of their properties. In Section 3, we propose an estimation procedure, allowing localized dimension reduction, and we establish a convergence result for this procedure. Section 4 deals with ensemble models based on the CORT estimator: bagging techniques and out-of-bag generalization statistics are developed in the field of copula density estimation, and applied to the CORT estimator. Finally, Section 5 investigates the performance of the model by applications on some simulated examples, and Section 6 concludes.

2. The piecewise linear copula

Let $\boldsymbol{X} = (X_1, \dots, X_d)^\top$ be a multivariate random vector of dimension d. We are interested in the dependence structure of $\boldsymbol{X}$. The concept of copula, whose formalization is due to [48], allows to study this dependence separately from the marginal distributions. Consider the distribution function (d.f.) F of the random vector $\boldsymbol{X}$ with marginal d.f.s $F_1, \dots, F_d$, and define the copula C as:

$$C(\boldsymbol{u}) = F\left\{F_1^{-1}(u_1), \dots, F_d^{-1}(u_d)\right\}, \quad \forall \boldsymbol{u} \in [0,1]^d,$$

where $F_i^{-1} : y \mapsto \inf\left\{x \in \bar{\mathbb{R}} : F_i(x) \geq y\right\}$ is the generalized inverse of F_i. Then the distribution F of $\boldsymbol{X}$ is characterized by $F_1, \dots, F_d$ and C. Sklar's Theorem [48] states that a copula C satisfying the previous equality always exists, and that it is uniquely defined provided that the marginal random variables are absolutely continuous. In particular, it is unique if the random vector is continuous. Note that C is the distribution function of a d-dimensional random vector with uniform margins.

The estimation of the full distribution can then be split into the estimation of one-dimensional margins, a widely treated subject, and the estimation of the copula for which we propose here to work with a piecewise linear distribution function. In the following, we define the piecewise linear copulas and then present some of their properties.

2.1. Definition

Let $\mathbb{I} = [0,1]^d$ be the unit hypercube, the domain of any copula, and let $\mathbb{1}_A$ be the indicator function of an event A.

Definition 1 (Piecewise linear copula)**.** Let $\mathcal{L}$ be a finite partition of $\mathbb{I}$ into subsets called leaves, usually denoted by ℓ. The piecewise linear copula with partition $\mathcal{L}$ and weighs p is defined by its distribution function:

$$\forall \boldsymbol{u} \in \mathbb{I},\ C_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u}) = \sum_{\ell \in \mathcal{L}} p_\ell \lambda_\ell(\boldsymbol{u}), \tag{1}$$

where λ denotes the Lebesgue measure of a set, $\lambda_\ell(\boldsymbol{u}) = \lambda(\ell)^{-1}\lambda([0, \boldsymbol{u}] \cap \ell)$, and $\boldsymbol{p}$ is a vector of non-negative weights summing to one. The corresponding copula density, which is piecewise constant, is given by:

$$c_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u}) = \sum_{\ell \in \mathcal{L}} \frac{p_\ell}{\lambda(\ell)} \mathbb{1}_{\boldsymbol{u} \in \ell}. \tag{2}$$

This type of histogram, where leaves might not all have same shape and size, has already been used in the case of density estimation with different construction schemes and leaves shapes, e.g. with a Voronoï diagram [7, 19] or a Delaunay tessellation [4, 25, 51] as partition, or more trivially with simple sets of hyper-boxes [2, 29, 30, 42].

Remark 1 (Existence)**.** Depending on the choice of the partition $\mathcal{L}$ and the weights $\boldsymbol{p}$, the distribution function $C_{\boldsymbol{p},\mathcal{L}}$ is not always a copula. However, if $\forall \ell \in \mathcal{L},\ p_\ell = \lambda(\ell)$, then $C_{\boldsymbol{p},\mathcal{L}}$ is the independence copula. Therefore, for any partition, there exists at least one set of weights making the model a proper copula. On the other hand, for a partition that is too complex, this is frequently the only solution: the assumption of uniform leaves and the marginals uniformity constraints restrict the shape of bins. For polytopic leaves that are not hyper-rectangles, we do not know if weighting them efficiently can be achieved inside the copula constraints.

Before looking more precisely at the copula constraints on these piecewise constant distribution functions, we therefore restrict ourselves to the case of hyper-rectangular leaves, leading to the following definition:

Definition 2 (Hyper-rectangles and suitable partitions)**.** Let $\boldsymbol{a}$ and $\boldsymbol{b}$ both be in $\mathbb{I}$. Then, if $\boldsymbol{a} \leq \boldsymbol{b}$ (component-wise), we define the hyper-rectangle $(\boldsymbol{a}, \boldsymbol{b}]$ as $(\boldsymbol{a}, \boldsymbol{b}] = (a_1, b_1] \times ... \times (a_d, b_d]$. We call suitable a partition where every leave is a hyper-rectangle with strictly positive Lebesgue measure.

Remark 1 partly drives the definition of a suitable partition. It is also why we chose to extend the density estimation trees from [42] instead of another piecewise constant density estimator: it produces a suitable partition. If not specified, we consider by default that partitions we are dealing with are suitable. In the next subsection, we propose some properties of the dependence structure induced by such a copula.

2.2. Properties

With the above formulation of a piecewise linear copula, we can easily obtain closed-form expressions for classical quantities of interest in copula modeling. We recall some of those quantities and then derive their expression for piecewise linear copulas. The Kendall τ and Spearman ρ (see [38]) are common dependence measures that can be computed from a copula. They are respectively defined for a copula C and its density c as:

$$\tau = 4 \int C(\boldsymbol{u})\, c(\boldsymbol{u})\, d\boldsymbol{u} - 1, \quad \rho = 12 \int C(\boldsymbol{u}) d\boldsymbol{u} - 3.$$

Note that both τ and ρ are always in $[-1, 1]$. The piecewise constant expression of the density in the piecewise linear class allows for simple computation of τ and ρ, although the expressions can be somewhat cumbersome.

Proposition 1 (Common dependence measures)**.** *Let $C_{\boldsymbol{p},\mathcal{L}}$ be a piecewise linear copula. Its Kendall τ and Spearman ρ are given in closed form by:*

$$\tau = -1 + 4 \sum_{\substack{\ell \in \mathcal{L} \\ k \in \mathcal{L}}} \frac{p_\ell p_k}{\lambda(\ell)\lambda(k)} \prod_{i=1}^{d} \left\{ \frac{(b \wedge d)^2 - (a \vee c)^2}{2} + c\left((a \vee c) - (b \wedge d)\right) + (d - c)\left(b - (a \vee d)\right) \right\},$$

$$\rho = -3 + 6 \sum_{\ell \in \mathcal{L}} p_\ell \prod_{i=1}^{d} (2 - b_i - a_i),$$

where we denote $\ell = (\boldsymbol{a}, \boldsymbol{b}]$ and $k = (\boldsymbol{c}, \boldsymbol{d}]$, and $\wedge$, $\vee$ denote respectively the minimum and maximum operator.

The proof is postponed to the appendix. Matrices of bivariate dependence measures can be obtained by projection of the partition on couples of dimensions and using the same formula on the projected models. The closed form expression for piecewise linear copulas facilitates greatly these kinds of computations. This property will be exploited later, when introducing penalization techniques. Furthermore, these closed form expressions will be used to assess the performance of the fitting procedure that we describe in the next section.

3. Estimation

Suppose that we have a dataset $\left(u_{i,j}\right)_{n\times d}$ of (pseudo-)observations from an unknown copula C. We seek parameters $(\boldsymbol{p}, \mathcal{L})$ of $c^{(n)}_{\boldsymbol{p},\mathcal{L}}$, an approximation of c in the piecewise linear copula class based on these n observations. To find the optimal parameters $(\boldsymbol{p}^*, \mathcal{L}^*)$, we will adopt a two stage mechanism, considering first that the partition $\mathcal{L}$ is known. From now on, we denote by $\|f\|_2^2$ the squared L_2 norm of a function f, given by $\|f\|_2^2 = \int f(x)^2 dx$.

3.1. Optimal weights $\boldsymbol{p}^*_{\mathcal{L}}$ knowing the partition $\mathcal{L}$

Suppose that a partition $\mathcal{L}$ is already constructed, and that we want to construct weights $\boldsymbol{p}$ to complete the approximation. As was done by [42] (see also [1, 3]), we will use an Integrated Square Error (ISE) loss to build the weights. Given a copula density c, the ISE of an estimator $\hat{c}$ is the squared L_2 distance to c:

$$I(\hat{c}) = \|\hat{c} - c\|_2^2 = \int (\hat{c}(\boldsymbol{u}) - c(\boldsymbol{u}))^2 d\boldsymbol{u}.$$

To approximate the true copula density c by a piecewise linear copula, we are looking to solve:

$$\begin{aligned}\underset{\boldsymbol{p},\mathcal{L}}{\arg\min}\ I(c^{(n)}_{\boldsymbol{p},\mathcal{L}}) &= \underset{\boldsymbol{p},\mathcal{L}}{\arg\min}\ \|c^{(n)}_{\boldsymbol{p},\mathcal{L}} - c\|_2^2 = \underset{\boldsymbol{p},\mathcal{L}}{\arg\min} \int (c^{(n)}_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u}) - c(\boldsymbol{u}))^2 d\boldsymbol{u} \\ &= \underset{\boldsymbol{p},\mathcal{L}}{\arg\min} \int c^{(n)}_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u})^2 - 2c(\boldsymbol{u})c^{(n)}_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u}) d\boldsymbol{u} = \underset{\boldsymbol{p},\mathcal{L}}{\arg\min}\ \|c^{(n)}_{\boldsymbol{p},\mathcal{L}}\|_2^2 - 2\left\langle c^{(n)}_{\boldsymbol{p},\mathcal{L}}, c\right\rangle,\end{aligned}$$

since $\|c\|_2^2$ is irrelevant to the minimization. Remark that $\langle c^{(n)}_{\boldsymbol{p},\mathcal{L}}, c\rangle = \mathbb{E}\left(c^{(n)}_{\boldsymbol{p},\mathcal{L}}(U)\right)$, so that, with a slight abuse of notation, we write $\langle c^{(n)}_{\boldsymbol{p},\mathcal{L}}, c\rangle$ even if C does not admit a density c. Furthermore, since the true copula C and its density c are unknowns, using empirical observations $(u_{i,j})$ from the copula, $\mathbb{E}\left(c^{(n)}_{\boldsymbol{p},\mathcal{L}}(U)\right)$ can be approximated by $n^{-1}\sum_{i=1}^n c^{(n)}_{\boldsymbol{p},\mathcal{L}}(u_i)$. We then define our empirical loss.

Definition 3 (Empirical Integrated Square Error)**.** Given observations $(u_{i,j})$ from a copula, define the Empirical Integrated Square Error (EISE) of an estimator $\hat{c}$ of the copula density as:

$$\hat{I}(\hat{c}) = \|\hat{c}\|_2^2 - \frac{2}{n}\sum_{i=1}^n \hat{c}(\boldsymbol{u_i}). \tag{3}$$

To minimize this loss, we find the weights $\boldsymbol{p}^*_{\mathcal{L}}$ knowing the partition $\mathcal{L}$ by solving the following problem:

$$\underset{p}{\arg\min}\ \hat{I}(c_{\boldsymbol{p},\mathcal{L}}), \quad \text{s.t. } c_{\boldsymbol{p},\mathcal{L}} \text{ is a copula.} \tag{4}$$

The copula constraints in Eq. (4) are classically expressed in the literature as in Definition 4:

Definition 4 (Copula constraints)**.** Let C be the distribution function of a signed measure μ_C. C is a copula if the following three conditions are satisfied:

$$\begin{aligned}&\forall \boldsymbol{u} \in \mathbb{I}, \forall i \in \{1,\dots,d\}, C(u_1,\dots,u_{i-1},0,u_{i+1},\dots,u_d) = 0, && \text{(Ground constraint)},\\ &\forall \boldsymbol{u} \in \mathbb{I}, \forall i \in \{1,\dots,d\}, C(1,\dots,1,u_i,1,\dots,1) = u_i, && \text{(Marginal uniformity)},\\ &\forall \boldsymbol{a} \le \boldsymbol{b} \in \mathbb{I}, \mu_C([\boldsymbol{a},\boldsymbol{b}]) \ge 0, && (d\text{-increasingness}).\end{aligned}$$

Note that, if C is a distribution function of some random vector, the first and third conditions are verified. It turns out that, under these constraints, our optimization problem is in fact a quadratic program, as Proposition 2 shows.

Proposition 2 (Quadratic program)**.** *Let $\mathcal{L}$ be a suitable partition. Then the weights minimizing the empirical integrated square error* (3) *under the copula constraints from Definition 4 are the unique solution of the quadratic program:*

$$\arg\min_{\boldsymbol{p}} \ \boldsymbol{p}'\boldsymbol{A}_{\mathcal{L}}\boldsymbol{p} - 2\boldsymbol{p}'\boldsymbol{A}_{\mathcal{L}}\mathbf{f}_{\mathcal{L}}, \quad \text{s.t. } \boldsymbol{B}_{\mathcal{L}}\boldsymbol{p} = \boldsymbol{g}_{\mathcal{L}} \text{ and } \boldsymbol{p} \geq \mathbf{0},$$

where the matrices $\boldsymbol{A}_{\mathcal{L}}$, $\boldsymbol{B}_{\mathcal{L}}$ and the vectors $\mathbf{f}_{\mathcal{L}}$, $\boldsymbol{g}_{\mathcal{L}}$ are given by:

$$\begin{aligned}
\boldsymbol{A}_{\mathcal{L}} &= \left(\lambda(\ell)^{-1}\mathbb{1}_{\ell=k}\right)_{\ell\in\mathcal{L},k\in\mathcal{L}}, && \textit{(size } |\mathcal{L}|\times|\mathcal{L}|\textit{)},\\
\boldsymbol{B}_1 &= \left(\lambda_{\ell_i}(u_i)\right)_{(i,\boldsymbol{u})\in\{1,\dots,d\}\times M_{\mathcal{L}},\ \ell\in\mathcal{L}}, && \textit{(size } nd\times|\mathcal{L}|\textit{)},\\
\boldsymbol{B}_{\mathcal{L}} &= (\boldsymbol{B}_1, \boldsymbol{B}_2), && \textit{(size } (nd+1)\times|\mathcal{L}|\textit{)},\\
\mathbf{f}_{\mathcal{L}} &= \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\boldsymbol{u}_i\in\ell}\right)_{\ell\in\mathcal{L}}, && \textit{(size } |\mathcal{L}|\textit{)},\\
\boldsymbol{B}_2 &= (1)_{\ell\in\mathcal{L}}, && \textit{(size } 1\times|\mathcal{L}|\textit{)},\\
\boldsymbol{g}_1 &= (u_i)_{(i,\boldsymbol{u})\in\{1,\dots,d\}\times M_{\mathcal{L}}}, && \textit{(size } nd\textit{)},\\
\boldsymbol{g}_{\mathcal{L}} &= (\boldsymbol{g}_1, 1), && \textit{(size } (nd+1)\textit{)},
\end{aligned}$$

where we denoted $|\mathcal{L}|$ the cardinal of $\mathcal{L}$, and $M_{\mathcal{L}}$ the set of middle-points of leaves.

Proof *This proof is in three parts. First, we show that the objective formulation is correct, then we discuss the constraints formulation and finally we prove existence and uniqueness of a solution. Introduce first a new scalar product on $\mathbb{R}^{|\mathcal{L}|}$:*

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} = \sum_{\ell\in\mathcal{L}} \frac{x_\ell y_\ell}{\lambda(\ell)}.$$

We denote by $\|.\|^2_{\mathcal{L}}$ its associated square norm, and by $d_{\mathcal{L}}$ its associated distance. Note that the associated bilinear symmetric operator has matrix $\boldsymbol{A}_{\mathcal{L}}$ (defined above), a diagonal and positive definite matrix.

Using the definition of $\boldsymbol{A}_{\mathcal{L}}$, $\mathbf{f}_{\mathcal{L}}$ and $\langle .,.\rangle_{\mathcal{L}}$, the objective function in Eq. (4) *rewrites:*

$$\hat{I}(c_{\boldsymbol{p},\mathcal{L}}) = \|\boldsymbol{p}\|^2_{\mathcal{L}} - 2\langle \boldsymbol{p}, \mathbf{f}_{\mathcal{L}}\rangle_{\mathcal{L}} = \boldsymbol{p}'\boldsymbol{A}_{\mathcal{L}}\boldsymbol{p} - 2\boldsymbol{p}'\boldsymbol{A}_{\mathcal{L}}\mathbf{f}_{\mathcal{L}}. \tag{5}$$

Then, the unconstrained version of the convex optimization problem from Eq. (4) *corresponds to the projection of $\mathbf{f}_{\mathcal{L}}$ onto $[0,1]^{|\mathcal{L}|}$ (convex, closed), with respect to the norm $\|.\|^2_{\mathcal{L}}$. Note that, if we set the weights to be equal to the empirical frequencies $\mathbf{f}_{\mathcal{L}}$, this result yields, for the optimization of $\mathcal{L}$, a loss of the form:*

$$-\|\mathbf{f}_{\mathcal{L}}\|^2_{\mathcal{L}} = -\sum_{\ell\in\mathcal{L}} \frac{\mathrm{f}_\ell^2}{\lambda(\ell)}, \tag{6}$$

which is the loss that was directly used by [42].

We now discuss the constraints. Denote by $C_{\mathcal{L}}$ the subset of $[0,1]^{|\mathcal{L}|}$ containing vectors $\boldsymbol{p}$ such that $C_{\boldsymbol{p},\mathcal{L}}$ satisfies the set of constraints from Definition 4, for a given $\mathcal{L}$. The claim is that:

$$C_{\mathcal{L}} = \left\{\boldsymbol{p}\in\mathbb{R}^{|\mathcal{L}|} : \ \boldsymbol{B}_{\mathcal{L}}\boldsymbol{p} = \boldsymbol{g}_{\mathcal{L}} \text{ and } \boldsymbol{p}\geq 0\right\}.$$

The first constraint of Definition 4 is trivially satisfied by our model. We show that the second constraint can be evaluated on only one point per leaf. Remember that the piecewise linear copula is uniformly distributed on each leaf. Hence, for all $i \in \{1, \dots, d\}$, if on some point $\boldsymbol{u} = (u_1, \dots, u_d)^\top$, $\sum_{\ell \in \mathcal{L}} p_\ell \lambda_{\ell_i}(u_i) = u_i$, then defining $\boldsymbol{x}$ such that $\boldsymbol{u} + \boldsymbol{x}$ is in the same leaf as $\boldsymbol{u}$ will give us that $\sum_{\ell \in \mathcal{L}} p_\ell \lambda_{\ell_i}(u_i + x_i) = u_i + x_i$ since only one leaf is active in the sum. Therefore, the marginal uniformity constraint may be evaluated on only one point per leaf, and hence is equivalent to:

$$\forall \ell \in \mathcal{L}, \exists\, \boldsymbol{u} \in \ell, \forall i \in \{1, \dots, d\}, \sum_{\ell \in \mathcal{L}} p_\ell \lambda_{\ell_i}(u_i) = u_i,$$

where for $\ell = [\boldsymbol{a}, \boldsymbol{b}]$ we still denote $\ell_i = [a_i, b_i]$ its marginals and $\lambda_{\ell_i}(u_i) = \frac{\lambda([0, u_i] \cap \ell_i)}{\lambda(\ell_i)}$. Then, if we choose evaluation points as middle-points of leaves to put the constraints in matrix-vector form, we have the expression $\boldsymbol{B}_1 \boldsymbol{p} = \boldsymbol{g}_1$ for these constraints. Furthermore, we need to force the sum of weights $\boldsymbol{p}$ to be equal to 1 (so that the total marginal mass is 1), giving a last line of ones to $\boldsymbol{B}_\mathcal{L}$ and a last value of one to $\boldsymbol{g}_\mathcal{L}$.

The third constraint states that the measure associated with the copula C has to be positive on any hyper-rectangle $[\boldsymbol{a}, \boldsymbol{b}]$. Recall that:

$$\forall \boldsymbol{a} \leq \boldsymbol{b},\ \mu_C([\boldsymbol{a}, \boldsymbol{b}]) = \int_{[\boldsymbol{a}, \boldsymbol{b}]} c(\boldsymbol{u}) d\boldsymbol{u} = \sum_{\ell \in \mathcal{L}} p_\ell \lambda_\ell([\boldsymbol{a}, \boldsymbol{b}])$$

If all weights are positive, then $\forall \boldsymbol{a} \leq \boldsymbol{b},\ \mu_C([\boldsymbol{a}, \boldsymbol{b}]) \geq 0$. On the other hand, if one weight is negative, then taking $\boldsymbol{a}, \boldsymbol{b}$ inside the corresponding leaf would make $\mu_C([a, b])$ negative. The last constraint is therefore equivalent to positivity of weights, which gives the last part of the wanted expression for $C_\mathcal{L}$. Finally, existence and uniqueness of a solution are trivial since the objective function is a quadratic function, $C_\mathcal{L}$ is a closed and convex set, and $C_\mathcal{L}$ is non-empty by Remark 1. □

Denoting $P_{\mathcal{L},S}(\boldsymbol{x})$ the orthogonal projection of a vector $\boldsymbol{x}$ onto a set S regarding the distance $d_\mathcal{L}$, the quadratic program from Proposition 2 gives the optimal weights knowing the partition as:

$$\boldsymbol{p}^*_\mathcal{L} = P_{\mathcal{L}, C_\mathcal{L}}(\mathbf{f}_\mathcal{L}).$$

The empirical frequencies $\mathbf{f}_\mathcal{L}$, which are the unconstrained solution, can then be used as a good starting point for a solver. We concentrate now on the construction of the partition $\mathcal{L}$.

3.2. Locally optimal splitting point $\boldsymbol{x}^*_{\ell, \boldsymbol{D}}$

Suppose that we have already a suitable partition $\mathcal{L}$ and associated weights $\boldsymbol{p}_\mathcal{L}$ such that $C_{\boldsymbol{p}_\mathcal{L}, \mathcal{L}}$ is a copula. For a given leaf $\ell \in \mathcal{L}$, denote $L = \ell_1, \dots, \ell_k$ a partition of ℓ into k new leaves such that $\mathcal{L}_2 = \mathcal{L} \setminus \{\ell\} \cup L$ is a new suitable partition. Then we have, as in [42], the following property:

Proposition 3 (Independence of surrogate loss)**.** *Define the surrogate loss associated to the additional split from $(\boldsymbol{p}^*_\mathcal{L}, \mathcal{L})$ to $(\boldsymbol{p}^*_{\mathcal{L}_2}, \mathcal{L}_2)$ as the difference of empirical integrated squared errors $\hat{I}(c_{\boldsymbol{p}^*_{\mathcal{L}_2}, \mathcal{L}_2}) - \hat{I}(c_{\boldsymbol{p}^*_\mathcal{L}, \mathcal{L}})$. Then the surrogate loss depends only on the partition L and data inside it.*

Proof *Index the objects of Proposition 2 by the partitions $\mathcal{L}$ and $\mathcal{L}_2$, and the part corresponding to $\mathcal{L} \setminus \{\ell\}$ cancels.* □

This locality of the loss allowed [42] to use a recursive partitioning algorithm. We then only perform simple splits.

Definition 5 (Simple split and splitting dimensions)**.** Denote $\boldsymbol{x}$ a given breakpoint in the leaf ℓ and $\boldsymbol{D} \subseteq \{1, \dots, d\}$ the set of splitting dimensions. Then the simple split of ℓ on $\boldsymbol{x}$ with dimensions $\boldsymbol{D}$ is defined as the partition $L(\ell, \boldsymbol{x}, \boldsymbol{D})$ given by:

$$L((\boldsymbol{a}, \boldsymbol{b}], \boldsymbol{x}, \boldsymbol{D}) = \bigtimes_{j \in \boldsymbol{D}} \left\{(a_j, x_j], (x_j, b_j]\right\} \bigtimes_{j \in \{1, \dots, d\} \setminus \boldsymbol{D}} \left\{(a_j, b_j]\right\}.$$

The full partition of $\mathbb{I}$ obtained after the split is denoted $\mathcal{L}_{\boldsymbol{x},\boldsymbol{D}} = \mathcal{L} \setminus \{\ell\} \cup L(\ell, \boldsymbol{x}, \boldsymbol{D})$. When $\boldsymbol{D} = \{1, \dots, d\}$, we might omit it in the subscripts.

Remark 2 (Degrees of freedom)**.** In a simple split on a set of dimension $\boldsymbol{D}$, the weighting of the new leaves is a quadratic program with $2^{|\boldsymbol{D}|}$ parameters responding to $|\boldsymbol{D}| + 1$ linear constraints. Hence, there exists a trade-off between complexity and expressiveness of the model in the dimension $|\boldsymbol{D}|$ of the breakpoints. We will exploit this characteristic of the recursive procedure for sparsity purposes later on.

Remark 3 (No one-dimensional splits)**.** Note that the copula constraints will not allow for estimation with only one-dimensional splits ($|\boldsymbol{D}| = 1$), as in [42], since there would be no degrees of freedom in the weights. This represents a huge problem as multivariate splits often imply bigger computational burden. Furthermore, the constraints themselves are not localized, but on the global scale, hence including them forbids a parallel implementation. We will see later that this issue can be avoided by delaying the constraint problem to a later stage of the optimization process.

Note that, neglecting the constraints, knowing $\boldsymbol{D}$, we are going to choose the splitting point $\boldsymbol{x}^*_{\ell,\boldsymbol{D}}$ as:

$$\boldsymbol{x}^*_{\ell,\boldsymbol{D}} = \underset{\boldsymbol{x} \in \ell}{\arg\min} - \sum_{k \in L(\ell, \boldsymbol{x}, \boldsymbol{D})} \frac{\mathrm{f}_k^2}{\lambda(k)} \tag{7}$$

In the next subsection, before giving a complete description of the fitting algorithm, we talk about the localized dimension reduction procedures that are possible with these simple splits, and describe the construction of splitting dimensions $\boldsymbol{D}$.

3.3. Optimal splitting dimensions $\boldsymbol{D}^*$

Suppose that we found $\boldsymbol{x}^*_\ell$ to split a leaf ℓ in all the dimensions $\{1, \dots, d\}$. Before effectively splitting the leaf, we will check if, by chance, some dimensions can be removed from the splitting dimensions. Indeed, we will choose the splitting dimensions $\boldsymbol{D}$ based on a statistical test whose hypothesis writes:

Hypothesis 1 (Sparsity hypothesis $\mathcal{H}_j$)**.** Denoting $\boldsymbol{U} \sim C$, define for a given dimension $j \in \{1, \dots, d\}$ the sparsity hypothesis as:

$$\mathcal{H}_j : \quad \left(U_j \perp\!\!\!\perp U_{-j}\right) \mid \boldsymbol{U} \in \ell \text{ and } U_j \mid \boldsymbol{U} \in \ell \sim \mathcal{U}(\ell_j),$$

where we denoted U_j the j^{th} marginal of the random vector U and U_{-j} the vector of all other marginals.

The hypothesis $\mathcal{H}_j$ literally says that, inside the leaf ℓ, the dimension j of the data is uniform and independent of the others. When $\mathcal{H}_j$ is accepted in a leaf ℓ, we will reduce the dimension of the breakpoint $\boldsymbol{x}^*_\ell$ in this leaf and in all child leaves accordingly, by removing j from the set of splitting dimensions $\boldsymbol{D}^*$. This will have the result that, inside the leaf ℓ, the final model will consider the dimension j to be uniform and independent of others.

To check this hypothesis using the integrated square error, as analyzed by [5], suppose without loss of generality that ℓ is rescaled to $\mathbb{I}$, containing n observations of the random vector $\boldsymbol{U} \sim F$, for F the restriction of C to ℓ, rescaled to $\mathbb{I}$ (note that F is not a copula). This removes the conditioning in the hypothesis. Then, define the test statistic as follows.

Definition 6 (Test statistic)**.** Denote by $f^{(n)}_{\mathbf{f},\mathcal{L}}$ the piecewise constant density that will be estimated on data $\boldsymbol{U} \sim F$, using the surrogate loss (7), and by $e_{j,n}(x) = \mathbb{E}\left(f^{(n)}_{\mathbf{f},\mathcal{L}}(x) | \mathcal{H}_j\right)$ the expectation of this estimate under $\mathcal{H}_j$. The test statistic is given by:

$$\mathcal{I}_j = \|e_{j,n} - f^{(n)}_{\mathbf{f},\mathcal{L}}\|_2^2,$$

where $\mathcal{L}$, $e_{j,n}$ and $f^{(n)}_{\mathbf{f},\mathcal{L}}$ are stochastic objects, depending on the independent and identically distributed (*i.i.d.*) random vectors $\boldsymbol{U}_1, \dots, \boldsymbol{U}_n$.

This test statistic does not test hypothesis $\mathcal{H}_j$ per se, but rather tests $\mathcal{H}_j$ under the hypothesis of piecewise constant density. This is a weaker assertion, but it is enough to decide if local dimension reduction is possible or not given the current estimation stage and the data. More classical tests for independence and/or uniformity can be founded in [12, 21, 24, 49, 53, 54].

The statistic $\mathcal{I}_j$ has the nice property that it is constructed as a square distance, and hence is always non-negative and is 0 only under $\mathcal{H}_j$. On the other hand, it requires that we compute the full patchwork distribution in the two cases (under $\mathcal{H}_j$ or not), which can be costly. Instead, we can only compute the next split, which will reduce the computation and simplify the statistic. The drawback is that the test is weakened. The next property gives an empirical form of this statistic, using this simplification.

Proposition 4 (Empirical form of the statistic)**.** *On a sample of data $(u_{i,j})_{n\times d}$, we can approximate the statistic $\mathcal{I}_j$ by:*

$$\widehat{\mathcal{I}_j} = \sum_{k\in\mathcal{L}_{x^*,\{1,\dots,d\}\setminus\{j\}}} \left\{ \frac{\mathrm{f}_k^2}{\lambda(k)} + \sum_{\substack{\ell\in\mathcal{L}_{x^*,\{1,\dots,d\}} \\ \ell\subset k}} \left(\frac{\mathrm{f}_\ell^2}{\lambda(\ell)} - 2\frac{\mathrm{f}_k\mathrm{f}_\ell}{\lambda(k)} \right) \right\}. \tag{8}$$

Proof *The estimator will cut on the same breakpoints on dimensions other than j whether we work under $\mathcal{H}_j$ or not. This gives the definition of the partitions, and hence the expression for $e_{j,n}$ and $f^{(n)}_{\mathbf{f},\mathcal{L}}$, giving the expression of $\widehat{\mathcal{I}_j}$.* □

The law of the statistic (8) under $\mathcal{H}_j$ cannot be computed explicitly. We use a Monte-Carlo simulation to compute the p-value of the test. To that purpose, simulate n uniform random variables to replace $\boldsymbol{u}_{\cdot,j}$, and recompute the statistic (8), T times. Indeed, note that under the null, the values of $\boldsymbol{u}_{\cdot,-j}$ can be held fixed.

The full localized dimension reduction procedure is formalized in Algorithm 1.

Algorithm 1: Localized dimension reduction.

Data: $T \in \mathbb{N}$, a leaf ℓ, observations $\boldsymbol{u}_1, .., \boldsymbol{u}_n \in \ell$, and a threshold probability α
Result: The splitting dimensions $\boldsymbol{D}^*$

Obtain $\boldsymbol{x}^*_\ell$ and $\mathcal{L}_{\boldsymbol{x}^*_\ell,\{1,\dots,d\}}$ by greedily minimizing the loss in (7) on ℓ.
foreach $j \in 1, \dots, d$ **do**
  Denote by s the statistic $\widehat{\mathcal{I}_j}$ given by (8).
  foreach $i \in 1, \dots, T$ **do**
    Simulate n uniform random variables on ℓ_j and replace the jth coordinate of the data by this simulation;
    Denote s_i the value of the statistic from (8) on this new dataset;
  end
  Set $p_j = \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{s<s_i}$
end
return $\boldsymbol{D}^* = \left\{ j \in 1, \dots, d : p_j > 1 - \alpha \right\}$

Note that the threshold α is a probability threshold for each individual test, and not a global threshold. To avoid multiple testing issues, it should be treated as a hyper-parameter and not a type 1 error. We now formalize in the next subsection the complete estimation procedure.

3.4. Full estimation procedure

Suppose that we start from the independence copula, which writes $C_{\{1\},\{\mathbb{I}\}}$ in our framework. Then, if the sample of observations belonging to the sole leaf is too far from independence, i.e., if $\mathcal{H}_j$ does not hold for all j, we construct the first breakpoint by greedily minimizing the loss (7) over the splitting point $\boldsymbol{x}$. Rescaling the new leaves to $\mathbb{I}$ allows starting over and split again, until a proper stopping condition is reached: either there are no points anymore in the leaf or the leaf passes the uniformity tests. A third stopping condition is that the loss is no more reduced by splitting.

To fasten the computation, we decided to ignore the copula constraints while splitting, and enforce them at the end on the constructed partition to correct the empirical weights. Later properties of convergence back up this decision, and this furthermore allows to parallelize the splitting process. Experiments showed that the algorithm that enforces the constraints at each split is much slower (since for the optimization of the breakpoint, sub-optimization corresponding to the quadratic program of Proposition 2 must be run for each evaluation) and does not provide much better results.

More formally, Algorithm 2 below states the complete estimation procedure.

Algorithm 2: CORT estimation.

Data: Observed ranks $\boldsymbol{u}_1, .., \boldsymbol{u}_n \in \mathbb{I}$
Result: Parameters $\boldsymbol{p}$ and $\mathcal{L}$ of the estimated piecewise linear copula
1 Initialize the tree by $\mathcal{L} = \{\mathbb{I}\}$ and $\boldsymbol{p}_{\mathcal{L}} = \{1\}$.
2 **while** *there exist leaves* $\ell \in \mathcal{L}$ *that are still splitable* **do**
3 **foreach** $\ell \in \mathcal{L}$ *that is splitable* **do**
4 Run Algorithm 1 in ℓ to find $\boldsymbol{D}^*$.
5 **if** $\boldsymbol{D}^* \neq \emptyset$ **then**
6 Find $\boldsymbol{x}^*_{\ell,\boldsymbol{D}^*}$ minimizing the surrogate loss (7).
7 Set $\mathcal{L} = \mathcal{L}_{\boldsymbol{x}^*_{\ell,\boldsymbol{D}^*},\boldsymbol{D}^*}$.
8 **end**
9 **end**
10 **end**
11 Compute $\boldsymbol{p}^*_{\mathcal{L}} = P_{\mathcal{L},\mathcal{C}_L}(\mathbf{f}_{\mathcal{L}})$ from Proposition 2.
12 **return** $(\boldsymbol{p}^*_{\mathcal{L}}, \mathcal{L})$.

The resulting estimator of the copula density, denoted by $c^{(n)}_{\boldsymbol{p}^*_{\mathcal{L}},\mathcal{L}}$, is called CORT for Copula Recursive Tree. The current implementation of this algorithm as an R package `cort` is available on `CRAN`. Note that the conditions for a leaf to be splitable can vary: by default, we consider that a leaf becomes non-splitable when it contains less than two observations. Options exist in the implementation to randomize the splitting dimensions (instead of optimizing them), or to constraint the maximum number of splitting dimensions to be inferior to some threshold. We do not provide the analysis of these parametrizations.

Remark 4. Before step 11 of Algorithm 2, the built model is simply the density estimate $f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}}$. This additional outcome of the fitting procedure will be used in the next sections for performance analysis.

Remark 5. If we restrict the breakpoint possibility to all points with coordinates in $\left(\frac{i}{m+1}\right)_{i\in\{1,\dots,m\}}$ where m divides the number of observations n, this gives us only m^d candidates. This corresponds to a form of checkerboard copula, see [10] for more details. Since the ISE loss we use is tractable enough, the breakpoint can be chosen by directly minimizing the criterion over the continuous space $\mathbb{I}$ if the dimension of the problem is not too big.

After discussing the consistency of this estimation procedure, Section 4 discusses potential extensions, mainly through bagging principles.

3.5. Consistency

We will show the consistency of the CORT estimator in the L_2 almost-sure sense. Recall from [42, Theorem 1] the following result about the unconstrained estimator.

Proposition 5 (Consistency of $f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}}$)**.** *Assuming the maximum diameter of leaves goes to* 0 *as n goes to* ∞*, we have:*

$$\Pr\left(\lim_{n\mapsto+\infty} \|f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}} - c\|_2^2 = 0\right) = 1$$

A detailed proof, based on a generalization by Vapnik-Chervonenkis [50] of the Glivenko-Cantelli Theorem, can be found in [42]. Denote now by $\boldsymbol{q}_{\mathcal{L}}$ the volumes given by the true copula on the leaves:

$$\forall \ell \in \mathcal{L},\ q_\ell = \int_\ell c(\boldsymbol{u}) d\boldsymbol{u}.$$

Then, one can easily check that $\boldsymbol{q}_{\mathcal{L}} \in C_{\mathcal{L}}$, and Proposition 5 leads to the following useful corollary:

Corollary 1. $d_{\mathcal{L}}(\mathbf{f}_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2 \to 0$*, a.s.*

Indeed, replacing c by a piecewise constant density with partition $\mathcal{L}$ and weights $\boldsymbol{q}_{\mathcal{L}}$ does not change the value of $\|f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}} - c\|_2^2$ and hence $\|f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}} - c\|_2^2 = d_{\mathcal{L}}(\mathbf{f}_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2$.

Definition 7 (Integrated constraint influence)**.** Define the integrated constraint influence as the following squared norm:

$$\|c^{(n)}_{\boldsymbol{p}^*_{\mathcal{L}},\mathcal{L}} - f^{(n)}_{\mathbf{f}_{\mathcal{L}},\mathcal{L}}\|_2^2 = d_{\mathcal{L}}(\boldsymbol{p}^*_{\mathcal{L}}, \mathbf{f}_{\mathcal{L}})^2 \tag{9}$$

In the simulation studies in Section 5, this quantity will be monitored via burn-in techniques, to see how it behaves as n grows. Proposition 6 below gives the corresponding theoretical result.

Proposition 6 (Asymptotic effect of constraints)**.** *As* $n \to \infty$*, the integrated constraint influence goes to* 0*.*

Proof *Recall from Proposition 2 that* $\boldsymbol{p}_{\mathcal{L}}*$ *is the orthogonal projection of* $\mathbf{f}_{\mathcal{L}}$ *into* $C_{\mathcal{L}}$*. Since* $\boldsymbol{q}_{\mathcal{L}} \in C_{\mathcal{L}}$*, we have that* $d_{\mathcal{L}}(\mathbf{f}_{\mathcal{L}}, \boldsymbol{p}^*_{\mathcal{L}})^2 \leq d_{\mathcal{L}}(\boldsymbol{f}_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2$*, which concludes the argument by Corollary 1.* □

The consistency of the estimator is now easy to obtain.

Proposition 7 (Consistency)**.** *For c the density of the true copula, assuming the diameter of the leaves goes to* 0 *as n goes to* ∞*, the estimator* $c^{(n)}_{\boldsymbol{p}^*_{\mathcal{L}},\mathcal{L}}$ *is consistent, i.e.:*

$$\Pr\left(\lim_{n\mapsto+\infty} \|c^{(n)}_{\boldsymbol{p}^*_{\mathcal{L}},\mathcal{L}} - c\|_2^2 = 0\right) = 1$$

Proof *Remark that* $\|c^{(n)}_{\boldsymbol{p}^*_{\mathcal{L}},\mathcal{L}} - c\|_2^2 = d_{\mathcal{L}}(\boldsymbol{p}^*_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2$*. Then,* $\boldsymbol{p}^*_{\mathcal{L}}$ *being the orthogonal projection of* $\mathbf{f}_{\mathcal{L}}$ *into* $C_{\mathcal{L}}$*, and* $\boldsymbol{q}_{\mathcal{L}}$ *being in* $C_{\mathcal{L}}$*, we have* $d_{\mathcal{L}}(\boldsymbol{p}^*_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2 \leq d_{\mathcal{L}}(\mathbf{f}_{\mathcal{L}}, \boldsymbol{q}_{\mathcal{L}})^2$ *and we conclude by Corollary 1.* □

4. Bagging of density estimators

While using the CORT algorithm, overfitting is probable since we do not use any kind of pruning. Although an implementation of pruning might be possible in our setting (remove one leaf, compute new weights through the adequate quadratic program, and compare the loss in the objective function to the final loss, then design a criterion to know where to stop), we preferred to augment the 'complexity' of the model by mixing (bagging) rather than reduce it by pruning, since the model is already quite simple. Furthermore, with a minimum node size (minimum number of points in a leaf to initiate a split) set to one, the model produces an average number of leaves growing exponentially with the dimension, a lot of them being empty. The bagging procedure we propose in this section should overcome these two issues by mixing different trees fitted on different resamples of the dataset.

Two estimates $\hat{g}_1$ and $\hat{g}_2$ of a function g can be bagged into an estimate $\frac{\hat{g}_1+\hat{g}_2}{2}$, candidate for the estimation of g if the estimates are close to be uncorrelated. This principle gave rise to the bagging algorithm in regression, developed by Breiman in [6]. In density estimation, bagging can also be exploited: if the estimator has a high variance and a small bias, then bagging it might yield a better result, regarding the bias/variance trade-off.

Recall that, while bootstrapping over n observations, the chance that an observation does not appear in the bootstrap sample is given by $\left(1-\frac{1}{n}\right)^n \to \frac{1}{e}$. Hence, asymptotically 36.8 percent of the dataset will end up out-of-bag. These samples can be used to check for accuracy of the model, and even set hyper-parameters when some are needed. Following the work of [6] in regression models, [43] formalized the cross-validation and bagging process for density estimation. Note that usually, the leave-one-out method is used in kernel density estimation to select hyper-parameters (mainly the bandwidth), see, e.g., [32], but the more involved out-of-bag procedure we propose is inspired by [52].

In the following, we denote by $\hat{C}$ the empirical copula of the whole dataset $\boldsymbol{u}$, a $n \times d$ matrix.

Definition 8 (CORT Forest)**.** Define $\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(N)}$ as N bootstrap resamples of the same size n, and $\hat{c}^{(1)}, ..., \hat{c}^{(N)}$ (resp. $\hat{C}^{(1)}, ..., \hat{C}^{(N)}$) the densities (resp. d.f.) of the CORT estimators on these resamples fitted by Algorithm 2. Define the CORT forest with weights $\boldsymbol{\omega} = (\omega_1, \dots, \omega_N)$ as the mixture distribution with density:

$$\hat{c}_{\boldsymbol{\omega}}(\boldsymbol{v}) = \sum_{j=1}^{N} \omega_j \hat{c}^{(j)}(\boldsymbol{v}) \ \forall \boldsymbol{v} \in \mathbb{I}.$$

For each observation $\boldsymbol{u}_i$ in the original training set, we recall the out-of-bag density estimate as:

$$\hat{c}^{oob}_{\boldsymbol{\omega}}(\boldsymbol{u}_i) = \frac{\sum_{j=1}^{N} \omega_j \hat{c}^{(j)}(\boldsymbol{u}_i) \mathbb{1}_{\boldsymbol{u}_i \notin \boldsymbol{u}^{(j)}}}{\sum_{j=1}^{N} \omega_j \mathbb{1}_{\boldsymbol{u}_i \notin \boldsymbol{u}^{(j)}}}.$$

We define $\hat{C}_{\boldsymbol{\omega}}$ and $\hat{C}^{oob}_{\boldsymbol{\omega}}$ accordingly. Note that $\hat{c}^{oob}_{\boldsymbol{\omega}}$ is not a proper density as it may fail to sum to 1 and it is only defined on the observation points. Because trees were fitted independently, this is however, on observation points, a good approximation of the forest density itself. From $\hat{c}^{oob}_{\boldsymbol{\omega}}$, based on [32], [52] defines out-of-bag version of common fitting statistics as follows.

Definition 9 (Out-of-bag statistics)**.** Define respectively the out-of-bag empirical integrated square error, the out-of-bag Kullback-Leibler divergence and two out-of-bag Cramer-Von-Mises distances associated to the forest as:

$$\hat{J}(\hat{c}_{\boldsymbol{\omega}}) = \|\hat{c}_{\boldsymbol{\omega}}\|_2^2 - \frac{2}{n}\sum_{i=1}^{n} \hat{c}^{oob}_{\boldsymbol{\omega}}(\boldsymbol{u}_i), \qquad \hat{K}(\hat{c}_{\boldsymbol{\omega}}) = -\frac{1}{n}\sum_{i=1}^{n} \ln\left(\hat{c}^{oob}_{\boldsymbol{\omega}}(\boldsymbol{u}_i)\right)$$

$$\hat{M}(\hat{C}_{\boldsymbol{\omega}}) = \frac{1}{n}\sum_{i=1}^{n} \left(\hat{C}^{oob}_{\boldsymbol{\omega}}(\boldsymbol{u}_i) - \hat{C}(\boldsymbol{u}_i)\right)^2, \qquad \hat{N}(\hat{C}_{\boldsymbol{\omega}}) = \frac{1}{n}\sum_{i=1}^{n} \hat{C}_{\boldsymbol{\omega}}(u_i)^2 - 2\hat{C}^{oob}_{\boldsymbol{\omega}}(\boldsymbol{u}_i)\hat{C}(\boldsymbol{u}_i).$$

Note that $\hat{K}(\hat{c}_\omega)$ is obtained as an empirical version of $\int \ln\left(\frac{c(\boldsymbol{u})}{\hat{c}_\omega^{oob}(\boldsymbol{u})}\right) dC(\boldsymbol{u})$. $\hat{M}(\hat{c}_\omega)$ estimates the Cramer-Von-Mises distance between the out-of-bag d.f. of the forest and the empirical copula. On the other hand, $\hat{N}(\hat{c}_\omega)$ keeps the true norm of the model, in the same spirit as $\hat{J}(\hat{c}_\omega)$, see [32, 52] for more details about these cross-validation tools.

We denote:

$$\Omega = \left\{ \omega \in \mathbb{R}^N, \ \sum_{j=0}^{N} \omega_j = 1 \text{ and } \omega_j \geq 0 \ \forall j \in \{1, \dots, N\} \right\}$$

the set of possible weights. Indeed, we want $\hat{c}_\omega$ to be a density, which implies that weights sum to one and are positives. We call optimal the forest with the weights minimizing $\hat{J}(\hat{c}_\omega)$. We use an optimization program to find weights given the resampling and the trees:

$$\boldsymbol{\omega}^* = \underset{\omega \in \Omega}{\arg\min} \ \hat{J}(\hat{c}_\omega).$$

Note that this method makes out-of-bag observations contribute to the final estimation through weights.

The forest estimation is studied in the next section. Note that the constructions of this section are the same if you use another base estimator than the CORT estimator: in the next section, we will use these tools to compare bagging of the CORT estimator with bagging of other copula estimators.

5. Investigation of performance

In this section, we investigate the performance of the proposed estimation procedure on several simulated datasets. To ensure reproducibility, we provide all the datasets in the R package `cort`, with the code and parameters needed to re-simulate them. We will compare our results with several other models:

- Deheuvel's empirical copula, hereafter denoted by "Empirical";
- The empirical beta copula [44], hereafter denoted "Beta";
- A Checkerboard copula [10] with $m = 10$, denoted by "Cb(m=10)";
- Another less-precise Checkerboard copula with $m = 5$, denoted by "Cb(m=5)".

Recall from [10] that a checkerboard copula with a parameter m is a piecewise linear copula with a partition $\mathcal{L}$ composed of a regular grid of hypercubes with side length m^{-1}. See the reference for details about the empirical beta copula. They are all non-parametric or semi-parametric models, with a straightforward estimation procedure. The two checkerboard copulas are provided by the R package `cort`, and the empirical beta copula, as well as the empirical copula, are from the R package `copula`.

We will compare results of different models in terms of dependence measures, Kendall tau and Spearman rho first. Then, we will look at predictive performance metrics defined in the previous section, $\hat{J}$, $\hat{K}$, $\hat{M}$ and $\hat{N}$, computed via a weighted bagging of each model.

Dataset 1 (Piecewise linear copula)**.** This dataset is a simple test: we simulate random samples from a density inside the piecewise copula class, and test whether the estimator can recover it. For that, we will use a 2-dimensional sample with 500 observations, uniform on $\mathbb{I}$, and apply the following function:

$$h_1(\boldsymbol{u}) = \left(u_1, \frac{u_2 + \mathbb{1}_{u_1 \leq \frac{1}{4}} + 2\mathbb{1}_{u_1 \leq \frac{1}{2}} + \mathbb{1}_{\frac{3}{4} \leq u_1}}{4} \right).$$

To illustrate the behavior of the algorithm, we propose in Fig. 1 to look at a running example of the first few splits. Fig. 2 (a) shows a simulation from the final (fully fitted) CORT estimator on Dataset 1, along with results from the CORT forest.

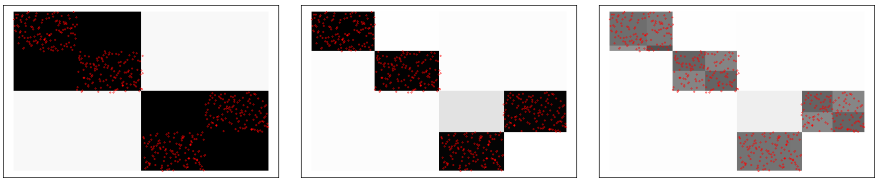
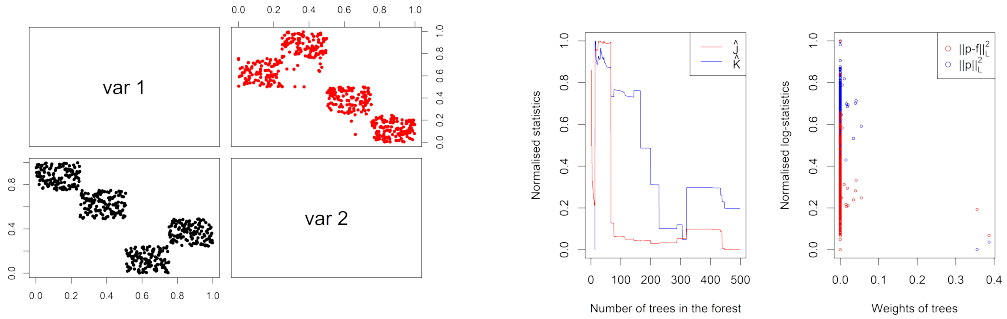


Fig. 1: (Dataset 1) Running example. Data points are in red, and darker zones mean boxes with higher weights (black is the maximum possible in each leaf). On the left, the model after the first split. In the middle, the model after the next round of splits, and on the right after the third round.



(a) **(b)**

Fig. 2: (Dataset 1) (a) The CORT estimator: in black, lower left, the input data. In red, upper-right, a simulation from the estimated tree. (b) Statistics of the forest: on the left, $\hat{K}$ and $\hat{J}$ in function of the number of trees. On the right, the Integrated Constraint Influence and square norm of each tree against the weight of the tree in the forest.

We observe on Fig. 2 (a) that the algorithm splits the space as requested. The few simulated points outside the four main boxes are there because the algorithm did not split exactly on $(\frac{1}{4}, \frac{3}{4})$, $(\frac{1}{2}, \frac{1}{2})$ and $(\frac{3}{4}, \frac{1}{4})$, and the constraints forced him to put some weight on some leaves that do not contain points.

Bagging the CORT algorithm on this dataset, we obtain statistics given by Fig. 2 (b). We observe that the out-of-bag statistics are decreasing in the number of trees fitted, although the weighting of the forest did select less than 10 trees over 500. Altogether, the algorithm succeeded into finding the right breakpoints. A comparison of the fit in terms of dependence measure to other models is available in Table 1.

Table 1: Obtained dependence measures (Kendall tau and Spearman rho) of several models on Dataset 1. The first column is the goal, others are concurrent models.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT	Bagged CORT
τ	-0.534	-0.515	-0.465	-0.527	-0.525	-0.393
ρ	-0.773	-0.757	-0.697	-0.766	-0.762	-0.604

On Table 1, we display pairwise dependence measures (Kendall τ and Spearman ρ) of the obtained fits. To read these measures, consider the first column, corresponding to the empirical copula, as the goal for other models. We

observe that all models perform correctly regarding dependence measures on this dataset, although the Checkerboard with $m = 5$ (which has a pretty rough partition, not including the $\frac{1}{4}$ multiples) has a Spearman ρ a little too high. Furthermore, bagging the CORT model gives the worst results.

Performing a standard weighted bagging procedure, we obtain fit statistics $\hat{K}, \hat{J}, \hat{M}$ and $\hat{N}$, displayed in Table 2. The experiment fits every model 500 times on resamples of the dataset and weights the resulting models to minimize $\hat{J}$.

Table 2: Results of the bagging of each model on Dataset 1. Each row represents a different performance metric: in all cases, lower is better.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT
$\hat{J}(\hat{c}_\omega)$	0.002	-3.16	-2.37	-2.98	-4.81
$\hat{K}(\hat{c}_\omega)$	Inf	-1.06	-0.743	-1.18	-0.837
$\hat{M}(\hat{c}_\omega)$	8.72e-06	4.13e-05	0.000281	1.58e-05	0.000633
$\hat{N}(\hat{c}_\omega)$	-0.0277	-0.0277	-0.0275	-0.0277	-0.0262

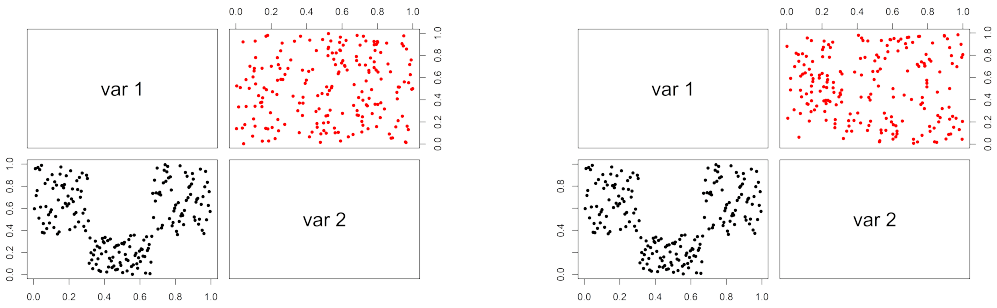
Note that the Kullback-Leibler out-of-bag divergence $\hat{K}$ is infinite for the empirical copula, since it does not assign weights to points it did not see. We observe in bagging results that the predictive performance of the checkerboards is quite poor, that the forest based on empirical Beta copulas is a lot better and that the CORT forest generalized even better regarding the density-based measures $\hat{J}$ and $\hat{K}$. Note that the bagging of empirical beta copula is a very powerful model.

Dataset 2 (Another piecewise linear copula)**.** As for Dataset 1, we simulate from a density inside the piecewise linear copula class, by applying the function:

$$h_2(\boldsymbol{u}) = \left(u_1, \frac{u_2}{2} + \frac{1}{2}\mathbb{1}_{u_1 \notin [\frac{1}{3}, \frac{2}{3})}\right)$$

to a 200×2 uniform sample, and taking ranks.

This second dataset is also in the piecewise linear class, but it splits the space in a ternary way, which the recursive splitting procedure of the CORT estimator cannot reproduce. In Fig. 3 (a), you can observe simulation from the CORT copula and the CORT forest fitted on Dataset 2.



(a) **(b)**

Fig. 3: (Dataset 2) (a) The estimated tree: in black, lower left, the input data. In red, upper-right, a simulation from the estimated tree. (b) The estimated forest: in black, lower left, the input data. In red, upper-right, a simulation from the estimated forest.

Remember that the algorithm splits recursively on only one breakpoint. If the data is split in a ternary way, as in Dataset 2, it will not succeed. Looking at details from the fitting procedure, we see that the constraints forced the

algorithm to return exactly the independence copula (by setting weights of each leaf equal to its volume). Indeed, while splitting, two splits in adjacent leaves are not synchronized: since the optimization routines are independent of each other, it is unlikely that they return the same breakpoint value for a given dimension, meaning that weights will not be transferable between the two zones: in our case, the constraints forced the weights back to independence. We see that the forest tried to correct this behavior, but the result is quite bad. The dependence measures (τ and ρ) are here structurally 0, and every model respected this correctly. Table 3 shows the same results as for the previous model.

Table 3: Results of the bagging of each model on Dataset 2. Each row represents a different performance metric: in all cases, lower is better.

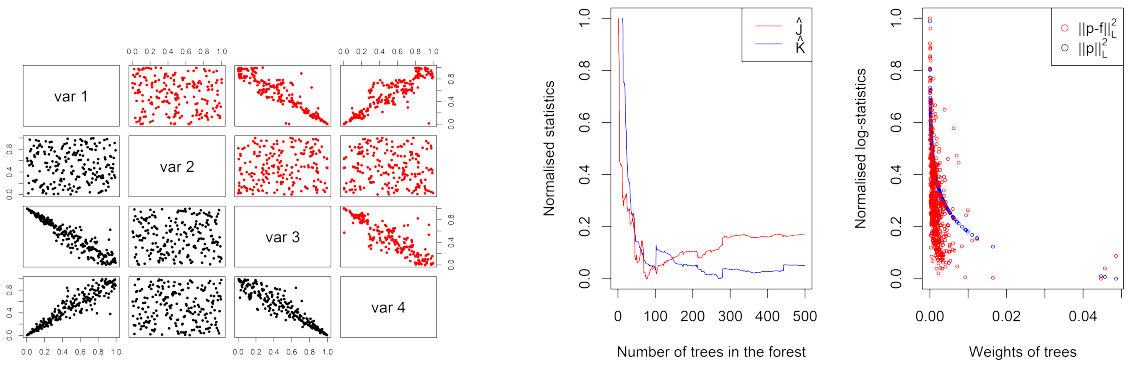
	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT
$\hat{J}(\hat{c}_\omega)$	0.00501	-1.48	-1.39	-1.21	-2.55
$\hat{K}(\hat{c}_\omega)$	Inf	-0.346	-0.271	-0.426	-0.311
$\hat{M}(\hat{c}_\omega)$	3.03e-05	5.16e-05	0.00016	7.38e-05	0.000907
$\hat{N}(\hat{c}_\omega)$	-0.134	-0.134	-0.134	-0.134	-0.134

Note that the out-of-bag ISE did not decrease during the fit of the forest, so that more complexity was added without any gain. We also observe that many trees in the forest have zero weights, they do not fit the data enough. Finally, Table 3 shows that the predictive performance of the CORT algorithm is quite bad.

We now turn ourselves to the third dataset.

Dataset 3 (Modified Clayton)**.** This dataset is a simulation of 200 points from a 3-dimensional Clayton copula [26] with $\theta = 7$ (hence highly dependent), for the first, third and fourth marginals. The second marginal is added as independent uniform draws. Lastly, the third marginal is flipped, inducing a negative dependence structure.

Dataset 3 is based on the Clayton copula, a commonly used dependence structure in many fields of application. The estimator developed in Algorithm 2 has several options: the most important one is the inclusion, or not, of the localized dimension reduction through Algorithm 1. Since here we have a completely independent dimension, this option is worth it: it reduces by a factor of 2 the number of leaves, and hence the complexity of the model, by setting the same second edge $[0, 1]$ to each leaf. Fig. 4 gives a representation of the tree and the statistics of the forest.



(a) **(b)**

Fig. 4: (Dataset 3) (a) Representation from the tree: in black, lower left, the input data. In red, upper-right, a simulation from the estimated tree. (b) Forest Statistics: on the left, $\hat{K}$ and $\hat{J}$ in function of the number of trees. On the right, the Integrated Constraint Influence and square norm of each tree against the weight of the tree in the forest.

On the left of Fig. 4 (b), the convex, decreasing shape of the Kullback-Leibler divergence with respect to the number of trees shows that the generalization error of the forest decreases with the number of trees. The decreasing trend of the constraint influence and the square norm of trees with respect to the assigned weights by the forest, on the right of Fig. 4 (b) shows how the weighting procedure selected trees. Table 4 shows dependence measures obtained from the different models.

Table 4: Obtained dependence measures of several models on Dataset 3. The first column is the goal, others are concurrent models.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT	Bagged CORT
Kendall Taus						
$\tau_{1,2}$	-0.003	0.010	0.006	-0.014	0.000	-0.020
$\tau_{1,3}$	-0.796	-0.750	-0.673	-0.799	-0.780	-0.493
$\tau_{1,4}$	0.779	0.732	0.659	0.779	0.707	0.474
$\tau_{2,3}$	0.015	0.010	0.011	0.029	0.000	0.031
$\tau_{2,4}$	-0.024	-0.009	-0.010	-0.038	0.000	-0.045
$\tau_{3,4}$	-0.775	-0.728	-0.654	-0.773	-0.695	-0.566
Spearman Rhos						
$\rho_{1,2}$	-0.005	0.013	0.010	-0.023	0.000	-0.029
$\rho_{1,3}$	-0.934	-0.915	-0.868	-0.936	-0.926	-0.648
$\rho_{1,4}$	0.924	0.903	0.857	0.925	0.872	0.626
$\rho_{2,3}$	0.023	0.014	0.016	0.045	0.000	0.047
$\rho_{2,4}$	-0.035	-0.016	-0.016	-0.057	0.000	-0.068
$\rho_{3,4}$	-0.922	-0.901	-0.853	-0.922	-0.862	-0.735

Table 5: Results of the bagging of each model on Dataset 3. Each row represents a different performance metric: in all cases, lower is better.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT
$\hat{J}(\hat{c}_\omega)$	0.00501	-9.27	-7.69	120	-54.6
$\hat{K}(\hat{c}_\omega)$	Inf	Inf	Inf	-0.582	-1.97
$\hat{M}(\hat{c}_\omega)$	2.84e-05	2.51e-05	5.25e-05	3.38e-05	9.41e-05
$\hat{N}(\hat{c}_\omega)$	-0.000666	-0.000669	-0.000641	-0.000657	-0.000639

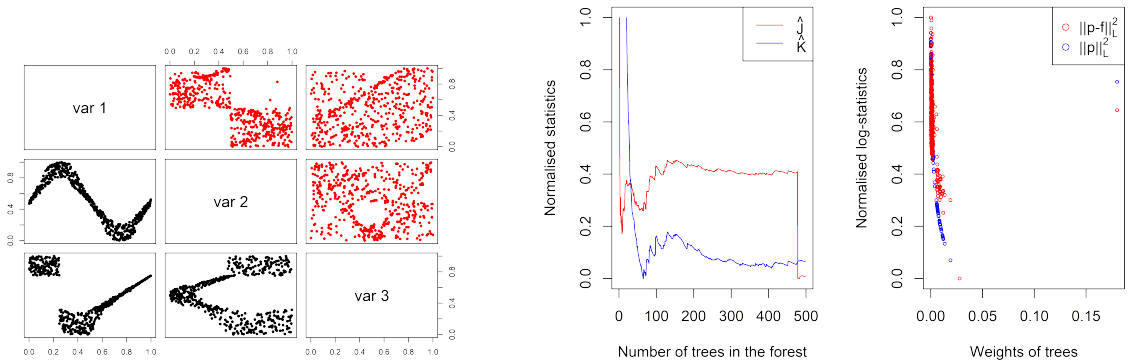
The comparison of Table 4 shows that, even if the CORT algorithm performs correctly, the forest tends to be biased in the dependence measures, toward more independence. On the other hand, the predictive performance of the model from Table 5 is really high on the density-based estimates ($\hat{J}$ and $\hat{K}$), and is less good on the distribution function based versions ($\hat{M}$ and $\hat{N}$).

The last dataset was produced based on a function h_3 defined by:

$$h_3(\boldsymbol{u}) = \left(u_1, \sin(2\pi u_1) - \frac{u_2}{\pi}, \left(1 + \frac{u_3}{\pi^2}\right)\left(\frac{u_3}{2}\mathbb{1}_{\frac{1}{4}\geq u_1} - \sin(\pi^{u_1})\mathbb{1}_{\frac{1}{4}<u_1}\right)\right).$$

Dataset 4 (Simulated functional)**.** We chose to produce a voluntarily hard to estimate dependence structure, by applying h_3 to uniformly drawn 3-dimensional random vectors. The dataset is the ranks of $(h_3(\boldsymbol{u}_i)))_{i\in 1,\dots,500}$.

Fig. 5 shows the CORT estimator and the bagging statistics on this dataset.



(a) **(b)**

Fig. 5: (Dataset 4) (a) Representation from the tree: in black, lower left, the input data. In red, upper-right, a simulation from the estimated tree. (b) Forest Statistics: On the left, $\hat{K}$ and $\hat{J}$ in function of the number of trees. On the right, the Integrated Constraint Influence and square norm of each tree against the weight of the tree in the forest.

Although the estimation on the second and third marginals is not very good, the statistics of the forests are good, showing that the estimator get better and better while adding trees. Indeed, on Fig. 5 (b), left, we observe decreasing out-of-bag errors, but we also observe on Fig. 5 (b), right, a very skewed and fat-tailed density for the constraint influence, meaning that certain trees did not produce very good partitions.

Table 6: Obtained dependence measures of several models on Dataset 4. The first column is the goal, others are concurrent models.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT	Bagged CORT
Kendall Taus						
$\tau_{1,2}$	-0.495	-0.491	-0.471	-0.500	-0.458	-0.451
$\tau_{1,3}$	0.089	0.042	-0.002	0.084	0.171	0.048
$\tau_{2,3}$	0.005	0.015	0.004	0.007	-0.109	0.013
Spearman Rhos						
$\rho_{1,2}$	-0.743	-0.737	-0.715	-0.747	-0.713	-0.674
$\rho_{1,3}$	-0.156	-0.154	-0.146	-0.159	0.246	0.060
$\rho_{2,3}$	-0.019	0.012	-0.001	-0.010	-0.177	0.020

Table 6 shows that the bivariate projections were not all treated as well as others by the CORT algorithm: the values of $\tau_{1,2}$ and $\rho_{1,2}$ are surprisingly quite good compared to $\tau_{1,3}, \tau_{2,3}, \rho_{1,3}, \rho_{2,3}$, for the CORT estimator. Hopefully, the bagging corrects this bias quite correctly.

Table 7: Results of the bagging of each model on Dataset 4. Each row represents a different performance metric: in all cases, lower is better.

	Empirical	Cb(m=10)	Cb(m=5)	Beta	CORT
$\hat{J}(\hat{c}_\omega)$	0.002	-15.7	-7.22	-23	-23.3
$\hat{K}(\hat{c}_\omega)$	Inf	-2.52	-1.79	-3.12	-1.72
$\hat{M}(\hat{c}_\omega)$	1.69e-05	0.000172	0.000736	4.76e-05	0.00233
$\hat{N}(\hat{c}_\omega)$	-0.0201	-0.0199	-0.0194	-0.0201	-0.0177

Finally, the predictive performance from Table 7 is still two-sided: the density-based results are quite good, but the distribution-function based ones are not very good.

More details and experiments are available in the supplementary section.

6. Conclusion

From a simple density estimation procedure designed by [42], we constructed a piecewise constant, tree-shaped, recursive copula density estimator. We computed several closed-form expressions for this estimator, and we gave an asymptotic result.

If, intuitively, constraining the space of potential weighting solutions will help the convergence of optimization routines, the copula constraints forced us to split the space in more than one dimension, making the resulting estimation procedure complex with the increasing dimension. The localized dimension reduction procedure helps to reduce the complexity.

The CORT estimator has good generalization performance and is straightforward to use since it does not have restrictive hypothesis on the true dependence structure. Although the implementation we provide is very fast, a balance between computation time and precision is available in the number of trees used in the bagging procedure. However, more work needs to be done to correct defaults of the splitting procedure, which is not able to understand certain kinds of dependence structures.

Acknowledgments

We are particularly grateful to the two referees, the associate editor and the editor who all made useful comments and suggestions, improving the manuscript. We would like to thank SCOR SE for funding this work through a CIFRE grant. Any errors are ours.

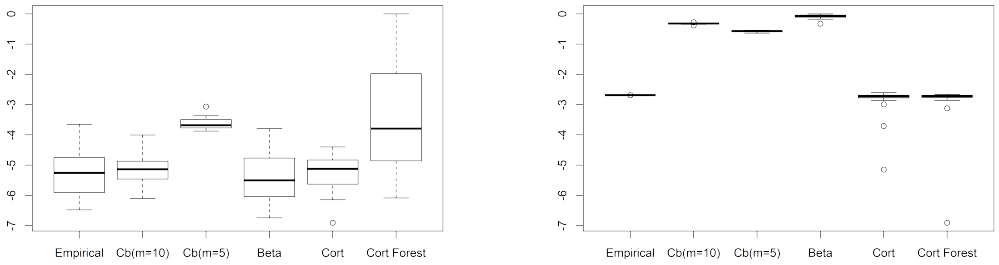
Supplementary material

On each dataset from Section 5, we ran an additional experiment. To observe the predictive performance of each model, we designed a cross-validation procedure: on 20 resamples of each dataset, we computed the Cramer-Von-Mises and ISE errors on test samples, given respectively by:

$$\hat{P}(\hat{D}) = \sum_{i \in T} \left(\hat{D}(\boldsymbol{u}_i) - \hat{C}(\boldsymbol{u}_i) \right)^2, \qquad \hat{Q}(\hat{d}) = \|\hat{d}\|_2^2 - \frac{2}{|T|} \sum_{i \in T} \hat{d}(\boldsymbol{u}_i),$$

where T is the test dataset used to obtain $\hat{D}$, a given copula estimator with density $\hat{d}$. Note that $\hat{P}$ is focused on the distribution function and $\hat{Q}$ on the density. Below are the resulting box plots for each of the datasets.

Fig. 6 gives a box plot of $\hat{P}$ Cramer-Von-Mises errors on the first dataset.

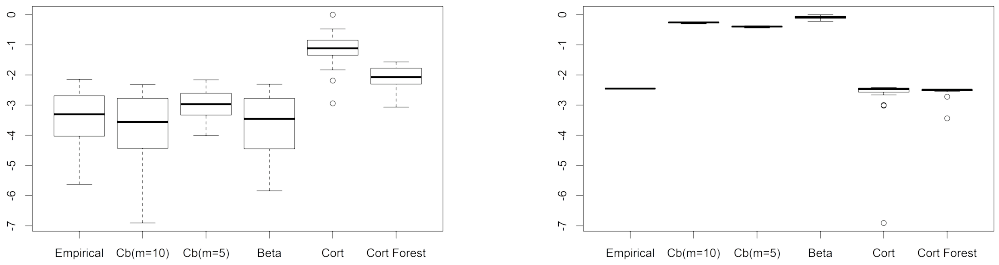


(a) Box plot of log-normalized $\hat{P}$

(b) Box plot of log-normalized $\hat{Q}$

Fig. 6: (Dataset 1) Box plots of resulting errors for 20 resamples for each model (the lower the better). $\hat{P}$ is focused on the distribution function and $\hat{Q}$ on the density.

On Fig. 6, note that smaller values of $\hat{P}$ (d.f. based) and $\hat{Q}$ (density based) mean a better model. We observe that, although the bagging procedure is not worth it, the CORT estimator is very good on this example, both for density and d.f. estimation. Unfortunately, it is not always the case, as shown by the experiments we did on Dataset 2.



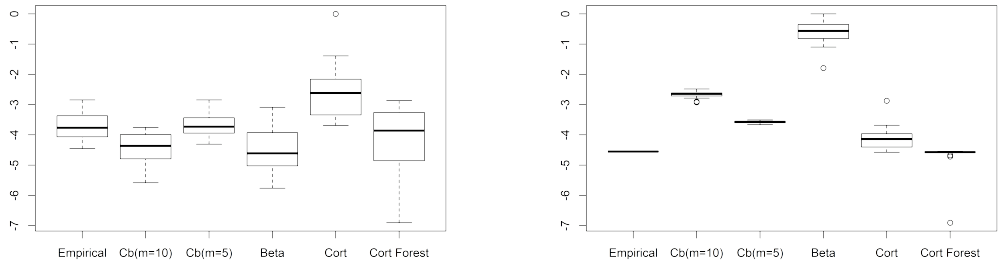
(a) Box plot of log-normalized $\hat{P}$

(b) Box plot of log-normalized $\hat{Q}$

Fig. 7: (Dataset 2) Box plots of resulting errors for 20 resamples for each model (the lower the better). $\hat{P}$ is focused on the distribution function and $\hat{Q}$ on the density.

The box plot of $\hat{P}$ error on Dataset 2 confirms the analysis we already had: in this case, the empirical beta copula performs a lot better.

For Dataset 3, however, the box plot of $\hat{P}$ and $\hat{Q}$ in Fig. 8 confirms the performance of the estimator:



(a) Box plot of log-normalized $\hat{P}$

(b) Box plot of log-normalized $\hat{Q}$

Fig. 8: (Dataset 3) Box plots of resulting errors for 20 resamples for each model (the lower the better). $\hat{P}$ is focused on the distribution function and $\hat{Q}$ on the density.

To understand more precisely what happened with Kendall's tau and Spearman's rho on Dataset 3, we ran a burn-in experiment: we fitted trees on subsamples of increasing size. We can then observe the burn in Kendall taus and Spearman rhos, represented on Figure 9.

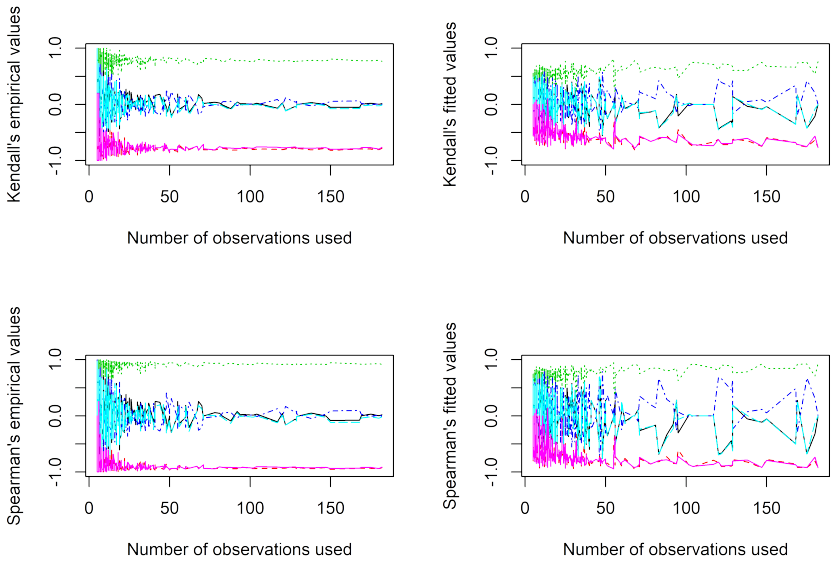
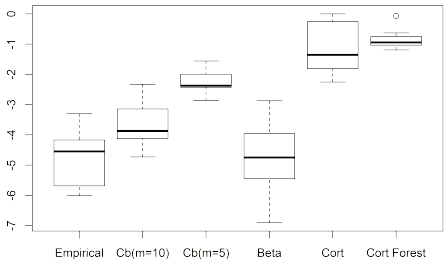


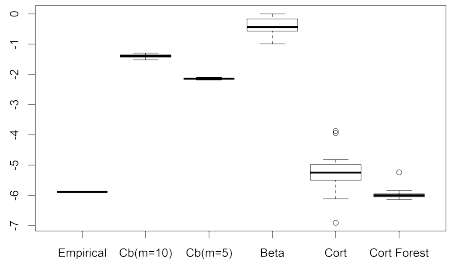
Fig. 9: (Dataset 3) Burn in of dependence measures: on the top row, Kendall's tau; on the bottom row, Spearman's rho; on the left column, empirical values from subsamples of increasing sizes; on the right column, values obtained by the fitted CORT estimators on the same subsamples. The size of the subsamples is in abscissa. Each line type corresponds to a couple of variables.

We see that the fitted values, on the right of Fig. 9, are convergent but biased compared to the empirical observed values of the dependence measures directly computed on resamples of the dataset, on the left. We also observe the high variance of the estimator, which is one of the good reasons to use a bagging procedure.

Finally, the box plots on Dataset 4 are given in Fig. 10.



(a) Box plot of log-normalized $\hat{P}$



(b) Box plot of log-normalized $\hat{Q}$

Fig. 10: (Dataset 4) Box plots of resulting errors for 20 resamples for each model (the lower the better). $\hat{P}$ is focused on the distribution function and $\hat{Q}$ on the density.

Proof of Proposition 1. We have:

$$\begin{aligned}\int c_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u})\, C_{\boldsymbol{p},\mathcal{L}}(\boldsymbol{u})\, d\boldsymbol{u} &= \sum_{\ell\in\mathcal{L}}\sum_{k\in\mathcal{L}} \frac{p_\ell p_k}{\lambda(\ell)\lambda(k)} \int_\ell \lambda([0,\boldsymbol{u}]\cap k)\, d\boldsymbol{u} \\ &= \sum_{\ell\in\mathcal{L}}\sum_{k\in\mathcal{L}} \frac{p_\ell p_k}{\lambda(\ell)\lambda(k)} \prod_{i=1}^{d} \int_{\ell_i} \lambda([0,u_i]\cap k_i)\, du_i\end{aligned}$$

where ℓ_i denotes the projection of ℓ onto the dimension i. Denote now $g(u,c,d) = \lambda([0,u]\cap[c,d])$, and remark that:

$$g(u,c,d) = \lambda([0,u]\cap[c,d]) = \begin{cases} 0, & u \le c \\ u-c, & c < u < d \\ d-c, & d \le u \end{cases}$$

Denote furthermore $G(u,a,b,c,d) = \int_a^b g(u,c,d)du$. We have:

$$\begin{aligned} G(u,a,b,c,d) = \int_a^b 0\,\mathbb{1}_{u<c}\,du \quad &+ \int_a^b (u-c)\,\mathbb{1}_{c<u<d}\,du \quad + \int_a^b (d-c)\,\mathbb{1}_{d<u}\,du \\ = \quad &\int_{a\vee c}^{b\wedge d} (u-c)\,\mathbb{1}_{c<u<d}\,du \quad + \int_{a\vee d}^{b} (d-c)\,\mathbb{1}_{d<u}\,du \\ = \quad &\left(\frac{(b\wedge d)^2}{2} - (b\wedge d)c\right) - \left(\frac{(a\vee c)^2}{2} - (a\vee c)c\right) \quad + (d-c)\,(b-(a\vee d))\,, \end{aligned}$$

where the two integrals are simply integrals of linear functions, regarding the univariate slack variable u. Hence,

$$G(u,a,b,c,d) = \frac{(b\wedge d)^2 - (a\vee c)^2}{2} + c\left((a\vee c) - (b\wedge d)\right) + (d-c)\left(b - (a\vee d)\right),$$

which provides the wanted expression for τ. For ρ, one only needs to compute the integral expression:

$$\begin{aligned}
\int C_{\boldsymbol{p},\mathcal{L}}(u)\,du &= \sum_{\ell\in\mathcal{L}} \frac{p_\ell}{\lambda(\ell)} \int \lambda([0,\boldsymbol{u}]\cap\ell)\,du \\
&= \sum_{\ell\in\mathcal{L}} \frac{p_\ell}{\lambda(\ell)} \prod_{i=1}^{d} \int_0^{a_i} 0\,du_i + \int_{a_i}^{b_i} (u_i - a_i)\,du_i + \int_{b_i}^{1} (b_i - a_i)\,du_i \\
&= \sum_{\ell\in\mathcal{L}} \frac{p_\ell}{\lambda(\ell)} \prod_{i=1}^{d} 0 + \frac{b_i^2 - a_i^2}{2} - (b_i - a_i)a_i + (1-b_i)(b_i - a_i) \\
&= \sum_{\ell\in\mathcal{L}} \frac{p_\ell}{\lambda(\ell)} \prod_{i=1}^{d} \frac{1}{2}(b_i - a_i)(2 - b_i - a_i)
\end{aligned}$$

which concludes the argument since $\lambda(\ell) = \prod_{i=1}^{d}(b_i - a_i)$.

□

References

[1] P. Alquier, Density estimation with quadratic loss: A confidence intervals method, ESAIM: Probability and Statistics 12 (2008) 438–463.
[2] L. Anderlini, Density Estimation Trees as fast non-parametric modelling tools, Journal of Physics: Conference Series 762 (2016) 012042.
[3] L. Birgé, Model selection for density estimation with L2-loss, arXiv:0808.1416 [math, stat] (2013).
[4] J.-D. Boissonnat, O. Devillers, K. Dutta, M. Glisse, Randomized incremental construction of Delaunay triangulations of nice point sets (2019) 28.
[5] A. Bowman, Density based tests for goodness-of-fit, Journal of Statistical Computation and Simulation 40 (1992) 1–13.
[6] L. Breiman, Out-of-bag estimation (1996).
[7] C. Cervellera, D. Macciò, Voronoi tree models for distribution-preserving sampling and generation, Pattern Recognition 97 (2020) 107002.
[8] S. X. Chen, T.-M. Huang, Nonparametric estimation of copula functions for dependence modelling, Canadian Journal of Statistics 35 (2007) 265–282.
[9] A. Criminisi, J. Shotton, E. Konukoglu, Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning, Foundations and Trends® in Computer Graphics and Vision 7 (2012) 81–227.
[10] A. Cuberos, E. Masiello, V. Maume-Deschamps, Copulas checker-type approximations: Application to quantiles estimation of sums of dependent random variables, Communications in Statistics - Theory and Methods (2019) 1–19.
[11] E. de Amo, M. Díaz Carrillo, F. Durante, J. Fernández Sánchez, Extensions of subcopulas, Journal of Mathematical Analysis and Applications 452 (2017) 1–15.
[12] T. De Wet, Cramér-von Mises tests for independence, Journal of Multivariate Analysis 10 (1980) 38–50.
[13] F. Durante, J. Fernández-Sánchez, J. J. Quesada-Molina, M. Úbeda-Flores, Convergence results for patchwork copulas, European Journal of Operational Research 247 (2015) 525–531.
[14] F. Durante, J. Fernández Sánchez, C. Sempi, Multivariate patchwork copulas: A unified approach with applications to partial comonotonicity, Insurance: Mathematics and Economics 53 (2013) 897–905.
[15] F. Durante, C. Sempi, Principles of Copula Theory, Chapman and Hall/CRC, 2015.
[16] J.-D. Fermanian, D. Radulovic, M. Wegkamp, Weak convergence of empirical copula processes, Bernoulli 10 (2004) 847–860.
[17] J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (2008) 432–441.
[18] M. Gavish, B. Nadler, R. R. Coifman, Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning., in: ICML, pp. 367–374.
[19] D. Geiß, R. Klein, R. Penninger, G. Rote, Optimally solving a transportation problem using Voronoi diagrams, Computational Geometry 46 (2013) 1009–1016.

[20] C. Genest, E. Masiello, K. Tribouley, Copula Density Estimation By Wavelet Methods (2012) 2.
[21] C. Genest, J. G. Nešlehová, B. Rémillard, O. A. Murphy, Testing for independence in arbitrary distributions, Biometrika 106 (2019) 47–68.
[22] S. T. Goh, C. Rudin, Cascaded High Dimensional Histograms: A Generative Approach to Density Estimation, arXiv:1510.06779 [stat] (2015).
[23] F. J. Hickernell, A generalized discrepancy and quadrature error bound, Mathematics of Computation of the American Mathematical Society 67 (1998) 299–322.
[24] M. Hofert, M. Mächler, A Graphical Goodness-of-Fit Test for Dependence Models in Higher Dimensions, Journal of Computational and Graphical Statistics 23 (2014) 700–716.
[25] S. Hornus, J.-D. Boissonnat, An efficient implementation of Delaunay triangulations in medium dimensions, Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique 37 (1992) 55–57.
[26] H. Joe, Dependence Modeling with Copulas, Chapman and Hall/CRC, 2014.
[27] D. Li, K. Yang, W. H. Wong, Density Estimation via Discrepancy Based Adaptive Sequential Partition, arXiv:1404.1425 [stat] (2018).
[28] Y. Li, X. Liu, F. Liu, PANDA: AdaPtive Noisy Data Augmentation for Regularization of Undirected Graphical Models, arXiv:1810.04851 [cs, stat] (2018).
[29] L. Lin, M. Drton, A. Shojaie, Estimation of high-dimensional graphical models using regularized score matching, Electronic journal of statistics 10 (2016) 806.
[30] E. Luini, P. Arbenz, Density estimation of multivariate samples using Wasserstein distance, Journal of Statistical Computation and Simulation 90 (2020) 181–210.
[31] D. W. Meyer, Density Estimation with Distribution Element Trees, Statistics and Computing 28 (2018) 609–632.
[32] J. C. Miecznikowski, D. Wang, A. Hutson, Bootstrap MISE estimators to obtain bandwidth for kernel density estimation, Communications in Statistics-Simulation and Computation 39 (2010) 1455–1469.
[33] P. A. Morettin, C. M. Toloi, C. Chiann, J. C. de Miranda, Wavelet-smoothed empirical copula estimators, Revista Brasileira de Finanças 8 (2010) 263–281.
[34] D. Müller, C. Czado, Selection of sparse vine copulas in high dimensions with the Lasso, Statistics and Computing 29 (2019) 269–287.
[35] D. T. Müller, Selection of Sparse Vine Copulas in Ultra High Dimensions, Ph.D. thesis, Technische Universität München, 2017.
[36] T. Nagler, Nonparametric Estimation in Simplified Vine Copula Models, Ph.D. thesis, Technische Universität München, 2018.
[37] T. Nagler, C. Czado, Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas, Journal of Multivariate Analysis 151 (2016) 69–89.
[38] R. B. Nelsen, An Introduction to Copulas, Springer Series in Statistics, Springer, New York, 2nd ed edition, 2006.
[39] S. Ning, N. Shephard, A nonparametric Bayesian approach to copula estimation, Journal of Statistical Computation and Simulation 88 (2018) 1081–1105.
[40] O. Okhrin, A. Ristig, Y.-F. Xu, Copulae in High Dimensions: An Introduction, in: W. K. Härdle, C. Y.-H. Chen, L. Overbeck (Eds.), Applied Quantitative Finance, Springer Berlin Heidelberg, Berlin, Heidelberg, 2017, pp. 247–277.
[41] B. Peherstorfer, D. Pflüge, H.-J. Bungartz, Density Estimation with Adaptive Sparse Grids for Large Data Sets, in: Proceedings of the 2014 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2014, pp. 443–451.
[42] P. Ram, A. G. Gray, Density estimation trees, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '11, ACM Press, San Diego, California, USA, 2011, p. 627.
[43] S. R. Sain, K. A. Baggerly, D. W. Scott, Cross-validation of multivariate densities, Journal of the American Statistical Association 89 (1994) 807–817.
[44] J. Segers, M. Sibuya, H. Tsukahara, The Empirical Beta Copula, Journal of Multivariate Analysis 155 (2017) 35–51.
[45] I. Siloko, C. Ishiekwene, Boosting and bagging in kernel density estimation, The Nigerian Journal of Science and Environment 14 (2016) 32–37.
[46] I. Siloko, C. Ishiekwene, F. Oyegue, New gradient methods for bandwidth selection in bivariate kernel density estimation, Mathematics and Statistics 6 (2018) 1–8.
[47] I. Siloko, E. Siloko, O. Ikpotokin, C. Ishiekwene, B. Afere, On asymptotic mean integrated squared error's reduction techniques in kernel density estimation, International Journal of Computational and Theoretical Statistics 6 (2019).
[48] A. Sklar, Fonctions de repartition à n dimension et leurs marges, Université Paris 8 (1959) 1–3.
[49] K. Song, Testing conditional independence via Rosenblatt transforms, The Annals of Statistics 37 (2009) 4011–4045.
[50] V. N. Vapnik, A. Y. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, in: Measures of Complexity, Springer, 2015, pp. 11–30.
[51] D. F. Watson, Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, The computer journal 24 (1981) 167–172.
[52] K. Wu, W. Hou, H. Yang, Density estimation via the random forest method, Communications in Statistics-Theory and Methods 47 (2018) 877–889.
[53] M. Yang, R. Modarres, Multivariate tests of uniformity, Statistical Papers 58 (2017) 627–639.
[54] S. Yao, X. Zhang, X. Shao, Testing mutual independence in high dimension via distance covariance, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80 (2018) 455–480.