# Accurate and robust Shapley Values for explaining predictions and focusing on local important variables

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Although Shapley Values (SV) are widely used in explainable AI, they can be poorly understood and estimated, which implies 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 efficiently the tree structure and are more accurate than state-of-the-art methods. For interpreting additive explanations, we recommend to filter the non-influential variables and to compute the Shapley Values only for groups of influential variables. For this purpose, we use the concept of "Same Decision Probability" (SDP) that evaluates the robustness of a prediction when some variables are missing. This prior selection procedure produces sparse additive explanations easier to visualize and analyse. Simulations and comparisons are performed with state-of-the-art algorithm, and show the practical gain of our approach.

## 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 in 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 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 importance measures have been proposed with a particular focus 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 [10], Individual Conditional Expectation [13], and local feature importance attribution measures such as Local Surrogate (LIME) [21]. With the same objective in mind, the Shapley Values [22], 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 [23, 19]. The Shapley Values (SV) are now massively used for identifying important variables at a local and global scale. As remarked in [18, 9], a lot of importance measures aim at analyzing the behavior of a prediction model $f$ based on $p$ features $X_1, \ldots, X_p$ by removing variables and considering reduced predictors. Typically, for any group of variables $\boldsymbol{X}_S = (X_i)_{i \in S}$, with any subset $S \subseteq [\![1, p]\!]$ and reference distribution $Q_{S,\boldsymbol{x}}$, reduced predictors are defined as

$$f_S(\boldsymbol{x}_S) \triangleq E_{Q_{S,\boldsymbol{x}}}\left[f\left(\boldsymbol{x}_S, \boldsymbol{X}_{\bar{S}}\right)\right] \tag{1.1}$$

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

where $Q_{S,\boldsymbol{x}}$ is the conditional distribution $P(\boldsymbol{X}_{\bar{S}}|\boldsymbol{X}_S = \boldsymbol{x}_S)$. Other SV can be defined with the marginal probabilities but their interpretation is different [14, 15, 6] and there are still active debates on using or not conditional probabilities [11]: we consider only conditional distributions as the so-called observational SV are widely used. The SV for local interpretability at $\boldsymbol{x}$ have been introduced in [19] and are based on a cooperative game with value function $v(f;S) \triangleq f_S(\boldsymbol{x}_S)$. For any group of variables $C \subseteq [\![1,p]\!]$ and $k \in [\![1, p-|C|]\!]$, we denote the set $\mathcal{S}_k(C) = \{S \subseteq [\![1,p]\!] \setminus C | \, |S| = k\}$, we define the Shapley Value (SV) of the coalition $C$ as

$$\phi_C(f;\boldsymbol{x}) \triangleq \frac{1}{p-|C|+1} \sum_{k=0}^{p-|C|} \frac{1}{\binom{p-|C|}{k}} \sum_{S \in \mathcal{S}_k(C)} (f_{S\cup C}(\boldsymbol{x}_{S\cup C}) - f_S(\boldsymbol{x}_S))) \tag{1.2}$$

This definition of the Shapley Value is a generalization of the classical SV for one variable: if we consider the singleton $C = \{i\}$ for $i \in [\![1,p]\!]$, we recover the standard definition for "player $X_i$". The same SV for a group of variables is considered in [4] in order to measure the importance of group of features. In the next section, we show how the definition (1.2) 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 $\phi_i(f;\boldsymbol{x})$ which is an active subject. We focus in this paper on tree-based models as the computational cost is reduced and the statistical problem is easier to address [18]. Indeed, our objective is to improve the estimates of the "true" Shapley Values based on a dataset $\mathcal{D}$ (such as the train set or simulated samples from a learned distribution $\hat{P}_{\boldsymbol{X}}$). Our objective is to reduce two sources of noise that can make the analysis of SV unreliable: we improve the estimation of the conditional expectations by statistically principled estimators and we provide a criterion for removing "noise variables" in order to obtain sparse additive explanations prior to the computation of SV. Our contributions are implemented in a Python package[1] that shows the bias reduction of our SV estimates, and highlights the improvement over state-of-the-art algorithms.

Our paper is organized as follows. In the next section, we derive invariance principles for SV under reparametrization or encoding that are particularly useful for dealing with categorical variables. In section 3, we introduce two estimators of reduced predictors and SV, and perform a detailed comparison with the dependent TreeExplainer algorithm [18]. In section 4, we propose the "Same Decision Probability" for focusing on the influential variables: after deriving a computational algorithm that builds on the estimators of section 3, we show how the so-called active coalition of variables gives a sparse additive explanation based on active Shapley Values.

## 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 categorial variables based on the notion of coalition and the general definition given in (1.2). This is useful in our case, as we are interested in the discretization of continuous variables in order 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 of the reduced predictor (1.1), there is no constraint on the dimension of $X_i$. We suppose that the $p$ variables are vector-valued ie. $X_i \in \mathbb{R}^{p_i}, p_i \geq 1$ and that they have a density $g_i$. We assume that we transform each variable $X_i$ with a diffeomorphism $\varphi_i : \mathbb{R}^{p_i} \longrightarrow \mathbb{R}^{p_i}$. We introduce the transformed variables $U_i \triangleq \varphi_i(X_i)$ and the reparametrized model $\tilde{f}$ defined by $\tilde{f}(U_1,\ldots,U_p) = f(X_1,\ldots,X_p)$, ie. $\tilde{f}(u_1,\ldots,u_p) = f \circ \boldsymbol{\varphi}^{(-1)}(\boldsymbol{u})$ where $\varphi = (\varphi_1,\ldots,\varphi_p)$. In general, we cannot relate the predictor $f_x$ learned from the real data set $\mathcal{D}_x^{Train} = \{(\boldsymbol{x}_i, z_i),\, i \in [\![1,n]\!]\}$ to the predictor $f_u$ learned from $\mathcal{D}_u^{Train} = \{(\boldsymbol{u}_i, z_i),\, i \in [\![1,n]\!]\}$ ($z$ 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 explanation, and we show below that the Shapley Values are invariant.

[1] Library "Active Coalition of Variables", github.com/acvneurips/ACV

**Proposition 2.1.** *Let $f$ and $\tilde{f} = f \circ \varphi$ its reparametrization, then we have for all $i \in [\![1, p]\!]$, for all $\boldsymbol{x}, \boldsymbol{u} = \boldsymbol{\varphi}(\boldsymbol{x})$:*

$$\phi_i(f; \boldsymbol{x}) = \phi_i(\tilde{f}; \varphi_i^{(-1)}(\boldsymbol{x})).$$

This identity indicates that the information provided by each feature $X_i$ in the explanation does not depend on any encoding. If the variables are grouped into $p$ groups of size $p_i$ (ideally by grouping correlated variables) and transformed by some feature engineering in order to improve the interpretability of each group, then we will keep the same SV $\phi_i(f, \boldsymbol{x})$. Another interest of identity (2.1) is to show that the SV depends essentially on the dependence structure of the features $X_i$. Indeed, if $p_i = 1$ and the functions $\varphi_i$ are the cumulative distribution functions $\varphi_i(x) = P(X_i \leq x)$, the variables $\boldsymbol{U} = (U_1, \ldots, U_p)$ are defined on the unit cube $[0, 1]^p$ with a distribution that corresponds to the copula $C(u_1, \ldots, u_p)$ of the distribution $P_{\boldsymbol{X}}$.

## 2.2 Invariance for encoded categorical variable

In the rest of the paper, continuous predictive variables are denoted with $X$ and the categorical predictive variables are denoted with $Y$ (the output to predict is denoted $Z$). There exists numerous encodings for a categorical variable $Y$ with modalities $\{1, \ldots, K\}$, but we focus on methods related to One-Hot-Encoding (OHE) and Dummy Encoding (DE) that corresponds to the introduction of indicator functions $Y_k$ ($Y_k = 1$ if $Y = k$, 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 popular practice that recommends to sum the SV of the indicator functions $Y_k$ for computing the SV of $Y$ 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 (1.2).

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

$$f(x, y) - E_P\left[f\left(X, Y\right)\right] = \phi_X(f; x, y) + \phi_Y(f; x, y) \tag{2.1}$$

In order to establish the link between the SV of the indicator functions $Y_k$, $k = 1$ and the SV of the variable $Y$, we need more notations. We focus on the Dummy Encoding (DE) $\varphi : y \mapsto (y_1, \ldots, y_{K-1})$. The variables $(X, Y_{1:K-1})$ are defined on $\mathbb{R} \times \{0, 1\}^{K-1}$, its distribution $\tilde{P}$ is the image probability of $P$ induced by the transformation $\varphi$. The initial predictor $f : \mathbb{R} \times \{1, \ldots, K\} \longrightarrow \mathbb{R}$ is reparametrized as a function $\tilde{f} : \mathbb{R} \times \{0, 1\}^{K-1} \longrightarrow \mathbb{R}$ such that $f(X, Y) \triangleq \tilde{f}(X, Y_1, \ldots, Y_{K-1})$. The function $\tilde{f}$ is not completely defined for all $(y_1 \ldots, y_{K-1}) \in \{0, 1\}^{K-1}$ and is only defined $\tilde{P}$-almost everywhere because of the deterministic dependence $\sum_{k=1}^{K-1} Y_k \leq 1$. Consequently, we need to extend $\tilde{f}$ to the whole space $\mathcal{X} \times \{0, 1\}^{K-1}$ by setting $\tilde{f}(x, y_1, \ldots y_{K-1}) = 0$ as soon as $\sum_{k=1}^{K-1} y_k > 1$. For the predictor $\tilde{f}(X, Y_1, \ldots, Y_{K-1})$, we can compute the SV of $X, Y_1, \ldots, Y_{K-1}$ and obtain the decomposition

$$\tilde{f}(x, y_{1:K-1}) - E_{\tilde{P}}\left[\tilde{f}\left(X, Y_{1:K-1}\right)\right] = \phi_X^{indiv}(\tilde{f}; x, y_{1:K-1}) + \sum_{k=1}^{K-1} \phi_{Y_k}(\tilde{f}; x, y_{1:K-1}) \tag{2.2}$$

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

$$\phi_X(f; x, y) + \phi_Y(f; x, y) = \phi_X^{indiv}(\tilde{f}; x, y_{1:K-1}) + \sum_{k=1}^{K-1} \phi_{Y_k}(\tilde{f}; x, y_{1:K-1}) \tag{2.3}$$

In general, we have $\phi_Y(f; x, y) \neq \sum_{k=1}^{K-1} \phi_{Y_k}(\tilde{f}; x, y_{1:K-1})$, because the SV depends on the number of variables. We show in the next proposition that $\phi_Y(f; x, y) = \phi_C(\tilde{f}; x, y_{1:K-1})$ where $\phi_C$ is computed with eq. (1.2) and $C$ is the coalition of variables $(Y_1, ..., Y_{K-1})$.

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

$$\left\{ \begin{array}{rcl} \phi_C(\tilde{f}; x, y_{1:K-1}) & = & \phi_Y(f; x, y) \\ \phi_X^{coal}(\tilde{f}; x, y_{1:K-1}) & = & \phi_X(f; x, y) \end{array} \right. \tag{2.4}$$

We refer to Appendix A for detailed derivations. In general, for cooperative games, the SV of a coalition $\phi_C(\tilde{f}; x, y_{1:K-1})$ is different from the sum of individual SV $\sum_{k\in C}\phi_{Y_k}(\tilde{f}; x, y_{1:K-1})$. We remark that we can compute two different SV for $X$ when we use the encoded predictor $\tilde{f}$: $\phi_X^{coal}(\tilde{f}; x, y_{1:K-1})$ and $\phi_X^{indiv}(\tilde{f}; x, y_{1:K-1})$. 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 $\phi_X^{coal}(\tilde{f}; x, y_{1:K-1})$ to $\phi_X^{indiv}(\tilde{f}; x, y_{1:K-1})$, as $\phi_X^{coal}$ is equal to the theoretical SV given in eq. (2.1). For this reason, we denote for simplicity $\phi_X(\tilde{f}; x, y_{1:K-1})$.

### 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 $(X_0, X_1, X_2)$, defined as $f(X, Y) = B_Y X$ with $X|Y = y \sim \mathcal{N}(\mu_y, \Sigma_y)$ and $\mathbb{P}(Y = y) = \pi_y$, $Y \in \{a, b, c\}$. The values of the parameters used in our experiments are found in Appendix I. In the left of 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 $X$. To quantify the global difference of the different methods, we compute the relative mean absolute error (R-MAE) of the SV defined as:

$$\text{R-MAE}(f, \tilde{f}) = \sum_{i=1}^{p} \frac{|\phi_i(f; \boldsymbol{x}) - \phi_i(\tilde{f}; \boldsymbol{x})|}{|\phi_i(f; \boldsymbol{x})|} \tag{2.5}$$

We compute the SV of 100 observations of the synthetic dataset. We observe in the right of figure 1 that the differences can be huge for almost all samples (DE is much worse than OHE in that example). Thus, we highly recommended to use the coalition as it is consistent with the true SV contrary to the sum. More examples on real dataset can be found in appendix F.

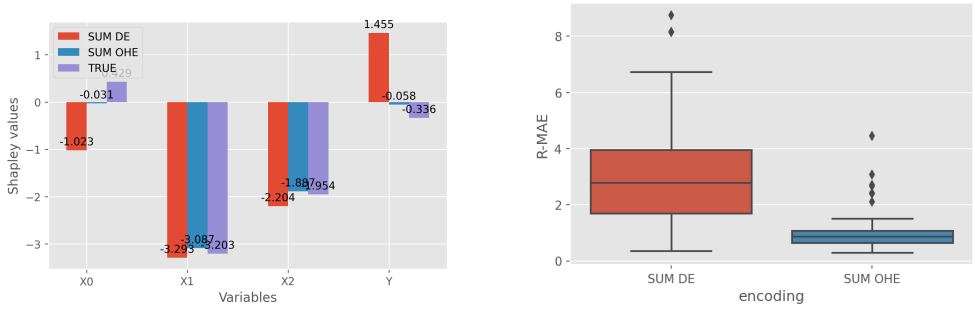


Figure 1: Left figure: SV with or without encoding (OHE - DE) for observation $x = [0.35, -1.61, -0.11], y = a$. Right figure: R-MAE distribution between the SV of the True and the corresponding encodings

## 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 $f_S(\boldsymbol{x}_S) = E\left[f(\boldsymbol{X})|\boldsymbol{X}_S\right], S \subseteq [\![1, p]\!]$. In current approaches, the estimation relies on several approximations, eg. that assumes independence [19] or more recently a modelling of the features with a gaussian distribution or vine copula [1, 2]. We focus on tree based models, as it has been exploited in [18] for deriving an algorithm "TreeExplainer" for computing exact SV: we can compute all the terms and the estimation of the conditional expectations is simplified. After a brief presentation of the limitations of "TreeExplainer", 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 model is straightforward by linearity.

### 3.1 Algorithms for computing Conditional Expectations and the TreeExplainer algorithm

We consider a tree-based model $f$ defined on $\mathbb{R}^p$ (categorical variables are one-hot encoded). We have $f(\boldsymbol{x}) = \sum_{m=1}^{M} f_m \mathbb{1}_{L_m}(\boldsymbol{x})$ 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 in the tree: a leaf $L_m$ is defined by a sequence of decision based on $d_m$ variables $X_{N_k}$, $k = 1, \ldots, d_m$. 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

$$L_m = \{\boldsymbol{x} \in \mathbb{R}^p : x_{N_1} \in I_{N_1}, \ldots, x_{N_{d_m}} \in I_{N_{d_m}}\}. \tag{3.1}$$

A crucial point is to identify the set of leaves compatible with the condition $\boldsymbol{X}_S = \boldsymbol{x}_S$: 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 $\boldsymbol{X}_S = \boldsymbol{x}_S$ the set of compatible leaves for each $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$ is

$$C(S, \boldsymbol{x}) = \left\{m \in [1 \ldots M] \,|\boldsymbol{x}_S \in L_m^S\right\} = \{m \in [1 \ldots M] \,|x_{N_i} \in I_{N_i},\, N_i \in S\}$$

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

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

When we have a model for $P_X$ from which we can derive a conditional density and evaluate directly the conditional probabilities $P_X(L_m|\boldsymbol{X}_S = \boldsymbol{x}_S)$, we can have an exact computation. This is typically the case when $\boldsymbol{X} \sim \mathcal{N}(\mu, \Sigma)$ and we can integrate the densities for deriving the conditional probabilities $P_X\left(\prod_{k=1}^{d_m} I_{N_k} | \boldsymbol{X}_S = \boldsymbol{x}_S\right)$. The derivation of conditional probabilities can become challenging, and assumptions about the factorization of the distribution can accelerate the computation: in [18], the authors introduce a recursive algorithm (TreeExplainer 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:

$$P_X^{SHAP}\left(\prod_{k=1}^{d_m} I_{N_k} | \boldsymbol{X}_S = \boldsymbol{x}_S\right) = \prod_{i=2 | N_i \notin S}^{d_m} P(X_{N_i} \in I_{N_i} | X_{N_{i-1}} \in I_{N_{i-1}}) \times \delta_S(N_1) \tag{3.2}$$

with $\delta_S(N_1) = P(X_{N_1} \in I_{N_1})$ if $N_1 \notin S$, and 1 otherwise. The underlying assumptions in (3.2) is that we have a Markov chain defined by the path in the tree, and the transition probabilities are estimated conditionally on $\{\boldsymbol{X}_S = \boldsymbol{x}_S\}$, e.g. each probability is replaced by 1 if $N_i \in S$, see algorithm description in Appendix D. As we will see in the simulations, this assumptions 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 $\phi_i(\hat{f}_S^{SHAP}; \boldsymbol{x})$ 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 make no assumption on the existence of the density or probability $p(\boldsymbol{x})$ as in [1, 2] and we have also $\mathcal{D}_x^{Explain}$ that corresponds to the (new) individuals on which we want to compute SV. We put emphasis on the fact that $\mathcal{D}_x^{Explain}$ can have a probability distribution different from $P_X$, or it can be deterministic (eg. uniform grid).

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

$$\hat{P}_X^{(D)}(L_m | \boldsymbol{X}_S = \boldsymbol{x}_S) \triangleq \frac{N(L_m, \boldsymbol{x}_S)}{N(\boldsymbol{x}_S)}. \tag{3.3}$$

When the variables $\boldsymbol{X}_S$ 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 rate of convergence 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 [5] and the binning of observations can help stabilizing the reduced predictors and SV such that we can improve the robustness of the explanation [3].
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 $C = \left(X_i^{(1)}, \dots, X_i^{(q-1)}\right)$. We define then the Discrete reduced predictor, that is denoted

$$\hat{f}^D(\boldsymbol{x}) = \sum_{m \in C(S,\boldsymbol{x})} f_m \hat{P}_X^{(D)}\left(L_m | \boldsymbol{x}_S\right) \tag{3.4}$$

and our estimates of the SV are $\phi_i(\hat{f}^D; \boldsymbol{x})$.
Although we lose some information with this pre-processing, the loss in performance is often minor with trees, see Appendix F. 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 variables. 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 $\{\boldsymbol{X}_S = \boldsymbol{x}_S\}$ by $\{\boldsymbol{X}_S \in L_m^S\}$. This change introduces a bias but it aims at improving the variance during estimation. We introduce the Leaf-based estimator

$$\hat{f}_S^{(Leaf)}(\boldsymbol{x}_S) = \frac{1}{Z(S,\boldsymbol{x})} \sum_{m \in C(S,\boldsymbol{x})} f_m \hat{P}_X^{(Leaf)}\left(L_m \left| \boldsymbol{X}_S \in L_m^S \right.\right) \tag{3.5}$$

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

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

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 $\boldsymbol{x}_S \in L_m^S$ across all the leaves of the tree.
We put emphasis on the correction needed for normalizing the probability: in general, we have $\sum_{m \in C(S,\boldsymbol{x})} \hat{P}_X^{(Leaf)}\left(L_m \left| \boldsymbol{X}_S \in L_m^S \right.\right) \neq 1$, because we do not condition by the same event (while we have $\sum_m P_X\left(L_m | \boldsymbol{X}_S = \boldsymbol{x}_S\right) = 1$). For this reason, the normalizing constant is defined as

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

The Leaf-based reduced predictor (3.5) 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 in the computation of $C(S, \boldsymbol{x})$, for every coalition $S$. We show in appendix B how the computational complexity of $\hat{f}_S^{(Leaf)}(\boldsymbol{x}_S)$ 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 Comparison of 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.

Let assume we have a dataset $\mathcal{D}_x^{Train} = \{(\boldsymbol{x}_i, z_i), i = 1, \dots, n\}$ with $n = 10000$ generated by a linear regression model with $X \in \mathbb{R}^p$, $X \in \mathcal{N}(\mu, \Sigma)$ and target $Z = B^t X$. We use a highly accurate

RandomForest $f$ trained on $\mathcal{D}_x^{Train}$, parameters can be found in Appendix I. Since we know the law of $\boldsymbol{X}$, we can compute exactly the SV with a Monte-Carlo estimator (MC).

We compare the true SV $\phi_i(f; \boldsymbol{x})$ and the SV of the different estimators $\phi_i(\hat{f}^{\alpha}; \boldsymbol{x}), \alpha = SHAP, Leaf, D$. To highlight the differences, we compute 3 metrics. For each estimator, we compute the R-MAE defined in (2.5), a True Positive Rate (TPR) to measure if the ranking of the top $k = 3$ highest and lowest SV is preserved.

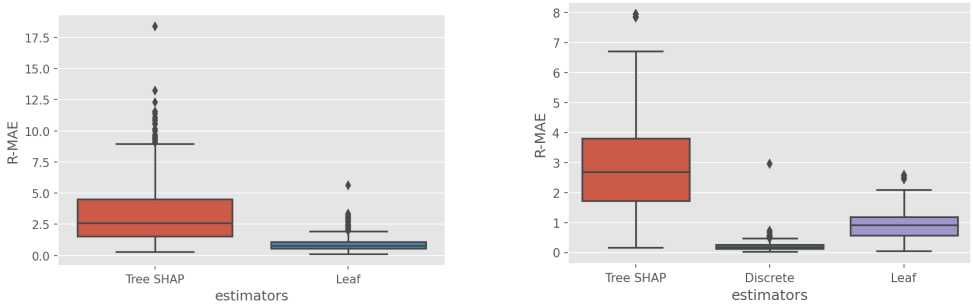


Figure 2: Left figure: R-MAE on 1000 new observations sampled from the synthetic model, p=5. Right figure: R-MAE on 1000 new observations sampled from the synthetic model, p=3

In the left of figure 3.3, we compute the SV $\phi_i(\hat{f}^{SHAP}; \boldsymbol{x})$, $\phi_i(\hat{f}^{Leaf}; x)$ on a $\mathcal{D}_x^{Explain}$ sampled 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. As also demonstrated in table 3.3, $\hat{f}^{Leaf}$ is capable of preserving the ranking of the top SV ($94\%$) outperforming $\hat{f}^{SHAP}$ ($86\%$).

We also measure the accuracy of the different estimators on out-of-distribution samples. We compute the metrics on observations sampled from a Uniform distribution. We observe in Table 3.3 (Uniform) that the precision has decreased showing that the estimator works less well on unlikely samples. Therefore, in order to reduce the uncertainty, we propose to use an IsolationForest [17] to detect the samples on which the estimators are bad.

We use an IsolationForest on $\mathcal{D}_x^{Explain}$ to detect anomaly samples. We see in the Table 3.3, after spliting the two groups that the precision has decreased on anomaly samples (Data w. Anomaly) and improved on the normal samples (Data w.o Anomaly). Indeed, the IsolationForest allows us to identify observations which fall into the leaves that are well covered, thus giving a better estimate.

| Dataset | $\mathcal{D}_x^{Explain}$ | | Uniform | | $\mathcal{D}_x^{Explain}$ w. Anomaly | | $\mathcal{D}_x^{Explain}$ w.o Anomaly | |
|---|---|---|---|---|---|---|---|---|
| Metrics | MSE | TPR | MSE | TPR | MSE | TPR | MSE | TPR |
| Tree SHAP | 3.31 | 86% (17%) | 7.73 | 77.46% (17%) | 6.52 | 82% (18%) | 2.86 | 86% (16%) |
| Leaf | 0.90 | 94% (12%) | 4.85 | 77.53% (19%) | 1.5 | 94% (13%) | 0.82 | 94% (12%) |

Table 1: Metrics of the estimators on the differents Datasets.

In the right of figure 3.3, we compare the SV of the Discrete unbiased estimator $\phi_i(\hat{f}^{(D)}; \boldsymbol{x})$, Tree SHAP $\phi_i^{\hat{SHAP}}(f; \boldsymbol{x})$ and Leaf estimator $\phi_i(\hat{f}^{(Leaf)}; \boldsymbol{x})$ with the True $\phi_i(f; \boldsymbol{x})$, where f was trained on the discretized version of $\mathcal{D}_x^{Train}$. As demonstrated in figure 3.3, the discrete estimator outperform Tree SHAP with a significant margin. However, the discretization makes all the estimators more sensible to out-of-distribution samples (see Appendix F). Indeed, the discretization can have a significant impact on the coverage of the leaves. Therefore, we suggest to used this estimators if the number of samples is large.

## 4 Focusing on influential variables with Same Decision Probabilities

An ideal aim of the SV analysis is to obtain a sparse additive explanation of the predictions, in order to get local simple and actionable rules for a complex model. We have derived in section 3 accurate

estimates of the SV in order to avoid attributing importance to a variable because of the estimation noise. Nevertheless, the estimation of sparse representation constitutes an additional estimation challenge: it is well known that the problem of variable selection is perturbed by the number of variables or their correlations. The same difficulties occur also for selecting important variables from SV's amplitudes and statistical accuracy is not enough, see for instance [20, 15, 16, 12]. Instead, we focus on locally influential variable as a way to identify sparse explanations. We describe below a piece-wise sparse model where the standard SV are perturbed by the global behavior of the model so that the sparsity is hidden. We propose a two-stage estimation procedure to correct this.

### 4.1 A motivating example for sparse explanations

We consider a binary classification $Y \sim \mathcal{B}(1, p(\boldsymbol{x}))$ with $\text{logit}(p(\boldsymbol{x})) = x_0 \times x_1$ if $x_4 < 0$, and $\text{logit}(p(\boldsymbol{x})) = x_2 \times x_3$ otherwise. We assume that $\boldsymbol{X} \in \mathbb{R}^8, \boldsymbol{X} \sim \mathcal{N}(0, I)$, so that the variables $X_5, X_6, X_7$ can be considered as noise variables. We fit then a Random Forest $f$. The accuracy of the model $f$ is good $90\%$, so we expect that it behaves like the true model $p(\boldsymbol{x})$. As a consequence, the SV $\phi_i(f; \boldsymbol{x})$ should reflect the role of the different variables. We choose an observation $\boldsymbol{x}_- = [-1.57, -4.15, -5.82, -5.90, -3.23, 0.71, -1.75, -1.27]$ with $x_4 < 0$ such that the active variables are: $X_0, X_1, X_4$. The estimated SV are given in table 2: the Leaf and TreeSHAP estimators are very close to the SV $\phi_i(f; \boldsymbol{x}_-)$ that is computed exactly by Monte-Carlo from $f$ (using the Gaussian assumption). In particular, we remark that TreeSHAP is consistent with the Leaf and theoretical values as the variables $\boldsymbol{X}$ are independent and in this case the assumption (eq. 3.2) is satisfied. In

| | $\phi_0$ | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ | $\phi_5$ | $\phi_6$ | $\phi_7$ |
|---|---|---|---|---|---|---|---|---|
| $f$ | -2.88 | -3.91 | -3.28 | -3.49 | 0.91 | -0.07 | -0.00 | -0.22 |
| $\hat{f}^{(Leaf)}$ | -2.91 | -3.86 | -3.35 | -3.53 | 0.90 | -0.10 | 0.043 | -0.24 |
| $\hat{f}^{(SHAP)}$ | -2.94 | -3.81 | -3.34 | -3.52 | 0.86 | -0.13 | 0.04 | -0.24 |

Table 2: SV of observation $\boldsymbol{x}_-$ given the different estimators

table (2), we see that the pure noise variables $X_5, X_6, X_7$ have very low SV, which confirms the ability of SV to detect variables with low impact on the prediction for observation $\boldsymbol{x}_-$. But the SV for $X_2, X_3$ seem unsatisfactory as they may suggest that they are as important as $X_0, X_1, X_4$, while they have no influence on the output (no impact in the formula, no correlation between features). The SV of $X_2, X_3$ seems important for $\boldsymbol{x}_-$ because these variables are influential for a sub-group of the population (when $x_4 > 0$): indeed, some contributions $f_{S\cup i}(\boldsymbol{x}_{S\cup i}) - f_S(\boldsymbol{x}_S)\, i = 2, 3$ are significantly different from zero while the decision for observation $\boldsymbol{x}_-$ does not depend on $X_2$ or $X_3$ locally (locality is measured by variable $X_4$). We emphasize that this paradox cannot be related to the effect of correlation between features, see Appendix E.
In the next section, we introduce a method for correcting this surprising behavior of the SV: we propose to identify first the local active variables, prior to compute the SV and to compute the SV only for this group of the active variables in order to highlight the individual effects of influential variables.

### 4.2 Detecting influential variables for computing Active Shapley Values

In order to recover the local sparsity of the model and obtain the corresponding importance attribution with the Shapley Values, we use the concept of the Same Decision Probability criterion, introduced in [7, 24] for measuring the robustness of classification decision. We show in our example how Same Decision Probabilities (SDP) can identify the group of influential variables on the decision.

**Definition 4.1.** ***(Same Decision Probability)*** *Let $f : \mathcal{X} \longrightarrow [0, 1]$ a probabilistic predictor and its classifier $D(\boldsymbol{x}) = \mathbb{1}_{f(\boldsymbol{x})\geq T}$ with threshold T, the Same Decision Probability of coalition $S \subset [\![1, p]\!]$, w.r.t $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$ is*

$$SDP_S\left(D; \boldsymbol{x}\right) = P\left(D(\boldsymbol{x}_S, \boldsymbol{X}_{\bar{S}}) = D(\boldsymbol{x}) \,|\boldsymbol{X}_S = \boldsymbol{x}_S\right)$$

SDP gives the probability to keep the same decision $D(\boldsymbol{x})$ when we do not observe the variables $\boldsymbol{X}_{\bar{S}}$. The higher is the probability, the better is the explanation based on $S$. Therefore, we focus on

the minimal subset of features such that the classifier makes the same decision with a given (high) probability $\pi$, given only them:

**Definition 4.2.** ***(Sufficient Coalition)****. Given $D$ a binary classifier, an observation $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$, $S \triangleq S^\star_\pi(\boldsymbol{x})$ is a Sufficient Coalition for probability $\pi$ if $SDP_{S^\star_\pi(\boldsymbol{x})}(D; \boldsymbol{x}) \geq \pi$ and no subset $Z$ of $S^\star_\pi(\boldsymbol{x})$ satisfies $SDP_Z(f; \boldsymbol{x}) \geq \pi$.*

In order to find the coalition $S^\star_\pi(\boldsymbol{x})$, we need to compute the SDP for any subset $S$. However, computing the SDP is known to be computationally hard: for simple Naive Bayes model and classifier, the computation SDP is known NP-hard [8]. Quite remarkably, we exploit the fact that the computation of SDP is related to Shapley Values: indeed, the SDP of tree-based models can be computed with reduced predictors. Based on the SDP of every coalition $S$, we can focus on the influential variables (for a given probability $\pi$) with a greedy algorithm that finds the sufficient coalition $S^\star_\pi(\boldsymbol{x})$, see Appendix C.

When we identify the influential variables $S^\star_\pi(\boldsymbol{x})$, we obtain simultaneously $N_\pi(\boldsymbol{x})$ formed by the remaining variables. By definition of SDP and $S^\star_\pi(\boldsymbol{x})$, the variables in $N_\pi(\boldsymbol{x})$ are not important for the prediction, because they don't change the prediction (with high probability): it is called the Null-Coalition. We will obtain a sparse explanation by computing the SV only for the active influential features. In order to get a formal definition of this procedure, we introduce a new XAI game where the SV of variables in $N_\pi(\boldsymbol{x})$ are fixed to zero.

**Definition 4.3.** ***(Active Shapley Values - ASV).*** *Let $f$ a model, $\boldsymbol{x}$ an instance, and the Sufficient and Null coalitions $S^\star_\pi(\boldsymbol{x})$ and $N_\pi(\boldsymbol{x})$ obtained for SDP $\geq \pi$. We define the new cooperative game with value function $v^\star$ such that for all $S$ in $S^\star_\pi(\boldsymbol{x})$, the value function is that*

$$v^\star(f; S) \triangleq f_{S \cup N_\pi(\boldsymbol{x})}(\boldsymbol{x}_{S \cup N_\pi(\boldsymbol{x})})$$

*and $v^\star(f; \emptyset) = E[f(\boldsymbol{X})]$. For all the variables $X_i$ in $S^\star_\pi$, we define the Active Shapley Value as*

$$\phi^\star_i(f; \boldsymbol{x}) = \frac{1}{|S^\star_\pi|} \sum_{k=0}^{|S^\star_\pi|-1} \frac{1}{\binom{|S^\star_\pi|-1}{k}} \sum_{S \in \mathcal{S}_k(S^\star_\pi(\boldsymbol{x}))} v^\star(S \cup i) - v^\star(S)$$

This game is different from the standard game (1.2) because we consider only the reduced predictors obtained by conditioning with the coalition $N_\pi(\boldsymbol{x})$. The accuracy formula gives a sparse additive explanation: $f(\boldsymbol{x}) - E[f(\boldsymbol{X})] = \sum_{i \in S^\star_\pi(\boldsymbol{x})} \phi^\star_i(f, \boldsymbol{x})$.

To illustrate ASV, we take the example and the observation above and compute the Sufficient Coalition $S^\star_\pi(\boldsymbol{x})$ with $\pi = 0.9$ and the Active SV $\phi^\star_i(f; \boldsymbol{x})$. We observe that the $S^\star$ found is $X_0, X_1, X_4$ with probability $SDP^\star_S(f; \boldsymbol{x}) = 0.95$ stating that the model does not rely on $X_2, X_3$ for this prediction. In addition, the ASV is $\phi_0 = -7.00$, $\phi_1 = -7.31$, $\phi_4 = 1.42$ and the SV of the remaining variables are zero. Its shows that ASV recover the individual effects of the important variables and they are consistent with the importance order of the variable of the classical SV while maintaining the variables in $N_\pi(\boldsymbol{x})$ to zero.

# 5 Conclusion

This paper is originally motivated by the societal impact of AI, and the need to build tools that promotes transparency in AI. In particular, Intelligibility and SV has attracted of lot of interest in the last year, but our detailed analysis show that there are still open questions for the transparent and reliable use of SV. We have put emphasis on the fact that SV are estimated from observations, and as estimators they can be derived from principled estimators. We can significantly reduce estimation noise and potential inferential errors. In addition, while the problem of variable selection has been very active in the statistics and ML community, their extension to Shapley Values is not straightforward and may even collapse. For this reason, we have adapted the concept of SDP for estimating the local influence of variables: this sheds a new light on SV, and it gives a way to sparsify additive explanation. Promising directions of research are to derive a way of estimating the uncertainty of SV estimators and to extend the SDP to the regression settings, as it considers only classification for the time being.

# References


[1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values, 2020.

[2] Kjersti Aas, Thomas Nagler, Martin Jullum, and Anders Løland. Explaining predictive models using shapley values and non-parametric vine copulas. *arXiv preprint arXiv:2102.06416*, 2021.

[3] David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.

[4] Quay Au, Julia Herbinger, Clemens Stachl, Bernd Bischl, and Giuseppe Casalicchio. Grouped feature importance and combined features effect plot. *arXiv preprint arXiv:2104.11688*, 2021.

[5] Clément Bénard, Gérard Biau, Sébastien Veiga, and Erwan Scornet. Interpretable random forests via rule extraction. In *International Conference on Artificial Intelligence and Statistics*, pages 937–945. PMLR, 2021.

[6] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020.

[7] S. Chen, Arthur Choi, and Adnan Darwiche. The same-decision probability: A new tool for decision making. 2012.

[8] Suming Chen, Arthur Choi, and Adnan Darwiche. An exact algorithm for computing the same-decision probability. IJCAI '13, page 2525–2531. AAAI Press, 2013.

[9] Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *arXiv preprint arXiv:2011.14878*, 2020.

[10] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5):1189–1232, 10 2001.

[11] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

[12] Daniel Fryer, Inga Strümke, and Hien Nguyen. Shapley values for feature selection: the good, the bad, and the axioms. *arXiv preprint arXiv:2102.10936*, 2021.

[13] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[14] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[15] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR, 2020.

[16] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR, 2020.

[17] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[18] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[19] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. 2017.

[20] Sisi Ma and Roshan Tourani. Predictive and causal implications of using shapley value for model interpretation. In Thuc Duy Le, Lin Liu, Kun Zhang, Emre Kiciman, Peng Cui, and Aapo Hyvärinen, editors, *Proceedings of the 2020 KDD Workshop on Causal Discovery (CD@KDD 2020), San Diego, CA, USA, 24 August 2020*, volume 127 of *Proceedings of Machine Learning Research*, pages 23–38. PMLR, 2020.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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, 2016.

[22] Lloyd S Shapley. Greedy function approximation: A gradient boosting machine. *Contribution to the Theory of Games*, 2:307–317, 1953.

[23] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11:1–18, 01 2010.

[24] Eric Wang, Pasha Khosravi, and Guy Van den Broeck. Towards probabilistic sufficient explanations. In *Extending Explainable AI Beyond Deep Models and Classifiers Workshop at ICML (XXAI)*, 2020.