Explaining probabilistic prediction on the simplex with Shapley compositions

Anonymous Authors[1]

Abstract

The concept of Shapley value has been widely use for measuring the contribution of each feature on a machine learning model's prediction. However, this has been designed for one-dimensional function's codomain. For multi-class probabilistic classifier, where the output is a discrete probability distribution over the set of more than two possible classes, the output lives on a multidimensional simplex. In this case, people have been applying the concept of Shapley value on each output dimension one-by-one, in an implicit one-vs-rest setting, ignoring the compositional nature of the output distribution where the relative information between probabilities matter. Using the Aitchison geometry of the simplex, coming from the field of compositional data analysis, this paper present a first initiative for a multidimensional extention of the concept of Shapley value, named Shapley composition, for explaining probabilistic predictions on the simplex in machine learning.

1. Introduction

Modern machine learning approaches like the one based on deep learning are often regarded as black-boxes making them not reliable for real-life application where the machine learnbing prediction has to be understood. These last years, the number of contribution to make models more explainable has therefore increased in the machine learning literature. One way to better understand a prediction would be to measure the contribution of each input features on the computation of the model output. The concept of Shapley value is now widely used for this purpose (Štrumbelj & Kononenko, 2014; Datta et al., 2016) especially since the release of the SHAP toolkit (Lundberg & Lee, 2017)[1]. The Shapley value came from cooperative game theory...

explain shapley in game theory,

How it is applied to ML,

Limitation,

...

1.1. Contributions

...

2. The Shapley value in machine learning

In this section, we recall the theoretical formulation of the Shapley value for measuring the contribution of each feature on a machine learning prediction.

Let $f : \mathcal{X} \to \mathbb{R}$ be a learned model one want to locally explain where $f(\boldsymbol{x})$ is the prediction on the instance $\boldsymbol{x} \in \mathcal{X}$. Let Pr be the probability distribution over $\mathcal{X}$ of the data[2]. Let $S \subseteq \mathcal{I} = \{1, 2, \dots d\}$, where d is the number of features that composes an instance $\boldsymbol{x} \in \mathcal{X}$, be a subset of indices. $\boldsymbol{x}_S$ refers to an instance $\boldsymbol{x}$ restricted to the features indicated by the indices in S.

When an instance $\boldsymbol{x}$ is observed, the expected value of the prediction is simply $\mathbb{E}[f(\boldsymbol{x}) \mid \boldsymbol{x}] = f(\boldsymbol{x})$. However, when only $\boldsymbol{x}_S$ is given, i.e. part of the features, there is uncertainty about the other features and we therefore compute the expected prediction given $\boldsymbol{x}_S$: $\mathbb{E}_{\mathrm{Pr}}[f(\boldsymbol{x}) \mid \boldsymbol{x}_S] = \int_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \mathrm{Pr}(\boldsymbol{x} \mid \boldsymbol{x}_S) d\boldsymbol{x}$. The contribution of the feature indexed by $i \notin S$ in the prediction $f(\boldsymbol{x})$ given the known features indexed by S is given by:

$$c_{f,\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S) = v_{f,\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) - v_{f,\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S), \quad (1)$$

where v is known as the value function:

$$\begin{aligned} v_{f,\boldsymbol{x},\mathrm{Pr}} : 2^{\mathcal{I}} &\to \mathbb{R}, \\ S &\mapsto \mathbb{E}_{\mathrm{Pr}}[f(\boldsymbol{x}) \mid \boldsymbol{x}_S], \end{aligned} \quad (2)$$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]https://github.com/shap/shap

[2]In practice, this is usually unknown but the expectation will be replaced by empirical samplings.

where $2^{\mathcal{X}}$ is the set of all subsets of I. This measure the contribution of the ith features with a particular coalition of features indexed by S. The whole contribution of the ith feature is computing by averaging this quantity over all possible coalitions as follow:

$$\phi_{f,\boldsymbol{x},\mathrm{Pr}}(i) = \frac{1}{d!}\sum_{\pi} c_{f,\boldsymbol{x},\mathrm{Pr}}(i, \pi_{\boldsymbol{X}}^{<i}), \tag{3}$$

where π is a permutation of the set I of indexes and $\pi_{\boldsymbol{X}}^{<i}$ is the features of $\boldsymbol{X}$ coming before the ith feature in the ordering given by π. For better clarity, the subscript ${}_{f,\boldsymbol{x},\mathrm{Pr}}$ wil be dropt from the equations.

This quantity is known as the Shapley value for the ith feature. It comes from cooperative game theory and is known to be only quantity respecting a set of desired axiomatic properties (Shapley et al., 1953). It is linear as a function of the model ($\alpha, \beta \in \mathbb{R}$): $\phi_{\alpha f + \beta g}(i) = \alpha\phi_f(i) + \beta\phi_g(i)$, and the "centered" learned model is additively separable with respect to the Shapley values: $(\boldsymbol{x}) - \mathbb{E}_{\mathrm{Pr}}[f(\boldsymbol{X})] = \sum_{i=1}^{d} \phi_f(i)$, which is known as the efficiency property.

Like originally developed in game theory, the Shapley value is designed for one-dimensinal codomain of the function f. For explaining machine learning models which output multidimensional discrete probability distribution, like in multiclass classification, people have been explaining each output dimension one-by-one, applying a logit transformation to the probabilities, resulting in a one-vs-rest comparison. However, this ignores the relative information between each probability and the compositional nature of probability distributions. Indeed, the probabilistic output of a classifier lives on a multidimensional simplex. The latter is the sample space of data refered as compositional data we briefly review in the next section.

3. Compositional data

Compositional data carries relative information. Each element of a composition describes a part of some whole (Pawlowsky-Glahn et al., 2015) like vectors of proportions, concentrations, and discrete probability distributions. A N-part composition is a vector of N non-zero positive real numbers that sum to a constant k. Each element of the vector is a part of the whole k. The sample space of compositional data is known as the simplex: $\mathcal{S}^N = \left\{\boldsymbol{x} = [x_1, x_2, \dots x_N]^T \in \mathbb{R}_+^{*N} \,\middle|\, \sum_{i=1}^{N} x_i = k\right\}$. In a composition, only the relative information between parts matters and John Aitchison introduced the use of log-ratios of components to handle this (Aitchison, 1982). He defined several operations on the simplex which leads to what is called the Aitchison geometry of the simplex.

3.1. The Aitchison geometry of the simplex

John Aitchison defined an internal operation called perturbation, an external one called powering and an inner product (Aitchison, 2001):

- a perturbation: $\boldsymbol{x} \oplus \boldsymbol{y} = \mathcal{C}\left([x_1 y_1, \dots x_N y_N]\right)$ seen as an addition between two compositions $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}^N$,
- a powering: $\alpha \odot \boldsymbol{x} = \mathcal{C}\left([x_1^\alpha, \dots x_N^\alpha]\right)$ seen as a multiplication by a scalar $\alpha \in \mathbb{R}$,
- an inner product:
$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle_a = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N} \log\frac{x_i}{x_j}\log\frac{y_i}{y_j}.$$

$\mathcal{C}(\cdot)$ is the closure operator. Since only the relative information matter, scaling factors are irrelevant and a composition $\boldsymbol{x}$ is equivalent to $\lambda\boldsymbol{x} = [\lambda x_1, \lambda x_2, \dots \lambda x_N]$ for all $\lambda > 0$. This equivalence is materialized by the closure operator defined for $k > 0$ as: $\mathcal{C}(\boldsymbol{x}) = \left[\frac{kx_1}{\|\boldsymbol{x}\|_1}, \frac{kx_2}{\|\boldsymbol{x}\|_1}, \dots \frac{kx_N}{\|\boldsymbol{x}\|_1}\right]^T$, where $\boldsymbol{x} \in \mathbb{R}_+^{*N}$ and $\|\boldsymbol{x}\|_1 = \displaystyle\sum_{i=1}^{N} |x_i|$.

This give to the simplex a $(N-1)$-dimensional Euclidean vector space structure called Aitchison geometry of the simplex. In this paper, since we are interested in classifiers' outputs as discrete probability distributions, we restrict ourselves to the probability simplex where $k = 1$.

3.2. The isometric log-ratio transformation

An $(N-1)$-dimensional orthonormal basis of the simplex, refered as an Aitchison orthonormal basis, can be built. The projection of a composition (a discrete probability distribution in our case) into this basis defined an isometric isomorphism between $\mathcal{S}^N$ and $\mathbb{R}^{N-1}$. This is known as an Isometric-Log-Ratio (ILR) transformation (Egozcue et al., 2003) and allows to express the compositions into a Cartesian coordinates system preserving the metric of the Aitchison geometry. Within this real space, the permutation, the powering and the Aitchison inner product defined above are respectively the standard addition, multiplication by a scalar and inner product.

Given a composition $\boldsymbol{p} = [p_1, \dots p_N]^T \in \mathcal{S}^N$ we express is ILR transformation as $\tilde{\boldsymbol{p}} = \mathrm{ilr}(\boldsymbol{p}) =$

$[\tilde{p}_1, \dots \tilde{p}_{N-1}]^T \in \mathbb{R}^{N-1}$. The ith element $\tilde{p}_i$ of $\tilde{\boldsymbol{p}}$ is be obtained as: $\tilde{p}_i = \langle \boldsymbol{p}, \boldsymbol{e}^{(i)} \rangle_a$ where set $\{\boldsymbol{e}^{(i)} \in \mathcal{S}^N, i = 1, \dots N-1\}$ forms an Aitchison orthonormal basis of the simplex. The choice of the basis will be discussed in Section 5.1.3.

4. Shapley on the simplex

In Section 2 we have briefly presented the standard formulation of Shapley value designed for one-dimensional prediction. In this Section, we will see how the Aitchison geometry can be used to extend this concept to the multidimensional simplex for explaning probabilistic predictions.

Let $\boldsymbol{f} : \mathcal{X} \to \mathcal{S}^N$ be a learned model, like a N-classes probabilistic classifier for instance, which outputs a probabilistic prediction on the $(N-1)$-dimensional probability simplex S^N. To properly consider the relative information between the probabilities, The outputs of the model must be treated as compositional data using the operators and metric defined by the Aitchison geometry of the simplex. We therefore rewrite the contribution and the value function of Equations 1 and 2 as follow:

$$\boldsymbol{c}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S) = \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \ominus \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S), \quad (4)$$

where $\boldsymbol{a} \ominus \boldsymbol{b}$ is the perturbation $\boldsymbol{a} \oplus ((-1) \odot \boldsymbol{b})$ which correspond to a substraction between compositions $\boldsymbol{a}$ and $\boldsymbol{b}$, and where:

$$\begin{aligned} \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}} &: 2^{\mathcal{X}} \to \mathcal{S}^N, \\ \boldsymbol{X}_S &\mapsto \mathbb{E}^{\mathcal{A}}_{\mathrm{Pr}}[\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x}_S]. \end{aligned} \quad (5)$$

The $\mathcal{A}$ in superscript highlight the fact that the expectation is done with respect to the Aitchison measure, rather than the Lebesgue measure, which can simply be computed as (Pawlowsky-Glahn et al., 2015): $\mathbb{E}^{\mathcal{A}}[\boldsymbol{Y}] = \mathrm{ilr}^{-1}(\mathbb{E}[\mathrm{ilr}(\boldsymbol{Y})])$, where $\mathbb{E}^{\mathcal{A}}$ refers to the expectation with respect to the Aitchison measure while $\mathbb{E}$ refers to the expectation with respect to the Lebesgue measure.

The Shapley quantity expressing the contribution of the ith feature on a prediction can simply be expressed on the simplex as the Shapley composition $\boldsymbol{\phi}(i)$ given by:

$$\boldsymbol{\phi}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i) = \frac{1}{d!} \odot \bigoplus_{\pi} \boldsymbol{c}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i, \pi^{<i}_{\boldsymbol{X}}). \quad (6)$$

It can be shown (in Appendix A) that the linearity and the efficiency properties naturally hold for the Shapley composition:

$$\begin{aligned} \boldsymbol{\phi}_{\alpha\odot\boldsymbol{f}(\boldsymbol{x})\oplus\beta\odot\boldsymbol{g}(\boldsymbol{x})}(i) &= \alpha \odot \boldsymbol{\phi}_{\boldsymbol{f}}(i) \oplus \beta \odot \boldsymbol{\phi}_{\boldsymbol{g}}(i), \\ \bigoplus_{i=1}^{d} \boldsymbol{\phi}_{\boldsymbol{f}}(i) &= \boldsymbol{f}(\boldsymbol{x}) \ominus \mathbb{E}^{\mathcal{A}}_{\mathrm{Pr}}[\boldsymbol{f}(\boldsymbol{X})]. \end{aligned} \quad (7)$$

This can be seen as a multidimensional extension of the Shapley value framework on the simplex. Here, the Shapley quantity is not a scalar anymore, this is a composition living on the probability simplex. In the next section, we will see in more details how this can be used to explain the contribution of the features on a multidimensional probabilistic prediction.

5. Explaining probabilistic prediction with Shapley compositions

Given a prediction $\boldsymbol{f}(\boldsymbol{x})$, the Shapley composition $\boldsymbol{\phi}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i)$ describes the contribution of the ith feature on the prediction. The efficiency property shows how the probability distribution moves from the base value, i.e. the expected prediction regardless of the current input, to the prediction $\boldsymbol{f}(\boldsymbol{x})$. In the standard Shapley formulation recalled in Section 2, the prediction is one-dimensional such that the Shapley quantity is a scalar. In application where there are more than two possible classes, the prediction is multidimensional such that the Shapley quantity is too. Both lives on the same space: the probability simplex. In this section, we discuss how the set of Shapley compositions can be analysed to better understand the contribution and influence of each features on the prediction.

5.1. Visualization

The Shapley compositions can be visualized in the Euclidean space isometric to the simplex thanks to the ILR transformation presented in Section 3.2. This space has the advantage of being intuitive since it is the standard real $(N-1)$-dimensional vector space we are used too.

5.1.1. Three classes

In the three classes case, the space is 2-dimensional. We illustrate this example with the well known Iris classification dataset consisting of a set of flowers descriped by 4 features: sepal length and width and petal length and width. The aim is to predict to which of the three species, setosa, versicolor and virginica, a flower belongs to. In our example we use a Support Vector Machine (SVM) classifier with a radial basis function (rbf) kernel as a classifier. Figure 1 shows the explanation of one versicolor instance where 1(a) shows the Shapley composition in the ILR space and

1(b) shows how they move the base distribution to the prediction. Having the highest norm, the petal length is the feature contributing the most on the prediction, moving the base to the versicolor maximum probability decision region. Being orthogonal to the virginica class-composition, this suggest that this features does not contribute on the predicted probability for this class. The Shapley composition for the petal width goes straight to the opposite direction of the setosa class vector suggesting that this feature contributes in rejecting this class. The other Shapley compositions have a low norm suggesting these features does not contribute in the prediction.

5.1.2. Four classes

In a four classes example, the simplex is 3-dimensional. We illustrate this with a simple digit recognition task[3]. It consists of classifying an 8×8 image as representing one of the digit among: 0, 1, 2 and 3. Since they are 64 pixels as a set of features, which would correspond to 64 Shapley composition, we reduce the number of features to 6 using a principal component analysis for better clarity. We again use a SVM classifier with an rbf kernel. The same explanation analysis as before can be applied here but within a 3-dimensional plot as illustrated by Figure 2. To better understand how this space is divided into four regions each representing the maximum probabity region for one class, one can think about the shape of a methane molecule. The hydrogens correspond to the vextices and the carbon to the center of a tetrahedron i.e. a 3-dimensional simplex. The relative position of the class-compositions in the ILR space are the same as the bonds between the corbon and a hydrogen: the angles are $\approx 109.5°$.

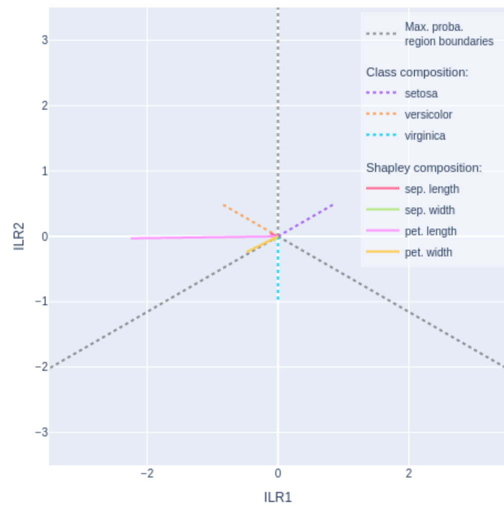
5.1.3. More classes

When there are more classes, all the dimension of the ILR space cannot be visualized at once but 2 or 3-dimensional subspaces can still be visualized. However, in order to choose which ILR components to visualize, one needs to understand what they refer too. The next section provides some intuition about that in the context of the explanation of a classifier prediction with more than 4 classes.

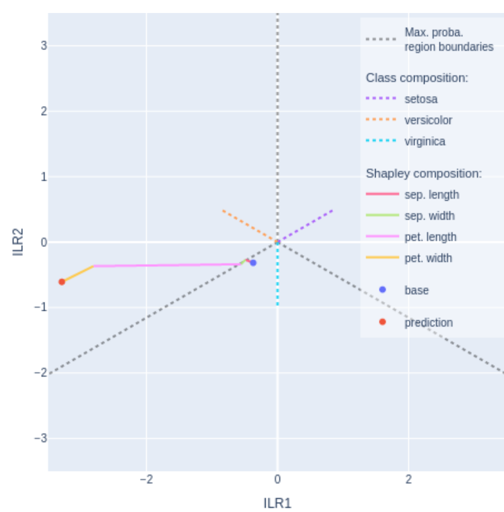
5.2. Groups of part, balances and sequential binary partition

....

[3] We use the scikit-learn's digits dataset(Pedregosa et al., 2011).



(a) Shapley compositions in the ILR space



(b) Sum of the Shapley compositions in the ILR space from the base to the prediction

Figure 1. Shapley explaination in the ILR space for the classification of an Iris instance.

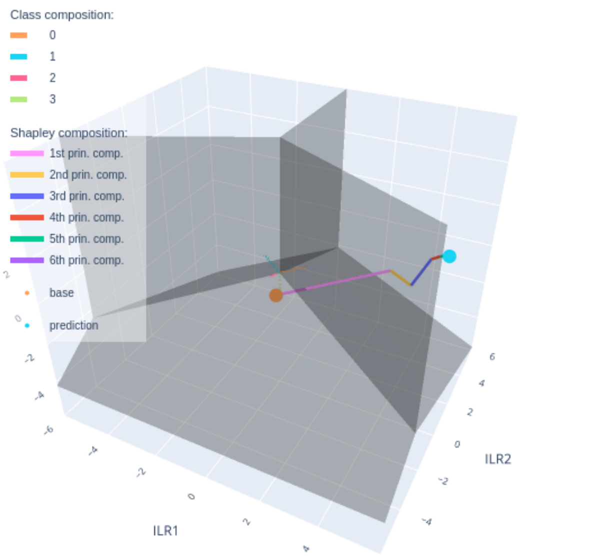


Figure 2. Shapley explaination in the 3-dimensional ILR space for a four classes digit recognition task. The Shapley compositions are summed in the ILR space from the base distribution to the prediction.

5.3. Angles, norms and projections

Some may find the fine analysis of the features contributions in cases with more than four classes tricky. Indeed, in this case, the ILR space cannot be visualized in a 2 or 3-dimensional plot and as discussed in Section 5.1.3, choosing which subspaces to visualize require a careful understanding of the sequential binary partition. However, as we already had the insight from the above visualization, the Shapley explaination can be summarized by sets of angles, norms and projections. Indeed the norm of a Shapley composition gives the strength of the feature's contribution in the prediction. The angle between two Shapley compositions can informs if features are redundant, orthogonal or contradictory. The projection of a Shapley composition on the set of class-compositions informs in favor of, or against, which classes a feature is contributing. Appendix C, provides some examples of summarizing a Shapley explanation using the norm of Shapley compositions, angles between them and their projection on the class-compositions.

5.4. Histograms

If one found hard to visualize the proposed Shapley explanation in the ILR space, the Shapley composition can be visualized as histograms like discrete probability distributions.

5.5. About our implementation

In this work, the estimation algorithm we used to compute the Shapley compositions is an adaptation of Algorithm 2 in (Štrumbelj & Kononenko, 2014). Since the resulting Shapley compositions are approximations, the efficiency property does not necessarily hold without an adjustement. Each Shapley compositions are therefore corrected following a similar method as in the sampling approximation in the SHAP toolkit (Lundberg & Lee, 2017)[4]. See Appendix D and E for more details.

6. Discussion and conclusion

Compare with standard Shapley ??

We know small expé..., sounds tricky,... first step for a theoretically founded multiclass problems explanations... ... Features INDEPENDENCE!!

Axiomatic formulation

References

Aitchison, J. The statistical analysis of compositional data. Journal of the Royal Statistical Society. Series B (Methodological), 44(2):139–177, 1982.

Aitchison, J. Simplicial inference. In Marlos A. G. Viana, D. S. P. R. e. A. S. S. o. A. M. i. S. (ed.), Algebraic Methods in Statistics and Probability, Contemporary Mathematics 287. American Mathematical Society, 2001.

Datta, A., Sen, S., and Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In 2016 IEEE Symposium on Security and Privacy (SP), pp. 598–617, 2016. doi: 10.1109/SP.2016.42.

Egozcue, J. J., Pawlowsky-Glahn, V., Mateu-Figueras, G., and Barcelo-Vidal, C. Isometric logratio transformations for compositional data analysis. Mathematical geology, 35(3):279–300, 2003.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc., 2017.

Pawlowsky-Glahn, V., Egozcue, J. J., and Tolosana-

[4] https://github.com/shap/shap/blob/master/shap/explainers/_sampling.py

Delgado, R. Modeling and Analysis of Compositional Data. John Wiley & Sons, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

Shapley, L. S. et al. A value for n-person games. pp. 307–317, 1953.

Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. Knowledge and information systems, 41:647–665, 2014.

A. Linearity and efficiency of the Shapley composition

In this section, we show the linearity of the Shapley composition with respect to the model prediction, and the efficiency property.

A.1. Linearity

The Shapley composition is linear, within the Aitchison geometry of the simplex, with respect to linear combination of models' predictions.

Proof. Let's consider the linear combination of predictions $\boldsymbol{h}(\boldsymbol{x}) = \alpha \odot \boldsymbol{f}(\boldsymbol{x}) \oplus \beta \odot \boldsymbol{g}(\boldsymbol{x})$. we want to check if:

$$\boldsymbol{\phi}_{\boldsymbol{h}}(i) = \alpha \odot \boldsymbol{\phi}_{\boldsymbol{f}}(i) \oplus \beta \odot \boldsymbol{\phi}_{\boldsymbol{g}}(i). \tag{8}$$

We have:

$$\begin{aligned}
\mathbb{E}^{\mathcal{A}}_{\mathrm{Pr}}[\boldsymbol{h}(\boldsymbol{x}) \mid \boldsymbol{x}_S] &= \mathrm{ilr}^{-1}\left(\mathbb{E}_{\mathrm{Pr}}[\mathrm{ilr}\left(\alpha \odot \boldsymbol{f}(\boldsymbol{x}) \oplus \beta \odot \boldsymbol{g}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S]\right), \\
&= \mathrm{ilr}^{-1}\left(\mathbb{E}_{\mathrm{Pr}}[\alpha\, \mathrm{ilr}\left(\boldsymbol{f}(\boldsymbol{x})\right) + \beta\, \mathrm{ilr}\left(\boldsymbol{g}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S]\right), \\
&= \mathrm{ilr}^{-1}\left(\alpha \mathbb{E}_{\mathrm{Pr}}[\mathrm{ilr}\left(\boldsymbol{f}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S] + \beta \mathbb{E}_{\mathrm{Pr}}[\mathrm{ilr}\left(\boldsymbol{g}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S]\right), \\
&= \alpha \odot \mathrm{ilr}^{-1}\left(\mathbb{E}_{\mathrm{Pr}}[\mathrm{ilr}\left(\boldsymbol{f}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S]\right) \oplus \beta \odot \mathrm{ilr}^{-1}\left(\mathbb{E}_{\mathrm{Pr}}[\mathrm{ilr}\left(\boldsymbol{g}(\boldsymbol{x})\right) \mid \boldsymbol{x}_S]\right), \\
&= \alpha \odot \mathbb{E}^{\mathcal{A}}_{\mathrm{Pr}}[\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x}_S] \oplus \beta \odot \mathbb{E}^{\mathcal{A}}_{\mathrm{Pr}}[\boldsymbol{g}(\boldsymbol{x}) \mid \boldsymbol{x}_S].
\end{aligned} \tag{9}$$

Therefore, $\boldsymbol{v}_{\boldsymbol{h},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S) = \alpha \odot \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S) \oplus \beta \odot \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S)$, meaning that $\boldsymbol{v}$ is linear with respect to the learned function or model. The linearity of the contribution $\boldsymbol{c}$ naturally follows:

$$\begin{aligned}
\boldsymbol{c}_{\boldsymbol{h},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S) &= \boldsymbol{v}_{\boldsymbol{h},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \ominus \boldsymbol{v}_{\boldsymbol{h},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S), \\
&= \left(\alpha \odot \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \oplus \beta \odot \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}})\right) \ominus \left(\alpha \odot \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S) \oplus \beta \odot \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S)\right), \\
&= \alpha \odot \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \oplus \beta \odot \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \ominus \alpha \odot \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S) \ominus \beta \odot \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S), \\
&= \alpha \odot \left(\boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \ominus \boldsymbol{v}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S)\right) \oplus \beta \odot \left(\boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_{S\cup\{i\}}) \ominus \boldsymbol{v}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(\boldsymbol{X}_S)\right), \\
&= \alpha \odot \boldsymbol{c}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S) \oplus \beta \odot \boldsymbol{c}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S).
\end{aligned} \tag{10}$$

And the linearity of the Shap composition:

$$\begin{aligned}
\boldsymbol{\phi}_{\boldsymbol{h}}(i) &= \frac{1}{d!} \bigoplus_{\pi} \boldsymbol{c}_{\boldsymbol{h},\boldsymbol{x},\mathrm{Pr}}(i, \pi^{<i}_{\boldsymbol{X}}), \\
&= \frac{1}{d!} \bigoplus_{\pi} \left(\alpha \odot \boldsymbol{c}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S) \oplus \beta \odot \boldsymbol{c}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S)\right), \\
&= \alpha \odot \left(\frac{1}{d!} \bigoplus_{\pi} \boldsymbol{c}_{\boldsymbol{f},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S)\right) \oplus \beta \odot \left(\frac{1}{d!} \bigoplus_{\pi} \boldsymbol{c}_{\boldsymbol{g},\boldsymbol{x},\mathrm{Pr}}(i, \boldsymbol{X}_S)\right), \\
&= \alpha \odot \boldsymbol{\phi}_{\boldsymbol{f}}(i) \oplus \beta \odot \boldsymbol{\phi}_{\boldsymbol{g}}(i).
\end{aligned} \tag{11}$$

□

A.2. Efficiency

The efficiency property naturally holds for Shapley compositions within the Aitchison geometry.

Proof.

$$
\begin{aligned}
\bigoplus_{i=1}^{d}\boldsymbol{\phi}_{\boldsymbol{f}}(i) &= \bigoplus_{i=1}^{d}\left(\frac{1}{d!}\odot\bigoplus_{\pi}\boldsymbol{c}(i,\pi_{\boldsymbol{X}}^{<i})\right), \\
&= \frac{1}{d!}\odot\bigoplus_{i=1}^{d}\left(\bigoplus_{\pi}\left(\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<i+1})\ominus\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<i})\right)\right), \\
&= \frac{1}{d!}\odot\bigoplus_{i=1}^{d}\left(\underbrace{\left(\bigoplus_{\pi}\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<i+1})\right)}_{\boldsymbol{A}_{i+1}}\ominus\underbrace{\left(\bigoplus_{\pi}\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<i})\right)}_{\boldsymbol{A}_{i}}\right), \\
&= \frac{1}{d!}\odot\bigoplus_{i=1}^{d}\left(\boldsymbol{A}_{i+1}\ominus\boldsymbol{A}_{i}\right), \\
&= \frac{1}{d!}\odot\left(\boldsymbol{A}_{d+1}\ominus\boldsymbol{A}_{1}\right),\ \text{since we have a telescoping perturbation}, \\
&= \frac{1}{d!}\odot\left(\left(\bigoplus_{\pi}\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<d+1})\right)\ominus\left(\bigoplus_{\pi}\boldsymbol{v}(\pi_{\boldsymbol{X}}^{<1})\right)\right), \\
&= \frac{1}{d!}\odot\left(\left(\bigoplus_{\pi}\boldsymbol{v}(\boldsymbol{X})\right)\ominus\left(\bigoplus_{\pi}\boldsymbol{v}(\boldsymbol{X}_{\emptyset})\right)\right), \\
&= \boldsymbol{v}(\boldsymbol{X})\ominus\boldsymbol{v}(\boldsymbol{X}_{\emptyset}),\ \text{since } d! \text{ is the number of different permutation}, \\
&= \boldsymbol{f}(\boldsymbol{x})\ominus\mathbb{E}_{\mathrm{Pr}}^{\mathcal{A}}[\boldsymbol{f}(\boldsymbol{X})].
\end{aligned}
\tag{12}
$$

□

B. Class-compositions

A k-class-compositions $\boldsymbol{c}^{(k)} \in \mathcal{S}^N$ is defined as an unit norm composition going straight to the direction of the kth class. This is a discrete probability distribution with maximum probability for the kth class and uniform values for the others. The ith part of the $\boldsymbol{c}^{(k)}$ is:

$$c_i^{(k)} = \begin{cases} 1-(N-1)p, & \text{if } i = k \\ p, & \text{otherwise,} \end{cases} \tag{13}$$

where $p < \frac{1}{N}$. We want the Aitchison norm of each class-composition to be one:

$$\begin{aligned}
\forall k \in \{1, \dots N\}, \qquad \|\boldsymbol{c}^{(k)}\|_a = 1 &\iff \sqrt{\frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\log\frac{c_i^{(k)}}{c_j^{(k)}}\right)^2} = 1, \\
\text{for clarity,}& \\
\text{we drop the } (k) \text{ from the equations,}& \\
&\iff \sqrt{\frac{1}{2N}\sum_{i=1}^{N}\left((N-1)\left(\log\frac{c_i}{p}\right)^2 + \left(\log\frac{c_i}{1-(N-1)p}\right)^2\right)} = 1, \\
&\iff \sqrt{\frac{1}{2N}2(N-1)\left(\log\frac{p}{1-(N-1)p}\right)^2} = 1, \\
\text{since } p < \frac{1}{N} \text{ and the norm should be positive:}& \\
&\iff \sqrt{\frac{N-1}{N}}\log\frac{1-(N-1)p}{p} = 1, \\
&\iff p = \frac{\exp\left(-\sqrt{\frac{N}{N-1}}\right)}{1+(N-1)\exp\left(-\sqrt{\frac{N}{N-1}}\right)}.
\end{aligned} \tag{14}$$

To summarize, the ith part of a k-class-compositions $\boldsymbol{c}^{(k)} \in \mathcal{S}^N$ is given by:

$$c_i^{(k)} = \frac{1}{1+(N-1)\exp\left(-\sqrt{\frac{N}{N-1}}\right)}\left(\begin{cases} 1, & \text{if } i = k \\ \exp\left(-\sqrt{\frac{N}{N-1}}\right), & \text{otherwise,} \end{cases}\right). \tag{15}$$

In this way, $\boldsymbol{c}^{(k)}$ is going straight to the direction of class k and uniformly against all the others.

C. Summarizing the explanation with norms, angles and projections of Shapley compositions

The example of Figure 1 can be summarized by computing the norms of the Shapley compositions, angles between them and their projection on the set of class-compositions.

D. Estimation of the Shapley compositions

In this work, we used an adaptation of Algorithm 2 in (Štrumbelj & Kononenko, 2014) we present in this section.

Let d be the number of features. We want to optimally distribute the $m_{\max}$ drawn samples over the d features. Let $\hat{\boldsymbol{\phi}}_i$ the estimation of the Shapley composition for the ith feature. We want to minimize the sum of squared errors or distances $\sum_{i=1}^{d} \|\hat{\boldsymbol{\phi}}_i \ominus \boldsymbol{\phi}_i\|_a^2$. Since $\hat{\boldsymbol{\phi}}_i$ is a (Aitchison) sample mean we have: $\tilde{\hat{\boldsymbol{\phi}}}_i \approx \mathcal{N}(\tilde{\boldsymbol{\phi}}_i, \frac{1}{m_i}\boldsymbol{\Sigma}^{(i)})$ and $\tilde{\hat{\boldsymbol{\phi}}}_i - \tilde{\boldsymbol{\phi}}_i \approx \mathcal{N}(\mathbf{0}, \frac{1}{m_i}\boldsymbol{\Sigma}^{(i)})$ where the tilde refers to the ILR transformation. Let $\boldsymbol{\Delta_i} = \tilde{\hat{\boldsymbol{\phi}}}_i - \tilde{\boldsymbol{\phi}}_i$ and $Z_i = \|\hat{\boldsymbol{\phi}}_i \ominus \boldsymbol{\phi}_i\|_a = \|\tilde{\hat{\boldsymbol{\phi}}}_i - \tilde{\boldsymbol{\phi}}_i\|_2 = \|\boldsymbol{\Delta}_i\|_2$. The expectation of the sum of squared errors is:

$$
\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{d} Z_i^2\right] &= \sum_{i=1}^{d} \mathbb{E}\left[Z_i^2\right], \\
&= \sum_{i=1}^{d} \mathbb{E}\left[\sum_{j=1}^{d-1} \Delta_{ij}^2\right], \\
&= \sum_{i=1}^{d}\sum_{j=1}^{d-1} \mathbb{E}\left[\Delta_{ij}^2\right], \\
&= \sum_{i=1}^{d}\sum_{j=1}^{d-1} \frac{1}{m_i}\Sigma_{jj}^{(i)}, \text{ since } \Delta_{ij} \approx \mathcal{Z}(0, \frac{1}{m_i}\Sigma_{jj}^{(i)}), \\
&= \sum_{i=1}^{d} \frac{1}{m_i} \operatorname{tr} \boldsymbol{\Sigma}^{(i)}.
\end{aligned} \tag{16}
$$

When a sample is drawn, the feature for which the sample will be used for improving the Shapley composition estimation is chosen to maximize $\frac{\operatorname{tr}\boldsymbol{\Sigma}^{(i)}}{m_i} - \frac{\operatorname{tr}\boldsymbol{\Sigma}^{(i)}}{m_i+1}$. Like in (Štrumbelj & Kononenko, 2014), this is summarized in Algorithm 2.

Algorithm 1 Adaptation of the Algorithm 1 in (Štrumbelj & Kononenko, 2014) for approximating the Shapley composition of the ith feature, with model $\boldsymbol{f}$, instance $\boldsymbol{x} \in \mathcal{X}$ and m drawn samples.

Initialize $\boldsymbol{\phi_i} \leftarrow \operatorname{ilr}^{-1}(\mathbf{0})$
for 1 to m do
  Randomly select a permutation π of the set of indexes $\mathcal{I}$,
  Randomly select a sample $\boldsymbol{w} \in \mathcal{X}$,
  Construct two instances:

- $\boldsymbol{b}_1$: which takes the values from $\boldsymbol{x}$ for the ith feature and the features indexed before i in the order given by π, and takes the values from $\boldsymbol{w}$ otherwise,
- $\boldsymbol{b}_2$: which takes the values from $\boldsymbol{x}$ the features indexed before i in the order given by π, and takes the values from $\boldsymbol{w}$ otherwise.

  $\boldsymbol{\phi}_i \leftarrow \boldsymbol{\phi}_i \oplus \boldsymbol{f}(\boldsymbol{b}_1) \ominus \boldsymbol{f}(\boldsymbol{b}_2)$
end for
$\boldsymbol{\phi}_i \leftarrow \frac{\boldsymbol{\phi}_i}{m}$

Algorithm 2 Adaptation of the Algorithm 2 in (Štrumbelj & Kononenko, 2014) for approximating all the Shapley compositions by optimally distributing a maximum number of samples $m_{\max}$ over the d features, with model $\boldsymbol{f}$, instance $\boldsymbol{x} \in \mathcal{X}$ and $m_{\min}$ the minimum number of samples each feature estimation.

Initialization: $m_i \leftarrow 0$, $\boldsymbol{\phi}_i \leftarrow \mathbf{0}$, $\forall i \in \{1, \ldots d\}$,
while $\sum_{i=1}^{d} m_i < m_{\max}$ do
  if $\forall i, m_i \leq m_{\min}$ then
    $j = \underset{i}{\operatorname{argmax}} \left(\frac{\operatorname{tr} \boldsymbol{\Sigma}^{(i)}}{m_i} - \frac{\operatorname{tr} \boldsymbol{\Sigma}^{(i)}}{m_i + 1} \right)$,
  else
    pick a j such that $m_j < m_{\min}$,
  end if
  $\boldsymbol{\phi}_j \leftarrow \boldsymbol{\phi}_j +$result of Algorithm 1 for the jth feature and $m = 1$,
  update $\boldsymbol{\Sigma}^{(j)}$ using an incremental algorithm,
  $m_j \leftarrow m_j + 1$
end while
$\boldsymbol{\phi}_i \leftarrow \frac{\boldsymbol{\phi}_i}{m_i}$, $\forall i \in \{1, \ldots d\}$.

E. Adjustement of the estimated Shapley compositions for efficiency

In practice, the computation of the Shapley values has an exponential time complexity and we do not have necessarily access to the true distribution of the data. The Shapley values are therefore approximated using estimation algorithms like for instance the one presented in the previous appendix. However, since the obtained values are approximations, they do not necessarily respect the desired efficiency property. This point is often overlooked in the literature. In this section we write down an adjustment strategy of the estimated Shapley compositions for them to respect the efficiency property. This is a similar strategy as in the sampling approximation of the Shapley values in the SHAP toolkit (Lundberg & Lee, 2017)[5].

Let $\{\hat{\boldsymbol{\phi}}_i\}_{1 \leq i \leq d}$ be the estimated Shapley compositions (given by the Algorithm 2 in our experiments). Let $\boldsymbol{s}_{err} = \boldsymbol{f}(\boldsymbol{x}) \ominus \boldsymbol{f}_0 \ominus \bigoplus_{i=1}^{d} \hat{\boldsymbol{\phi}}_i$, where is the base $\boldsymbol{f}_0$ composition, be the error composition on the pertubation of all Shapley compositions, i.e. the error making the efficiency property unfulfilled. In order to respect the efficiency property, we want this error to be the neutral element of the perturnation, i.e. the "zero" in the sense of the Aitchison geometry, i.e the uniform distribution. We could simply perturb each estimated Shapley compositions by $\frac{1}{d} \odot \boldsymbol{s}_{err}$ however this would move each Shapley composition by the same amount while we want to allow the compositions with a higher estimation variance (i.e. with a precision likely to be lower) to move more than the ones with a smaller variance (i.e. with a precision likely to be higher).

We therefore weight the ith adjustment by a scalar $w_i = w\left(\operatorname{tr}\left(\boldsymbol{\Sigma}^{(i)}\right)\right)$, where w is an increasing function, and where $\sum_{i=1}^{d} w_i = 1$. Note that the vector of weight is actually a composition too. Similarly to the SHAP toolkit implementation, we choose w as:

$$w_i = w\left(\operatorname{tr}\left(\boldsymbol{\Sigma}^{(i)}\right)\right) = \frac{v_i}{1 + \sum_{j=1}^{d} v_j}, \text{ where } v_i = \frac{\operatorname{tr}\left(\boldsymbol{\Sigma}^{(i)}\right)}{\epsilon \max_j \operatorname{tr}\left(\boldsymbol{\Sigma}^{(j)}\right)}. \tag{17}$$

The ith estimated Shapley composition is then asjusted as follow:

$$\hat{\boldsymbol{\phi}}_i \leftarrow \hat{\boldsymbol{\phi}}_i \oplus \left(w_i \odot \boldsymbol{s}_{err}\right). \tag{18}$$

[5] https://github.com/shap/shap/blob/master/shap/explainers/_sampling.py

In this way, when ϵ goes to zero, the efficiency property is respected for the adjusted Shapley compositions and more weight is given to the adjusments of the Shapley compositions with a higher estimation variance.