
Accurate Shapley Values for explaining tree-based models

Salim I. Amoukou
University Paris Saclay
LaMME
Stellantis

Tangi Salaün
Quantmetry

Nicolas J.B. Brunel
University Paris Saclay, ENSIIE
LaMME
Quantmetry

Abstract

Although Shapley Values (SV) are widely used in explainable AI, they can be poorly understood and estimated, implying that their analysis may lead to spurious inferences and explanations. As a starting point, we remind an invariance principle for SV and derive the correct approach for computing the SV of categorical variables that are particularly sensitive to the encoding used. In the case of tree-based models, we introduce two estimators of Shapley Values that exploit the tree structure efficiently and are more accurate than state-of-the-art methods. Simulations and comparisons are performed with state-of-the-art algorithms and show the practical gain of our approach. Finally, we discuss the ability of SV to provide reliable local explanations. We also provide a Python package that compute our estimators at <https://github.com/salimamoukou/acv00>.

1 Introduction

The explainability and interpretability of Machine Learning (ML) models are now central topics in Machine Learning Research due to their increasing ubiquity in Industry, Business, Sciences, and Society. As ML models are usually considered as black-box models, scientists, practitioners, and citizens call for the development of tools that could provide better insights into the important variables in a prediction or in identifying biases for some individuals, or sub-groups. Typically, standard global importance measures such

as permutation importance measures (Breiman, 2001) are not sufficient for explaining individual or local predictions, and new methodologies are developed in the very active field of Explainable AI (XAI). Indeed, various local explanations have been proposed, focusing on model-agnostic methods that can be applied to the most successful ML models, typically ensemble methods (such as random forests, gradient boosted trees) and deep learning. The most used are, for instance, Partial Dependence Plot (Friedman, 2001), Individual Conditional Expectation (Goldstein et al., 2015), and local feature attributions such as Local Surrogate (LIME) (Ribeiro et al., 2016). With the same objective in mind, the Shapley Values (Shapley, 1953), a concept primarily developed in Cooperative Game Theory, has been adapted to XAI for evaluating the "fair" contribution of a variable $X_i = x_i$ in a prediction (Strumbelj and Kononenko, 2010; Lundberg and Lee, 2017). The Shapley Values (SV) are now massively used for identifying important variables at a local and global scale. As remarked by Lundberg et al. (2020); Covert et al. (2020b), a lot of importance measures aim at analyzing the behavior of a prediction model f based on p features X_1, \dots, X_p by removing variables and considering reduced predictors. Typically, for any group of variables $\mathbf{X}_S = (X_i)_{i \in S}$, with any subset $S \subseteq \llbracket 1, p \rrbracket$ and reference distribution $Q_{S, \mathbf{x}}$, reduced predictors are defined as:

$$f_S(\mathbf{x}_S) \triangleq E_{Q_{S, \mathbf{x}}} [f(\mathbf{x}_S, \mathbf{X}_{\bar{S}})], \quad (1)$$

where $Q_{S, \mathbf{x}}$ is the conditional distribution $P(\mathbf{X}_{\bar{S}} | \mathbf{X}_S = \mathbf{x}_S)$. Other SV can be defined with the marginal probabilities but their interpretation is different (Heskes et al., 2020; Janzing et al., 2020; Chen et al., 2020) and there are still active debates on using or not conditional probabilities (Frye et al., 2020): we consider only conditional SV as, in this case, the estimation is very challenging. The SV for local explanations at \mathbf{x} have been introduced in (Lundberg and Lee, 2017) and are based on a cooperative game with value function $v(f; S) \triangleq f_S(\mathbf{x}_S)$. For

any group of variables $C \subseteq \llbracket 1, p \rrbracket$ and $k \in \llbracket 1, p - |C| \rrbracket$, we denote the set $\mathcal{S}_k(C) = \{S \subseteq \llbracket 1, p \rrbracket \setminus C \mid |S| = k\}$ and we introduce a straightforward generalization of the SV for coalition C as

$$\phi_C(f; \mathbf{x}) \triangleq \frac{1}{p - |C| + 1} \sum_{k=0}^{p-|C|} \frac{1}{\binom{p-|C|}{k}} \sum_{S \in \mathcal{S}_k(C)} (f_{S \cup C}(\mathbf{x}_{S \cup C}) - f_S(\mathbf{x}_S)). \quad (2)$$

This definition of the Shapley Value is a generalization of the classical SV for one variable: if we consider the singleton $C = \{i\}$ for $i \in \llbracket 1, p \rrbracket$, we recover the standard definition for "player X_i ". In the next section, we show how the definition 2 appears naturally for measuring the impact of a group of variables C , and in particular categorical variables.

Our paper proposes several solutions to the problem of the computation and the estimation of the Shapley Values $\phi_i(f; \mathbf{x})$ which is an active subject. We focus on tree-based models only as the computational cost is reduced and the statistical problem is easier to address. Indeed, we show that the current state-of-the-art algorithm for tree-based models that is Tree SHAP (Lundberg et al., 2020) is highly biased when the features are dependent. Thus, we improve the estimation of the SV by statistically principled estimators. In addition, we address the theoretical computation of SV for categorical variables when we use standard encodings, which motivates the use of equation 2. In particular, we show that the true SV of the categorical variable is different from the sum of SV of encoded variables. Moreover, using the sum of the encoded variables as the SV of a categorical variable provides wrong estimates of all the SV in the model and implies spurious interpretations. Note that this is currently the only way to handle categorical variables with Tree SHAP. Therefore, we highlighted the correct way of computing the SV of encoded variables and implemented it with our estimators. Our contributions are implemented in a Python package¹.

The paper is organized as follows. In the next section, we derive invariance principles for SV under reparametrization or encoding that is particularly useful for dealing with categorical variables. In section 3, we introduce two estimators of reduced predictors and SV. In section 4, we highlight the improvement over the dependent Tree SHAP. Finally, we discuss the ability of SV to provide reliable local explanations.

2 Coalition and Invariance for Shapley Values

We derive in this section a unifying property of invariance for the Shapley Values of continuous and categorical variables: the explanation provided by a variable should not depend on the way it is coded in a model. We show that this invariance property gives a natural way of computing the SV of categorical variables based on the notion of coalition and the general definition given in Eq. 2. This is useful in our case, as we are also interested in the discretization of continuous variables to facilitate the estimation of Shapley Values and enhance their stability, as we will see in section 3.

2.1 Invariance under reparametrization for continuous variables

From the definition 1 of the reduced predictor, there is no constraint on the dimension of X_i . We suppose that the p variables are vector-valued i.e., $X_i \in \mathbb{R}^{p_i}$, $p_i \geq 1$ and that they have a density g_i . We assume that we transform each variable X_i with a diffeomorphism $\varphi_i : \mathbb{R}^{p_i} \rightarrow \mathbb{R}^{p_i}$. We introduce the transformed variables $U_i \triangleq \varphi_i(X_i)$ and the reparametrized model \tilde{f} defined by $\tilde{f}(U_1, \dots, U_p) = f(X_1, \dots, X_p)$, i.e., $\tilde{f}(u_1, \dots, u_p) = f \circ \varphi^{(-1)}(\mathbf{u})$ where $\varphi = (\varphi_1, \dots, \varphi_p)$. In general, we cannot relate the predictor f_x learned from the real data set $\mathcal{D}_x^{Train} = \{(\mathbf{x}_i, y_i), i \in \llbracket 1, n \rrbracket\}$ to the predictor f_u learned from $\mathcal{D}_u^{Train} = \{(\mathbf{u}_i, y_i), i \in \llbracket 1, n \rrbracket\}$ (y is the label to predict). Indeed, estimation procedures are not invariant with respect to reparametrization that's why we obtain different predictors after "diffeomorphic feature engineering": $f_u \neq f_x \circ \varphi$. For this reason, we focus only on the impact of reparametrization on explanations, and we show below that the Shapley Values are invariant.

Proposition 2.1. *Let f and $\tilde{f} = f \circ \varphi^{(-1)}$ its reparametrization, then we have $\forall i \in \llbracket 1, p \rrbracket$, and $\mathbf{u} = \varphi(\mathbf{x})$:*

$$\phi_i(f, \mathbf{x}) = \phi_i(\tilde{f}, \varphi(\mathbf{x})).$$

We refer to Appendix A for detailed derivations. This identity indicates that the information provided by each feature X_i in the explanation does not depend on any encoding, as mentioned by Covert et al. (2020a).

Suppose we transformed the variables by some feature engineering. In that case, we will keep the same SV $\phi_i(f, \mathbf{x})$. Another interest of identity (2.1) is to show that the SV depends essentially on the dependence structure of the features X_i .

¹<https://github.com/salimamoukou/acv00>.

2.2 Invariance for encoded categorical variable

In the rest of the paper, continuous predictive variables are denoted with X , categorical predictive variables are denoted with Z , and the output to predict is denoted Y . There exist numerous encodings for a categorical variable Z with modalities $\{1, \dots, K\}$. Still, we focus on methods related to One-Hot-Encoding (OHE), and Dummy Encoding (DE) that corresponds to the introduction of some indicator variables Z_k ($Z_k = 1$ if $Z = k$, 0 otherwise). Contrary to the continuous case, the introduction of indicators changes the number of "players" in the game defined for computing the Shapley Value. Unlike the diffeomorphic reparametrization, this change has dramatic consequences on the computation of the SV of all the variables in the models. As a consequence, the widespread practice that recommends summing the SV of the indicator variables Z_k for computing the SV of Z is not justified and false in general: if we want to benefit from a similar invariance result to proposition 2.1, we need to deal with the coalition of indicators and use the general expression of SV introduced in Eq. 2.

For the sake of simplicity, we assume that the model has only two variables $\mathbf{X} = (X, Z)$, where $X \in \mathbb{R}$ and $Z = 1, \dots, K$ is a categorical variable. The efficiency property of SV gives the decomposition

$$f(x, z) - E_P[f(X, Z)] = \phi_X(f; x, z) + \phi_Z(f; x, z) \quad (3)$$

In order to establish the link between the SV of the indicator variables Z_k and the SV of the variable Z , we need more notations. We focus on the Dummy Encoding (DE) $\varphi : z \mapsto (z_1, \dots, z_{K-1})$. The variables $(X, Z_{1:K-1})$ are defined on $\mathbb{R} \times \{0, 1\}^{K-1}$, its distribution \tilde{P} is the image probability of P induced by the transformation φ . The initial predictor $f : \mathbb{R} \times \{1, \dots, K\} \rightarrow \mathbb{R}$ is reparametrized as a function $\tilde{f} : \mathbb{R} \times \{0, 1\}^{K-1} \rightarrow \mathbb{R}$ such that $f(X, Z) \triangleq \tilde{f}(X, Z_1, \dots, Z_{K-1})$. The function \tilde{f} is not completely defined for all $(z_1, \dots, z_{K-1}) \in \{0, 1\}^{K-1}$ and is only defined \tilde{P} -almost everywhere because of the deterministic dependence $\sum_{k=1}^{K-1} Z_k \leq 1$. Consequently, we need to extend \tilde{f} to the whole space $\mathcal{X} \times \{0, 1\}^{K-1}$ by setting $\tilde{f}(x, z_1, \dots, z_{K-1}) = 0$ as soon as $\sum_{k=1}^{K-1} z_k > 1$. For the predictor $\tilde{f}(X, Z_1, \dots, Z_{K-1})$, we can compute the SV of X, Z_1, \dots, Z_{K-1} and obtain the following decomposition thank to the efficiency property

$$\begin{aligned} & \tilde{f}(x, z_{1:K-1}) - E_{\tilde{P}}[\tilde{f}(X, Z_{1:K-1})] \\ &= \phi_X(\tilde{f}; x, z_{1:K-1}) + \sum_{k=1}^{K-1} \phi_{Z_k}(\tilde{f}; x, z_{1:K-1}) \quad (4) \end{aligned}$$

where $\phi_{Z_k}(\tilde{f}; x, z_{1:K-1})$ are the SV of the variable Z_k computed with distribution \tilde{P} . Consequently, we have

$$\begin{aligned} \phi_X(f; x, z) + \phi_Z(f; x, z) &= \phi_X(\tilde{f}; x, z_{1:K-1}) \\ &+ \sum_{k=1}^{K-1} \phi_{Z_k}(\tilde{f}; x, z_{1:K-1}) \quad (5) \end{aligned}$$

In general, we have $\phi_Z(f; x, z) \neq \sum_{k=1}^{K-1} \phi_{Z_k}(\tilde{f}; x, z_{1:K-1})$, because the SV depends on the number of variables. We show in the next proposition that $\phi_Z(f; x, z) = \phi_{Z_C}(\tilde{f}; x, z_C)$ where ϕ_{Z_C} is computed with equation 2 and C is the coalition of variables (Z_1, \dots, Z_{K-1}) .

Proposition 2.2. *For all $x \in \mathcal{X}$, and if $z_{1:K-1} = \varphi(z)$ then*

$$\begin{cases} \phi_{Z_C}(\tilde{f}; x, z_C) &= \phi_Z(f; x, z) \\ \phi_X(\tilde{f}; x, z_C) &= \phi_X(f; x, z), \end{cases} \quad (6)$$

where $\phi_X(\tilde{f}; x, z_C)$ is the SV of X when the variables (Z_1, \dots, Z_{K-1}) are considered as a single variable. We refer to Appendix A for detailed derivations. In general, for cooperative games, the SV of a coalition $\phi_{Z_C}(\tilde{f}; x, z_C)$ is different from the sum of individual SV $\sum_{k \in C} \phi_{Z_k}(\tilde{f}; x, z_{1:K-1})$. We note that we can compute two different SV for X when we use the encoded predictor \tilde{f} : $\phi_X(\tilde{f}; x, z_C)$ and $\phi_X(\tilde{f}; x, z_{1:K-1})$. These two SV are different in general as they involve different number of variables and different conditional expectations. Proposition 2.2 shows that we should prefer $\phi_X(\tilde{f}; x, z_C)$ as it is equal to the theoretical SV given in equation 3.

2.3 Coalition or Sum: numerical comparisons

We give numerical examples illustrating the differences between coalition or sum and the corresponding explanations. We consider a linear predictor f , with 1 categorical and 3 continuous variables $\mathbf{X} = (X_1, X_2, X_3)$, defined as $f(\mathbf{X}, Z) = B_Z \mathbf{X}$ with $\mathbf{X}|Z = z \sim \mathcal{N}(\mu_z, \Sigma_z)$ and $\mathbb{P}(Z = z) = \pi_z$, $Z \in \{a, b, c\}$. The values of the parameters used in our experiments are found in Appendix G. In figure 1, we remark that the SV change dramatically for a single observation. The sign changes given the encoding (DE or OHE) and is often different from the sign of the true SV of Y without encoding. We can also note important differences in the SV of the quantitative variable \mathbf{X} .

To quantify the global difference of the different methods, we compute the relative absolute error (R-AE) of the SV defined as:

$$\text{R-AE}(f, \tilde{f}) = \sum_{i=1}^p \frac{|\phi_i(f; \mathbf{x}) - \phi_i(\tilde{f}; \mathbf{x})|}{|\phi_i(f; \mathbf{x})|} \quad (7)$$

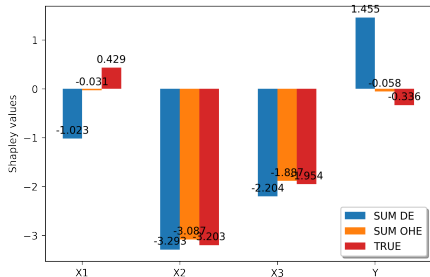


Figure 1: SV with or without encoding (OHE - DE) for observation $x = [0.35, -1.61, -0.11]$, $y = a$

We compute the SV of 1000 observations of the synthetic dataset. We observe in figure 2 that the differences can be huge for almost all samples (DE is much worse than OHE in this example). Thus, we highly recommend using the coalition as it is consistent with the true SV contrary to the sum. More examples on real datasets can be found in Appendix D.

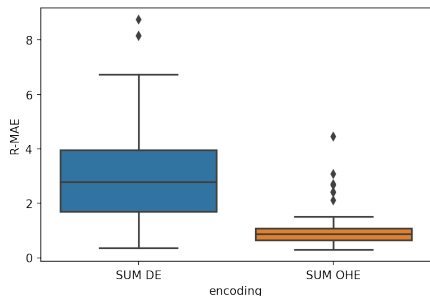


Figure 2: R-AE distribution between the SV without encoding and the corresponding encodings

3 Shapley Values for tree-based models

There are two challenges for the computation of SV: the combinatorial explosion with 2^p coalitions to consider and the estimation of the conditional expectations $f_S(\mathbf{x}_S) = E[f(\mathbf{X}) | \mathbf{X}_S = \mathbf{x}_S]$, $S \subseteq \llbracket 1, p \rrbracket$. In current approaches, the estimation relies on several approximations and sampling procedures that assume independence (Lundberg and Lee, 2017; Covert and Lee, 2020) or more recently Aas et al. (2020, 2021) proposed to model the features with a gaussian distribution or vine copula to draw samples from the conditional distributions. Besides, Williamson and Feng (2020) trained one model for each selected subset S of variables that is accurate but computationally costly. Moreover, their final objective differs from ours since we are interested in local estimates and exact com-

putations (i.e., no sampling of the subsets). Thus, we focus on tree-based models, as it has been exploited by Lundberg et al. (2020) for deriving an algorithm Tree SHAP for exact computing of SV: we can compute all the terms (no sampling of the subsets $S \subseteq \llbracket 1, p \rrbracket$) and the estimation of the conditional expectations is simplified. After a brief presentation of the limitations of Tree SHAP, we introduce two new estimators that use the tree structure. For the sake of simplicity, we do not consider ensemble of trees (Random Forests, Gradient Tree Boosting,...) as the extension of our estimators to these more complex models is straightforward by linearity.

3.1 Algorithms for computing Conditional Expectations and the Tree SHAP algorithm

We consider a tree-based model f defined on \mathbb{R}^p (categorical variables are one-hot encoded). We have $f(\mathbf{x}) = \sum_{m=1}^M f_m \mathbb{1}_{L_m}(\mathbf{x})$ where L_m represents a leaf. The leaves form a partition of the input space, and each leaf can be written as $L_m = \prod_{i=1}^p [a_i^m, b_i^m]$ (with $-\infty \leq a_i^m < b_i^m \leq +\infty$). Alternatively, we write the leaf with the decision path perspective: a leaf L_m is defined by a sequence of decision based on d_m variables X_{N_k} , $k = 1, \dots, d_m$. For each node N_k in the path of the leaf L_m , we associate the region I_{N_k} (defined by a split: it is either $]-\infty, t_k]$ or $[t_k, +\infty[$) and the leaf can be rewritten as

$$L_m = \{\mathbf{x} \in \mathbb{R}^p : x_{N_1} \in I_{N_1}, \dots, x_{N_{d_m}} \in I_{N_{d_m}}\}. \quad (8)$$

A crucial point is to identify the set of leaves compatible with the condition $\mathbf{X}_S = \mathbf{x}_S$: we can partition the leaf according to a coalition S : $L_m = L_m^S \times L_m^{\bar{S}}$ with $L_m^S = \prod_{i \in S} [a_i^m, b_i^m]$ and $L_m^{\bar{S}} = \prod_{i \in \bar{S}} [a_i^m, b_i^m]$. Thus, for each condition $\mathbf{X}_S = \mathbf{x}_S$ the set of compatible leaves of $\mathbf{x} = (\mathbf{x}_S, \mathbf{x}_{\bar{S}})$ is

$$\begin{aligned} C(S, \mathbf{x}) &= \{m \in [1 \dots M] | \mathbf{x}_S \in L_m^S\} \\ &= \{m \in [1 \dots M] | x_{N_i} \in I_{N_i}, N_i \in S\} \end{aligned}$$

and the reduced predictor $f_S(\mathbf{x}_S)$ has the simple expression

$$f_S(\mathbf{x}_S) = \sum_{m \in C(S, \mathbf{x})} f_m P_X(L_m | \mathbf{X}_S = \mathbf{x}_S)$$

When we have a model for P_X from which we can derive a conditional density and evaluate directly the conditional probabilities $P_X(L_m | \mathbf{X}_S = \mathbf{x}_S)$, we can have an exact computation. This is typically the case when $\mathbf{X} \sim \mathcal{N}(\mu, \Sigma)$ and we can integrate the densities for deriving the conditional probabilities $P_X\left(\prod_{k=1}^{d_m} I_{N_k} | \mathbf{X}_S = \mathbf{x}_S\right)$. The derivation of conditional probabilities can become challenging, and assumptions about the factorization of the distribution

can accelerate the computation: in (Lundberg et al., 2020), the authors introduce a recursive algorithm (Tree SHAP with path-dependent feature perturbation, Algorithm 1) that assumes that the probabilities for every compatible leaf L_m can be factored with the decision tree:

$$P_X^{SHAP} \left(\prod_{k=1}^{d_m} I_{N_k} | \mathbf{X}_S = \mathbf{x}_S \right) = \delta_S(N_1) \times \prod_{i=2|N_i \notin S}^{d_m} P \left(X_{N_i} \in I_{N_i} \mid \prod_{k=2|N_k \notin S}^i X_{N_{k-1}} \in I_{N_{k-1}} \right) \quad (9)$$

with $\delta_S(N_1) = P(X_{N_1} \in I_{N_1})$ if $N_1 \notin S$, and 1 otherwise. The underlying assumption in equation (9) is that we have a Markov chain defined by the path in the tree, and the transition probabilities are estimated conditionally on $\{\mathbf{X}_S = \mathbf{x}_S\}$, e.g., each probability is replaced by 1 if $N_i \in S$, see algorithm description in Appendix C. As we will see in the simulations, this assumption is not satisfied in general and we can observe a bias in the estimation produced by this algorithm. We denote \hat{f}_S^{SHAP} and $\phi_i(\hat{f}_S^{SHAP}; \mathbf{x})$ the corresponding estimators. Therefore, we propose two estimators that do not make assumptions on the probability P_X .

3.2 Statistical Estimation of Conditional Expectations

Discrete case We want to solve the statistical problem of estimating probabilities from the dataset $\mathcal{D}_x^{Train} \sim P_X$. We do not assume the existence of the density or probability $p(\mathbf{x})$ as in Aas et al. (2020, 2021). We have $\mathcal{D}_x^{Explain}$ that corresponds to the (new) individuals on which we want to compute SV.

We first assume that all the variables are categorical: in that case, we can estimate directly $P_X(L_m | \mathbf{X}_S = \mathbf{x}_S)$. For every $\mathbf{x} \in \mathcal{D}_x^{Explain}$, a straightforward estimation is based on $N(\mathbf{x}_S)$: the number of observations in \mathcal{D}_x^{Train} such that $\mathbf{X}_S = \mathbf{x}_S$ (across all the leaves of the tree) and $N(L_m, \mathbf{x}_S)$: the number of observations of \mathcal{D}_x^{Train} in leaf L_m that satisfies the condition $\mathbf{X}_S = \mathbf{x}_S$. We have

$$\hat{P}_X^{(D)}(L_m | \mathbf{X}_S = \mathbf{x}_S) \triangleq \frac{N(L_m, \mathbf{x}_S)}{N(\mathbf{x}_S)}. \quad (10)$$

When the variables \mathbf{X}_S are continuous, the estimation is more challenging, and a standard approach is to use kernel smoothing estimators (with Parzen-Rosenblatt kernels). The main drawbacks are the low convergence rate in high dimensions or the derivation and the selection of appropriate bandwidths, which might add complexity and instability to the whole estimation procedure.

We suggest a simple approach based on quantile-discretization of the continuous variables: such processing is common for easing model explainability (typically for tree-based models), see for instance (Bénard et al., 2021b) and the binning of observations can help to stabilize the reduced predictors and SV such that we can improve the robustness of the explanation (Alvarez-Melis and Jaakkola, 2018).

In our experiments, we take usually $q = 10$ quantiles (estimated with the empirical cdf) and the discretized variable X_i is encoded with indicator functions $X_i^{(r)}$, $r = 1, \dots, q-1$. Following our previous section, the SV of X_i are computed by using the coalition of variables $C = (X_i^{(1)}, \dots, X_i^{(q-1)})$. We define then the Discrete reduced predictor, that is denoted

$$\hat{f}_S^D(\mathbf{x}_S) = \sum_{m \in C(S, \mathbf{x})} f_m \hat{P}_X^{(D)}(L_m | \mathbf{x}_S) \quad (11)$$

and our estimates of the SV are $\phi_i(\hat{f}_S^D; \mathbf{x})$.

Although we lose some information with this preprocessing, the loss in performance is often minor with trees, see Appendix D. With only $q = 10$, the input space is a fine grid of p^{10} cells that can provide a great richness. Obviously, this is also a limitation, as the number of cells grows very fast with p and the number of categories per variable. There is a risk of obtaining a high variance with cells having low frequencies. For this reason, we propose another estimator that uses the leaf estimated by the tree.

Continuous and mixed-case Instead of discretizing the variables, we use the leaves of the estimated tree. Essentially, we replace the conditions $\{\mathbf{X}_S = \mathbf{x}_S\}$ by $\{\mathbf{X}_S \in L_m^S\}$. This change introduces a bias, but it aims at improving the variance during estimation. We introduce the Leaf-based estimator

$$\hat{f}_S^{(Leaf)}(\mathbf{x}_S) = \frac{1}{Z(S, \mathbf{x})} \sum_{m \in C(S, \mathbf{x})} f_m \hat{P}_X^{(Leaf)}(L_m | \mathbf{X}_S \in L_m^S) \quad (12)$$

where $\hat{P}_X^{(Leaf)}(L_m | \mathbf{X}_S \in L_m^S)$ is an estimate of the conditional probability, and $Z(S, \mathbf{x})$ is a normalizing constant. The definition of every probability estimate is

$$\hat{P}_X^{(Leaf)}(L_m | \mathbf{X}_S \in L_m^S) = \frac{N(L_m)}{N(L_m^S)}$$

where $N(L_m)$ is the number of observations (of \mathcal{D}_x^{Train}) in the leaf L_m , and $N(L_m^S)$ is the number of observations satisfying the conditions $\mathbf{x}_S \in L_m^S$ across all the leaves of the tree.

We put emphasis on the correction needed for normalizing the probability: in general, we have $\sum_{m \in C(S, \mathbf{x})} \hat{P}_X^{(Leaf)}(L_m | \mathbf{X}_S \in L_m^S) \neq 1$, because we do not condition by the same event (while we have

$\sum_m P_X(L_m | \mathbf{X}_S = \mathbf{x}_S) = 1$). For this reason, the normalizing constant is defined as

$$Z(S, \mathbf{x}) = \sum_{m \in C(S, \mathbf{x})} \frac{N(L_m)}{N(L_m^S)}.$$

The Leaf-based reduced predictor (12) can be computed for continuous and categorical variables, and hence we can compare it with $\hat{f}_S^{(D)}$ in order to evaluate the bias. We see that in both cases, the main challenge is the computation of $C(S, \mathbf{x})$, for every coalition S . We show in the next section how the computational complexity of $\hat{f}_S^{(Leaf)}(\mathbf{x}_S)$ is drastically reduced. Indeed, when we consider the leaf L_m , we only have to compute the SV for d_m variables, and not for p variables.

3.3 Fast Algorithm for the computation of Shapley Values with the Leaf estimator

We have introduced a plug-in estimator of the reduced predictor that is based on an approximation of the conditional expectation on event $\{\mathbf{X}_S = \mathbf{x}_S\}$ by a conditional expectation based on event $\{\mathbf{X}_S \in L_m^S\}$. Indeed, the Leaf estimator $\hat{f}_S^{(Leaf)}$ defined in equation 12 is an unbiased estimator of

$$f_S^{(Leaf)}(\mathbf{x}_S) = \sum_{m=1}^M f_m P_X(L_m | \mathbf{X}_S \in L_m^S(\mathbf{x})) \quad (13)$$

For sake of notational simplicity, we write simply $L_m^S = L_m^S(\mathbf{x})$ and we remove the dependence on \mathbf{x} . Thanks to this approximation, we can propose a straightforward estimate based on empirical frequencies. Here, we focus on the computational efficiency offered by this approximation. It is well-known that the complexity of the computation of a Shapley value is exponential as we need to compute 2^p different coalitions for each observation \mathbf{x} . We show below that the complexity can be made much lower for the Leaf estimator $\hat{f}_S^{(Leaf)}$. Indeed, we derive an algorithm with complexity exponential in the depth of the tree instead of being exponential in the total number of variable p . This is very interesting as the depth of the tree is rarely above 10 in practice, while p can be very large (different order of magnitudes). The idea is to split the original game into the sum of smaller games, as described by the following proposition.

Proposition 3.1. *Let $f(\mathbf{x}) = \sum_{m=1}^M f_m \mathbf{1}_{L_m}(\mathbf{x})$ be a tree-based models and $\mathbf{X} = (X_1, \dots, X_p)$. We introduce the set of variables $S_m = \{X_{N_1}, X_{N_2}, \dots, X_{N_{d_m}}\}$ of the path of each leaf L_m . For any variable X_i , the SV $\phi_i(f^{(Leaf)}, \mathbf{x})$ can be decomposed into the sum of M cooperative games defined on each leaf L_m , and we*

have

$$\phi_i(f^{(Leaf)}, \mathbf{x}) = \sum_{m=1}^M \phi_i^m(f^{(Leaf)}, \mathbf{x}) \quad (14)$$

where $\phi_i^m(f^{(Leaf)}, \mathbf{x})$ is a reweighted version of the Shapley Value of the cooperative game with players S_m and value function $v(f^{(Leaf)}, S) = P_X(L_m | \mathbf{X}_S \in L_m^S(\mathbf{x}))$.

Proof. By definition, we have for a single variable i

$$\begin{aligned} \phi_i(f^{(Leaf)}, \mathbf{x}) &= \frac{1}{p} \sum_{S \subseteq [p] \setminus \{i\}} \binom{p-1}{|S|}^{-1} \left(f_{S \cup i}^{(Leaf)}(\mathbf{x}_{S \cup i}) - f_S^{(Leaf)}(\mathbf{x}_S) \right) \\ &= \frac{1}{p} \sum_{S \subseteq [p] \setminus \{i\}} \binom{p-1}{|S|}^{-1} \left(\sum_{m=1}^M f_m [P(L_m | \mathbf{X}_{S \cup i} \in L_m^{S \cup i}) - P(L_m | \mathbf{X}_S \in L_m^S)] \right) \\ &= \frac{1}{p} \sum_{m=1}^M \sum_{S' \subseteq S_m \setminus \{i\}} \left[\binom{p-1}{|S'|}^{-1} f_m [P(L_m | \mathbf{X}_{S' \cup i} \in L_m^{S' \cup i}) - P(L_m | \mathbf{X}_{S'} \in L_m^{S'})] \right. \\ &\quad \left. + \sum_{Z \neq \emptyset, Z \subseteq \bar{S}_m \cup i} \binom{p-1}{|Z| + |S'|}^{-1} f_m [P(L_m | \mathbf{X}_{S' \cup Z \cup i} \in L_m^{S' \cup Z \cup i}) - P(L_m | \mathbf{X}_{S' \cup Z} \in L_m^{S_m \cup Z})] \right] \end{aligned}$$

However, if $Z \subseteq \bar{S}_m$ and $S \subseteq S_m$:

$$P_X(L_m | \mathbf{X}_{Z \cup S} \in L_m^{Z \cup S}) = P_X(L_m | \mathbf{X}_S \in L_m^S). \quad (15)$$

Note that the identity of equation 15 is not true anymore if we consider the conditional probability $\mathbf{X}_S = \mathbf{x}_S$. Therefore, the SV $\phi_i(f^{(Leaf)}, \mathbf{x})$ can be rewrite as:

$$\begin{aligned} \frac{1}{p} \sum_{m=1}^M \sum_{S' \subseteq S_m \setminus \{i\}} \left[\binom{p-1}{|S'|}^{-1} f_m [P(L_m | \mathbf{X}_{S' \cup i} \in L_m^{S' \cup i}) - P(L_m | \mathbf{X}_{S'} \in L_m^{S'})] \right. \\ \left. + \sum_{Z \neq \emptyset, Z \subseteq \bar{S}_m \cup i} \binom{p-1}{|Z| + |S'|}^{-1} f_m [P(L_m | \mathbf{X}_{S' \cup i} \in L_m^{S' \cup i}) - P(L_m | \mathbf{X}_{S'} \in L_m^{S'})] \right] \\ = \frac{1}{p} \sum_{m=1}^M \sum_{S' \subseteq S_m \setminus \{i\}} \left[\binom{p-1}{|S'|}^{-1} + \sum_{Z \neq \emptyset, Z \subseteq \bar{S}_m \cup i} \binom{p-1}{|Z| + |S'|}^{-1} \right] f_m [P(L_m | \mathbf{X}_{S' \cup i} \in L_m^{S' \cup i}) - P(L_m | \mathbf{X}_{S'} \in L_m^{S'})] \\ \triangleq \sum_{m=1}^M \phi_i^m(\mathbf{x}) \end{aligned}$$

Therefore, we suggest computing SV leaf by leaf thanks to equation 14. In that case, the computation of the SV for the p variables is done by summing over M games (leaves), each of them having a number of variables $|S_m|$ lower than D the maximum depth of any tree. Consequently, the complexity is $\mathcal{O}(p \times M \times 2^D)$ in worst cases. The *Multi-Games algorithm* described below dramatically improves the computational complexity as D is often much lower than p . Moreover, the algorithm is linear in the number of observations. However, it is still higher than the complexity of Tree SHAP (Lundberg et al. (2020)) which is polynomial $\mathcal{O}(M \times D^2)$.

The algorithm is described below, we use the following notations $N(L_m^\emptyset) = \sum_{m=1}^M N(L_m)$ and $\mathbf{1}_{L_m^\emptyset}(\mathbf{x}_\emptyset) = 1$.

Algorithm 1: *Multi-Games Algorithm*

Inputs: $\mathbf{x}, f(\mathbf{x}) = \sum_{m=0}^M f_m \mathbf{1}_{L_m}(\mathbf{x});$
 $p = \text{length}(\mathbf{x});$
 $\phi = \text{zeros}(p);$
for $m = 1$ **to** M **do**
 for i **in** $[p]$ **do**
 if i **not in** S_m **then**
 continue ; /* skip to next variable */
 end
 for $S \subseteq S_m$ **do**
 $\phi[i]_+ =$
 $\left(\binom{p-1}{|S|}^{-1} + \sum_{k=1}^{p-|S_m|} \binom{p-|S_m|}{k} \binom{p-1}{k+|S|}^{-1} \right) \times$
 $\left(\mathbf{1}_{L_m^{S \cup i}}(\mathbf{x}_{S \cup i}) \frac{N(L_m)}{N(L_m^{S \cup i})} - \mathbf{1}_{L_m^S}(\mathbf{x}_S) \frac{N(L_m)}{N(L_m^S)} \right)$
 end
 end
end
return ϕ

Remark: The algorithm is easily parallelizable as it can be vectorized to compute SV of several observations at the same time.

4 Comparison of the estimators

To compare the different estimators, we need a model where conditional expectations can be calculated exactly. If $X \sim \mathcal{N}(\mu, \Sigma)$ then $X_{\bar{S}}|X_S$ is also multivariate gaussian with explicit mean vector $\mu_{\bar{S}|S}$ and covariance matrix $\Sigma_{\bar{S}|S}$, see Appendix A. Note that we do not include any comparisons with KernelSHAP as our main goal is to improve upon TreeSHAP which is the SOTA for tree-based models. In addition, most implementation of KernelSHAP is based on marginal distribution as its aims to be model-agnostic. However, recently Aas et al. (2020, 2021) proposed a conditional version of KernelSHAP but it assumes Gaussian distribution

and samples the subsets. Therefore, the comparisons with our synthetic data would be unfair.

Experiment 1. In the first experiment, let assume we have a dataset $\mathcal{D}_x^{Train} = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$ with $n = 10^4$ generated by a linear regression model with $\mathbf{X} \in \mathbb{R}^p, \mathbf{X} \in \mathcal{N}(0, \Sigma), \Sigma = \rho J_p + (\rho - 1)I_p$ with $p = 5, \rho = 0.7, I_p$ is the identity matrix, J_p is all-ones matrix and a linear predictor $Y = B^t \mathbf{X}$. We use a RandomForest f trained on \mathcal{D}_x^{Train} with a MSE= 4.28, parameters can be found in Appendix G. Since we know the law of \mathbf{X} , we can compute exactly the SV of f with a Monte-Carlo estimator (MC).

We compare the true SV $\phi_i(f; \mathbf{x})$ and the SV of the different estimators $\phi_i(\hat{f}^\alpha; \mathbf{x}), \alpha = SHAP, Leaf, D$. To highlight the differences, we compute 3 metrics. For each estimator, we compute the R-AE defined in equation 7 and a True Positive Rate (TPR) to measure if the ranking of the top $k = 3$ highest and lowest SV is preserved.

In figure 3, we compute the SV $\phi_i(\hat{f}^{SHAP}; \mathbf{x}), \phi_i(\hat{f}^{Leaf}; \mathbf{x})$ on a dataset $\mathcal{D}_x^{Explain}$ of size 1000 generated by the synthetic model. We observe that the estimator \hat{f}^{Leaf} is more accurate than Tree SHAP \hat{f}^{SHAP} by a large margin. TreeSHAP has an average R-AE= 3.31 and TPR= 86%(±17%) while Leaf estimator gets R-AE= 0.90 and TPR= 94%(12 ± %).

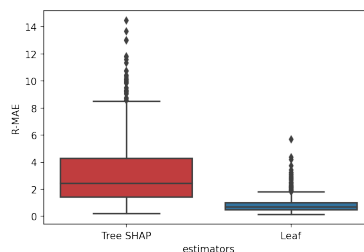


Figure 3: R-AE on 1000 new observations sampled from the synthetic model, $p=5$.

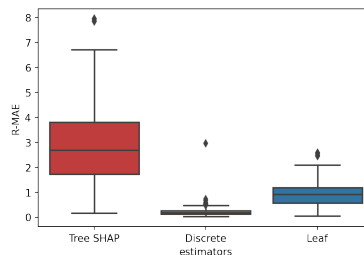


Figure 4: R-AE on 1000 new observations sampled from the synthetic model with discretized variables, $p=5$

In figure 4, we compare the SV of the Discrete unbiased estimator $\phi_i(\hat{f}^{(D)}; \mathbf{x})$, Tree SHAP $\phi_i^{\hat{SHAP}}(f; \mathbf{x})$ and

Leaf estimator $\phi_i(\hat{f}^{(Leaf)}; \mathbf{x})$ with the True $\phi_i(f; \mathbf{x})$, where f was trained on the discretized version of \mathcal{D}_x^{Train} . As demonstrated in figure 3, the Discrete estimator also outperform Tree SHAP with a significant margin.

Experiment 2. Here, we evaluate the impact of the dependence between the features on the different estimators. We use the toy model of experiment 1 but the correlation coefficient ρ varies between 0 and 0.99, representing an increasing positive correlation among the features. As demonstrated in figure 5, Tree SHAP works well when the features are independent ($\rho = 0$), but it is outperformed by Leaf when the dependence increases.

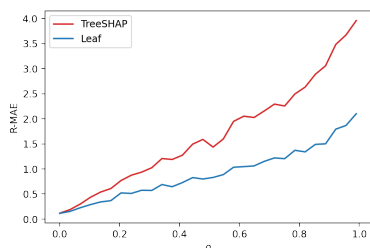


Figure 5: R-AE of the different estimators given the correlation coefficient $\rho \in [0, 0.99]$

Finally, we conduct a run-time comparison of the computation of SV with Leaf and TreeSHAP. We used 3 datasets with different shapes: Boston ($n=506$, $p=13$), Adults ($n=32561$, $p=12$), and a toy linear model with ($n=10000$, $p=500$) where n is the number of observations and p the number of variables. We trained on these datasets XGBoost with default parameters ($ntree = 100$, $maxdepth = 6$). We compute the SV of 1000 observations for Adults, the toy model, and 506 for Boston. As expected in table 1, TreeSHAP is much faster than Leaf estimator. This difference in run-time can be partly explained by the fact that Leaf estimator has to go through all the data for each leaf, whereas TreeSHAP uses the information stored in the trees. However, the Leaf estimator is not very affected by the dimension of the variables as it succeeds in computing the SV when $p = 500$ in a reasonable time.

Table 1: Run-time of TreeSHAP and Leaf estimator on Adults (A), Boston (B) and the toy (T) datasets.

DATASETS	LEAF	TREE SHAP
A (P=12)	1 MIN 4 S \pm 1.73 S	3.33 S \pm 39.9 MS
B (P=13)	8.82 S \pm 204 MS	129 MS \pm 6.91 MS
T (P=500)	1MIN 5S \pm 1.73 S	101 MS \pm 4.54 MS

5 Discussion and Future works

We have shown that the Shapley Values used in XAI and one of its common implementation cannot provide reliable explanations because of the use of biased estimates or because of inappropriate management of categorical variables. We have introduced new estimators and derived the correct way of handling categorical variables, and we show that, even in simple models, the difference can be very significant. Despite this, the impact of such inaccuracies in explanation is poorly addressed, while there is an ever-increasing interest for a trustworthy AI. This leak might be due to the difficulty of evaluating - systematically and quantitatively - the SV and the corresponding explanations. Indeed, it is often hard to have a ground truth and to be able to evaluate precisely the quality of an explanation (as it depends on the law of \mathbf{X} that can be difficult to approach). Moreover, such analysis can be altered by a confirmation bias.

Nevertheless, we think that the quality of the estimates is not the only drawback of SV. Indeed, it can be shown that the explanations of SV are not local, but they remain global, as it is shown in the following proposition

Proposition 5.1. *Let us assume that we have $X \in \mathbb{R}^p$, $\mathbf{X} \in \mathcal{N}(0, I_p)$ independent Gaussian features, and a linear predictor f defined as:*

$$f(\mathbf{X}) = (a_1X_1 + a_2X_2)\mathbb{1}_{X_5 \leq 0} + (a_3X_3 + a_4X_4)\mathbb{1}_{X_5 > 0}. \quad (16)$$

Even if we choose an observation \mathbf{x} such that $x_5 \leq 0$ and the predictor only uses X_1, X_2 , the SV of ϕ_3, ϕ_4 is not necessarily zero. Indeed, $\forall i \in \{3, 4\}$

$$\begin{aligned} \phi_i &= \frac{1}{p} P(\mathbf{X}_5 > 0) \sum_{S \subseteq [p] \setminus \{i, 5\}} \binom{p-1}{|S|}^{-1} (a_i(\mathbf{x}_i - E[\mathbf{X}_i])) \\ &= K (a_i(\mathbf{x}_i - E[\mathbf{X}_i])) \quad K \text{ a constant} \end{aligned}$$

The proof is in Appendix B.

Proposition 5.1 shows that the SV is not really local but is, in fact, global. Such results raise important difficulties in the interpretation of SV, and we think that they are hidden in practice by the lack of precision and understanding of Shapley Values. Moreover, when the model has numerous variables, the number of non-vanishing SV is very high (even in the case similar to Proposition 5.1), thus it is challenging to select the relevant variables. For this reason, we aim at developing an algorithm based on Shapley Values that gives better insight into the local behavior of the model.

References

- Aas, K., Jullum, M., and Løland, A. (2020). Explaining individual predictions when features are dependent: More accurate approximations to shapley values.
- Aas, K., Nagler, T., Jullum, M., and Løland, A. (2021). Explaining predictive models using shapley values and non-parametric vine copulas. *arXiv preprint arXiv:2102.06416*.
- Alvarez-Melis, D. and Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.
- Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2021a). Sirius: Stable and interpretable rule set for classification. *Electronic Journal of Statistics*, 15(1):427–505.
- Bénard, C., Biau, G., Veiga, S., and Scornet, E. (2021b). Interpretable random forests via rule extraction. In *International Conference on Artificial Intelligence and Statistics*, pages 937–945. PMLR.
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.
- Chen, H., Janizek, J. D., Lundberg, S., and Lee, S.-I. (2020). True to the model or true to the data? *arXiv preprint arXiv:2006.16234*.
- Covert, I. and Lee, S. (2020). Improving kernelshap: Practical shapley value estimation via linear regression. *CoRR*, abs/2012.01536.
- Covert, I., Lundberg, S., and Lee, S. (2020a). Understanding global feature contributions through additive importance measures. *CoRR*, abs/2004.00668.
- Covert, I., Lundberg, S., and Lee, S.-I. (2020b). Explaining by removing: A unified framework for model explanation. *arXiv preprint arXiv:2011.14878*.
- Dua, D. and Graff, C. (2017). UCI machine learning repository.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232.
- Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., and Feige, I. (2020). Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*.
- Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65.
- Heskes, T., Sijben, E., Bucur, I. G., and Claassen, T. (2020). Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Janzing, D., Minorics, L., and Blöbaum, P. (2020). Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR.
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839.
- Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ” why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.
- Shapley, L. S. (1953). Greedy function approximation: A gradient boosting machine. *Contribution to the Theory of Games*, 2:307–317.
- Strumbelj, E. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18.
- Williamson, B. D. and Feng, J. (2020). Efficient non-parametric statistical inference on population feature importance using shapley values.

Supplementary Material: Accurate Shapley Values for explaining tree-based models

A Proofs

This section gathers all the proofs of the propositions and claims of the paper.

2. Coalition and Invariance for Shapley Values

2.1 Invariance under reparametrization for continuous variables

Proposition A.1. *Let f and $\tilde{f} = f \circ \varphi^{(-1)}$ its reparametrization, then we have for all $i \in \llbracket 1, p \rrbracket$, for all $\mathbf{x}, \mathbf{u} = \varphi(\mathbf{x})$:*

$$\phi_i(f, \mathbf{x}) = \phi_i(\tilde{f}, \varphi(\mathbf{x})).$$

Proof. It is a direct application of the change of variables formula. If $g(\mathbf{x})$ is the joint density of X_1, \dots, X_p (X_i has density g_i), the transformed variable $\mathbf{U} = (\varphi_1(X_1), \dots, \varphi_p(X_p))$ has density $\tilde{g}(\mathbf{u}) = g(\varphi^{(-1)}\mathbf{u}) \times \prod_i |J(\varphi_i^{(-1)})(u_i)|$. With obvious notations, we have

$$\tilde{g}(u_{\bar{S}}|u_S) = \frac{\tilde{g}(u_{\bar{S}}, u_S)}{\tilde{g}_S(u_S)} = g\left(\varphi_{\bar{S}}^{(-1)}(u_{\bar{S}}|\varphi_S^{(-1)}(u_S))\right) \times \prod_{i \in \bar{S}} |J(\varphi_i^{(-1)})(u_i)|.$$

The computation of the reduced predictor is straightforward

$$\begin{aligned} E[f(\mathbf{X})|\mathbf{x}_S] &= \int f(\mathbf{x}_S, \mathbf{x}_{\bar{S}})g(\mathbf{x}_{\bar{S}}|\mathbf{x}_S)d\mathbf{x}_{\bar{S}} \\ &= \int f(\varphi_S^{(-1)}(\varphi_S(\mathbf{x}_S)), \varphi_{\bar{S}}^{(-1)}\varphi_{\bar{S}}(\mathbf{x}_{\bar{S}}))g(\mathbf{x}_{\bar{S}}|\mathbf{x}_S)d\mathbf{x}_{\bar{S}} \\ &= \int \tilde{f}(\mathbf{u}_S, \mathbf{u}_{\bar{S}})g\left(\varphi_{\bar{S}}^{(-1)}(\mathbf{u}_{\bar{S}}|\varphi_S^{(-1)}(\mathbf{u}_S))\right) \prod_{i \in \bar{S}} |J\varphi^{(-1)}(u_i)| d\mathbf{u}_{\bar{S}} \\ &= E\left[\tilde{f}(\mathbf{U}_S, \mathbf{U}_{\bar{S}})|\mathbf{U}_S = \mathbf{u}_S\right]. \end{aligned}$$

The equality of Shapley Values is then a direct consequence of the equality of reduced predictors.

2.2 Invariance for encoded categorical variable

We recall the expression of the SV for 2 variables for all $x \in \mathbb{R}$ and $Y \in \{1, \dots, K\}$. The role of variable X, Y are symmetric and the categorical or quantitative nature of the variable does not have any impact on the computation of SV given:

$$\begin{cases} \phi_X(f; x, y) = \frac{1}{2} (E[f(X, Y)|X = x] - E[f(X, Y)]) + \frac{1}{2} (f(x, y) - E[f(X, Y)|Y = y]) \\ \phi_Y(f; x, y) = \frac{1}{2} (E[f(X, Y)|Y = y] - E[f(X, Y)]) + \frac{1}{2} (f(x, y) - E[f(X, Y)|X = x]) \end{cases} \quad (17)$$

Proposition A.2. *For all $x \in \mathcal{X}$, and if $y_{1:K-1} = \mathcal{C}(y)$ then*

$$\begin{cases} \phi_{Y_C}(\tilde{f}; x, y_C) = \phi_Y(f; x, y) \\ \phi_X(\tilde{f}; x, y_C) = \phi_X(f; x, y) \end{cases} \quad (18)$$

Proof. As we consider only doable $(x, y_{1:K-1})$, then $\exists! y \in \{1, \dots, K\}$ such that $\mathcal{C}(y) = y_{1:K-1}$. We have the coalition $C = \{1, \dots, K-1\}$, and number of variables $p = K$, meaning

$$\phi_{Y_C}(\tilde{f}; x, y_C) = \frac{1}{2} \left\{ \frac{1}{\binom{1}{0}} \Delta(\tilde{f}; \{\emptyset\}, Y_C) + \frac{1}{\binom{1}{1}} \Delta(\tilde{f}; \{X\}, Y_C) \right\}$$

where

$$\begin{aligned} \Delta(\tilde{f}; \{\emptyset\}, Y_C) &= E_{\tilde{P}} \left[\tilde{f}(X, Y_{1:K-1}) | Y_{1:K-1} = y_{1:K-1} \right] - E_{\tilde{P}} \left[\tilde{f}(X, Y_{1:K-1}) | \emptyset \right] \\ &= E_P \left[\tilde{f}(X, \varphi(Y)) | Y = y \right] - E_P \left[\tilde{f}(X, \varphi(Y)) \right] \\ &= E_P [f(X, Y) | Y = y] - E_P [f(X, Y)] \end{aligned}$$

Indeed

$$\begin{aligned} E_{\tilde{P}} \left[\tilde{f}(X, Y_{1:K-1}) | Y_{1:K-1} = y_{1:K-1} \right] &= \int \tilde{f}(x, y_{1:K-1}) dP(x | y_{1:K-1}) \\ &= \int \tilde{f}(x, y_{1:K-1}) \frac{dP(x, y_{1:K-1})}{P(y_{1:K-1})} \\ &= \int \tilde{f}(x, \varphi(y)) \frac{dP(x, \varphi(y))}{P(\varphi(y))} \\ &= \int f(x, y) \frac{dP(x, y)}{P(y)} \end{aligned}$$

In addition,

$$\begin{aligned} \Delta(\tilde{f}; \{X\}, C) &= E_{\tilde{P}} \left[\tilde{f}(X, Y_{1:K-1}) | X = x, Y_{1:K-1} = y_{1:K-1} \right] - E_{\tilde{P}} \left[\tilde{f}(X, Y_{1:K-1}) | X = x \right] \\ &= \tilde{f}(x, y_{1:K-1}) - E_P \left[\tilde{f}(X, \varphi(Y)) | X = x \right] \\ &= \tilde{f}(x, \varphi(y)) - E_P \left[\tilde{f}(X, \varphi(Y)) | X = x \right] \\ &= f(x, y) - E_P [f(X, y) | X = x] \end{aligned}$$

$$\begin{aligned} \phi_{Y_C}(\tilde{f}; x, y_C) &= \frac{1}{2} (E_P [f(X, Y) | Y = y] - E_P [f(X, Y)]) \\ &\quad + \frac{1}{2} (f(x, y) - E_P [f(X, y) | X = x]) \end{aligned}$$

We can recognize that we have exactly $\phi_{Y_C}(\tilde{f}; x, y_C) = \phi_Y(f; x, y)$. From Equation 2.1, we derive that $\phi_X(\tilde{f}; x, y_C) = \phi_X(f; x, y)$.

Proposition A.3. *If $X \sim \mathcal{N}(\mu, \Sigma)$, then $X_{\bar{S}} | X_S = x_S$ is also multivariate gaussian with mean $\mu_{\bar{S} | S}$ and covariance matrix $\Sigma_{\bar{S} | S}$ equal:*

$$\mu_{\bar{S} | S} = \mu_{\bar{S}} + \Sigma_{\bar{S}, S} \Sigma_{S, S}^{-1} (x_S - \mu_S) \quad \text{and} \quad \Sigma_{\bar{S} | S} = \Sigma_{\bar{S} \bar{S}} - \Sigma_{\bar{S} S} \Sigma_{S S}^{-1} \Sigma_{S, \bar{S}}$$

B Focus on influential variables on Linear regression

Proposition B.1. *Let us assume that we have $X \in \mathbb{R}^p$, $\mathbf{X} \in \mathcal{N}(0, I_8)$ and a linear predictor f defined as:*

$$f(X) = (a_1X_1 + a_2X_2)\mathbb{1}_{X_5 \leq 0} + (a_3X_3 + a_4X_4)\mathbb{1}_{X_5 > 0}. \quad (19)$$

Even if we choose an observation \mathbf{x} such that $x_5 \leq 0$ and the predictor only uses X_1, X_2 , the SV of ϕ_3, ϕ_4 is not necessarily zero.

Proof.

$$\begin{aligned} \phi_3 &= \frac{1}{p} \sum_{S \subseteq [p] \setminus \{3\}} \binom{p-1}{|S|}^{-1} \left(f_{S \cup 3}(\mathbf{x}_{S \cup 3}) - f_S(\mathbf{x}_S) \right) \\ &= \frac{1}{p} \sum_{S \subseteq [p] \setminus \{3,5\}} \binom{p-1}{|S|}^{-1} \left(f_{S \cup 3}(\mathbf{x}_{S \cup 3}) - f_S(\mathbf{x}_S) \right) + \frac{1}{p} \sum_{S \subseteq [p] \setminus \{3,5\}} \binom{p-1}{|S|+1}^{-1} \left(f_{S \cup \{3,5\}}(\mathbf{x}_{S \cup \{3,5\}}) - f_{S \cup 5}(\mathbf{x}_{S \cup 5}) \right) \end{aligned} \quad (20)$$

(21)

The second term is zero. Indeed, $\forall S \subseteq [p] \setminus \{3,5\}$

$$f_{S \cup \{3,5\}}(\mathbf{x}_{S \cup \{3,5\}}) - f_{S \cup 5}(\mathbf{x}_{S \cup 5}) = 0$$

Because, if we condition on the event $\{X_5 = \mathbf{x}_5\}$ with $x_5 \leq 0$

$$\begin{aligned} f_{S \cup \{3,5\}}(\mathbf{x}_{S \cup \{3,5\}}) &= E \left[(a_1X_1 + a_2X_2)\mathbb{1}_{X_5 \leq 0} + (a_3X_3 + a_4X_4)\mathbb{1}_{X_5 > 0} \mid X_{S \cup \{3,5\}} = \mathbf{x}_{S \cup \{3,5\}} \right] \\ &= E \left[(a_1X_1 + a_2X_2)\mathbb{1}_{X_5 \leq 0} \mid X_{S \cup \{3,5\}} = \mathbf{x}_{S \cup \{3,5\}} \right] && \text{because } x_5 \leq 0 \\ &= E \left[(a_1X_1 + a_2X_2) \mid X_{S \cup 5} = \mathbf{x}_{S \cup 5} \right] && \perp\!\!\!\perp \text{ of } X_3 \\ &= f_{S \cup 5}(\mathbf{x}_{S \cup 5}) \end{aligned}$$

The first term of 3.3 is the classic marginal contribution of SV in linear model. $\forall S \subseteq [p] \setminus \{3,5\}$

$$\begin{aligned} f_{S \cup 3}(\mathbf{x}_{S \cup 3}) &= E \left[a_1X_1 + a_2X_2 \mid X_{S \cup 3} = \mathbf{x}_{S \cup 3} \right] P(X_5 \leq 0 \mid X_{S \cup 3} = \mathbf{x}_{S \cup 3}) \\ &\quad + E \left[a_3X_3 + a_4X_4 \mid X_{S \cup 3} = \mathbf{x}_{S \cup 3} \right] P(X_5 > 0 \mid X_{S \cup 3} = \mathbf{x}_{S \cup 3}) \\ &= E \left[a_1X_1 + a_2X_2 \mid X_S = \mathbf{x}_S \right] P(X_5 \leq 0) + (E \left[a_2X_2 \mid X_S = \mathbf{x}_S \right] + a_3\mathbf{x}_3) P(X_5 > 0) \\ &= f_S(\mathbf{x}_S) + P(X_5 > 0) \left(a_3(\mathbf{x}_3 - E[X_3]) \right) \end{aligned}$$

Therefore,

$$\begin{aligned} \phi_3 &= \frac{1}{p} \sum_{S \subseteq [p] \setminus \{3,5\}} \binom{p-1}{|S|}^{-1} P(X_5 > 0) \left(a_3(\mathbf{x}_3 - E[X_3]) \right) \\ &= K \left(a_3(\mathbf{x}_3 - E[X_3]) \right) \end{aligned} \quad K \text{ is a constant}$$

The computation of ϕ_4 is obtained by symmetry.

C Link between the Algorithm 1 (TreeSHAP with path-dependent) and \hat{f}^{SHAP}

In section 3.1, we have said that the recursive algorithm 1 introduced in Lundberg et al. (2020) and shows in figure 2 assumes that the probabilities can be factored with the decision tree as:

$$P_X^{SHAP} \left(\prod_{k=1}^{d_m} I_{N_k} | \mathbf{X}_S = \mathbf{x}_S \right) = \delta_S(N_1) \times \prod_{i=2|N_i \notin S}^{d_m} P \left(X_{N_i} \in I_{N_i} \mid \prod_{k=2|N_k \notin S}^i X_{N_{k-1}} \in I_{N_{k-1}} \right) \quad (22)$$

with $\delta_S(N_1) = P(X_{N_1} \in I_{N_1})$ if $N_1 \notin S$, and 1 otherwise.

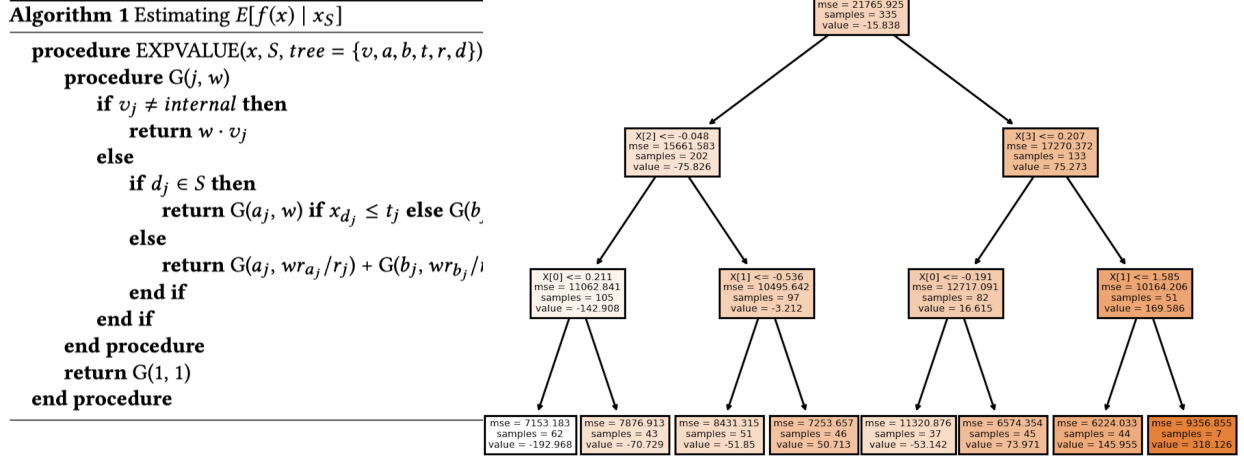


Figure 6: Left figure: Algorithm 1 in Lundberg et al. (2020) (Tree SHAP). Right figure: A sample decision tree used to illustrate the link between \hat{f}^{SHAP}

To show the link between between \hat{f}^{SHAP} and the Algorithm 1, we choose an observation $x = (2, 3, 0.5, -1)$ and compute $E[\hat{f}^{SHAP}(X) | X_0 = 2, X_2 = 0.5]$ where f is the tree in the left of figure C. x is compatible with Leaf 6, 7, 11, 13, 14, we denote $f_6, f_7, f_{11}, f_{13}, f_{14}$ the value of each leaf respectively.

The output of the algorithm is on step 4, and its corresponds to:

$$\begin{aligned}
 \hat{f}^{SHAP}(x) &= P(X_1 \leq 0.305)P(X_2 > -0.048 | X_1 \leq 0.305, \dots) * P(X_1 \leq -0.536 | X_2 > -0.048, \dots) f_6 \\
 &+ P(X_1 \leq 0.305)P(X_2 > -0.048 | X_1 \leq 0.305, \dots) * P(X_1 > -0.536 | X_2 > -0.048, \dots) f_7 \\
 &+ P(X_1 > 0.305)P(X_3 \leq 0.207 | X_1 > 0.305, \dots) * P(X_0 > -0.191 | X_3 \leq 0.207, \dots) f_{11} \\
 &+ P(X_1 > 0.305)P(X_3 > 0.207 | X_1 > 0.305, \dots) * P(X_1 \leq 1.585 | X_3 > 0.207, \dots) f_{13} \\
 &+ P(X_1 > 0.305)P(X_3 > 0.207 | X_1 > 0.305, \dots) * P(X_1 > 1.585 | X_3 > 0.207, \dots) f_{14} \\
 &= (202/335) * 1 * (51/97) * (-51.85) + (202/335) * 1 * (46/97) * (50.716) \\
 &+ (133/335) * (82/133) * 1 * (73.971) + (133/335) * (51/133) * (44/51) * (145.955) \\
 &+ (133/335) * (51/133) * (7/51) * (318.125) \\
 &= 41.98
 \end{aligned}$$

Step	Calculus
0	G(0, 1)
1	G(1, 202/335) + G(8, 133/335)
2	G(5, 202/335) + G(9, 88/335) + G(12, 51/335)
3	G(6,(202/335)*(51/97)) + G(7,(202/335)*(46/97)) + G(11,82/335) + G(13,44/335) + G(14,7/335)
4	-(202/335)*(51/97)*51,85 + (202/335)*(46/97)*50,713 + (82/335)*73,971 + (44/335)*145,955 + (7/335)*318,126
5	= 41.98

D Additional examples

D.1 Impact of quantile discretization

The table below shows the impact of discretization on the performance of a Random Forest on UCI datasets.

Dataset	Breiman’s RF	q=2	q=5	q=10	q=20
Authentication	0.0002	0.08	0.002	0.0005	0.0004
Diabetes	0.17	0.23	0.18	0.18	0.18
Haberman	0.32	0.35	0.30	0.32	0.30
Heart Statlog	0.10	0.10	0.10	0.10	0.10
Hepatitis	0.13	0.15	0.14	0.14	0.13
Ionosphere	0.02	0.07	0.03	0.02	0.02
Liver Disorders	0.23	0.32	0.27	0.25	0.24
Sonar	0.07	0.09	0.07	0.07	0.07
Spambase	0.01	0.14	0.03	0.02	0.01
Titanic	0.13	0.15	0.14	0.14	0.13
Wilt	0.007	0.15	0.03	0.02	0.02

Table 2: Accuracy, measured by 1-AUC on UCI datasets, for two algorithms: Breiman’s random forests and random forests with splits limited to q-quantiles, for $q \in \{2, 5, 10, 20\}$. Table 5 in [Bénard et al. \(2021a\)](#)

D.2 The differences between Coalition and sum on Census Data

We use UCI Adult Census Dataset [Dua and Graff \(2017\)](#). We keep only 4 highly-predictive categorical variables: Marital Status, Workclass, Race, Education and use a Random Forest which has a test accuracy of 86%. We compare the Global SV by taking the coalition or sum of the modalities. Global SV are defined as:

$$I_j = \sum_{i=0}^N |\phi_j^{(i)}|$$

In figure 7, we see differences between the global SV with coalition and sum with N=5000. The ranking of the variables changes, e.g. Education goes from important with sum to not important with the coalition. We also compute the proportion of order inversion over N=5000 observations choose randomly. The ranking of variables is changed in 10% of the cases. Note that this difference may increase or diminish depending on the data.

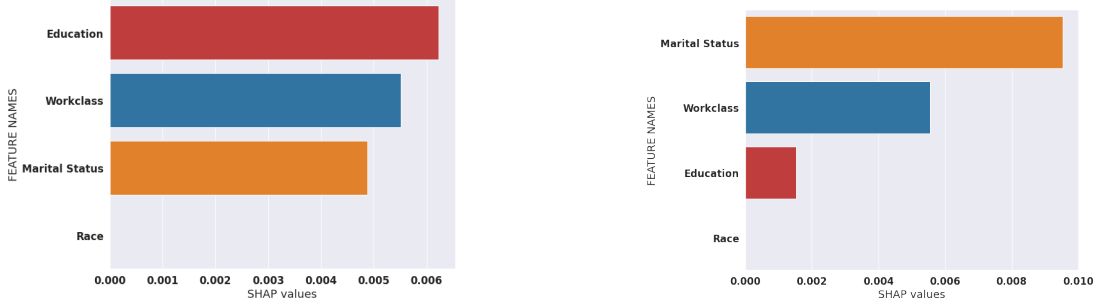


Figure 7: Difference between the global absolute value of SV: sum (left) vs coalition (right) of dummies of individual with modalities: Married, local gov, others, 1rst-4th.

E Individual Shapley values for indicator variables in Dummy Encodings

We give some partial results for the Shapley Values of the modalities $Y = k$, based on the dummy encoding considered in section 2. Indeed equation 2.4 introduces $\phi_k(\tilde{f}, x, y_{1:K-1})$, and proposition 2.1 claims that their sum is different in all generality of the SV of Y . In this section, we give a deeper insight into these values and show that are related multiple comparisons between modalities.

We compute the Shapley Value at point $(x, y = i) = (x, 0, 0, \dots, 1, \dots, 0) = (x, \mathcal{C}(y))$: for ease of notation, we set $Y_0 = X$, and we compute also the Shapley values $\phi_k(\tilde{f}; x, y_{1:K-1})$ for $k = 1, \dots, K - 1$. We recall that we need to compute

$$\frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{\binom{K-1}{k}} \sum_{\substack{Z \subseteq \llbracket 1..K \rrbracket \setminus i \\ |Z| = k}} \Delta(\tilde{f}; Z, i).$$

where Δ denotes the difference between the value function evaluated at $Z \cup \{i\}$ and Z . If we examine the terms $\Delta(\tilde{f}; Z, i)$, the computation needs to take into account if $X = Y_0$ is part of the conditioning variable of not. Indeed, we have for each $k \geq 1$,

$$\sum_{\substack{Z \subseteq \llbracket 0..K-1 \rrbracket \setminus i \\ |Z| = k}} \Delta(\tilde{f}; Z, i) = \sum_{\substack{Z \subseteq \llbracket 1..K-1 \rrbracket \setminus i \\ |Z| = k}} \Delta(\tilde{f}; Z, i) + \sum_{\substack{Z' \subseteq \llbracket 1..K-1 \rrbracket \setminus i \\ |Z'| = k-1}} \Delta(\tilde{f}; Z' \cup \{0\}, i). \quad (23)$$

We start by computing the first term in the right hand side, and it involves only the dummies, and not the quantitative variable.

Proposition E.1 (Computation of Contributions in Shapley without X). *We compute the Shapley values of the variable Y_i , when we have the observations $(x, y_{1:K-1}) = (x, \mathcal{C}(i))$ for $i \in \{1, \dots, K\}$. We consider any $Z' \subseteq \llbracket 1..K-1 \rrbracket \setminus i$, with $|Z'| = k \geq 1$ and $Z' = \{j_1, \dots, j_k\}$. In that case,*

$$\Delta(\tilde{f}; Z, i) = E_P[f(X, Y)|Y = i] - E_P[f(X, Y)|Y \notin \{j_1, \dots, j_k\}] \quad (24)$$

Proof. We have $Y_i = 1 \Leftrightarrow Y = i$, and for $Z' \subseteq \llbracket 1..K-1 \rrbracket \setminus \{0, i\}$, we consider $Z' = \{j_1, \dots, j_k\}$, with $1 \leq j_1 < \dots < j_k \leq K-1$,

$$\begin{aligned} E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_{j_1} = 0, \dots, Y_{j_k} = 0, Y_i = 1 \right] &= E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_i = 1 \right] \\ &= E_{\tilde{P}} \left[\tilde{f}(Y_0, \mathcal{C}(Y)) | Y_i = 1 \right] \\ &= E_P[f(Y_0, Y) | Y = i] \end{aligned}$$

because for all $j_1, \dots, j_{k-1} \neq i$, we have $\{Y_{j_1} = 0, \dots, Y_{j_k} = 0, Y_i = 1\} = \{Y_i = 1\}$.

Moreover,

$$E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_{j_1} = 0, \dots, Y_{j_k} = 0 \right] = E_P \left[\tilde{f}(Y_0, \mathcal{C}(Y)) | Y \neq j_1, \dots, j_k \right]$$

Hence for $Z \subseteq \llbracket 1..K-1 \rrbracket \setminus i$, we have

$$\Delta(\tilde{f}; Z, i) = E_P [f(X, Y) | Y = i] - E_P [f(X, Y) | Y \notin \{j_1, \dots, j_k\}].$$

The second term of the right hand side is given below.

Proposition E.2 (Computation of Contributions in Shapley with X). *We compute the Shapley values only for the variable Y_i , when we have the observations doable $(x, y_{1:K-1}) = (x, \mathcal{C}(i))$ for $i \in \{1, \dots, K\}$. We consider any $Z' \subseteq \llbracket 1..K-1 \rrbracket \setminus i$, with $|Z'| = k-1 \geq 1$, and $Z' = \{j_1, \dots, j_{k-1}\}$. In that case,*

$$\Delta(\tilde{f}; Z' \cup \{0\}, i) = E_P [f(X, Y) | X = x, Y = i] - E_P [f(X, Y) | X, Y \notin \{j_1, \dots, j_{k-1}\}] \quad (25)$$

Proof. We assume that we have a subset $|Z'| = k-1$, such that $Z' \subseteq \llbracket 1..K-1 \rrbracket \setminus i$. This means that $Z' = \{j_1, \dots, j_{k-1}\}$, with $1 \leq j_1, \dots, j_{k-1} \leq K-1$. We

$$\begin{aligned} E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_0 = x, Y_{j_1} = 0, \dots, Y_{j_{k-1}} = 0, Y_i = 1 \right] &= E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_0 = x, Y_i = 1 \right] \\ &= E_P \left[\tilde{f}(Y_0, \mathcal{C}(Y)) | Y_0 = x, Y = i \right] \\ &= E_P [f(Y_0, \mathcal{C}(Y)) | Y_0 = x, Y = i] \end{aligned}$$

and

$$\begin{aligned} E_{\tilde{P}} \left[\tilde{f}(Y_0, Y_{1:K-1}) | Y_0 = x, Y_{j_1} = 0, \dots, Y_{j_{k-1}} = 0 \right] &= E_P \left[\tilde{f}(Y_0, \mathcal{C}(Y)) | Y_0 = x, Y \notin \{j_1, \dots, j_{k-1}\} \right] \\ &= E_P [f(Y_0, Y) | Y_0 = x, Y \notin \{j_1, \dots, j_{k-1}\}] \end{aligned}$$

Finally, we can give several examples of the different increments involved in the Shapley values of each variable X or Y_k . If $k = 0$, then $Z' = \emptyset$ and

$$\Delta(\tilde{f}; Z', i) = \Delta(\tilde{f}; \emptyset, i) = E_P [f(X, Y) | Y = i] - E_P [f(X, Y)]$$

If $k = 1$, then $Z' = \{0\}$ or $Z' = \{j\} \neq \{i\}$,

$$\begin{aligned} \Delta(\tilde{f}; Z', i) &= \Delta(\tilde{f}; \{0\}, i) = E_P [f(X, Y) | X = x, Y = i] - E_P [f(X, Y) | X = x] \\ \Delta(\tilde{f}; Z', i) &= \Delta(\tilde{f}; \{j\}, i) = E_P [f(X, Y) | Y = i] - E_P [f(X, Y) | Y \neq j] \end{aligned}$$

For $k = K-1$, $Z' = \{1, \dots, K-1\}$,

$$\Delta(\tilde{f}; \{1, \dots, K-1\}, i) = E_P [f(X, Y) | X = x, Y = i] - E_P [f(X, Y) | X = x, Y \neq i]$$

The propositions E.1 and E.2 show that the individual Shapley value for the variable (modality) Y_i is a weighted mean of the difference between class i and group of classes:

$$\begin{cases} E_P [f(X, Y) | Y = i] - E_P [f(X, Y) | Y \notin \{j_1, \dots, j_k\}] \\ E_P [f(X, Y) | X = x, Y = i] - E_P [f(X, Y) | X, Y \notin \{j_1, \dots, j_{k-1}\}] \end{cases}$$

Finally, we can also compute the Shapley values of the other variables Y_j at point $(x, y = i)$, for $j \neq i$. In that case, the difference $\Delta(\tilde{f}; Z', j), j \neq i$ are of the type of

$$\begin{cases} E_P [f(X, Y) | Y \notin \{j, j_1, \dots, j_k\}] - E_P [f(X, Y) | Y \notin \{j_1, \dots, j_k\}] \\ E_P [f(X, Y) | Y = i] - E_P [f(X, Y) | Y = j] \\ E_P [f(X, Y) | X = x, Y \notin \{j, j_1, \dots, j_k\}] - E_P [f(X, Y) | X, Y \notin \{j_1, \dots, j_{k-1}\}] \\ E_P [f(X, Y) | X = x, Y = i] - E_P [f(X, Y) | X, Y = j] \end{cases}$$

The Shapley values computes a mean of the difference between different aggregation of modalities, that contains or not the variable of interest.

As a conclusion of this part, we see that the individual Shapley values $\phi_k(\tilde{f}; x, y_{1:K-1})$ perform a multiple comparison of the means obtained by aggregating the classes or modalities in various ways, looking at the presence or not of the modality k . These differences of means have weights $\frac{1}{\binom{K-1}{k}}$ where k is basically the number of classes of the variable Y that we aggregate.

Consequently the sum $\sum_{k=1}^K \phi_k(\tilde{f}; x, y_{1:K-1})$ is clearly different from the

$$\phi_Y(f; x, y) = \frac{1}{2} (E[f(X, Y)|Y = y] - E[f(X, Y)]) + \frac{1}{2} (f(x, y) - E[f(X, Y)|X = x]).$$

This latter has a much more global analysis that aims at measuring how the mean $E[f(X, Y)|Y = y]$ in the various classes changes w.r.t $E[f(X, Y)]$, while the individual Shapley focus on the difference between subgroups of classes.

F Plug-In estimator of Marginal expectation

As we have indicated in the paper, the Shapley Values can be computed with different probability $Q_{S, \mathbf{x}}$. In that section, we show that when we use the marginal distribution (as in the so-called interventional case), the previous estimators for tree-based models can be adapted straightforwardly.

We consider then decision tree

$$f(x) = \sum_{m=1}^M f_m \mathbb{1}_{L_m}(x)$$

and remark that the Marginal Shapley coefficients involve the computations of the marginal expectations $E_P[\mathbb{1}_{L_m}(\mathbf{X}_{\bar{Z}}, \mathbf{x}_Z)]$ for any subgroup of variables Z . On real data, we need to compute the conditional expectations, but we use the Tree approximations in order to replace

$$\begin{aligned} E_P[\mathbb{1}_{L_m}(\mathbf{X}_{\bar{Z}}, \mathbf{x}_Z)] &= \int \int \mathbb{1}_{L_m}(\mathbf{u}_{\bar{Z}}, \mathbf{x}_Z) p(\mathbf{u}_{\bar{Z}}, \mathbf{u}_Z) d\mathbf{u}_{\bar{Z}} d\mathbf{u}_Z \\ &= \int \int \mathbb{1}_{L_m}(\mathbf{u}_{\bar{Z}}, \mathbf{x}_Z) p(\mathbf{u}_Z | \mathbf{u}_{\bar{Z}}) p(\mathbf{u}_{\bar{Z}}) d\mathbf{u}_Z d\mathbf{u}_{\bar{Z}} \\ &= \int \left\{ \int p(\mathbf{u}_Z | \mathbf{u}_{\bar{Z}}) d\mathbf{u}_Z \right\} \mathbb{1}_{L_m}(\mathbf{u}_{\bar{Z}}, \mathbf{x}_Z) p(\mathbf{u}_{\bar{Z}}) d\mathbf{u}_{\bar{Z}} \\ &= \int \mathbb{1}_{L_m}(\mathbf{u}_{\bar{Z}}, \mathbf{x}_Z) p(\mathbf{u}_{\bar{Z}}) d\mathbf{u}_{\bar{Z}} \end{aligned}$$

This means that we just need the marginal distributions of the variables $\mathbf{X}_{\bar{Z}}$ in order to compute the expectations of the leaf. In the case of quantitative data, the leaf can be written $L_m = \prod_{i=1}^p [a_i^m, b_i^m]$, and we have by definition

$$\exists k \in Z, x_k \notin [a_k, b_k] \implies \mathbb{1}_{L_m}(\mathbf{u}_{\bar{Z}}, \mathbf{x}_Z) = 0$$

We define the set of leafs compatible with condition $\mathbf{X}_Z = \mathbf{x}_Z$ as

$$C(Z, \mathbf{x}) = \left\{ m \in [1 \dots M] \mid L_m = \prod_{i=1}^p [a_i^m, b_i^m], \forall k \in Z, x_k \in [a_k^m, b_k^m] \right\}$$

We write for $m \in C(Z, \mathbf{x})$, $L_m = L_m^{\bar{Z}} \times L_m^Z$, with $L_m^{\bar{Z}} = \prod_{i \in \bar{Z}} [a_i^m, b_i^m]$ and $L_m^Z = \prod_{i \in Z} [a_i^m, b_i^m]$, this means that for all $m \in C(Z, \mathbf{x})$ we have

$$E_P[\mathbb{1}_{L_m}(\mathbf{X}_{\bar{Z}}, \mathbf{x}_Z)] = E_P[\mathbb{1}_{L_m^{\bar{Z}}}(\mathbf{X}_{\bar{Z}})]$$

As an approximation, the conditional probability for $m \in C(Z, \mathbf{x})$ is computed as

$$\begin{aligned} E_P[\mathbb{1}_{L_m^{\bar{Z}}}(\mathbf{X}_{\bar{Z}})] &= P(X_i \in [a_i^m, b_i^m], i \in \bar{Z}) \\ &\simeq \frac{N(L_m^{\bar{Z}})}{N} \end{aligned}$$

where $N(L_m^{\bar{Z}})$ is the number of observations in the (partial) leaf $L_m^{\bar{Z}}$. As a consequence we have

$$\begin{aligned}
 E_P[f(\mathbf{X}_Z, \mathbf{x}_Z)] &= \sum_{m=1}^M \hat{y}_m E_P[\mathbb{1}_{L_m}(\mathbf{X}_Z, \mathbf{x}_Z)] \\
 &= \sum_{m \in C(Z, \mathbf{x})} \hat{y}_m E_P[\mathbb{1}_{L_m}(\mathbf{X}_Z, \mathbf{x}_Z)] \\
 &= \sum_{m \in C(Z, \mathbf{x})} \hat{y}_m E_P[\mathbb{1}_{L_m^{\bar{Z}}}(\mathbf{X}_{\bar{Z}})] \\
 &\simeq \sum_{m \in C(Z, \mathbf{x})} \hat{y}_m \frac{N(L_m^{\bar{Z}})}{N}
 \end{aligned}$$

G EXPERIMENTAL SETTINGS

All our experiments are reproducible and can be found on the github repository *Active Coalition of Variables*, <https://github.com/salimamoukou/acv00>

A.1 Toy model of Section 2.3

Recall that the model is a linear predictor with categorical variables define as $f(X, Y) = B_Y X$ with $X|Y = y \sim \mathcal{N}(\mu_y, \Sigma_y)$ and $\mathbb{P}(Y = y) = \pi_y$, $Y \in \{a, b, c\}$.

For the experiments in Figure 1 and 2, we set $\pi_y = \frac{1}{3}$, $\mu_y = 0 \forall y \in \{a, b, c\}$. We use a random matrices generated from a Wishart distribution. The covariance matrices are:

$$\Sigma_a = \begin{bmatrix} 0.41871254 & -0.790061361 & 0.46956991 \\ -0.79006136 & 1.90865098 & -0.82571655 \\ 0.46956991 & -0.82571655 & 0.95835472 \end{bmatrix}, \quad \Sigma_b = \begin{bmatrix} 0.55326081 & 0.11811951 & -0.70677924 \\ 0.11811951 & 2.73312979 & -2.94400196 \\ -0.70677924 & -2.94400196 & 4.22105088 \end{bmatrix},$$

$$\Sigma_c = \begin{bmatrix} 9.2859966 & 1.12872646 & 2.4224434 \\ 1.12872646 & 0.92891237 & -0.14373393 \\ 2.4224434 & -0.14373393 & 1.81601676 \end{bmatrix} \text{ for } y \in \{a, b, c\} \text{ respectively.}$$

The coefficients are $B_a = [1, 3, 5]$, $B_b = [-5, -10, -8]$, $B_c = [6, 1, 0]$ and the selected observation in figure 1 values is $x = [0.35, -1.61, -0.11, 1., 0., 0.]$

A.2 Toy model of Section 4

The data $\mathcal{D} = (x_i, z_i)_{1 \leq i \leq n}$ are generated from a linear regression $Z = B^t X$ with $n = 10000$, $\mathbf{X} \in \mathbb{R}^p$, $\mathbf{X} \in \mathcal{N}(0, \Sigma)$, $\Sigma = \rho J_p + (\rho - 1)I_p$ with $p = 5$, $\rho = 0.7$, I_p is the identity matrix, J_p is all-ones matrix and a linear predictor $Z = B^t \mathbf{X}$. $B = [6.49, -2.44, -2.11, -4.29, 3.46]$ for the continuous case and $d=3$, $B = [6.49, -2.44, 0]$ for the discrete case.

We used the decision tree of sklearn trained on \mathcal{D} with the defaults parameters. The Mean Squared Error (MSE) are $\text{MSE} = 4.39$ for the continuous case and $\text{MSE} = 2.88$ for the discrete case.