HAL open science

Estimation of high-order moment-independent importance measures for Shapley value analysis

Gabriel Sarazin, Pierre Derennes, Jérôme Morio

To cite this version:

Gabriel Sarazin, Pierre Derennes, Jérôme Morio. Estimation of high-order moment-independent importance measures for Shapley value analysis. Applied Mathematical Modelling, 2020, 88, pp.396-417. 10.1016/j.apm.2020.06.036 . hal-02912961

HAL Id: hal-02912961

https://hal.science/hal-02912961v1

Submitted on 7 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimation of high-order moment-independent importance measures for Shapley value analysis

Gabriel Sarazin[a,b], Pierre Derennes[a,b], Jérôme Morio[b,*]

[a] *Université de Toulouse, UPS IMT, F-31062 Toulouse Cedex 9, France*
[b] *ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France*

Abstract

Moment-independent importance measures are increasingly used by practitioners to understand how output uncertainty may be shared between a set of stochastic inputs. Computing Borgonovo's sensitivity indices for a large group of inputs is still a challenging problem due to the curse of dimensionality and it is addressed in this article. An estimation scheme taking the most of recent developments in copula theory is developed. Furthermore, the concept of Shapley value is used to derive new sensitivity indices, which makes the interpretation of Borgonovo's indices much easier. The resulting importance measure offers a double advantage compared with other existing methods since it allows to quantify the impact exerted by one input variable on the whole output distribution after taking into account all possible dependencies and interactions with other variables. The validity of the proposed methodology is established on several analytical examples and the benefits in terms of computational efficiency are illustrated with real-life test cases such as the study of the water flow through a borehole. In addition, a detailed case study dealing with the atmospheric re-entry of a launcher first stage is completed.

Keywords: Global sensitivity analysis, input-output models, dependent inputs, moment-independent importance measures, multivariate copula distributions, Shapley value.

1. Introduction

In diverse disciplines, system modeling is often achieved by considering a black-box model for which the observation is expressed as a deterministic function of external parameters representing some physical variables. These basic variables are usually assumed random in order to take phenomenological uncertainties into account. Then, sensitivity analysis (SA) techniques play a crucial role in the handling of these uncertainties and in the comprehension of the system behavior. These techniques aim at ranking inputs according to the impact they have on the output distribution. In addition, two other objectives are pursued: (a) targeting the input

*Corresponding author
Email addresses: `Gabriel.Sarazin@onera.fr` (Gabriel Sarazin), `Pierre.Derennes@onera.fr` (Pierre Derennes), `Jerome.Morio@onera.fr` (Jérôme Morio)

variables that most reduce the output variability when they are set to prescribed values, and (b) simplifying the model complexity by omitting the contribution of low influential inputs. The influence criterion depends on the considered SA approach. There are various SA techniques in the literature and two essential families emerge with a clear separation between local and global methods. The reader is referred to [1, 2] and associated references for a review. Local methods aim at studying the behavior of the output in the vicinity of specified input nominal values. In contrast, global sensitivity analysis (GSA) consider the whole variation range of input variables.

This paper focuses on the GSA method suggested by Borgonovo [3], which is a more and more popular alternative to the variance-based Sobol approach. Indeed, Borgonovo's methodology relies on considering moment-independent global sensitivity indices (also called δ-sensitivity measures) which take the entire output distribution into account, by opposition to Sobol indices that only seek to explain how the second-order moment of the output variable is impacted by input uncertainties. It is important to distinguish between the first-order and high-order indices. First-order indices measure the influence, in the sense of Borgonovo, of any single input variable which is isolated from others. On the contrary, when Borgonovo's importance measure is applied to a larger set of r inputs, an r-order sensitivity index is derived and it accounts for the joint impact of the whole r-dimensional set of variables. In practice, many studies settle for estimating first-order indices and draw some conclusions from them even if this can be misleading. In fact, if the first-order index associated to a given input is very small, two situations may be envisioned. On the one hand, the input variable may actually be of very small influence. On the other hand, this variable can become influential once regrouped with others because of a dependence pattern that was not unclosed by the computation of first-order indices.

The estimation problem in the case of first-order indices has been investigated in many works. The PDF-based methods [3, 4, 5] attracted most attention since they stem from the conceptual ideas of the definition. However, as they lead to implement a double-loop estimation algorithm, other techniques were developed such as the CDF-based estimator [6], the single-loop estimator [7] and the importance sampling estimator [8]. Another approach is put forth in [9] and relies on the Nataf transform, that is on the assumption that the joint distribution between any group of inputs and the output incorporates a Gaussian copula. When the latter situation is not justified, a copula-based estimator may be employed instead. It consists in learning a copula distribution with either an appropriate kernel family [10] or the maximum entropy method [11].

As far as high-order Borgonovo's indices are concerned, estimation is harder to perform. The double-loop method [3] is still applicable but it is illusory to think that such an expensive estimation scheme can be used for both first-order and high-order indices. Most alternatives to the double-loop approach depend on techniques that do not cope with the curse of dimensionality. The estimator based on the prior application of Nataf transform can be seamlessly generalized to any high-order index but the Gaussian assumption becomes increasingly controversial if it is made without proper validation. Extending the copula-based estimator to high-order indices needs to be considered by fully exploiting recent breakthroughs

in dependence modelling [12].

In this context, the contribution of this paper is twofold. Firstly, the authors developed an estimation scheme for high-order indices where the copula-based expression of Borgonovo's indices is coupled with numerical procedures for copula density estimation in high dimension [13]. Taking the most of copula distribution models in order to refine the estimation of Borgonovo's δ-sensitivity measures has been partly addressed in [14], but only within a very particular context where Y is multidimensional. In addition, this previous work focuses only on the specific case of parametric regular vine copulas whereas the present paper includes a comparative study of the estimators resulting from different copula distribution models. Secondly, for ease of interpretation, the authors come up with a new importance measure which appraises the overall impact of each input variable on the output distribution. If one can estimate δ-sensitivity measures at all orders, $2^d - 1$ indices have to be taken into account, where d denotes the problem dimension. It is thus necessary to synthesize the information provided by this huge set of Borgonovo's indices. In a similar way to what was proposed for Sobol indices in [15], the present article considers the notion of Shapley effects as an added value to sum up Borgonovo's indices. Shapley values were initially introduced in cooperative game theory to share total gains between players with a fair tribute to all fruitful collaborations. They were later used in sensitivity analysis where they prove to be a simple solution to remove the conceptual problem of dependent inputs.

The remainder of this article is organized as follows. Section 2 briefly reviews Borgonovo's δ-sensitivity measures and describes the estimation method due to connections with copula theory. Section 3 surveys different alternatives to learn copula distributions in high dimension. Furthermore, the definition of Shapley values as well as their application in the framework of Borgonovo's GSA method is discussed in Section 4. The different results developed in this paper are illustrated with numerical test cases in Section 5.

2. Borgonovo's moment-independent importance measures

This work is made in the context of a general input-output model $Y = \mathcal{M}(\boldsymbol{X})$ where $\mathcal{M} : \mathbb{R}^d \to \mathbb{R}$ is a deterministic function which represents a computationally expensive simulation code but is considered as a black box. The scalar output variable Y is computed from a random vector $\boldsymbol{X} = (X_1, \ldots, X_d)$ lying in $\mathbb{R}^d$. As extracting subvectors from $\boldsymbol{X}$ is a task repeated throughout this work, the following notations are introduced:

- $\mathcal{P}_d$ is the power set of $[d] := \{1, \ldots, d\}$.
- If $\boldsymbol{u} = (u_1, \ldots, u_r) \in \mathcal{P}_d$, $|\boldsymbol{u}| = r$ and $\boldsymbol{X}_{\boldsymbol{u}} = (X_{u_1}, \ldots, X_{u_r})$.
- The absolute complement of $\boldsymbol{u}$ is denoted by $\bar{\boldsymbol{u}}$.
- $\mathcal{H}_r = [0\,,1]^r$ is the hypercube associated to $\mathbb{R}^r$.

In addition, it is assumed that all manipulated random vectors are absolutely continuous with respect to Lebesgue measure. The PDFs of $\boldsymbol{X}$, $\boldsymbol{X}_{\boldsymbol{u}}$ and Y are respectively denoted by $f_{\boldsymbol{X}}$, $f_{\boldsymbol{X}_{\boldsymbol{u}}}$ and f_Y.

2.1. Definition of Borgonovo's sensitivity indices

The moment-independent SA method, first introduced by [16], focuses on finding inputs (or groups of inputs) that, if fixed within their distribution ranges, lead to the most significant modification of the entire output distribution. For any given set of values $\boldsymbol{x_u} \in \mathbb{R}^r$, the discrepancy between the unconditional output PDF f_Y and its conditional version $f_{Y|\boldsymbol{X_u}=\boldsymbol{x_u}}$ is measured through the L^1-norm of their difference. This number is referred to as the shift $s(\boldsymbol{x_u})$ and, graphically speaking, is nothing but the area enclosed between the two representative curves:

$$s(x_{\boldsymbol{u}}) = \left\| f_Y - f_{Y|\boldsymbol{X_u}=\boldsymbol{x_u}} \right\|_{L^1(\mathbb{R})} = \int_{\mathbb{R}} \left| f_Y(y) - f_{Y|\boldsymbol{X_u}=\boldsymbol{x_u}}(y) \right| \mathrm{d}y. \tag{1}$$

So as to consider the whole range of values $\boldsymbol{X_u}$ may take, Borgonovo's δ-sensitivity measure was defined as the normalized expectation of $s(\boldsymbol{X_u})$:

$$\delta_{\boldsymbol{u}}^Y = \frac{1}{2}\mathbb{E}\left[s(\boldsymbol{X_u})\right] = \frac{1}{2}\mathbb{E}\left[\left\| f_Y - f_{Y|\boldsymbol{X_u}} \right\|_{L^1(\mathbb{R})} \right]. \tag{2}$$

For the sake of brevity, if there is no point of ambiguity, denoting $\delta_{\boldsymbol{u}}$ will be preferred to $\delta_{\boldsymbol{u}}^Y$. Unlike variance-based sensitivity measures such as Sobol indices, Borgonovo's δ-sensitivity measures attempt to take into account the entire variability of the output distribution f_Y instead of only focusing on how $\mathbb{V}(Y)$ is impacted. Moreover, and it is perhaps one of their most valuable advantages over Sobol indices, they are perfectly suited for non-independent input variables. The properties verified by this family of importance measures were thoroughly investigated and the most essential results are recalled in the proposition below.

Proposition 2.1 ($\boldsymbol{\delta}$-sensitivity measures)

- $\forall \boldsymbol{u} \in \mathcal{P}_d\,, 0 \leqslant \delta_{\boldsymbol{u}} \leqslant 1$.
- *The total index* $\delta_{[d]}$ *is equal to* 1.
- *Let* $\boldsymbol{v} \in \mathcal{P}_d$ *be such that* $\boldsymbol{v} \subset \boldsymbol{u}$. *Then, one has always* $\delta_{\boldsymbol{v}} \leqslant \delta_{\boldsymbol{u}}$.
- *All indices* $\delta_{\boldsymbol{u}}$ *are invariant under monotonic transformations.*
  If $\tilde{Y} = \varphi(\boldsymbol{X})$ *with* φ *a* C^1*-diffeomorphism,* $\delta_{\boldsymbol{u}}^Y = \delta_{\boldsymbol{u}}^{\tilde{Y}}$ *for any* $\boldsymbol{u} \in \mathcal{P}_d$.

The three first points are demonstrated in [3] while the last one is addressed in [4]. Estimating δ-sensitivity measures has been actively explored over the last decade and the forthcoming subsections aim at introducing the two estimation strategies that will be carried out throughout the paper.

2.2. The double-loop estimation method

A very intuitive estimation method stemming from (2) consists in replacing the expectation operator by its empirical counterpart, which gives:

$$\hat{\delta}_{\boldsymbol{u}} = \frac{1}{2N} \sum_{n=1}^{N} \left\| f_Y(\cdot) - f_{Y|\boldsymbol{X_u}}\left(\cdot \mid \boldsymbol{X}_{\boldsymbol{u}}^n\right) \right\|_{L^1(\mathbb{R})} \tag{3}$$

where:

$$\left\{\boldsymbol{X}_{\boldsymbol{u}}^n = \left(X_{u_1}^n, \ldots, X_{u_r}^n\right)\right\}_{1 \leqslant n \leqslant N} \overset{\text{i.i.d.}}{\sim} f_{\boldsymbol{X}_{\boldsymbol{u}}} \ . \tag{4}$$

As no prior information is available on the output distribution, any PDF involving Y is unknown and needs to be estimated before summing L^1-norms. On the one hand, estimating f_Y seems fairly straightforward provided that one is given a set of input observations $\boldsymbol{X}^k \overset{\text{i.i.d.}}{\sim} f_{\boldsymbol{X}}$. The associated outputs Y^k are computed with $\mathcal{M}$ and kernel density estimation (KDE) yields $\hat{f}_Y$:

$$\hat{f}_Y(y) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{h}K\left(\frac{y - Y^k}{h}\right) \text{ with } K(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2} \tag{5}$$

where h is called the bandwidth and regulates how smooth the resulting curve $\hat{f}_Y$ is. The kernel method is discussed with full details in [17] while a review of fully automatic bandwidth selectors can be found in [18]. On the other hand, and this is not surprising, estimating the conditional distributions $f_{Y|\boldsymbol{X}_{\boldsymbol{u}}}$ is a much more thorny issue since it requires being able to sample Y knowing that $\boldsymbol{X}_{\boldsymbol{u}}$ is constrained to take prescribed values. For any $\boldsymbol{x}_{\boldsymbol{u}} \in \mathbb{R}^r$, random generation according to $Y \mid \boldsymbol{X}_{\boldsymbol{u}} = \boldsymbol{x}_{\boldsymbol{u}}$ may be achieved as follows:

1. Generate samples $\boldsymbol{X}_{\bar{\boldsymbol{u}}}^k$ according to $\boldsymbol{X}_{\bar{\boldsymbol{u}}} \mid \boldsymbol{X}_{\boldsymbol{u}} = \boldsymbol{x}_{\boldsymbol{u}}$. It is not something trivial in the general case but two particular cases are worth mentioning. First, if $\boldsymbol{X}_{\boldsymbol{u}}$ and $\boldsymbol{X}_{\bar{\boldsymbol{u}}}$ are independent, it amounts to simulate according to $f_{\boldsymbol{X}_{\bar{\boldsymbol{u}}}}$. Secondly, in the Gaussian case, the conditional distribution $f_{\boldsymbol{X}_{\bar{\boldsymbol{u}}}|\boldsymbol{X}_{\boldsymbol{u}}}$ can be derived easily, as explained in [19].
2. Compute the associated outputs $Y_{\boldsymbol{x}_{\boldsymbol{u}}}^k = \mathcal{M}\left(\boldsymbol{x}_{\boldsymbol{u}}, \boldsymbol{X}_{\bar{\boldsymbol{u}}}^k\right)$.

If one succeeds in drawing samples from $f_{Y|\boldsymbol{X}_{\boldsymbol{u}}}(\cdot \mid \boldsymbol{x}_{\boldsymbol{u}})$, a density estimator can be constructed exactly in the same spirit as (5):

$$\hat{f}_{Y|\boldsymbol{X}_{\boldsymbol{u}}}(y \mid \boldsymbol{x}_{\boldsymbol{u}}) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{h_{\boldsymbol{x}_{\boldsymbol{u}}}}K\left(\frac{y - Y_{\boldsymbol{x}_{\boldsymbol{u}}}^k}{h_{\boldsymbol{x}_{\boldsymbol{u}}}}\right) \tag{6}$$

where the bandwidth $h_{\boldsymbol{x}_{\boldsymbol{u}}}$ is updated every time a new conditioning set $\boldsymbol{x}_{\boldsymbol{u}}$ is encountered. After replacing f_Y and $f_{Y|\boldsymbol{X}_{\boldsymbol{u}}}$ by their respective estimators (5) and (6), the naive estimator set out in (3) becomes:

$$\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}} = \frac{1}{2N}\sum_{n=1}^{N}\left\|\hat{f}_Y(\cdot) - \hat{f}_{Y|\boldsymbol{X}_{\boldsymbol{u}}}\left(\cdot \mid \boldsymbol{X}_{\boldsymbol{u}}^n\right)\right\|_{L^1(\mathbb{R})} \ . \tag{7}$$

For now, $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ is called the double-loop estimator and the estimation procedure comprises the five steps below:

1. Generate an input dataset $\boldsymbol{X}_{\mathbf{obs}}$ from the input joint distribution $f_{\boldsymbol{X}}$:

$$\boldsymbol{X}_{\mathbf{obs}} := \left[\boldsymbol{X}^n\right]_{1 \leqslant n \leqslant N} \text{ with } \boldsymbol{X}^n = \left(\boldsymbol{X}_{\boldsymbol{u}}^n, \boldsymbol{X}_{\bar{\boldsymbol{u}}}^n\right).$$

2. Compute the associated outputs:

$$\boldsymbol{Y}_{\mathbf{obs}} := \left[Y^n\right]_{1 \leqslant n \leqslant N} \text{ with } Y^n = \mathcal{M}\left(\boldsymbol{X}^n\right).$$

3. Use output samples Y^n and (5) to estimate f_Y.
4. For every observed subvector $\boldsymbol{X}_{\boldsymbol{u}}^n$, generate conditional output observations:

$$\left\{Y_{\boldsymbol{X}_{\boldsymbol{u}}^n}^k\right\}_{1 \leqslant k \leqslant N} \overset{\text{i.i.d.}}{\sim} f_{Y|\boldsymbol{X}_{\boldsymbol{u}}}(\cdot \mid \boldsymbol{X}_{\boldsymbol{u}}^n) \ .$$

   and estimate $f_{Y|\boldsymbol{X}_{\boldsymbol{u}}}\big(\cdot \mid \boldsymbol{X}_{\boldsymbol{u}}^n\big)$ with (6).
5. Compute $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ with (7).

This approach will be considered as the reference in the simulation study of Section 5. As far as norms are concerned, since the metric space is $L^1(\mathbb{R})$, the Monte Carlo method is not the only workable way of computing the associated integrals and vectorized adaptive quadrature techniques may be used [20]. The second and fourth steps respectively imply N and N^2 calls to $\mathcal{M}$, which raises the required computational budget to $N(N+1)$ model evaluations. A variant of $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ was developped in [5] in order to secure convergence of the resulting estimator. The trick lies in constructing a finely grained partition of the sample space covered by $\boldsymbol{X}_{\boldsymbol{u}}$. Moreover, if one is willing to estimate all first-order Borgonovo's indices $\hat{\delta}_i^{\mathcal{K}}$ without multiplying by d the computation budget, a permuted column sampling plan developed in [21] may be deployed. Because of the computational burden it implies, $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ is often regarded as a brute-force estimator that should be considered only as a last resort. The following subsection provides a quick overview of existing alternatives.

2.3. Single-loop alternatives

Another kernel estimation method is introduced in [22] with actual computational gains, but the most promising idea can be found in [5]. Basic calculations allow to rewrite $\delta_{\boldsymbol{u}}$ under the following form:

$$\delta_{\boldsymbol{u}} = \frac{1}{2} \int_{\mathbb{R}^{r+1}} \left| f_{\boldsymbol{X}_{\boldsymbol{u}}}(\boldsymbol{x}_{\boldsymbol{u}}) f_Y(y) - f_{\boldsymbol{X}_{\boldsymbol{u}}Y}(\boldsymbol{x}_{\boldsymbol{u}}, y) \right| \mathrm{d}\boldsymbol{x}_{\boldsymbol{u}} \, \mathrm{d}y \tag{8}$$

$$= \frac{1}{2} \left\| f_{\boldsymbol{X}_{\boldsymbol{u}}} f_Y - f_{\boldsymbol{X}_{\boldsymbol{u}}Y} \right\|_{L^1(\mathbb{R}^{r+1})} \ . \tag{9}$$

From a given input-output dataset, PDF estimates can be constructed with KDE for both f_Y and $f_{\boldsymbol{X}_{\boldsymbol{u}}Y}$ and they give birth to a plug-in estimator:

$$\hat{\delta}_{\boldsymbol{u}} = \frac{1}{2} \left\| f_{\boldsymbol{X}_{\boldsymbol{u}}} \hat{f}_Y - \hat{f}_{\boldsymbol{X}_{\boldsymbol{u}}Y} \right\|_{L^1(\mathbb{R}^{r+1})} \ . \tag{10}$$

The remaining L^1-norm must be computed with the classical Monte Carlo method in $\mathbb{R}^{r+1}$. Using $\hat{f}_{\boldsymbol{X}_{\boldsymbol{u}}Y}$ as sampling distribution is advised in [7] and a nonparametric importance sampling approach is proposed in [8]. However, all these methods depend on how the kernel method resists the curse of dimensionality. Indeed, even if resorting to state-of-the-art techniques such as those in [23], nonparametric density estimation suffers from finding optimal smoothing parameters for kernel functions. A way forward to keep improving single-loop methods is offered by copula theory. The potential that copula functions offer from the perspective of estimating Borgonovo's indices, as well as currently ongoing works on this subject, are dealt with in the following subsection.

2.4. The copula-based estimation method

Since its very beginning, copula theory is concerned with identifying, learning or creating relevant dependence patterns for multivariate distributions when their margins are provided. Sklar's theorem [24] states that any absolutely continuous multivariate joint PDF could be expressed as the product of its marginal PDFs and a copula term draining all information regarding dependence among the involved variables. Let it be written for the random vector $(\boldsymbol{X_u}, Y)$:

$$f_{\boldsymbol{X_u}Y} = c_{\boldsymbol{X_u}Y}\Big(F_{X_{u_1}}, \dots, F_{X_{u_r}}, F_Y\Big) \times \left[\prod_{i=1}^{r} f_{X_{u_i}}\right] \times f_Y \ , \tag{11}$$

where $c_{\boldsymbol{X_u}Y}$ is defined on $\mathcal{H}_{r+1}$ and fulfills the requirements of a copula function, that is:

$$\Big(U_1, \dots, U_r, V\Big) \sim c_{\boldsymbol{X_u}Y} \text{ where } \begin{cases} U_i := F_{X_{u_i}}(X_{u_i}) \\ V := F_Y(Y) \end{cases} . \tag{12}$$

When $c_{\boldsymbol{X_u}Y}$ is a Gaussian copula, $f_{\boldsymbol{X}Y}$ is said to follow a Nataf distribution, even if the marginal distributions are unspecified. This particular case is examined in [9] where it is also assumed that all input variables are mutually independent. The integral expression in (8) thus becomes:

$$\delta_{\boldsymbol{u}} = \frac{1}{2}\int_{\mathbb{R}^{r+1}} \Big| c_{\mathcal{N}}\Big(F_{\boldsymbol{X_u}}(\boldsymbol{x_u}), F_Y(y) \,\big|\, \boldsymbol{\rho}_0\Big) - 1\Big| \, f_Y(y) \left[\prod_{i=1}^{r} f_{X_{u_i}}(x_{u_i})\right] \mathrm{d}\boldsymbol{x_u}\, \mathrm{d}y \ , \tag{13}$$

where $\boldsymbol{\rho}_0$ is the correlation matrix of the probability-transformed vector $(\boldsymbol{U}, V)$. Applying the Nataf transform consists in using a change of variables based on the standard normal CDF Φ:

$$\begin{cases} \Phi(S_i) = F_{X_{u_i}}(X_{u_i}) \\ \Phi(T) = F_Y(Y) \end{cases} . \tag{14}$$

Integration by substitution then yields:

$$\delta_{\boldsymbol{u}} = \frac{1}{2}\int_{\mathbb{R}^{r+1}} \left| \frac{\phi(s_1)\dots\phi(s_r)\,\phi(t)}{\phi_2\big(\boldsymbol{s}, t \,|\, \boldsymbol{\rho}_0\big)} - 1 \right| \phi_2\big(\boldsymbol{s}, t \,|\, \boldsymbol{\rho}_0\big)\, \mathrm{d}\boldsymbol{s}\, \mathrm{d}t \tag{15}$$

$$= \frac{1}{2}\mathbb{E}\left[\left| \frac{\phi(S_1)\dots\phi(S_r)\,\phi(T)}{\phi_2\big(\boldsymbol{S}, T \,|\, \boldsymbol{\rho}_0\big)} - 1 \right|\right] , \tag{16}$$

where ϕ and ϕ_2 denote respectively the PDFs of $\mathcal{N}(0,1)$ and $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\rho}_0)$. Then, the Nataf-based estimator $\hat{\delta}^{\mathcal{T}}_{\boldsymbol{u}}$ comes naturally:

$$\hat{\delta}^{\mathcal{T}}_{\boldsymbol{u}} = \frac{1}{N}\sum_{k=1}^{N} \left| \frac{\phi(S_1^k)\dots\phi(S_r^k)\,\phi(T^k)}{\phi_2\big(\boldsymbol{S}^k, T^k \,|\, \boldsymbol{\rho}_0\big)} - 1 \right| \quad \text{with} \quad (\boldsymbol{S}^k, T^k) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\rho}_0) \ . \tag{17}$$

Nevertheless, this methods suffers two major drawbacks. On the one hand, $\boldsymbol{\rho}_0$ is unknown, subject to certain exceptions, and each correlation coefficient between S_i

and T has to be determined by solving numerically a nonlinear equation [25]. On the other hand, the construction of $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{T}}$ is underpinned by two strong assumptions. Regarding mutual independence, one could easily consider correlated inputs and adapt $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{T}}$ accordingly. On the contrary, as far as Gaussianity is concerned, there is no doubt that it is too restrictive to estimate accurately high-order Borgonovo's indices in a general context. In order to develop an estimator that goes beyond this assumption, one needs to make the most of the opportunities offered by copula theory. As suggested for the first time in [10], combining (8) and twice (11) leads to:

$$\delta_{\boldsymbol{u}} = \frac{1}{2} \int_{\mathcal{H}_{r+1}} \left| c_{\boldsymbol{X}_{\boldsymbol{u}} Y}\big(u_1, \ldots, u_r, u_{r+1}\big) - c_{\boldsymbol{X}_{\boldsymbol{u}}}\big(u_1, \ldots, u_r\big) \right| \mathbf{d}\boldsymbol{u} \tag{18}$$

after operating a simple change of variables so as to fall back on an integral defined over $\mathcal{H}_{r+1}$. It should be noted that copula PDF $c_{\boldsymbol{X}_{\boldsymbol{u}}}$ becomes 1 if either $\boldsymbol{u}$ is a singleton or $\boldsymbol{X}_{\boldsymbol{u}}$ is made of mutually independent variables. A copula-based estimation $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}}$ of $\delta_{\boldsymbol{u}}$ may thus be derived by first learning $c_{\boldsymbol{X}_{\boldsymbol{u}} Y}$ from data and secondly computing a Monte Carlo estimate of the integral:

$$\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}} = \frac{1}{2N} \sum_{k=1}^{N} \left| \hat{c}_{\boldsymbol{X}_{\boldsymbol{u}} Y}\big(U_1^k, \ldots, U_{r+1}^k\big) - c_{\boldsymbol{X}_{\boldsymbol{u}}}\big(U_1^k, \ldots, U_r^k\big) \right| , \tag{19}$$

where:

$$\Big\{ \boldsymbol{U}^k = \big(U_1^k, \ldots, U_{r+1}^k\big) \Big\}_{1 \leqslant k \leqslant N} \overset{\text{i.i.d.}}{\sim} \mathcal{U}\Big([0\,,1]\Big)^{\otimes r+1} . \tag{20}$$

Copulas are gaining increasing interest in sensitivity analysis, as underlined by [26] and [27]. Indeed, Borgonovo's indices are not the only association measures that can be rewritten in terms of distance in a well-chosen metric space between the intrinsic dependence structure underlying $(\boldsymbol{X}_{\boldsymbol{u}}, Y)$ and independence. Moreover, from a statistical learning viewpoint, as copula functions belong to the space of multivariate distributions, inference can be performed with most estimation methods whether they are parametric or not. Insights into bivariate estimation methods are provided by [28] and [29] while [30] strives to compare parametric, semiparametric and nonparametric approaches. It was ascertained that the curse of dimensionality is somewhat easier to face from the angle of copula estimation. Developments in copula theory produced key results, most notably the pair-copula factorization, which enabled coming up with more flexible and more tractable copula models to tackle high dimensional problems [13]. A brief survey of recents advances and current limits in the field of dependence modeling is proposed in Section 3. As far as Borgonovo's δ-sensitivity measures are concerned, copula estimation was conducted with Gaussian kernels in the first copula-focused paper [10] and by resorting to a maximum entropy approach in [11]. More recently, and in a slightly different context where Y is multivariate, one can find in [14] a first attempt to connect vine copula models and Borgonovo's indices. Before going further, let us recall the procedure leading to $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}}$:

1. Generate an input dataset $\boldsymbol{X}_{\mathbf{obs}}$ from the input joint distribution $f_{\boldsymbol{X}}$:

$$\boldsymbol{X}_{\mathbf{obs}} := \Big[\boldsymbol{X}^n\Big]_{1 \leqslant n \leqslant N} \text{ with } \boldsymbol{X}^n = \Big(\boldsymbol{X}_{\boldsymbol{u}}^n, \boldsymbol{X}_{\bar{\boldsymbol{u}}}^n\Big).$$

2. Compute the associated outputs:

$$\boldsymbol{Y}_{\mathbf{obs}} := \Big[Y^n\Big]_{1 \leqslant n \leqslant N} \text{ with } Y^n = \mathcal{M}\big(\boldsymbol{X}^n\big).$$

3. Rescale input samples with marginal CDFs:

$$\boldsymbol{U}_{\mathbf{obs}} := \Big[\boldsymbol{U}^n\Big]_{1 \leqslant n \leqslant N} \text{ with } \boldsymbol{U}^n = \Big(F_1(X_1^n)\,,\,\ldots\,,\,F_d(X_d^n)\Big).$$

4. Compute ranks on output samples as suggested in [29] to offset the lack of knowledge about F_Y:

$$\boldsymbol{V}_{\mathbf{obs}} := \Big[V^n\Big]_{1 \leqslant n \leqslant N} \text{ with } V^n = \frac{1}{N+1}\sum_{i=1}^{N} \mathbb{1}_{\{Y^i \leqslant Y^n\}}\,.$$

5. Estimate $\hat{c}_{\boldsymbol{X_u}Y}$ from $\boldsymbol{U}_{\mathbf{obs}}$ and $\boldsymbol{V}_{\mathbf{obs}}$ with any copula estimation method.
6. Compute $\hat{\delta}^{\mathcal{C}}_{\boldsymbol{u}}$ with (19) where $c_{\boldsymbol{X_u}}$ is given by:

$$c_{\boldsymbol{X_u}}(\boldsymbol{u}) = f_{\boldsymbol{X_u}}\Big(F^{-1}_{X_{u_1}}(u_1), \ldots, F^{-1}_{X_{u_r}}(u_r)\Big).$$

Unlike what was noted in Section 2.2, model evaluations are only necessary at the second step. The computational budget is significantly alleviated since only N calls to $\mathcal{M}$ yield $\hat{\delta}^{\mathcal{C}}_{\boldsymbol{u}}$. Another salient feature lies in the fact that the first four steps do not depend on the subset $\boldsymbol{u}$ under consideration. Therefore, and it was not the case with $\hat{\delta}^{\mathcal{K}}_{\boldsymbol{u}}$, the computational burden remains equal to N if one needs to estimate a collection of several Borgonovo's δ-sensitivity measures. Because of its numerical efficiency, the copula-based estimation method seems to be an appealing alternative to $\hat{\delta}^{\mathcal{K}}_{\boldsymbol{u}}$. However, its accuracy directly hinges on how well the copula density is estimated at the fifth step. The next section seeks to show that, whether copula density estimation may sometimes be an easy task, it is more often a tricky problem, especially when intricate patterns of dependence have to be learnt.

3. A few words on copula estimation

3.1. Inference with bivariate copulas

3.1.1. Parametric families

Since the boom of copula theory, a surge of parametric models have been designed to satisfy three complementary criteria: interpretability, flexibility and tractability [30]. An allegedly exhaustive compendium of existing models is available in [31]. Elliptical copulas arise from elliptical distributions [32] and thus encompass both the Gaussian and t-Student copulas. The Archimedean class of copulas comprises many classical families such as Clayton, Frank, Gumbel and Joe. For each one, the copula CDF is obtained from a convex decreasing function $\psi(\cdot)$ which is called the generator and gives notable properties [33]. Another class deserving interest consists of extreme-value copulas [34] since they gave birth to prominent concepts such as max-stability and tail dependence.

3.1.2. Model selection

In the light of all those candidate models, a practitioner may wonder which one is the most relevant in view of given data, especially if no prior idea about the underlying distribution is available. Most often, fitting a chosen parametric model to data is performed through a maximum likelihood approach. Then, a validation step requires testing the zero-assumption according to which the true copula distribution belongs to the prescribed parametric family. It can be done with one of the many methods discussed in [35] and [36]. A method to retain the best model in terms of goodness-of-fit consists in maximizing a likelihood-based information criterion such as Akaike or Bayes. Before that, a model shortlist may be constituted from a qualitative study of the graphical properties observed on copula samples. In the case where several Archimedean copulas are competiting, a nonparametric graphical procedure was proposed in [37] to guide copula selection.

3.1.3. Nonparametric tools

If one is willing to outright evade the parametric framework, many nonparametric procedures have been explored [28]. To cite but a few, common practices include the mirror image method [38] to offset any possible bias introduced by the boundaries, transformed kernels [39] to be brought back to KDE in $\mathbb{R}^2$, and beta kernels [40] to directly handle kernels with compact support. Another frequently used method is based on the fact that Bernstein polynomials are dense in the space of all continuous functions [41].

3.2. Graph models to face the curse of dimensionality

A substantial majority of copula models can be easily generalized as d goes up. However, increasing dimensionality makes the minimum number of required parameters inflate exponentially and classical parametric families are outperformed since they were not designed to this end. To remedy this problem, tree-based hierarchical constructions have been developped with pair-copulas as building blocks. The main idea is to build up on bivariate dependence modeling, which is nowadays a properly processed task, to elaborate flexible dependence structures that should be able to face the growth of polymorphism.

3.2.1. Nested Archimedean copulas

Nested Archimedean copulas [42] are generalizations of Archimedean copulas that allow for asymmetry and provides more flexibility. Such a copula distribution is defined recursively since its CDF is obtained from an initial Archimedean copula CDF by replacing one or several of its arguments by another (possibly nested) Archimedean copula. In the case where there are $d-1$ nesting levels, it results in a hierarchical structure that exactly looks like a cluster dendogram [43]. A bivariate Archimedean copulas is affected to each node and models how the related clusters statistically depend on each other. Those copula models aroused interest in finance, especially in portfolio management where hierarchical structures often naturally emerge from real-life data. Commonly known Archimedean generators lead to an abundance of nested Archimedean copulas but a sufficient "nesting" condition has nevertheless to be fulfilled before combining copulas. An estimation algorithm to perform inference into this rich repertoire may be found in [44].

3.2.2. Product of bivariate copulas

Another class of multivariate copula distributions is proposed by Mazo [45] who simplifies what was done by Liebscher [46] in order to envision applications. This approach based on the product of transformed bivariate copulas. In general, multiplying several copula CDFs does not yield another copula CDF because the resulting margins are not uniform. In this model, dependence is described by a set of pairwise associations which are summarized in a tree E. Then, each variable u_i in the copula CDF expression is raised to the power $1/n_i$ where n_i is the number of edges connected to the node i in the graphical representation E. This makes the margins be uniform and a theorem by Liebscher [46] ensures the constructed function is indeed a copula CDF.

3.3. Pair-copula constructions

Among multivariate copula distributions, pair-copula constructions have gained increasing popularity over the last decade because of their notable level of flexibility. The first idea is due to Joe [47] but the current formalism appeared a few years laters with Bedford and Cooke [48, 49]. The main idea lies in a factorization result on the copula PDF obtained after applying recursively Sklar's theorem and Bayes' rule. The copula PDF may thus be written as a product of conditional pair-copula PDFs. Their arrangement is governed by a sequence of nested trees $\{E_1, ..., E_{d-1}\}$ consisting of nodes and edges. E_1 is a spanning tree connecting variables $U_1, \ldots, U_d$. Then, E_{i+1} may be derived from E_i provided that a "proximity condition" is obeyed. This is compulsory to be sure that the ensuing tree sequence coincides with a feasible copula factorization. Thus, any edge e involved in a tree E_i joins two nodes, denoted by j_e and k_e, and is subject to a conditioning set u_{D_e}. Accordingly, one has:

$$c_{\boldsymbol{X}}(\boldsymbol{u}) = \prod_{i=1}^{d-1} \prod_{e \in E_i} c_{j_e k_e | D_e} \left(u_{j_e|D_e}, u_{k_e|D_e} \middle| \boldsymbol{u}_{D_e} \right) , \tag{21}$$

with:

$$u_{j_e|D_e} = F_{j_e|D_e}(x_{j_e} \mid \boldsymbol{x}_{D_e}) \text{ and } u_{k_e|D_e} = F_{k_e|D_e}(x_{k_e} \mid \boldsymbol{x}_{D_e}) . \tag{22}$$

To enable fast and robust inference, the so-called "simplifying assumption" is made in most cases. The covariate term $\boldsymbol{u}_{D_e}$ involved in the conditioning mechanism is considered uninformative and is omitted. A simplified version of (21) is then used:

$$c_{\boldsymbol{X}}(\boldsymbol{u}) = \prod_{i=1}^{d-1} \prod_{e \in E_i} c_{j_e k_e | D_e} \left(u_{j_e|D_e}, u_{k_e|D_e} \right) . \tag{23}$$

For a given tree sequence, the dependence structure is organized as a cascade of pair-copulas which can be chosen independently. Models of this kind are referred to as regular vine copulas (R-vines). For the sake of simplicity, two particular subclasses were popularized [50] since the start of research on this topic: "canonical" vine copulas (C-vines) and "drawable" vine copulas (D-vines) depending on whether E_1 has a star-like structure or a chain-like structure. Aas [51] paved the way through inference with an estimation procedure applicable to simplified C-vines and D-vines. Dissmann [52] bridged the remaining gaps and came up with an automated model

selection technique in the simplified R-vine framework. For similar reasons as in the bivariate case, R-vine models have been recently adapted to embed nonparametric tools. This is a promising advance since it allows to deploy kernel methods without incurring the curse of dimensionality. Benefits in terms of convergence are shown in [53]. Among all available inference procedures, a comparative study led in [54] reveals that none is uniformly better than all others when one has to estimate a nonparametric R-vine copula.

The focus of this section was to stress the fact that many copula models were developped to date and that they often compete with each other. However, in many application fields, including financial econometrics, insurance, engineering, hydrology, earth sciences or marketing, R-vine models have become a multipurpose copula-based solution to construct, learn and simulate dependence patterns in high dimension. Their reputation is strengthened by their user-friendly software availability under the R packages `CDVine` and `VineCopula`. Incorporating nonparametric copulas within an R-vine tree structure is now possible with the R package `kdevine` and offers hope for even more flexibility. In this view, R-vine copulas must be regarded as a relevant option for the copula-based estimation of δ-sensitivity measures introduced in Section 2.4. Besides, they will be an essential ingredient throughout the simulation study proposed in Section 5.

4. Shapley values

4.1. Definition and properties of Shapley values

In cooperative game theory, Shapley values [55] allow to fairly apportion earnings among players knowing that some of them may have collaborated to enhance their chances of success. A cooperative game requires a finite set of players $[d] = \{1, \dots, d\}$ and a cost function $\tilde{c}(\cdot)$ which maps any coalition of players $\boldsymbol{u} \in \mathcal{P}_d$ to the total payoff expected after working collaboratively. Mathematically speaking, $\tilde{c}(\cdot)$ is such that:

$$\tilde{c} : \left| \begin{array}{ccc} \mathcal{P}_d & \longrightarrow & \mathbb{R} \\ \boldsymbol{u} & \longmapsto & \tilde{c}(\boldsymbol{u}) \end{array} \right. \quad \text{with } \tilde{c}(\varnothing) = 0 \ , \tag{24}$$

Then, the Shapley value assigned to the i-th player is computed as follows:

$$\phi_i := \phi_i(\tilde{c}) = \frac{1}{d} \sum_{\boldsymbol{u} \subseteq [d] \setminus \{i\}} \binom{d-1}{|\boldsymbol{u}|}^{-1} \Big(\tilde{c}(\boldsymbol{u} \cup \{i\}) - \tilde{c}(\boldsymbol{u}) \Big). \tag{25}$$

The i-th player might join $\binom{d-1}{|\boldsymbol{u}|}$ coalitions of size $|\boldsymbol{u}|$. Insofar as joining a coalition is a uniformly random choice, $\binom{d-1}{|\boldsymbol{u}|}^{-1}$ is the probability that the player joins one particular coalition of size $|\boldsymbol{u}|$. The cost difference $\tilde{c}(\boldsymbol{u} \cup \{i\}) - \tilde{c}(\boldsymbol{u})$ must be seen as the marginal contribution of the i-th player in the coalition $\boldsymbol{u}$. Hence, ϕ_i is the average gain obtained after taking into account all coalitions this player might be involved in. It was shown in [55] that the values defined in (25) satisfy several axioms that sound rather natural in a cooperative game context.

Proposition 4.1 (Shapley values)

1. ***Efficiency.*** *Shapley values sum to the total gain of the great coalition:*

$$\sum_{i=1}^{d} \phi_i = \tilde{c}\big([d]\big) \ .$$

2. ***Symmetry.*** *If $\tilde{c}(\boldsymbol{u} \cup \{i\}) = \tilde{c}(\boldsymbol{u} \cup \{j\})$ for all $\boldsymbol{u} \subseteq [d] \smallsetminus \{i, j\}$, then $\phi_i = \phi_j$.*

3. ***Dummy.*** *If $\tilde{c}(\boldsymbol{u} \cup \{i\}) = \tilde{c}(\boldsymbol{u})$ for all $\boldsymbol{u} \subseteq [d] \smallsetminus \{i\}$, then $\phi_i = 0$.*

4. ***Linearity.*** *For any cost functions $\tilde{c}_1$ and $\tilde{c}_2$, one has:*

$$\forall i \in [d], \quad \phi_i(\tilde{c}_1 + \tilde{c}_2) = \phi_i(\tilde{c}_1) + \phi_i(\tilde{c}_2) \ .$$

4.2. Application in sensitivity analysis

In sensitivity analysis, Shapley values were not used until a fairly recent work due to Owen [15]. To understand analogy with cooperative game theory, the set of single input variables $X_1, \ldots, X_d$ must be thought as a set of players and the chosen importance measure as a cost function. This approach has already been studied in the framework of variance-based sensitivity analysis methods [56] and especially in the correlated case [57, 58] where the use of Shapley values is all the more justified. In this paper, the novel idea is to consider the following cost function:

$$\tilde{c}(\boldsymbol{u}) = \delta_{\boldsymbol{u}} \ , \tag{26}$$

which maps any coalition $\boldsymbol{u}$ to its sensitivity measure $\delta_{\boldsymbol{u}}$ established in Section 2. Adopting this definition complies with Shapley value theory since:

$$\tilde{c}(\varnothing) = \delta_{\varnothing} = 0 \ . \tag{27}$$

Applying the new cost function (26) defines Shapley values with respect to Borgonovo's δ-sensitivity measure:

$$\mathrm{Sh}_i := \frac{1}{d} \sum_{\boldsymbol{u} \subseteq [d] \smallsetminus \{i\}} \binom{d-1}{|\boldsymbol{u}|}^{-1} (\delta_{\boldsymbol{u} \cup \{i\}} - \delta_{\boldsymbol{u}}) \ . \tag{28}$$

The efficiency axiom stated in Proposition 4.1 implies the following result:

$$\sum_{i=1}^{d} \mathrm{Sh}_i = 1 \ . \tag{29}$$

Shapley values Sh_i allow an easier interpretation of Borgonovo's indices. On the one hand, the information provided by Borgonovo's indices at all orders, that is $2^d - 1$ values, is now summed up by a much smaller set consisting of only d sensitivity indicators. On the other hand, (29) suggests a Sobol-like interpretation in terms of percentage. And finally, computing Shapley values from the complete set of Borgonovo's indices does not require any additional call to $\mathcal{M}$ and must be regarded as an easily achievable GSA byproduct.

4.3. Synthesis on sensitivity analysis with Shapley values

The objective of this section is not to review all the sensitivity analysis techniques. An exhaustive review is notably available here [1]. Nevertheless, it is of interest to analyse the principal merits and demerits of Sobol indices, Borgonovo's indices and their respective Shapley value extensions. For that purpose, we summarize and compare some of their different characteristics in Table 1. Sobol indices focus on the impact of inputs on the variance output whereas Borgonovo's indices consider the entire distribution. In many cases, both indices give the same tendencies unless the variance is not a representative quantity of the output distribution. To ease the interpretation of GSA indices, it is preferable that they sum to 1 which is unfortunately not the case of Borgonovo's indices. Dependent inputs have long been a major deficiency of variance-based GSA. Shapley values were recently coupled with Sobol indices to overcome this issue [15]. Borgonovo's importance measure can be applied on dependent variables but, when a group of inputs is considered, it does not allow to distinguish between dependence effects (due to the copula) and interaction effects (due to how $\mathcal{M}$ works). In some engineering applications, it is also relevant to only have one scalar index by input, which is the case for Shapley values. All these GSA indices can be estimated with a single input-output dataset. This type of low-cost estimation with Sobol indices and Shapley values of Sobol indices is very recent [59].

Table 1: Main features of various GSA indices: Sobol indices S_i, Borgonovo's indices δ_i and their respective Shapley value extensions denoted by $\mathrm{Sh}_i^{\mathrm{sobol}}$ and $\mathrm{Sh}_i^{\mathrm{borgo}}$.

GSA features	S_i	$\mathrm{Sh}_i^{\mathrm{sobol}}$	δ_i	$\mathrm{Sh}_i^{\mathrm{borgo}}$
Quantity of interest	$\mathbb{V}(Y)$	$\mathbb{V}(Y)$	f_Y	f_Y
Sum equal to 1	$+$	$+$	$-$	$+$
Adapted to dependent inputs	$-$	$+$	$+$	$+$
Adapted to grouped inputs	$+$	$-$	$+$	$-$
Estimation from $\{\boldsymbol{X}_{\mathbf{obs}}, \boldsymbol{Y}_{\mathbf{obs}}\}$	$+$	$+$	$+$	$+$

5. Simulation study

In this section, the proposed methodology is applied to several numerical test cases with increasing complexity. A very simple situation is first considered where $\mathcal{M}$ is a linear mapping which takes Gaussian inputs. Having a look at such a trivial example is however important because it is the only case presented in this work where one could derive analytical expressions of the copula densities highlighted in (18). Hence, the theoretical values of both Borgonovo's indices and Shapley values can be estimated with as much accuracy as desired. Having access to those

values enable to check if both the double-loop estimation method and the copula-based one are properly tuned. Then, two other analytical test cases are examined in higher dimension, with d being respectively equal to 6 and 8. Both are widely used by the uncertainty analysis research community and may be regarded as benchmark examples involving a significant level of nonlinearity in the way they transform the input variables. Finally, in order to be more consistent with the initial assumption according to which $\mathcal{M}$ is a black-box model, the proposed approach is confronted with a simulation code which is truly representative of the underlying physics.

The main objective of the following numerical simulations is to compare the Shapley values estimators $\hat{\mathrm{Sh}}_i$ resulting from the prior computation of all Borgonovo's indices with either the double-loop estimator $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ or a copula-based estimator $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}}$. The corresponding Shapley value estimators for input X_i are given respectively by $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ and $\hat{\mathrm{Sh}}_i^{\mathcal{C}}$. The real issue is actually to assess the extent to which $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}}$, despite the lack of theoretical guarantees regarding its convergence, can be a suitable substitute for $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ in order to compute Shapley values. Let us now denote by $N_{\mathcal{K}}$ and $N_{\mathcal{C}}$ the number of input samples respectively used for the double-loop and the copula-based estimation procedures described in Sections 2.2 and 2.4. Every time an averaged estimation of Shapley values has to be performed for a given estimation method $\hat{\delta}_{\boldsymbol{u}}$ of Borgonovo's indices, the same computational procedure is applied. Further details are specified in Appendix A.

5.1. Gaussian linear model

5.1.1. Description

Let us consider the case where the model output may be expressed as:

$$Y = \mathcal{M}(\boldsymbol{X}) = \boldsymbol{A}\boldsymbol{X} \tag{30}$$

with $\boldsymbol{A} = 0.1 \times \begin{bmatrix} 17 & 18 & 19 & 20 \end{bmatrix}$. The input random vector $\boldsymbol{X}$ lies in $\mathbb{R}^4$ and follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that:

$$\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1 & 1/2 & 1/3 \\ 1/3 & 1/2 & 1 & 1/2 \\ 1/4 & 1/3 & 1/2 & 1 \end{bmatrix}.$$

It is straightforward to see that the augmented vector $\boldsymbol{Z}_{\boldsymbol{u}}$ which brings together $\boldsymbol{X}_{\boldsymbol{u}}$ and Y is also a centered Gaussian distribution since:

$$\boldsymbol{Z}_{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{X}_{\boldsymbol{u}} \\ Y \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{\boldsymbol{u}} \\ \hline \boldsymbol{A} \end{bmatrix} \boldsymbol{X} \tag{31}$$

where:

$$\boldsymbol{S}_{\boldsymbol{u}} = \begin{bmatrix} \alpha_{ij} \end{bmatrix} \in \{0,1\}^{|\boldsymbol{u}| \times d} \text{ with } \alpha_{ij} = \begin{cases} 1 \text{ if } u_i = j \\ 0 \text{ otherwise} \end{cases}. \tag{32}$$

Thus, $\boldsymbol{Z}_{\boldsymbol{u}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Lambda}_{\boldsymbol{u}})$ and the covariance matrix $\boldsymbol{\Lambda}_{\boldsymbol{u}}$ may be divided into four blocks:

$$\boldsymbol{\Lambda}_{\boldsymbol{u}} = \left[\begin{array}{c|c} \boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}}} & \boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}} Y} \\ \hline \boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}} Y}^{\intercal} & \mathbb{V}(Y) \end{array} \right] \in \mathrm{M}_{|\boldsymbol{u}|+1}(\mathbb{R}) \tag{33}$$

with $\boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}}}$ the covariance matrix of $\boldsymbol{X}_{\boldsymbol{u}}$ and:

$$\boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}}Y} := \Big[\mathrm{Cov}\big(X_{u_i}, Y\big)\Big]_{1 \leqslant i \leqslant |\boldsymbol{u}|} \in \mathbb{R}^{|\boldsymbol{u}|} \,. \tag{34}$$

As $\boldsymbol{X}_{\boldsymbol{u}}$ and $\boldsymbol{Z}_{\boldsymbol{u}}$ are Gaussian, the underlying copulas are both Gaussian. Moreover, they are fully-parametrized by the correlation matrices $\boldsymbol{\Gamma}_{\boldsymbol{X}_{\boldsymbol{u}}}$ and $\boldsymbol{\Gamma}_{\boldsymbol{u}}$ which can respectively be extracted from the covariance matrices $\boldsymbol{\Lambda}_{\boldsymbol{X}_{\boldsymbol{u}}Y}$ and $\boldsymbol{\Lambda}_{\boldsymbol{u}}$. In light of that fact, the copula-based integral expression of Borgonovo's importance measure $\delta_{\boldsymbol{u}}$ established in (18) becomes:

$$\delta_{\boldsymbol{u}} = \frac{1}{2} \int_{\mathcal{H}_{r+1}} \left| c_{\mathcal{N}}\Big(u_1, \ldots, u_r, u_{r+1} \,\Big|\, \boldsymbol{\Gamma}_{\boldsymbol{u}}\Big) - c_{\mathcal{N}}\Big(u_1, \ldots, u_r \,\Big|\, \boldsymbol{\Gamma}_{\boldsymbol{X}_{\boldsymbol{u}}}\Big) \right| \mathbf{d}\boldsymbol{u} \tag{35}$$

and a natural Monte Carlo estimator of $\delta_{\boldsymbol{u}}$ is thus adapted from (19):

$$\hat{\delta}^{\mathcal{I}}_{\boldsymbol{u}} = \frac{1}{2N} \sum_{k=1}^{N} \left| c_{\mathcal{N}}\Big(U_1^k, \ldots, U_r^k, U_{r+1}^k \,\Big|\, \boldsymbol{\Gamma}_{\boldsymbol{u}}\Big) - c_{\mathcal{N}}\Big(U_1^k, \ldots, U_r^k \,\Big|\, \boldsymbol{\Gamma}_{\boldsymbol{X}_{\boldsymbol{u}}}\Big) \right| . \tag{36}$$

The above estimator $\hat{\delta}^{\mathcal{I}}_{\boldsymbol{u}}$ may be regarded as a reference one, in the sense that its variability does not come from the estimation of copula densities but only from stochastic integration over the hypercube $\mathcal{H}_{r+1}$. Such a variability may be reduced as much as desired by increasing the amount N of points $\boldsymbol{U}^k$ sampled over $\mathcal{H}_{r+1}$ and it is worth noting that it does not entail any call to the black-box model $\mathcal{M}$. Moreover, $\hat{\delta}^{\mathcal{I}}_{\boldsymbol{u}}$ inherits all the statistical properties associated with Monte Carlo integration and is therefore unbiased, consistent and asymptotically Gaussian. As far as the double-loop method is concerned, the estimation algorithm is run with $N_{\mathcal{K}} = 100$ input samples. Moreover, a natural choice for the copula-based estimator consists in directly learning a Gaussian copula from a dataset $\boldsymbol{X}_{\text{obs}}$ bringing together $N_{\mathcal{C}} = 5\,000$ input samples. It can be done with both the maximum likelihood method and the method of moments [60]. The reader is referred to [61] for general considerations about parametric copula estimation and to [62] for a specific focus on elliptical copulas. The resulting estimator may be written as follows:

$$\hat{\delta}^{\mathcal{C}}_{\boldsymbol{u}} = \frac{1}{2N} \sum_{k=1}^{N} \left| \hat{c}_{\mathcal{N}}\Big(U_1^k, \ldots, U_r^k, U_{r+1}^k \,\Big|\, \hat{\boldsymbol{\Gamma}}_{\boldsymbol{u}}\Big) - c_{\mathcal{N}}\Big(U_1^k, \ldots, U_r^k \,\Big|\, \boldsymbol{\Gamma}_{\boldsymbol{X}_{\boldsymbol{u}}}\Big) \right| , \tag{37}$$

where $\hat{\boldsymbol{\Gamma}}_{\boldsymbol{u}}$ is a maximum-likelihood estimation of the correlation matrix $\boldsymbol{\Gamma}_{\boldsymbol{u}}$. One major criticism could be to point out the fact that the whole correlation matrix $\hat{\boldsymbol{\Gamma}}_{\boldsymbol{u}}$ is estimated whereas most correlation coefficients Γ_{ij} can be directly found in $\boldsymbol{\Gamma}_{\boldsymbol{X}_{\boldsymbol{u}}}$. However, if estimation is limited to the missing coefficients, there is no assurance that $\hat{\boldsymbol{\Gamma}}_{\boldsymbol{u}}$ remains positive semidefinite. Moreover, it can be numerically verified that this provides less accurate estimates.

5.1.2. Results

Shapley values are estimated with $M = 100$ times for the three aforementioned estimators. The computational procedure is recalled in Appendix A. As expected, the reference estimators $\hat{\mathrm{Sh}}_i^{\mathcal{I}}$ are unbiased and evidence very small dispersion. This

level of accuracy is achieved with $N = 10^6$ points in (36). Estimation is also accurate but with slightly larger coefficients of variation for the two other estimators. It is worth noting that rankings are unchanged for the three methods. It has to be stressed because $\boldsymbol{A}$ and $\boldsymbol{\Gamma}$ were chosen so that hierarchy among sensitivity indices would be hard to retrieve. From now, the values given to $N_{\mathcal{K}}$ and $N_{\mathcal{C}}$ are considered as validated tuning setups for the next test cases. The outstanding results obtained for this first application should be put in perspective as all copulas are Gaussian and perfectly suitable for a maximum-likelihood parametric estimation. Two examples presenting more complex dependence structures are studied in the next sections.

Table 2: Estimation of Shapley values Sh_i for the Gaussian linear model defined in (30).

Inputs	$\hat{\mathrm{Sh}}_i^{\mathcal{I}}$			$\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ with $N_{\mathcal{K}} = 100$			$\hat{\mathrm{Sh}}_i^{\mathcal{C}}$ with $N_{\mathcal{C}} = 5\,000$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.2213	(4)	1.0	0.2234	(4)	2.9	0.2212	(4)	3.0
X_2	0.2532	(3)	0.7	0.2515	(3)	2.5	0.2536	(3)	2.6
X_3	0.2678	(1)	0.5	0.2671	(1)	2.6	0.2678	(1)	2.5
X_4	0.2578	(2)	0.8	0.2580	(2)	2.5	0.2574	(2)	2.8

5.2. Single Degree of Freedom (SDoF) oscillator

5.2.1. Description

A non-linear oscillator is now addressed. Detailed explanations and a schematic may be found in [63]. The system consists of a mass M connected by two springs at a stationary support. Both have the same free length R but differ by their stiffness coefficients C_1 and C_2. The system is subject to a rectangular load pulse with random duration T and amplitude F. The model output is defined by:

$$Y = \mathcal{M}(\boldsymbol{X}) = -3R + \left| \frac{2F}{C_1 + C_2} \sin\left(\sqrt{\frac{C_1 + C_2}{m}} \frac{T}{2} \right) \right| , \tag{38}$$

and stands for the difference between the maximum displacement response and $3R$. The six input variables C_1, C_2, R, M, T and F are mutually independent and lognormally distributed with respective parameters given in Table 3 according to what can be found in Table 5 from [63]. Such a two-spring-mass test case is often examined in uncertainty analysis when the accuracy of a newly-designed surrogate model needs to be assessed [64, 65, 66]. For this model, the underlying copulas do not have a closed-form expression. If one assumes they belong to a prescribed parametric family, it could lead to serious misestimations of Borgonovo's indices. In order not to incur this, copula PDFs have to be learnt with a multivariate class of distribution tailored to resist high dimension, like those introduced previously in Section 3. The copula-based estimator $\hat{\mathrm{Sh}}_i^{\mathcal{C}}$ is coupled with parametric R-vine copula models since they offer an interesting balance between learning flexibility and implementation easiness.

The double-loop estimator $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ is now regarded as a reference because it does not make any model assumption regarding how Y may be tied to an input subvector

Table 3: Input probabilistic model attached to the SDoF oscillator. Each marginal distribution f_{X_i} is assumed lognormal, that is $\log(X_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Input	**Description (unit)**	**Parameters**
X_1	Stiffness coefficient C_1 (N.m^{-1})	$\mu_1 = 2.0$ $\sigma_1 = 0.20$
X_2	Stiffness coefficient C_2 (N.m^{-1})	$\mu_2 = 0.2$ $\sigma_2 = 0.02$
X_3	Free length R (m)	$\mu_3 = 0.6$ $\sigma_3 = 0.05$
X_4	Mass M (kg)	$\mu_4 = 1.0$ $\sigma_4 = 0.05$
X_5	Duration T (s)	$\mu_5 = 1.0$ $\sigma_5 = 0.20$
X_6	Amplitude F (N)	$\mu_6 = 1.0$ $\sigma_6 = 0.20$

$\boldsymbol{X}$. For purposes of comparison, the first-order and total-effect Sobol indices are also computed with a "Pick-and-Freeze" estimation scheme [67]. The expected goal is to discuss whether the summary provided by Shapley values match the conclusions one could have drawn after glancing at Sobol indices.

5.2.2. Results

The computational procedure presented in Appendix A is repeated $M = 100$ times to produce the results shown in Tables 4 and 5. Generally speaking, $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ and $\hat{\mathrm{Sh}}_i^{\mathcal{C}}$ seem to lead to the same kind of conclusions. Rankings are slightly different but an overall trend appears with three groups. X_4 and then X_1 emerge as the most influential variables while X_5 and X_6 form a second group of competing variables. Very small Shapley values are observed for X_2 and X_3. From a computational viewpoint, it must be noted that the double-loop estimator proves to be somewhat less robust than its copula-based contender. However, the associated simulation budget $\mathcal{B}_{\mathcal{K}}$ is computed as follows:

$$\begin{aligned}
\mathcal{B}_{\mathcal{K}} &= \text{number of } \delta\text{-indices} \times \text{number of calls to } \mathcal{M} \text{ per } \delta\text{-index} \\
&= \sum_{i=1}^{d-1} \binom{d}{i} \times N_{\mathcal{K}}(N_{\mathcal{K}}+1) \\
\hline
\mathcal{B}_{\mathcal{K}} &= (2^d - 2) \times N_{\mathcal{K}} \times (N_{\mathcal{K}}+1) = 626\,200 \text{ calls to } \mathcal{M}\ .
\end{aligned}$$

On the contrary, the simulation budget $\mathcal{B}_{\mathcal{C}}$ required by the copula-based method is limited to $N_{\mathcal{C}} = 5\,000$ calls to $\mathcal{M}$. As can be seen in Section 2.4, the copula paradigm allows to derive all Borgonovo's indices from a single input-output dataset.

Examining total-effect Sobol indices in Table 5 sheds a different light on this sensitivity analysis problem. Within the pair (X_1,X_4) of roughly equal influence, total-effect Sobol indices designate X_1 as the main variance contributor, well ahead of all others variables. In this sense, focusing on variance may lead to wrongly identify X_1 as the only prevailing variable. High values are observed for some coefficients of variation, in particular those concerning $\hat{S}_2$, $\hat{S}_3$ and $\hat{S}_2^T$. It must be noticed that the corresponding mean values are all very small. In fact, the Monte Carlo estimation of small Sobol indices often suffers from a loss of accuracy due to the fact that the variance of the conditional expectation becomes too small with respect to the square of the output mean value. As this problem is often deplored in practice,

Table 4: Estimation of Shapley values Sh_i for the input-output function introduced in (38).

Inputs	$\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ with $N_{\mathcal{K}} = 100$			$\hat{\mathrm{Sh}}_i^{\mathcal{C}}$ with $N_{\mathcal{C}} = 5\,000$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.3315	(2)	8.1	0.2890	(2)	2.7
X_2	0.0419	(5)	14.8	0.0914	(6)	3.7
X_3	0.0254	(6)	16.6	0.0919	(5)	3.6
X_4	0.3442	(1)	8.0	0.2923	(1)	3.1
X_5	0.1845	(3)	7.8	0.1168	(4)	4.4
X_6	0.0725	(4)	12.6	0.1187	(3)	3.9

Table 5: Estimation of both first-order and total-effect Sobol indices for (38).

Inputs	$\hat{S}_i$			$\hat{S}_i^T$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.4867	(1)	0.7	0.6615	(1)	0.5
X_2	0.0027	(6)	72.4	0.0002	(6)	48.3
X_3	0.0029	(5)	71.4	0.0062	(5)	4.9
X_4	0.2703	(2)	1.4	0.2704	(2)	1.0
X_5	0.0313	(4)	10.7	0.1880	(3)	1.2
X_6	0.0344	(3)	10.1	0.0618	(4)	3.5

better estimators have been constructed [68, 69] but their implementation is beyond the scope of this paper. For small indices, standard deviations should rather be examined and most people would agree that there is no cause of worry since the rankings of Sobol indices are not endangered.

5.3. Water flow through a borehole

5.3.1. Description

Let us now consider the physical model used in the literature to describe the water flow between an upper aquifer and a lower aquifer separated by an impermeable rock layer. Water can though be transferred from one to the other through a borehole. The flow rate Y is thus computed from eight scalar parameters describing either geological layout or hydrological properties:

$$Y = \mathcal{M}(\boldsymbol{X}) = \frac{2\pi T_u \big(H_u - H_l\big)}{\log\left(\dfrac{R}{R_w}\right)\left(1 + \dfrac{2LT_u}{\log\left(\frac{R}{R_w}\right) R_w^2 K_w} + \dfrac{T_u}{T_l}\right)} \,. \tag{39}$$

A schematic and further details on physical assumptions, especially in fluid mechanics, can be found in [70] where (39) seems to have been introduced for the first time. As a general remark, it can be said that this model is sometimes used during the prospecting of nuclear waste repositories. In the research community, it is widely used to validate various computer experiment techniques [71, 72]. The input

probabilistic model is given in Table 6 and follows what is stated in Table B.1.* from [73]. For many variables, expert knowledge limits to providing the upper and lower bounds of the distribution support. Uniform distributions are derived and this is sort of common practice in this application field. The resulting probabilistic model may be subject to possible updates in the future with the availability of new operational data.

Table 6: Input probabilistic model attached to the borehole function defined in (39). Distribution types include the normal ($\mathcal{N}$), lognormal ($\mathcal{L}$) and uniform ($\mathcal{U}$) distributions.

Input	Description (unit)	Parameters	
X_1 ($\mathcal{N}$)	Radius of borehole R_w (m)	$\mu_1 = 1.0 \times 10^{-1}$	$\sigma_1 = 1.6 \times 10^{-2}$
X_2 ($\mathcal{L}$)	Radius of influence R (m)	$\mu_2 = 7.7$	$\sigma_2 = 1.0$
X_3 ($\mathcal{U}$)	Tansmissivity of upper aquifer T_u (m^2/yr)	$a_3 = 6.3 \times 10^4$	$b_3 = 1.2 \times 10^5$
X_4 ($\mathcal{U}$)	Potentiometric head of upper aquifer H_u (m)	$a_4 = 9.9 \times 10^2$	$b_4 = 1.1 \times 10^3$
X_5 ($\mathcal{U}$)	Tansmissivity of lower aquifer (m^2/yr)	$a_5 = 6.3 \times 10^1$	$b_5 = 1.2 \times 10^2$
X_6 ($\mathcal{U}$)	Potentiometric head of lower aquifer H_l (m)	$a_6 = 7.0 \times 10^2$	$b_6 = 8.2 \times 10^2$
X_7 ($\mathcal{U}$)	Length of borehole L (m)	$a_7 = 1.1 \times 10^3$	$b_7 = 1.7 \times 10^3$
X_8 ($\mathcal{U}$)	Hydraulic conductivity of borehole K_w (m/yr)	$a_8 = 9.9 \times 10^3$	$b_8 = 1.2 \times 10^4$

5.3.2. Results

The computational procedure found in Appendix A is applied $M = 10$ times to get averaged estimates for $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ and $\hat{\mathrm{Sh}}_i^{\mathcal{C}}$. Just as in Section 5.2, estimating all copula PDFs $c_{\boldsymbol{X_u}Y}$ with R-vine models is the key step that makes the copula-based method work. The results are presented in Table 7 where X_1 is identified as the most influential variable by both estimation methods. Then X_4, X_6 and X_7 can be regarded as equally influential on Y behind X_1. In this test case, the results are even more convincing as rankings are exactly the same between $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ and $\hat{\mathrm{Sh}}_i^{\mathcal{C}}$. However, it may be noted that the coefficient of variation for $\hat{\mathrm{Sh}}_5^{\mathcal{C}}$ is abnormally high. Similarly to what was observed in Table 5, this occurs for an index with a small mean value, but the explanation must be sought elsewhere since Table 7 deals with Shapley values. If a variable X_i has no influence on Y, one must expect to have $\hat{\delta}_{\boldsymbol{u}\cup\{i\}} \approx \hat{\delta}_{\boldsymbol{u}}$ for almost all subsets $\boldsymbol{u}$ that do not include i. As a result, all difference terms involved in (28) are hard to estimate accurately and may even be wrongly negative. That is why it is not very surprising to observe a deterioration of the coefficient of variation in this situation. Moreover, from a computational viewpoint, here is yet another example proving the relevancy of the copula-based method since the simulation budget $\mathcal{B}_{\mathcal{C}}$ only amounts to $N_{\mathcal{C}} = 5\,000$ calls to $\mathcal{M}$ while:

$$\mathcal{B}_{\mathcal{K}} = \quad (2^d - 2) \times N_{\mathcal{K}} \times (N_{\mathcal{K}} + 1) \quad = \quad 2\,565\,400 \text{ calls to } \mathcal{M} \ .$$

Comparison with total-effect Sobol indices is quite reassuring since the four more influential variables are common in Tables 7 and 8. However, as in the previous test case, Sobol indices tend to give too much importance to some variables, as if sharing the variance was a too discriminating approach. Furthermore, one can see that $\hat{S}_2$,

$\hat{S}_3$, $\hat{S}_5$ and their total-effect counterparts feature much larger coeffients of variation. Once again, this must be imputed to the difficulties inherent to the estimation of small sensitivity indices. Up to now, $\mathcal{M}$ has always been an analytical input-output performance function that could be written as a simple composition of classical math tools. A fully non-transparent simulation code is under study in Section 5.4.

Table 7: Estimation of Shapley values Sh_i for the input-output function introduced in (39).

Inputs	$\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ with $N_{\mathcal{K}} = 100$			$\hat{\mathrm{Sh}}_i^{\mathcal{C}}$ with $N_{\mathcal{C}} = 5\,000$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.4513	(1)	6.0	0.4660	(1)	2.2
X_2	0.0571	(6)	5.4	0.0287	(6)	19.1
X_3	0.0625	(5)	0.1	0.0299	(5)	24.8
X_4	0.1190	(3)	11.3	0.1347	(3)	4.9
X_5	0.0515	(7)	9.7	0.0162	(7)	222.8
X_6	0.1192	(2)	11.5	0.1362	(2)	6.1
X_7	0.1105	(4)	10.9	0.1279	(4)	6.8
X_8	0.0291	(8)	27.8	0.0604	(8)	8.0

Table 8: Estimation of both first-order and total-effect Sobol indices for (39).

Inputs	$\hat{S}_i$			$\hat{S}_i^T$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.6635	(1)	0.4	0.6939	(1)	0.5
X_2	0.0029	(7)	66.8	0.0001	(8)	91.6
X_3	0.0029	(8)	66.8	0.0001	(7)	74.9
X_4	0.0959	(3)	3.8	0.1057	(3)	1.5
X_5	0.0029	(6)	66.8	0.0001	(6)	65.8
X_6	0.0954	(2)	3.5	0.1059	(2)	1.5
X_7	0.0909	(4)	3.5	0.1027	(4)	2.0
X_8	0.0225	(5)	14.7	0.0251	(5)	4.1

5.4. A launcher stage fallout model

5.4.1. Description

Estimating Shapley values is attempted for a much more complex test case, where $\mathcal{M}$ is no longer an analytical input-output performance function. This application is taken from the field of aerospace engineering. During the launch of a satellite, the payload is put into orbit after a highly technological process. Before the spacecraft is released, several tasks have to be carried out, including first stage separation, fairing jettisoning and second stage separation. Even if the mission's main goal leans on having the satellite successfully placed in orbit, monitoring how the propulsion module subsystems are dropped is of utmost importance to guarantee that the fall-back in the atmosphere do not jeopardize human life and activities. $\mathcal{M}$ consists of

a simplified trajectory simulation code of the dynamic fallout phase of a generic launcher first stage. It has also been used in [74] for an extensive sensitivity analysis study. The simulation code needs to be given flight parameters describing the system state at the instant when the first-stage rocket boosters are ejected. Running $\mathcal{M}$ allows to predict the re-entry trajectory and the impact point in the ocean. Input flight parameters $\boldsymbol{X}$ are subject to aleatory uncertainties because of the physical perturbations that may affect the real flight before its arrival at the separation position. The model output Y is the distance between the theoretical and experimental impact positions. The former is computed from nominal input values. The latter is computed from measured data forwarded by embedded systems. The input probabilistic model is presented in Table 9 and it has already been deemed appropriate in previous works, see Table 1 from [74]. All variables are assumed to be independent and normally distributed.

Table 9: Input probabilistic model chosen to describe perturbations occurring at flight separation. All marginal distributions are assumed normal and centered.

Input	**Description (unit)**	**Std**
X_1	Stage altitude at separation Δa (m)	$\sigma_1 = 1.7 \times 10^2$
X_2	Velocity perturbation Δv (m.s^{-1})	$\sigma_2 = 3.7$
X_3	Flight path angle perturbation $\Delta\gamma$ (rad)	$\sigma_3 = 1.0 \times 10^{-3}$
X_4	Azimuth angle perturbation $\Delta\psi$ (rad)	$\sigma_4 = 1.8 \times 10^{-3}$
X_5	Propellant mass residual perturbation Δm (kg)	$\sigma_5 = 7.0 \times 10^1$
X_6	Drag force error perturbation ΔC_d (1)	$\sigma_6 = 1.0 \times 10^{-1}$

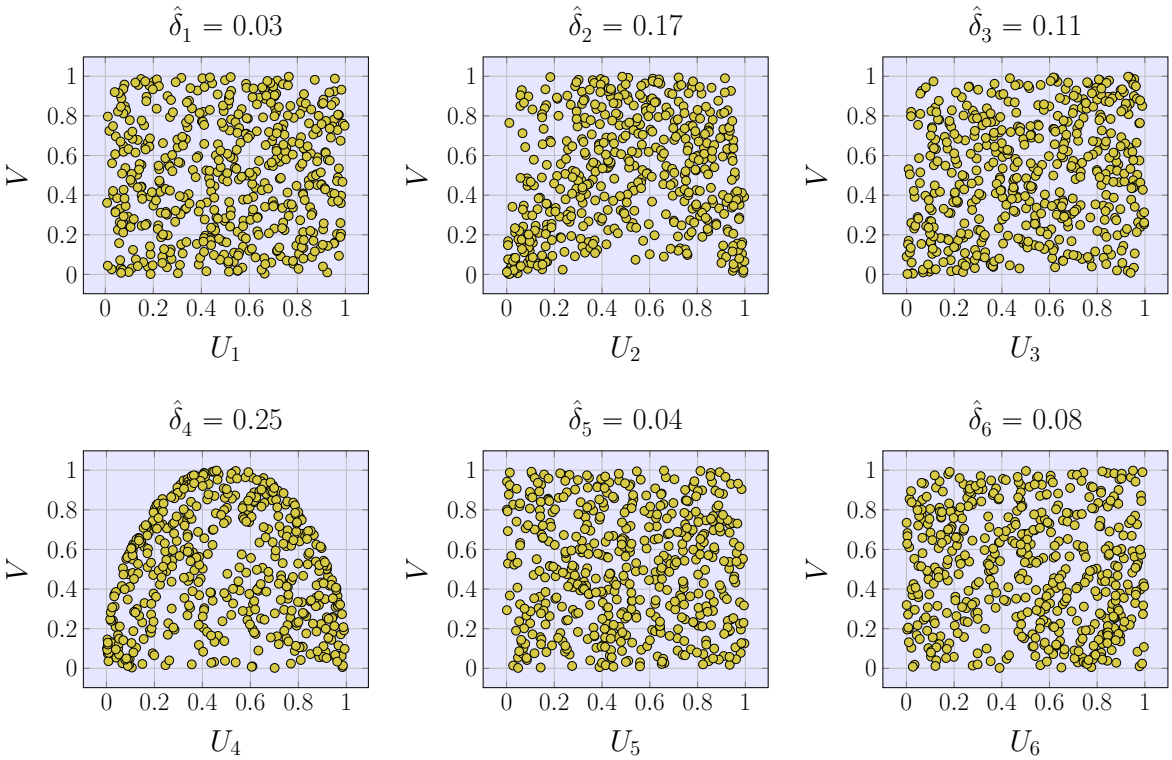
5.4.2. Dependence structure

To raise interest about the additional level of difficulty brought by the simulation code, pairwise dependencies are plotted in Figure 1. Remember that c_{X_iY} is nothing but the joint PDF arising from $(U_i\,, V)$ where $U_i = F_{X_i}(X_i)$ and $V = F_Y(Y)$. Of course, pair-copulas are only a first glimpse of the real dependence structure underlying $(\boldsymbol{X}, Y)$ and they do not allow to take definitive decision in terms of sensitivity analysis. One should recall the copula-based expression (18) and notice that any first-order index may be written as:

$$\delta_i = \frac{1}{2}\int_{\mathcal{H}_2} \Big| c_{\boldsymbol{X}_iY}\big(u_i\,, v\big) - 1 \Big| \,\mathrm{d}u_i\,\mathrm{d}v \;. \tag{40}$$

As a consequence, any pairplot that provides visual evidence that X_i and Y are almost independent must coincide with a small value of $\hat{\delta}_i$. It is confirmed in Figure 1 where $\hat{\delta}_1$, $\hat{\delta}_3$, $\hat{\delta}_5$ and $\hat{\delta}_6$ are classified as the four smallest first-order Borgonovo's indices. On the contrary, sloped data clusters are observable in the pairplot $(U_2\,, V)$, especially in the lower left corner and, in a lesser extent, in the lower right corner. It suggests that c_{X_2Y} features non-zero tail dependencies [75] that deserve specific attention [76]. Likewise, according to the fourth pairplot, c_{X_4Y} seems to lie within a strict subdomain of $\mathcal{H}_2$ because of some geometrical constraints coming from how $\mathcal{M}$ transforms $\boldsymbol{X}$ into Y. As may be intuited from visual considerations, $\hat{\delta}_4$ and $\hat{\delta}_2$ emerge as the two largest first-order indices in Figure 1.

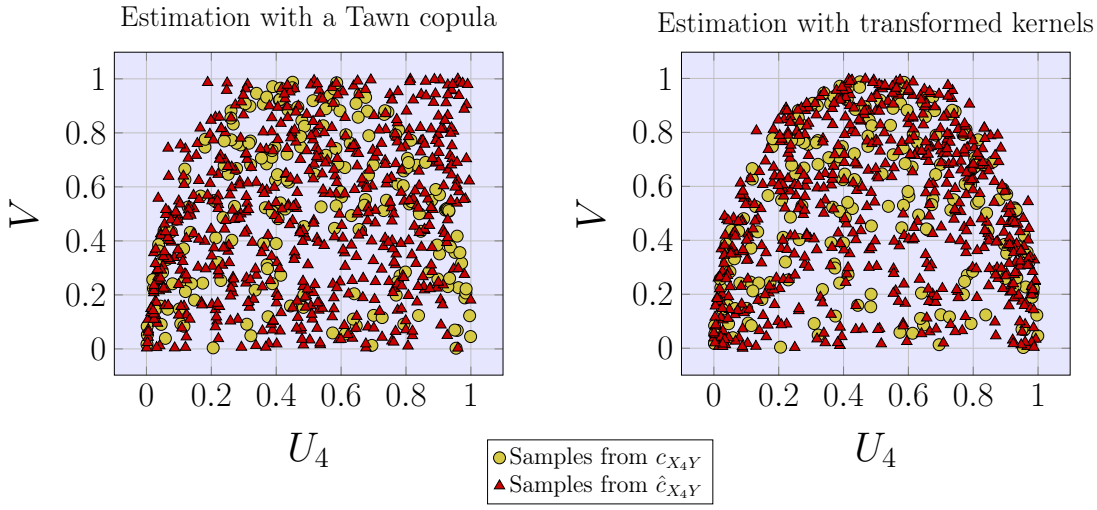
Figure 1: Pair-copula samples $(U_i^n, V^n) \sim c_{X_iY}$



In addition of the benchmark double-loop algorithm, two copula-based estimation methods are used in this test case to compute Shapley values. They are respectively denoted by $\hat{\mathrm{Sh}}_i^{\mathcal{C}_1}$ and $\hat{\mathrm{Sh}}_i^{\mathcal{C}_2}$. They only differ by the inference procedure they use to estimate copula PDFs. $\hat{\mathrm{Sh}}_i^{\mathcal{C}_1}$ resorts to parametric R-vines whereas $\hat{\mathrm{Sh}}_i^{\mathcal{C}_2}$ adopts a nonparametric variant based on the transformed KDE of all pair-copulas. The reason why a second copula-based estimator is deployed may be found in the fact that parametric R-vines are feared not to be enough flexible to capture properly the subtleties of the dependence structure brought into play by $\mathcal{M}$. As already touched upon, the dependence pattern observed for c_{X_4Y} looks unusual. The associated copula might be non-exchangeable [77], that is the PDF is not arranged symmetrically along one of the two bisectors. Most parametric copula families are exchangeable[1] and hence are very likely to fail to reconstruct c_{X_4Y}. After scanning all parametric families, the one maximizing Akaike information criterion is the non-exchangeable Tawn copula [78]. Even if this family is tailored to deal with non-exchangeability, one can see in Figure 2 that the estimated copula does not share the same support than initial simulated data. On the contrary, trying out a nonparametric approach gives compelling results since the fitted model precisely matches the contours of the targeted subdomain.

[1] A copula is said to be exchangeable if $\forall (u,v) \in \mathcal{H}_2, C(u,v) = C(v,u)$.

Figure 2: Copula estimation for the pair (X_4, Y)



5.4.3. Results

The computational procedure described in Appendix A is applied $M = 10$ times and the achieved results for all three estimation methods may be consulted in Table 10. Comparison with total-effect Sobol indices is also possible with Table 11. As in previous cases, the "Pick-and-Freeze" estimation scheme used to compute Sobol indices does not produce robust estimates for small indices. For instance, $\hat{S}_1$ and $\hat{S}_5$ show larger coefficients of variation compared with all other indices. The double-loop estimator $\hat{\mathrm{Sh}}_i^{\mathcal{K}}$ gives results which are in reasonable agreement with total-effect Sobol indices $\hat{S}_i^T$. X_2 is identified as the most influential input variable while X_3, X_4 and X_6 make up a second group of less impactful variables. As noticed in the previous test cases, Sobol indices tend to help the leading variable to pull away from others whereas Shapley values seem to share contributions with fewer differences between variables. One can notice that both copula-based methods are failing to correctly estimate Shapley values. For those estimators, the relevancy of Shapley value analysis could almost be questioned, with all variables being roughly of equal influence.

Table 10: Estimation of Shapley values when $\mathcal{M}$ is a realistic trajectory simulation code.

Inputs	$\hat{\mathrm{Sh}}_i^{\mathcal{K}}$			$\hat{\mathrm{Sh}}_i^{\mathcal{C}_1}$			$\hat{\mathrm{Sh}}_i^{\mathcal{C}_2}$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.0402	(5)	9.6	0.1417	(6)	2.9	0.1135	(6)	2.7
X_2	0.3150	(1)	6.6	0.2210	(1)	4.2	0.2261	(2)	3.6
X_3	0.2134	(3)	5.5	0.1775	(2)	5.5	0.1637	(3)	3.7
X_4	0.2425	(2)	13.0	0.1566	(4)	2.6	0.2494	(1)	1.6
X_5	0.0391	(6)	10.8	0.1438	(5)	3.4	0.1137	(5)	3.5
X_6	0.1498	(4)	8.6	0.1593	(3)	4.9	0.1336	(4)	4.5

Figure 3: Global method comparison for the estimation of Borgonovo's indices at all orders. It is recalled that (a) $\delta_{\varnothing} = 0$, (b) $\delta_{[d]} = 1$ and (c) $\boldsymbol{u} \subseteq \boldsymbol{v} \Rightarrow \delta_{\boldsymbol{u}} \leqslant \delta_{\boldsymbol{v}}$.

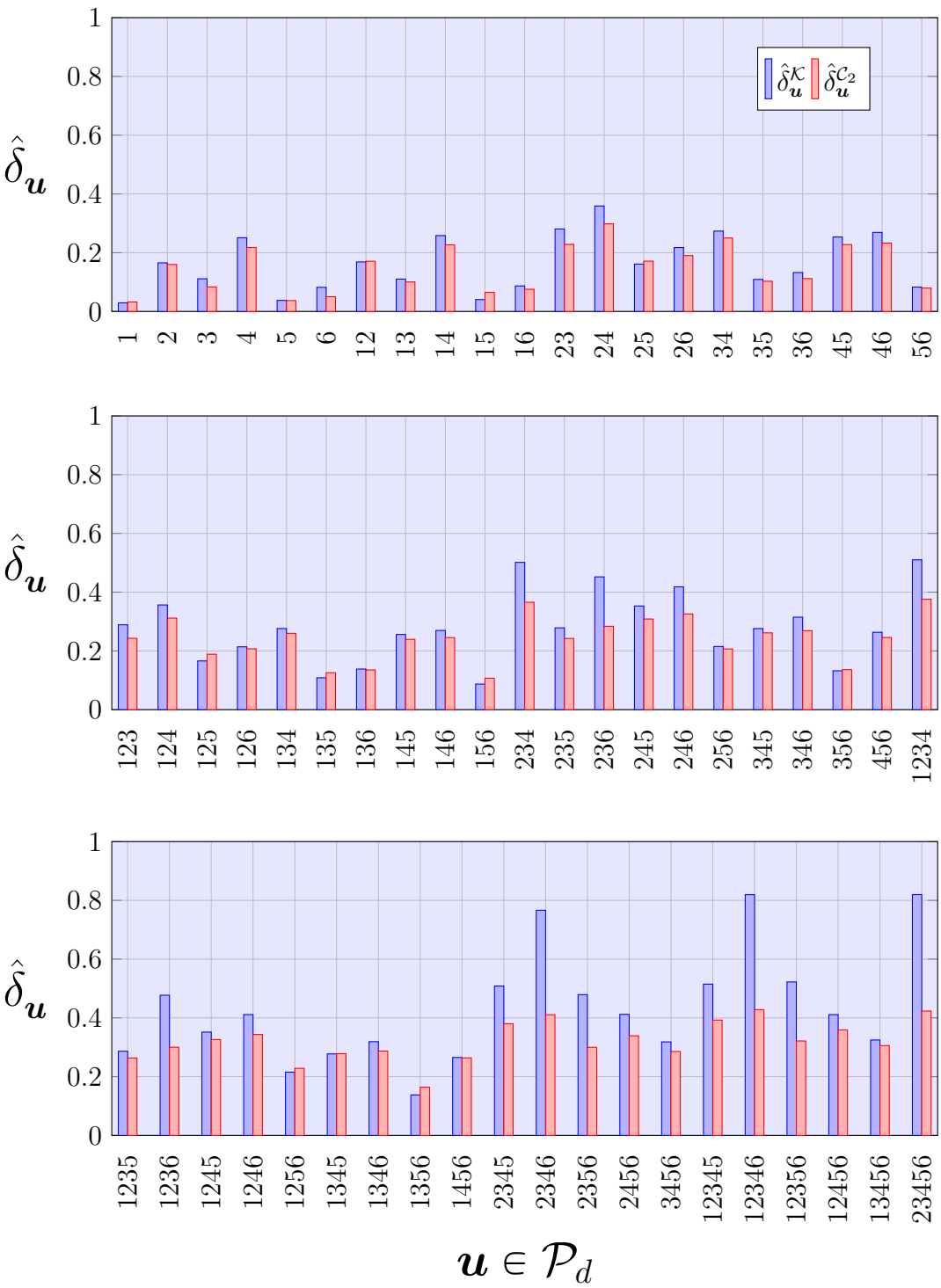


Table 11: Estimation of Sobol indices when $\mathcal{M}$ is a realistic trajectory simulation code.

Inputs	$\hat{S}_i$			$\hat{S}_i^T$		
	Mean	**Rank**	**CV**(%)	**Mean**	**Rank**	**CV**(%)
X_1	0.0009	(6)	143.2	0.0151	(6)	3.0
X_2	0.2649	(1)	1.1	0.6594	(1)	0.6
X_3	0.0664	(3)	4.6	0.3820	(2)	0.6
X_4	0.1871	(2)	1.5	0.2105	(4)	0.8
X_5	0.0012	(5)	106.3	0.0171	(5)	2.6
X_6	0.0186	(4)	20.6	0.2198	(3)	0.6

5.4.4. Analysis

In order to understand what is going on with Shapley values estimated with copula-based methods, a natural idea is to check whether all Borgonovo's indices are estimated properly. That is what is done in Figure 3 where the values obtained with $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{K}}$ and $\hat{\delta}_{\boldsymbol{u}}^{\mathcal{C}_2}$ are compared. One needs to investigate which are the δ-indices responsible for the lack of accuracy evidenced by copula-based estimators of Shapley values. Whether both estimators seem to match as long as $|\boldsymbol{u}| \leqslant 3$, it is no longer the case for some high-order indices. For instance, δ_{346} is well estimated by the copula-based method whereas δ_{2346} is not. It suggests that nonparametric copula estimation succeeds in learning $c_{X_{346}Y}$ but fails to reiterate with $c_{X_{2346}Y}$. Let us recall that Bayes rule for densities allows to write:

$$f_{\boldsymbol{X}_{2346}Y}(\boldsymbol{x}_{2346}, y) = f_{X_2Y|\boldsymbol{X}_{346}}(x_2, y \mid \boldsymbol{x}_{346}) \times f_{\boldsymbol{X}_{346}}(\boldsymbol{x}_{346}) \ . \tag{41}$$

Then, the conditional version of Sklar's theorem is applied on $f_{X_2Y|\boldsymbol{X}_{346}}$ and one has:

$$\begin{aligned} f_{X_2Y|\boldsymbol{X}_{346}}(x_2, y \mid \boldsymbol{x}_{346}) = f_{X_2|\boldsymbol{X}_{346}}(x_2 \mid \boldsymbol{x}_{346}) \times f_{Y|\boldsymbol{X}_{346}}(y \mid \boldsymbol{x}_{346}) \times \ldots \\ c_{X_2Y|\boldsymbol{X}_{346}}\Big(F_{X_2|\boldsymbol{X}_{346}}(x_2 \mid \boldsymbol{x}_{346}) \, , \, F_{Y|\boldsymbol{X}_{346}}(y \mid \boldsymbol{x}_{346}) \Big| \boldsymbol{x}_{346}\Big) \ . \end{aligned} \tag{42}$$

After noting that X_2 is independent from $\boldsymbol{X}_{3456}$, combining (41) and (42) yields:

$$\begin{aligned} f_{\boldsymbol{X}_{2346}Y}(\boldsymbol{x}_{2346}, y) = \left[\prod_{i \in \{2,3,4,6\}} f_{X_i}(x_i) \right] \times f_Y(y) \times \ldots \\ c_{X_2Y|\boldsymbol{X}_{346}}\Big(F_{X_2}(x_2) \, , \, F_{Y|\boldsymbol{X}_{346}}(y \mid \boldsymbol{x}_{346}) \Big| \boldsymbol{x}_{346}\Big) \times \ldots \\ c_{\boldsymbol{X}_{346}Y}\Big(F_{X_3}(x_3), F_{X_4}(x_4), F_{X_6}(x_6), F_Y(y)\Big) \ . \end{aligned} \tag{43}$$

The second and third lines in (43) account for the copula density term and thus $c_{\boldsymbol{X}_{2346}Y}$ and may be factorized as follows:

$$\begin{aligned} c_{\boldsymbol{X}_{2346}Y}(u_2, \ldots, u_6, v) = c_{\boldsymbol{X}_{346}Y}(u_3, u_4, u_6, v) \times \ldots \\ c_{X_2Y|\boldsymbol{X}_{346}}\Big(u_2 \, , \, h(u_3, \ldots, u_6, v) \Big| \underbrace{F_{X_3}^{-1}(u_3), \ldots, F_{X_6}^{-1}(u_6)}_{\boldsymbol{x}_{346}}\Big) \ , \end{aligned} \tag{44}$$

where:

$$h(u_3, u_4, u_6, v) = \int_0^v c_{\boldsymbol{X}_{346}Y}(u_3, u_4, u_6, t)\,\mathrm{d}t \ . \tag{45}$$

As pointed out just before, R-vine models are likely to learn very accurately $c_{\boldsymbol{X}_{346}Y}$. To derive an estimation of $c_{\boldsymbol{X}_{2346}Y}$, the factorization (44) shows that all boils down to learning the conditional pair-copula $c_{X_2Y|\boldsymbol{X}_{346}}$. For inference purposes, parametric and nonparametric R-vines handle conditional pair-copulas as if the conditioning process exerted no impact on the intrinsic dependency they hold. It is the so-called "simplifying assumption" previously discussed in Section 3. In the case of $c_{X_2Y|\boldsymbol{X}_{346}}$, it implies that the dependence structure between X_2 and Y remains unchanged regardless of the values $\boldsymbol{X}_{346}$ might take. To rule on the validity of such a risky assumption in the context under study, it would be necessary to observe several samples of the random pair $(X_2\,, Y)$ for different values of the conditioning set. This can be achieved by proceeding as indicated below:

1. Set $\boldsymbol{x}_{346} = (x_3, x_4, x_6) \in \mathbb{R}^3$.
2. Generate input samples of the remaining variables:

$$\left\{\boldsymbol{X}_{125}^n = \left(X_1^n, X_2^n, X_5^n\right)\right\}_{1 \leqslant n \leqslant N} \overset{\text{i.i.d.}}{\sim} f_{X_{125}} \ .$$

3. Compute the associated output samples:

$$Y^n = \mathcal{M}(X_1^n, X_2^n, x_3, x_4, X_5^n, x_6) \ .$$

4. Any pairwise observation (X_2^n, Y^n) needs to be rescaled:

$$U_2^n = F_{X_2}(X_2^n) \text{ and } V^n = \frac{1}{N+1}\sum_{i=1}^{N} \mathbb{1}_{\{Y^i \leqslant Y^n\}} \ .$$

5. Plot the resulting copula samples:

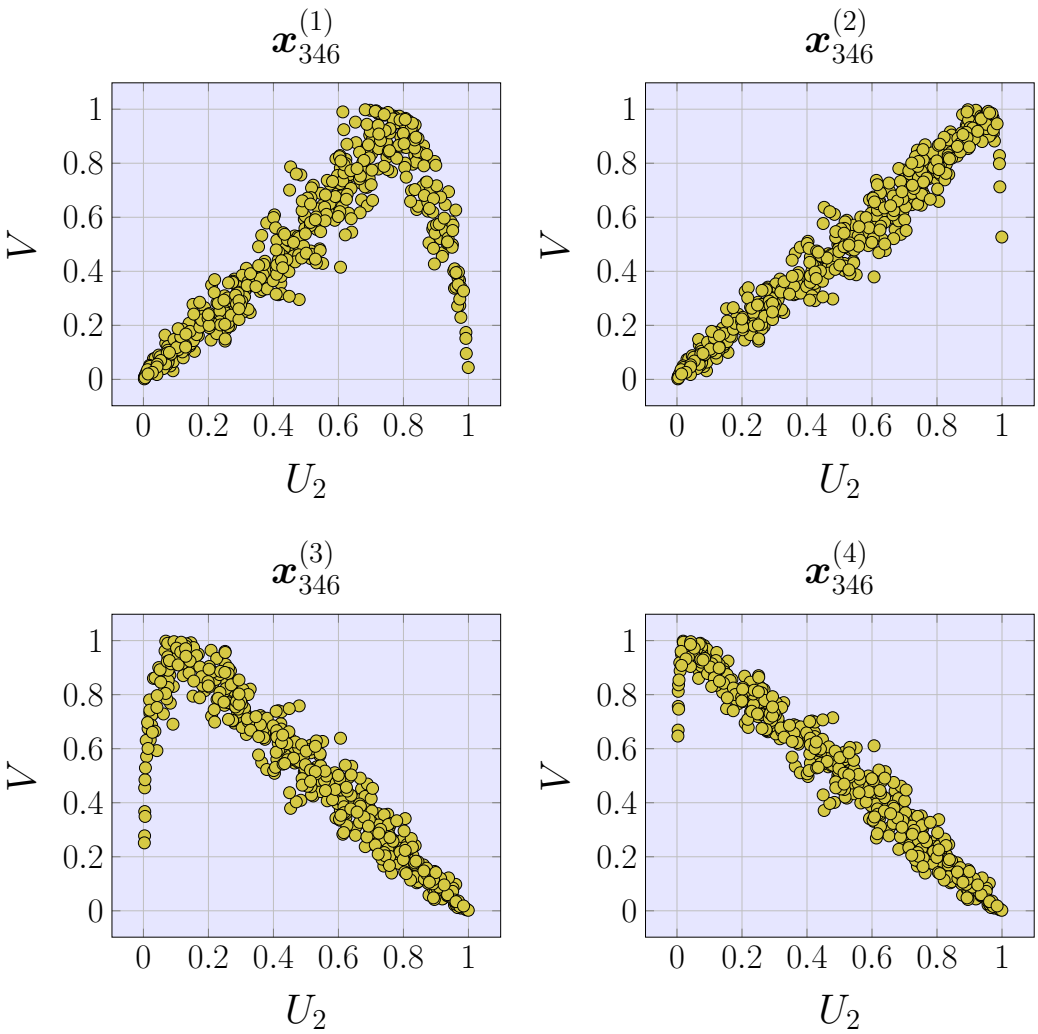
$$\left\{(U_2^n, V^n)\right\}_{1 \leqslant n \leqslant N} \ .$$

The procedure is repeated several times, as shown in Figure 4, and it is glaring clear that the dependence structure underlying $c_{X_2Y|\boldsymbol{X}_{346}}$ evolves according to the values given to covariates. It can thus be concluded that some of the copula distributions induced by the black-box model do not obey the simplifying assumption. Therefore, in this particular context, copula estimation with R-vine models can not be considered as a reliable approach.

The simplifying assumption has long aroused a lively debate [79, 80, 81], especially about how misleading it might be if wrongly accepted. Ongoing research initiatives to address this issue have to be noted. A review of recent advances in this field is available in [12]. The main ideas were put forth in the parametric framework and they ambition to let copula parameters depend on the covariates [82, 83, 84]. The copula factorization given by (21) becomes:

$$c_{\boldsymbol{X}}(\boldsymbol{u}) = \prod_{i=1}^{d-1}\prod_{e \in E_i} c_{j_e k_e|D_e}\left(u_{j_e|D_e}, u_{k_e|D_e}\middle|\theta\left(\boldsymbol{u}_{D_e}\right)\right) \ , \tag{46}$$

Figure 4: Conditional pair-copula samples $(U_2^n, V^n) \sim c_{X_2Y|\boldsymbol{X}_{346}}(\cdot \mid \boldsymbol{x})$ with $\boldsymbol{x} \in \left\{\boldsymbol{x}_{346}^{(i)}\right\}_{1 \leqslant i \leqslant 4}$



where $\theta\big(\boldsymbol{u}_{D_e}\big)$ stands for an arbitrary parametrization of the conditional pair-copula $c_{j_e k_e|D_e}$. Unfortunately, those improvements are highly insufficient to seize the entire volatility of the conditional copula $c_{X_2Y|\boldsymbol{X}_{346}}$ displayed at Figure 4. A fully non-simplified and nonparametric approach has to be used and it is yet to be explored.

5.4.5. Further discussion

This aerospace test case reveals the limits of the copula-based method in a context where the dependence structure extracted from real-life data contains intricacies. There may be a debate about the most relevant copula model in this situation. Even resorting to nonparametric R-vines is not always a quick-fix approach. On the contrary, in some cases, estimating a Gaussian copula works fine. In the current situation, selecting an appropriate copula distribution model is mainly based on graphical considerations and remains an open problem. A promising research avenue for future work would consist in coupling the copula-based estimation of high-order Borgonovo's indices with the wide literature on goodness-of-fit tests for copulas [35, 36]. In particular, goodness-of-fit tests have recently been designed for the special class of vine copulas [85] and shed new light on this topic compared with previous works based on the Rosenblatt transform [51]. The simplifying assumption can also be tested [86] for purposes of validation and, if it is rejected, this may indicate the need to deploy more sophisticated copula models.

6. Conclusion

In this paper, the authors advocate to compute Shapley values with Borgonovo's importance measure as cost function. The resulting sensitivity indicators appear to be an all-in-one solution since they are easily interpretable, take into account the entire output distribution and do not require independent inputs. However, estimating Borgonovo's indices at all orders is a mandatory first step. It can not be properly done with a double-loop estimator because it would force the simulation budget to be an exponential function of the problem dimension. To circumvent this pitfall, copula-based methods are gaining increasing interest since they allow to derive all Borgonovo's indices from a single input-output dataset. Nonetheless, one is brought back to the thorny topic of copula estimation in possibly high dimension. One important point of criticism about the copula-based approach lies in the heavy influence that the accuracy of copula estimation has on the resulting Borgonovo's indices. To date, even high-dimensional copula models are not able to guarantee convergence to the true underlying copula distribution without operating strong assumptions. In that context, copula model selection remains an open question that may be addressed with appropriate goodness-of-fit tests. It was shown that using multivariate copula distributions, in particular R-vine copulas, offers enough flexibility to some extent. The numerical study highlights that existing copula models fail to neatly estimate high-order Borgonovo's indices in the case of a complex aerospace dataset. The so-called "simplifying assumption" does not hold and yields misestimations. Furthermore, improvements or extensions of this work include the following points:

1. In the light of recent research effort to develop non-simplified regular vine copulas, there is no doubt that the shortcomings identified in the aerospace

application will be overcome very soon with the advent of even more flexible copula models.

2. As briefly discussed when examining the Gaussian linear model, one may regret that the estimation procedure of $c_{\boldsymbol{X}Y}$ is merely data-driven and does not better incorporate the information brought by $c_{\boldsymbol{X}}$ since:

$$c_{\boldsymbol{X}}(\boldsymbol{u}) = \int_0^1 c_{\boldsymbol{X}Y}(\boldsymbol{u}, t)\,\mathrm{d}t \ . \tag{47}$$

3. The copula-based estimation method could be easily extended to the case where the input distribution $f_{\boldsymbol{X}}$ is unknown and subject to statistical learning. The only extra task would consist in estimating the copula PDF between input variables. The resulting copula-based estimator would be given by:

$$\hat{\delta}^{\mathcal{C}}_{\boldsymbol{u}} = \frac{1}{2N}\sum_{k=1}^{N}\left|\hat{c}_{\boldsymbol{X}_{\boldsymbol{u}}Y}\big(U_1^k,\ldots,U_{r+1}^k\big) - \hat{c}_{\boldsymbol{X}_{\boldsymbol{u}}}\big(U_1^k,\ldots,U_r^k\big)\right| \ . \tag{48}$$

4. Shapley values can always be rewritten as a sum over all possible permutations of the players. Moreover, it was shown that only using a small set of randomly selected permutations is enough to produce accurate estimates of Shapley values. This idea was introduced in sensitivity analysis in order to reduce the number of Sobol indices that have to be estimated to compute Shapley values. The same could be done here with Borgonovo's δ-sensitivity measures instead of Sobol indices.

Acknowledgements The first and second authors contributed equally to this work. The first two authors are currently enrolled in a PhD program, respectively funded by ONERA - The French Aerospace Lab, the Occitanie region and Université Toulouse III - Paul Sabatier. Their financial supports are gratefully acknowledged. The authors also thank Florian Simatos (ISAE) for fruitful discussions.

Appendix A. Computational procedure

The whole computational procedure for the estimation of Shapley values in the numerical study is detailed hereafter.

1. A sequence of M input datasets $\boldsymbol{X}^{(k)}_{\mathbf{obs}}$ is generated and must be regarded as independent N-samples from the input joint distribution $f_{\boldsymbol{X}}$:

$$\boldsymbol{X}^{(k)}_{\mathbf{obs}} := \left[X_1^n,\ldots,X_d^n\right]^{(k)}_{1\leqslant n\leqslant N} \ .$$

2. The associated collections of Borgonovo's indices $\hat{\Delta}^{(k)}$ are estimated. To construct $\hat{\Delta}^{(k)}$, the only required material consists of $\boldsymbol{X}^{(k)}_{\mathbf{obs}}$ and $\mathcal{M}$:

$$\hat{\Delta}^{(k)} := \left\{\hat{\delta}^{(k)}_{\boldsymbol{u}} : \boldsymbol{u}\in\mathcal{P}_d\right\} \ .$$

3. Every Shapley value Sh_i is estimated M times, as expanded in (28):

$$\hat{\mathrm{Sh}}_i^{(k)} = \frac{1}{d} \sum_{\boldsymbol{u} \subset \{1,\ldots,d\} \setminus \{i\}} \binom{d-1}{|\boldsymbol{u}|}^{-1} \left(\hat{\delta}_{\boldsymbol{u} \cup \{i\}}^{(k)} - \hat{\delta}_{\boldsymbol{u}}^{(k)} \right) .$$

4. The empirical mean $\overline{\mathrm{Sh}}_i$ of each Shapley value estimator (MEAN column in the different result tables) $\hat{\mathrm{Sh}}_i$ is computed:

$$\overline{\mathrm{Sh}}_i = \frac{1}{M} \sum_{k=1}^{M} \hat{\mathrm{Sh}}_i^{(k)} .$$

5. In a similar way, the variability of each $\hat{\mathrm{Sh}}_i$ is assessed through classical unbiased estimators of both the variance and the coefficient of variation (CV column in the different result tables):

$$\hat{\sigma}_i^2 = \frac{1}{M-1} \sum_{1=1}^{M} \left(\hat{\mathrm{Sh}}_i^{(k)} - \overline{\mathrm{Sh}}_i \right)^2 \text{ and } \hat{\nu}_i = \frac{\hat{\sigma}_i}{\overline{\mathrm{Sh}}_i} .$$

References

[1] B. Iooss, P. Lemaître, A review on global sensitivity analysis methods, in: Uncertainty management in simulation-optimization of complex systems, Springer, 2015, pp. 101–122.

[2] E. Borgonovo, E. Plischke, Sensitivity analysis: a review of recent advances, European Journal of Operational Research 248 (2016) 869–887.

[3] E. Borgonovo, A new uncertainty importance measure, Reliability Engineering & System Safety 92 (2007) 771–784.

[4] E. Borgonovo, W. Castaings, S. Tarantola, Moment independent importance measures: New results and analytical test cases, Risk Analysis 31 (2011) 404–428.

[5] E. Plischke, E. Borgonovo, C. L. Smith, Global sensitivity measures from given data, European Journal of Operational Research 226 (2013) 536–550.

[6] Q. Liu, T. Homma, A new computational method of a moment-independent uncertainty importance measure, Reliability Engineering & System Safety 94 (2009) 1205–1211.

[7] P. Wei, Z. Lu, X. Yuan, Monte carlo simulation for moment-independent sensitivity analysis, Reliability Engineering & System Safety 110 (2013) 60–67.

[8] P. Derennes, J. Morio, F. Simatos, A nonparametric importance sampling estimator for moment independent importance measures, Reliability Engineering & System Safety 187 (2019) 3–16.

[9] L. Zhang, Z. Lu, L. Cheng, C. Fan, A new method for evaluating borgonovo moment-independent importance measure with its application in an aircraft structure, Reliability Engineering & System Safety 132 (2014) 163–175.

[10] P. Wei, Z. Lu, J. Song, Moment-independent sensitivity analysis using copula, Risk Analysis 34 (2014) 210–222.

[11] P. Derennes, J. Morio, F. Simatos, Estimation of moment independent importance measures using a copula and maximum entropy framework, in: 2018 Winter Simulation Conference (WSC), IEEE, 2018, pp. 1623–1634.

[12] C. Czado, Recent developments in vine copula based modeling, in: Analyzing Dependent Data with Vine Copulas, Springer, 2019, pp. 203–226.

[13] D. Kurowicka, R. M. Cooke, Uncertainty analysis with high dimensional dependence modelling, John Wiley & Sons, 2006.

[14] Y. Zhou, Z. Lu, Y. Shi, K. Cheng, A vine copula–based method for analyzing the moment-independent importance measure of the multivariate output, Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability (2018) 1748006X18781121.

[15] A. B. Owen, Sobol' indices and Shapley value, SIAM/ASA Journal on Uncertainty Quantification 2 (2014) 245–251.

[16] E. Borgonovo, Measuring uncertainty importance: investigation and comparison of alternative approaches, Risk analysis 26 (2006) 1349–1361.

[17] A. W. Bowman, A. Azzalini, Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations, volume 18, OUP Oxford, 1997.

[18] N.-B. Heidenreich, A. Schindler, S. Sperlich, Bandwidth selection for kernel density estimation: a review of fully automatic selectors, AStA Advances in Statistical Analysis 97 (2013) 403–433.

[19] T. J. Page Jr, Multivariate statistics: A vector space approach, JMR, Journal of Marketing Research (pre-1986) 21 (1984) 236.

[20] L. F. Shampine, Vectorized adaptive quadrature in matlab, Journal of Computational and Applied Mathematics 211 (2008) 131–140.

[21] W. Castaings, E. Borgonovo, M. Morris, S. Tarantola, Sampling strategies in density-based sensitivity analysis, Environmental Modelling & Software 38 (2012) 13–26.

[22] X. Luo, Z. Lu, X. Xu, A fast computational method for moment-independent uncertainty importance measure, Computer Physics Communications 185 (2014) 19–27.

[23] D. W. Scott, Multivariate density estimation: theory, practice, and visualization, John Wiley & Sons, 2015.

[24] M. Sklar, Fonctions de répartition à n dimensions et leurs marges, Publ. inst. statist. univ. Paris 8 (1959) 229–231.

[25] A. Der Kiureghian, P.-L. Liu, Structural reliability under incomplete probability information, Journal of Engineering Mechanics 112 (1986) 85–104.

[26] E. Plischke, E. Borgonovo, Copula-based sensitivity measures of computer experiments, Safety and Reliability of Complex Engineered Systems (2015).

[27] E. Plischke, E. Borgonovo, Copula theory and probabilistic sensitivity analysis: Is there a connection?, European Journal of Operational Research 277 (2019) 1046–1059.

[28] A. Charpentier, J.-D. Fermanian, O. Scaillet, The estimation of copulas: Theory and practice, Copulas: From theory to application in finance (2007) 35–64.

[29] C. Genest, A.-C. Favre, Everything you always wanted to know about copula modeling but were afraid to ask, Journal of hydrologic engineering 12 (2007) 347–368.

[30] P. Jaworski, F. Durante, W. K. Hardle, T. Rychlik, Copula theory and its applications, volume 198, Springer, 2010.

[31] S. Nadarajah, E. Afuecheta, S. Chan, A compendium of copulas, Statistica 77 (2018) 279–328.

[32] H.-B. Fang, K.-T. Fang, S. Kotz, The meta-elliptical distributions with given marginals, Journal of multivariate analysis 82 (2002) 1–16.

[33] A. J. McNeil, J. Nešlehová, et al., Multivariate archimedean copulas, d-monotone functions and l1-norm symmetric distributions, The Annals of Statistics 37 (2009) 3059–3097.

[34] M. Ribatet, M. Sedki, Extreme value copulas and max-stable processes, Journal de la Société Française de Statistique 154 (2013) 138–150.

[35] J.-D. Fermanian, et al., Goodness-of-fit tests for copulas, Journal of multivariate analysis 95 (2005) 119–152.

[36] C. Genest, B. Rémillard, D. Beaudoin, Goodness-of-fit tests for copulas: A review and a power study, Insurance: Mathematics and economics 44 (2009) 199–213.

[37] C. Genest, L.-P. Rivest, Statistical inference procedures for bivariate archimedean copulas, Journal of the American statistical Association 88 (1993) 1034–1043.

[38] I. Gijbels, J. Mielniczuk, Estimating the density of a copula function, Communications in Statistics-Theory and Methods 19 (1990) 445–464.

[39] K. Wen, X. Wu, Transformation-kernel estimation of the copula density, Preprint http://agecon2. tamu. edu/people/faculty/wu-ximing/agecon2/public/copula. pdf (2015).

[40] S. X. Chen, Beta kernel estimators for density functions, Computational Statistics & Data Analysis 31 (1999) 131–145.

[41] A. Sancetta, S. Satchell, The bernstein copula and its applications to modeling and approximations of multivariate distributions, Econometric theory 20 (2004) 535–562.

[42] O. Okhrin, Y. Okhrin, W. Schmid, Properties of hierarchical archimedean copulas, Statistics & Risk Modeling with Applications in Finance and Insurance 30 (2013) 21–54.

[43] F. Nielsen, Hierarchical clustering, in: Introduction to HPC with MPI for Data Science, Springer, 2016, pp. 195–211.

[44] O. Okhrin, Y. Okhrin, W. Schmid, On the structure and estimation of hierarchical archimedean copulas, Journal of Econometrics 173 (2013) 189–204.

[45] G. Mazo, S. Girard, F. Forbes, A class of multivariate copulas based on products of bivariate copulas, Journal of Multivariate Analysis 140 (2015) 363–376.

[46] E. Liebscher, Construction of asymmetric multivariate copulas, Journal of Multivariate analysis 99 (2008) 2234–2250.

[47] H. Joe, Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters, Lecture Notes-Monograph Series (1996) 120–141.

[48] T. Bedford, R. M. Cooke, Probability density decomposition for conditionally dependent random variables modeled by vines, Annals of Mathematics and Artificial intelligence 32 (2001) 245–268.

[49] T. Bedford, R. M. Cooke, Vines: A new graphical model for dependent random variables, The Annals of Statistics (2002) 1031–1068.

[50] U. Callies, D. Kurowicka, R. Cooke, Graphical models for the evaluation of multisite temperature forecasts: comparison of vines and independence graphs, Proc. of ESREL 2003, Safety and Reliability 1 (2003) 363–371.

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

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

[53] 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.

[54] T. Nagler, C. Schellhase, C. Czado, Nonparametric estimation of simplified vine copula models: comparison of methods, Dependence Modeling 5 (2017) 99–120.

[55] L. S. Shapley, A value for n-person games, Contributions to the Theory of Games 2 (1953) 307–317.

[56] A. B. Owen, C. Prieur, On shapley value for measuring importance of dependent inputs, SIAM/ASA Journal on Uncertainty Quantification 5 (2017) 986–1002.

[57] E. Song, B. L. Nelson, J. Staum, Shapley effects for global sensitivity analysis: Theory and computation, SIAM/ASA Journal on Uncertainty Quantification 4 (2016) 1060–1083.

[58] B. Iooss, C. Prieur, Shapley effects for sensitivity analysis with dependent inputs: comparisons with sobol'indices, numerical estimation and applications, arXiv preprint arXiv:1707.01334 (2017).

[59] F. Gamboa, P. Gremaud, T. Klein, A. Lagnoux, Global sensitivity analysis: a new generation of mighty estimators based on rank statistics (2020).

[60] E. Bouyé, V. Durrleman, A. Nikeghbali, G. Riboulet, T. Roncalli, Copulas for finance-a reading guide and some applications, Available at SSRN 1032533 (2000).

[61] A. J. McNeil, R. Frey, P. Embrechts, Quantitative risk management: concepts, techniques and tools-revised edition, Princeton university press, 2015.

[62] L. Hernández, J. Tejero, J. Vinuesa, Maximum likelihood estimation of the correlation parameters for elliptical copulas, arXiv preprint arXiv:1412.6316 (2014).

[63] C. Bucher, Y. Chen, G. Schuëller, Time variant reliability analysis utilizing response surface approach, in: Reliability and optimization of structural systems' 88, Springer, 1989, pp. 1–14.

[64] B. Echard, N. Gayton, M. Lemaire, Ak-mcs: an active learning reliability method combining kriging and monte carlo simulation, Structural Safety 33 (2011) 145–154.

[65] B. Echard, N. Gayton, M. Lemaire, N. Relun, A combined importance sampling and kriging reliability method for small failure probabilities with time-demanding numerical models, Reliability Engineering & System Safety 111 (2013) 232–240.

[66] N. Lelièvre, P. Beaurepaire, C. Mattrand, N. Gayton, Ak-mcsi: A kriging-based method to deal with small failure probabilities and time-consuming models, Structural Safety 73 (2018) 1–11.

[67] F. Gamboa, A. Janon, T. Klein, A. Lagnoux, C. Prieur, Statistical inference for sobol pick-freeze monte carlo method, Statistics 50 (2016) 881–902.

[68] E. Myshetskaya, et al., Monte carlo estimators for small sensitivity indices, Monte Carlo Methods and Applications 13 (2008) 455–465.

[69] A. B. Owen, Better estimation of small sobol'sensitivity indices, ACM Transactions on Modeling and Computer Simulation (TOMACS) 23 (2013) 1–17.

[70] W. V. Harper, S. K. Gupta, Sensitivity/uncertainty analysis of a borehole scenario comparing Latin hypercube sampling and deterministic sensitivity approaches, Office of Nuclear Waste Isolation, Battelle Memorial Institute, 1983.

[71] M. D. Morris, T. J. Mitchell, D. Ylvisaker, Bayesian design and analysis of computer experiments: use of derivatives in surface prediction, Technometrics 35 (1993) 243–255.

[72] J. An, A. Owen, Quasi-regression, Journal of complexity 17 (2001) 588–607.

[73] B. A. Worley, Deterministic uncertainty analysis, Technical Report, Oak Ridge National Lab., TN (USA), 1987.

[74] P. Derennes, V. Chabridon, J. Morio, M. Balesdent, F. Simatos, J.-M. Bourinet, N. Gayton, Nonparametric importance sampling techniques for sensitivity analysis and reliability assessment of a launcher stage fallout, in: Modeling and Optimization in Space Engineering, Springer, 2019, pp. 59–86.

[75] G. G. Venter, Tails of copulas, in: Proceedings of the Casualty Actuarial Society, volume 89, 2002, pp. 68–113.

[76] G. Frahm, M. Junker, R. Schmidt, Estimating the tail-dependence coefficient: properties and pitfalls, Insurance: mathematics and Economics 37 (2005) 80–100.

[77] F. Durante, Construction of non-exchangeable bivariate distribution functions, Statistical Papers 50 (2009) 383–391.

[78] J. A. Tawn, Bivariate extreme value theory: models and estimation, Biometrika 75 (1988) 397–415.

[79] I. H. Haff, K. Aas, A. Frigessi, On the simplified pair-copula construction—simply useful or too simplistic?, Journal of Multivariate Analysis 101 (2010) 1296–1310.

[80] J. Stoeber, H. Joe, C. Czado, Simplified pair copula constructions—limitations and extensions, Journal of Multivariate Analysis 119 (2013) 101–118.

[81] F. Spanhel, M. S. Kurz, Simplified vine copula models: Approximations based on the simplifying assumption, arXiv preprint arXiv:1510.06971 715 (2015).

[82] D. Han, K. S. Tan, C. Weng, Vine copula models with glm and sparsity, Communications in Statistics-Theory and Methods 46 (2017) 6358–6381.

[83] T. Vatter, T. Nagler, Generalized additive models for pair-copula constructions, Journal of Computational and Graphical Statistics 27 (2018) 715–727.

[84] C. Schellhase, F. Spanhel, Estimating non-simplified vine copulas using penalized splines, Statistics and Computing 28 (2018) 387–409.

[85] U. Schepsmeier, A goodness-of-fit test for regular vine copula models, Econometric Reviews 38 (2019) 25–46.

[86] M. S. Kurz, F. Spanhel, Testing the simplifying assumption in high-dimensional vine copulas, arXiv preprint arXiv:1706.02338 (2017).