Shapley Values

Imagine the following example: You have several features, these could be columns in a table or pixels in an image. Consider the case where two features j and i transmit the same information. Furthermore, imagine that the signal the features transmit is critical for the model's prediction task. A first naive approach for determining the importance of each feature would be to calculate the marginal contribution of the feature $f_\theta(n) - f_\theta(n \backslash i)$, where n is the set of all features. Surely important features when removed will result in large changes to the model output? The equation says that we should calculate the model's output with and without the feature we are analyzing, and that the difference is directly proportional to that feature's importance. But this intuition quickly breaks down for our hypothetical example. If we remove feature i, since feature j transmits the same information, the predictive weight will be picked up by our feature that remains in the set, resulting in $f_\theta(n) - f_\theta(n \backslash i) = 0$. Similarly, when we remove feature j, feature i will compensate again, resulting in $f_\theta(n) - f_\theta(n \backslash j) = 0$. In this situation, removing features one by one and replacing them can result in the two most important features being ranked as least important. The calculation fails in this case because of the existence of a strong correlation between our two features. The simple calculation is not sufficient to extract feature importance when features are correlated. We need a method that respects all the possible intercorrelations between features.

The idea of the Shapley value was derived from coalitional game theory, where participants collaborating in a game had to fairly distribute the rewards based on their individual contribution. The mathematics governing this fair distribution can easily be mapped onto the machine learning domain in the case of classification problems and is a remedy for the above situation. Instead of players participating in a game, we have a collection of either feature columns, pixels, or words. The value function v which ordinarily could refer to some black box economic model is replaced by the machine learning model f_θ, and finally the reward itself gets mapped to the model's classification score. The Shapley value of a particular feature i of instance x is given by

$$\phi_i(x) = \sum_{S \subseteq N \backslash \{i\}} \frac{|S|! \times (|N| - |S| - 1)!}{|N|!} \left(f_\theta(S \cup \{i\}) - f_\theta(S) \right),$$

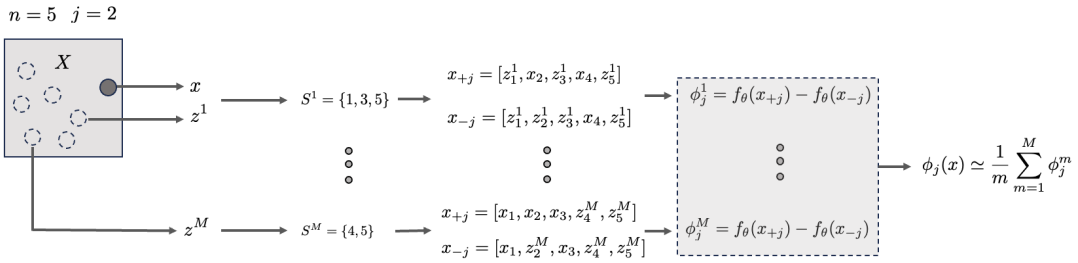
where:
- N is the set of all features.
- S is a subset of N that does not include feature i.
- $f_\theta(S)$ is the prediction of the model using only the features in subset S.
- $|S|$ is the number of features in subset S.
- $|N|$ is the total number of features.

The formula says that the Shapley value is the average marginal contribution of a feature value across all possible coalitions. Here we have assumed that the value function, telling us the worth of the coalition, is given by the model's prediction, i.e., $v \to f_\theta$, which can be computationally expensive for large models. In practice, this function need not be the model's forward function.

There exists a variety of methods for approximating the above equation. This variety is due in part to computational restraints as well as constraints imposed on the calculation process by the model's architecture. Calculating Shapley values exactly is oftentimes not computationally feasible when dealing with large numbers of features. Therefore, we often reduce the computational demands by either subsampling, using Monte Carlo approximations, or replacing the value function with the model gradients. Although this speeds up the calculation, the desirable axioms of fairness, such as efficiency, where the set of Shapley values over all features sums to the model's output $\sum \phi_i(x) = \hat{y}$, is no longer guaranteed to hold. Additionally, ML models usually require all features to pipe information through its channels at all times. Think for example of a simple neural

network. In these settings, it is unclear what it means to create a subset of feature coalitions, resulting in the use of baselines to artificially emulate the turning off of features in the presence of information still flowing through the channels.
The Shapley value formalism might be the only explanatory formalism with a solid mathematical foundation that offers "axiom's of fairness". These axioms include the efficiency property, where the sum of the attributions of the features is equivalent to the models output deviation from the norm, and the ability to generate contrastive explanation that can be compared on an instance level.
Because Shapley values are local, it becomes difficult to plot the results. One cannot visually capture all the relevant information within a single plot, leading to many different plotting methods that reveal different insights and information. Additionally, the local nature of the calculation can, if packaged correctly, be plotted in such a way to tell us something global. We now look at different methods that allow us to quickly approximate Shapley values.

Monte Carlo Method for Calculating Shapley Values



Using the Monte Carlo method, we can approximate the Shapley values by randomly sampling permutations of the feature set and computing the marginal contributions. The algorithm can be seen above.
Procedure: Select an instance $\mathbf{x}$ and a feature (say $j = 2$) out of the $n = 5$ possible features. We then randomly sample an instance $\mathbf{z}_m$ from the dataset and randomly generate a subset $S_m = \{\cdot\}$ where $0 \leq |S| \leq n - 1$, not including the index $j = 2$. This randomly selects some collection of columns/features in the case of tabular data. Then, construct two new instances:

1. $\mathbf{x}_{+j}$, which replaces all feature values at the indices in S with the corresponding $\mathbf{z}_m$ values,
2. $\mathbf{x}_{-j}$, which is the same as $\mathbf{x}_{+j}$ but additionally the $j = 2$ feature has also been substituted.

Calculate the marginal contribution $\phi_j^m = f_\theta(\mathbf{x}_{+j}) - f_\theta(\mathbf{x}_{-j})$ and repeat the process M times to calculate the approximate Shapley value $\phi_j(x) \approx \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$.
The intuition for this method relies on the idea of shuffling to remove a feature channels net effect on the models output. If we feed our model an instance and calculate the score multiple time while randomly selecting and exchanging the value of a particular feature to the value of a random instance in the dataset, then then average effect from that channel cancels out. In this way random sampling and taking expectation values over a large amount of iterations is an effective candidate for turning feature channels off, thus allowing us to generate the subsets seen in the formal equation. Note that the normalization factor is simply $1/M$, which due to the law of large numbers guarantees that, over many samples, each feature's contribution will be fairly averaged in a manner that mirrors the exact Shapley value calculation.

Although Shapley values are mathematically attractive with great properties their are some disadvantages with the method. 1) Shapley values do not produce a prediction model, meaning that one cannot ask hypothetical questions of what would happen to the output if variable j increased its value, this can be detrimental for clients that prefer

explanations of how they could adjust their behaviour to achieve a particular outcome. The explanations are dense and are at the level of the number of inputs, where many times sparse explanations are preferred. To calculate explanations one always needs access to the entire dataset, or a substantial subset in order to evaluate a new instance, since the instances values are calculated using the dataset as background. Finally Shapley values methods may include unrealistic data instances.

Axioms of Fairness

Shapley values are desirable because they obey four axioms of fairness

- Efficiency: The sum of all feature attributions equals the output prediction of the model $\sum_i \phi_i(x) = f_\theta(x)$ allowing for the comparison of attributions between instances. This axiom ensures that the entire prediction is distributed among the features without any leftover. It guarantees that the attributions account for the full effect of the prediction, allowing for a complete and comprehensive explanation of the model's output. Without this axiom, we could end up with an incomplete or misleading interpretation where the sum of the contributions does not match the model's prediction.
- Symmetry: If two features (i, j) are equally critical for the model output, they will have equivalent Shapley values $\phi_i = \phi_j$. This axiom enforces fairness by ensuring that features which play an identical role in contributing to the output are treated equally. This prevents bias in attributing more importance to one feature over another when their contributions are identical, thus ensuring a fair and unbiased distribution of importance.
- Null player: If a feature has zero marginal contribution to any coalition (i.e., it does not change the prediction regardless of whether it is included), then its Shapley value should be zero. This axiom ensures that irrelevant features are assigned no importance. If a feature does not influence the prediction, it should not be credited with any of the prediction value. This prevents noise or redundant features from being incorrectly attributed importance, maintaining the integrity and relevance of the attributions.
- Additivity/linearity: The additivity axiom (also known as linearity) states that if you have two models f and g, the Shapley values for their combined model $f + g$ should be the sum of the Shapley values for each individual model. Mathematically, $\phi_i(f+g) = \phi_i(f)+\phi_i(g)$. This axiom ensures that the Shapley values are consistent across models and combinations of models. It supports the idea that the contribution of each feature should be additive and consistent, allowing for straightforward interpretation even when dealing with complex models that can be decomposed into simpler parts.