# The Many Shapley Values for Model Explanation

**Mukund Sundararajan** [1] **Amir Najmi** [1]

## Abstract

The Shapley value has become the basis for several methods that attribute the prediction of a machine-learning model on an input to its base features. The use of the Shapley value is justified by citing the uniqueness result from (Shapley, 1953), which shows that it is the only method that satisfies certain good properties (**axioms**). There are, however, a multiplicity of ways in which the Shapley value is operationalized for model explanation. These differ in how they reference the model, the training data, and the explanation context. Hence they differ in output, rendering the uniqueness result inapplicable. Furthermore, the techniques that rely on they training data produce non-intuitive attributions, for instance unused features can still receive attribution. In this paper, we use the axiomatic approach to study the differences between some of the many operationalizations of the Shapley value for attribution. We discuss a technique called Baseline Shapley (BShap), provide a proper uniqueness result for it, and contrast it with two other techniques from prior literature, Integrated Gradients (Sundararajan et al., 2017) and Conditional Expectation Shapley (Lundberg & Lee, 2017).

## 1. Motivation and Related Work

We discuss the **attribution problem**, i.e., the problem of distributing the prediction score of a model for a specific input to its base features (cf. (Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017)); the attribution to a base feature can be interpreted as the importance of the feature to the prediction. For instance, when attribution is applied to a model that makes loan decisions, the attributions tell you how influential a feature was to the loan decision for a specific loan applicant. Attributions thus have explanatory value.

One of the leading approaches to attribution is based on the Shapley value (Shapley, 1953), a construct from cooperative game theory. In cooperative game theory, a group of players come together to consume a service, and this incurs some cost. The Shapley value distributes this cost among the players. There is a correspondence between cost-sharing and the attribution problem: The cost function is analogous to the model, the players to base features, and the cost-shares to the attributions.

The Shapley value is known to be the unique method that satisfies certain properties (see Section 2.1 for more details). The desirability of these properties, and the uniqueness result make a strong case for using the Shapley value. Unfortunately, despite the uniqueness result, there are a multiplicity of Shapley values that differ in how they refer to the model, the training data, and the explanation context. Here is a sampling of the literature to illustrate the variety of approaches:

1. (Lindeman et al., 1980; Grömping, 2007)) uses the Shapley value to attribute the goodness of fit ($R^2$) of a linear regression model to its features by retraining the model on different feature subsets.

2. (Owen, 2014; Owen & Prieur, 2017) apply the Shapley value to study the importance of a feature to a given function, by using it to identify the "variance explained" by the feature; no retraining involved.

3. (Štrumbelj et al., 2009; Štrumbelj & Kononenko, 2014) use the Shapley value to solve the attribution problem, i.e., feature importance for a specific prediction. The first paper applies the Shapley value by retraining the model on every possible subset of the features. The second paper applies the Shapley value to the conditional expectation of a specific model (no retraining) (see Section 2.1.1 for a formal definition of the conditional expectation approach). They assume that features are distributed uniformly and independently.

4. (Datta et al., 2016) applies the Shapley value to the conditional expectations of the model's function with a contrived distribution that is the product of the marginals of the underlying feature distribution.

[1]Google LLC. Correspondence to: Mukund Sundararajan <mukunds@google.com>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Online, PMLR 119, 2020. 

5. (Lundberg & Lee, 2017) also investigates the Shapley value with conditional expectations; it constructs various approximations that make assumptions about either the function, or the distribution, and applies it compositionally on modules of a deep network.

6. (Lundberg et al., 2018) computes the Shapley value with conditional expectations efficiently for trees; however, it is not very clear about its assumptions on the feature distribution [1].

7. (Aas et al., 2019) generalizes one of the approaches in (Lundberg & Lee, 2017) to the case when the distributions are not independent, either by assuming that the features are generated by a mixture of Gaussians or by a non-parametric, heuristic approach that applies the Mahalanobis distance to the empirical distribution.

8. Unlike the methods above that either delete or marginalize over a feature, (Sun & Sundararajan, 2011; Sundararajan et al., 2017; Agarwal et al., 2019) apply the Shapley value, by using a different approach to 'turn features off'. This approach takes an auxiliary input called a baseline, and switches the explicand's feature value to the value of the feature in the baseline (see Section 2.2 for details).

9. (Sundararajan et al., 2017) proposes a technique called Integrated Gradients, that is based on the Aumann-Shapley (Aumann & Shapley, 1974) cost-sharing technique. Aumann-Shapley is one of the several extensions of the discrete Shapley value to continuous settings. Also, this technique is applicable only when the gradient of the prediction score with respect to the base features is well-defined, and is therefore not applicable to models like tree ensembles.

The first and second approaches solve a different problem (of feature importance across all the training data), and we will ignore them for the most part. Notice that the rest are solving the same attribution problem, and are reflective of the non-uniqueness of the Shapley value for model explanation. (Lundberg & Lee, 2017) unifies several of these methods (excluding Integrated Gradients) under a common framework based on certain conditional expectations over feature distributions. However, as we point out later in the paper, the choice of feature distribution influences the attributions significantly, not just in quantity, but also in quality.

[1] In an email exchange, Scott Lundberg clarified that the implicit assumption is that the features are distributed according to "the distribution generated by the tree".

### 1.1. Our Results

We identify two reasons for the multiplicity of Shapley values. The first is whether training data plays a role (in addition to the model) in the definition of the Shapley value or not. Section 3 discusses a previously proposed approach from (Lundberg & Lee, 2017) where the training data does play a role. We show that **sparsity** of the training data obscures properties of the model; for instance, an unused feature may still accrue importance. (A recent paper by (Janzing et al., 2020) discusses the same issue using the vocabulary of Pearl causality; they discuss that this approach relies on observational conditional probabilities instead of 'interventional' conditional probabilities.)

In Section 2.2, we study definitions of the Shapley value that do not depend on the training data. Here, the multiplicity arises because there are multiple extensions of the Shapley value to continuous features; recall that the Shapley value is implicitly defined for binary features. We study two such methods. A simple extension of the Shapley value that we call Baseline Shapley, and Integrated Gradients (Sundararajan et al., 2017). We provide uniqueness results for these methods via a reductions to prior results from cooperative game theory.

## 2. Preliminaries

We model the machine-learning model as a real-valued function $f$ that takes a vector of real-valued features as input. If the problem is a classification problem, the function models the score of a class. The set of features is denoted by $N$. We designate the input to be explained, i.e., the **explicand**, by the vector $x$ of features; when we say $x_S$ we mean the sub-vector of a vector $x$ restricted to the features in the set $S$.

At times, we may assume that the features are generated according to a distribution $D$; this distribution could be a posited distribution as in Section 4.3. Often, it is the empirical distribution of the training data, wherein it is written as $\hat{D}$. Of special significance are independent feature distributions. The product of marginals of distribution $D$ is written as $\Pi(D)$. The conditional expectation $E[f(x)|x_S]$ is the expected value of the function over the distribution with the features in $S$ fixed at the explicand's value.

### 2.1. Shapley value

The Shapley value takes as input a set function $v : 2^N \to R$. The Shapley value produces attributions $s_i$ for each player $i \in N$ that add up to $v(N)$. The Shapley value of a player $i$ is given by:

$$s_i = \sum_{S \subseteq N \setminus i} \frac{|S|! * (|N| - |S| - 1)!}{N!} (v(S \cup i) - v(S)) \tag{1}$$

There is an alternate permutation-based description of the Shapley value: Order the players uniformly at random, add them one at a time in this order, and assign to each player $i$ its expected marginal contribution $V(S \cup i) - v(S)$; here $S$ is the set of players that precede $i$ in the ordering.

In this paper, we study three extensions of the Shapley value to model explanation.

#### 2.1.1. Conditional Expectations Shapley (CES)

This approach takes three inputs: an explicand $x$, a function $f$, and a distribution $D$. The set function is defined by the conditional expectation

$$v(S) = E_D[f(x')|x'_S = x_S] \tag{2}$$

We denote the CES attribution for feature $i$ with explicand $x$, distribution $D$ and function $f$ by $ces_i(x, D, f)$. This approach has been used by (Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017; Datta et al., 2016) and was proposed in this specific form by (Lundberg & Lee, 2017), where it is called Shapley Additive Explanations, or SHAP. (The associated library employs variants of the Shapley value that resemble the Baseline Shapley approach discussed next. Therefore to prevent confusion, we call this approach CES and not SHAP. Furthermore, a lot of the criticism that we apply to CES do not carry over to the SHAP library.) When CES is carried out with the empirical distribution of the training $\hat{D}$, it will be denoted as CES($\hat{D}$).

### 2.2. Baseline Shapley (BShap)

This approach takes as input an explicand $x$, the function $f$ and an auxiliary input called the baseline $x'$. The set function is defined as:

$$v(S) = f(x_S; x'_{N \setminus S}) \tag{3}$$

That is, we model a feature's absence using its value in the baseline. We call this the Baseline Shapley (BShap) approach. We denote BShap attribution by $bs_i(x, x', f)$. Variants of this approach have been used by (Sun & Sundararajan, 2011; Agarwal et al., 2019; Lundberg & Lee, 2017).

#### 2.2.1. Random Baseline Shapley (RBShap)

This approach is a variant of BShap that takes three inputs: An explicand $x$, a function $f$, and a distribution $D$. The attributions are the expected BShap values, where the baseline $x'$ is drawn randomly according to the distribution $D$.

This approach is implicit in (Lundberg & Lee, 2017) (see Equation 11).

$$v(S) = E_{x' \sim D} f(x_S; x'_{N \setminus S}) \tag{4}$$

#### 2.2.2. Integrated Gradients (IG)

This approach takes as input an explicand $x$, the function $f$ and an auxiliary input called the baseline $x'$. We consider the straight-line path (in $\mathsf{R}^{|N|}$) from the baseline $x'$ to the input $x$, and compute the gradients at all points along the path. The path can be parameterized as $\gamma(x, \alpha) = x' + \alpha \cdot (x - x')$. Integrated gradients are obtained by accumulating these gradients. The integrated gradients attribution for an explicand $x$ and baseline $x'$, for a variable $x_i$ is:

$$IG_i(x, x', f) = (x_i - x'_i) \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha \tag{5}$$

IG is an analog of the Aumann-Shapley method from cost-sharing (Aumann & Shapley, 1974). We will discuss the sense in which IG is an extension of the Shapley value in Section 4.1.

### 2.3. Axioms

We now list several desirable properties of an attribution technique and discuss why each property is desirable. Later, we will use these properties as a framework to compare and contrast various attribution methods. Variants of these axioms have appeared in prior cost-sharing literature (cf. (Friedman & Moulin, 1999)).

An attribution method satisfies:

- **Dummy** if dummy features get zero attributions. A feature $i$ is dummy in a function $f$ if for any two values $x_i$ and $x'_i$ and every value $x_{N \setminus i}$ of the other features, $f(x_i; x_{N \setminus i}) = f(x'_i; x_{N \setminus i})$; this is just a formal way of saying that the feature is not referenced by the model, and it is natural to require such variables to get zero attributions.
- **Efficiency** if for every explicand $x$, and baseline $x'$, the attributions add up to the difference $f(x) - f(x')$ for the baseline approach. For the conditional expectation approach, $f(x')$ is replaced by $E_{x' \sim D}[f(x')]$. This axiom can be seen as part of the framing of the attribution problem; we would like to apportion blame of the entire difference $f(x) - f(x')$ to the features.
- **Linearity** if, feature by feature, the attributions of the linear combination of two functions $f_1$ and $f_2$ is the linear combination of the attributions for each of the two functions. Attributions represent a kind of forced linearization of the function. It is therefore desirable to preserve the existing linear structure in the function.

- **Symmetry** if for every function $f$ that is symmetric in two variables $i$ and $j$, if the explicand $x$ and baseline $x'$ are such that $x_i = x_j$ and $x'_i = x'_j$, then the attributions for $i$ and $j$ should be equal. This is a natural requirement with obvious justification.

- **Affine Scale Invariance (ASI)** if the attributions are invariant under a simultaneous affine transformation of the function and the features. That is, for any $c, d$, if $f_1(x_1, \ldots, x_n) = f_2(x_1, ..., (x_i - d)/c, ..., x_n)$, then for all $i$ we have $attr_i(x, x', f_1) = attr_i((x_1, \ldots, c * x_i + d, \ldots x_n), (x'_1, \ldots, c * x'_i + d, \ldots x'_n), f_2)$. ASI conveys the idea that the zero point and the units of a feature should not determine its attribution. Here is a concrete example: Imagine a model that takes temperature as a feature. ASI dictates that whether temperature is measured in Celsius or Farenheit, the attribution to the feature should be identical.

- **Demand Monotonicity** if the model is monotone in a feature (e.g. a linear model has a positive coefficient for the 'age') then the attributions for a feature increase with increasing feature values (say when 'age=20' is changed to 'age=21'), all else being fixed, including the baseline value for the feature.

- **Proportionality** If the function $f$ can be rewritten as a function of $\sum_i x_i$, and the baseline (x') is zero, then the attributions are proportional to the explicand values(x).

### 2.4. An Empirical Case Study: Diabetes Prediction

While the bulk of this paper is axiomatic and theoretical, we will replicate some of our observations on a diabetes prediction task. The motivation is to show that many of the issues we identify theoretically show up in practice, and that too in the simplest possible setting, indicating that the issues are commonplace. We train our diabetes prediction models on a data set from the Scikit learning library (Pedregosa et al., 2011); this data set was originally used in (Efron et al., 2004). The data has ten base features, age, sex, body mass index (BMI), average blood pressure (BP), and six blood serum measurements. Data is obtained for each of 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after the time of measurement of the base features.

We train a linear model using Scikit's implementation of Lasso regression (Tibshirani, 1996); we used the standard settings of the fitting algorithm and 75%-25% train-test split. The variance explained by the model is 35%. The model coefficients are 399 for BMI, 4.9 for BP and 291 for the fifth blood serum measurement (s5). The intercept is 154.15, which closely matches the data set average of response.

## 3. An Analysis of CES

While CES was proposed by (Lundberg & Lee, 2017), and justified axiomatically (by citing the original Shapley axiomatization), the justification did not cover the choice of using conditional expectations as the set function (recall the definition of CES in Section 2.1.1). Furthermore, it appears that CES has only been applied with modification: For instance, (Štrumbelj & Kononenko, 2014), assumes an independent feature distribution while (Lundberg & Lee, 2017) applies it to modules of a deep network rather than end-to-end. Consequently, the properties have CES have not been carefully studied. This is what we remedy in this section.

### 3.1. Comprehending CES($\hat{D}$)

As discussed in Sections 1 and 2.1.1, CES attributions depend crucially on the choice of the distribution $D$. Arguably, the most obvious choice is to use the training data distribution $\hat{D}$; we call this CES($\hat{D}$).

Unfortunately, the properties of CES($\hat{D}$) are not immediately apparent from its definition; the functional forms of the Shapley value and conditional expectations are sufficiently complex as to prevent direct understanding. We therefore begin by redefining CES($\hat{D}$) as an intuitive procedure.

The input is the (training) data, i.e., a list of examples $T = \{x^t\}$. (We use superscripts to index examples and subscripts to index features.) Given an explicand $x$, $T_S$ is the subset of $T$ that agrees with $x$ on the features in the set $S$, i.e., $T_S = \{x^t | \forall i \in S, x^t_i = x_i\}$. Notice that $T_{\{\}} = T$, and $T_N = \{x_i\}$. The value of the set function $v(S)$ is the average value of the function over inputs in the set $T_S$; this corresponds to computing the conditional expectation $E[f(x)|x_S)]$ in the CES approach (Section 2.1.1).

We now use the procedural definition of the Shapley value (see Section 2.1), i.e., we average the marginal contribution of 'adding' variable $i$ (i.e., conditioning on it) over permutations of the variables. We notice that conditioning on an additional variable $i$ only reduces examples that 'agree' with $x$ on the conditioned features. We call this the Downward Closure property[2]:

**Lemma 3.1** (Downward Closure)**.** *For every pair of sets of features $S, S'$ , if $S \subseteq S'$ then $T_{S'} \subseteq T_S$. (Proof in Section 5)*

Putting these observations together, we have Algorithm 1. (If needed, the computation can be further sped up by sam-

[2]We borrow the term from the frequent itemset mining literature (cf. (Agrawal & Srikant, 1994)). In itemset mining, the downward closure property reflects that every subset of a frequent itemset is also frequent. Analogously, for every feature set $S$, every row in $T_S$ is also in $T_{S'}$ for every subset $S' \subseteq S$.

pling over permutations as is common with the Shapley value, and by caching values of the sets $T_i$ for all $i$.)

**Algorithm 1** Computing CES($\hat{D}$)

Inputs: explicand $x$ and examples $T$, each over feature set $N$
{Compute Shapley values via permutations}
$s_{\sigma_i} \leftarrow 0$ for all $i$
**for all** permutations $\sigma$ of $N$ **do**
  $v_{new} \leftarrow \frac{1}{|T|}\sum_{x \in T} f(x)$
  $T' \leftarrow T$
  **for all** $i \in 1 \ldots |N|$ **do**
    $v_{old} \leftarrow v_{new}$
    {Use the Downward Closure Lemma to update $T'$}
    **for all** $t \in T'$ **do**
      {$\sigma_i$ is the $i$th feature in the ordering $\sigma$}
      **if** $x^t_{\sigma_i} \neq x_i$ **then**
        delete $t$ from $T'$
      **end if**
    **end for**
    $v_{new} \leftarrow \frac{1}{|T'|}\sum_{x \in T'} f(x)$
    {Update Shapley value of $i$th feature in ordering $\sigma$}
    $s_{\sigma_i} \leftarrow s_{\sigma_i} + \frac{1}{|N|!}(v_{new} - v_{old})$
  **end for**
**end for**

### 3.2. The Effect of Sparsity on CES($\hat{D}$)

Our main motivation for providing a procedure for CES($\hat{D}$) is to understand its properties. Our first observation is that CES($\hat{D}$) is extremely sensitive to the degree of sparsity; sparsity arises naturally when the variables are continuous because it unlikely that data points share feature values precisely.

**Remark 3.2.** *Suppose we have an explicand $x$ such that every feature value $x_i$ is unique, i.e., it does not occur elsewhere in the training data. Then, notice that $T_S = \{x\}$ for all non-empty sets $S$. Therefore in each permutation, the first feature gets attribution $f(x) - E_{x' \; D}[f(x')]$ while all the other features get an attribution of zero. Therefore all the variables get **equal** attributions, even if the function is not symmetric in the variables!*

A practical implication of Remark 3.2 is that the attributions would be very sensitive to noise in the data. For instance, Figure 1 shows the distribution of attributions across 20 explicands for CES($\hat{D}$) (the second column in each plot) on the linear model. The attributions vary across features—for instance BMI has a larger variation than Sex. If we add a tiny amount of noise, and recompute attributions, then **all** the features (including BMI and Sex) will get **identical** attributions (we don't show this in the figure).

One way to deal with this sensitivity is to **smooth** the data. We can simulate smoothing within Algorithm 1. When we condition on a set $S$ of features in the computation of CES, we average the prediction over all the training data points that are **close** to the explicand in each of the features in $S$; two data points are close in a certain feature if their difference is within a certain fraction of the standard deviation. In our experiments, we use two settings $0.1$ and $0.2$. Figure 1 shows how different amounts of smoothing change the attributions (see for instance the attributions of the feature S2). Thus while smoothing mitigates sensitivity, it is still unclear how much smoothing to do.

There are some other approaches to dealing with sparsity. One approach is to use the distribution $\Pi(\hat{D})$, i.e., the product of empirical marginal distributions, as in (Datta et al., 2016), or to assume that the function is somewhat smooth, and to compute the function's value at a point outside the training data using a weighted sum of nearby points in the training data as in (Aas et al., 2019). Again, these will undoubtedly give different results, and we cannot easily pick between them.

### 3.3. An Axiomatic Analysis of CES($\hat{D}$)

As discussed in Section 2.1, the Shapley value and its variants were conceived in the context of cooperative game-theory where there is no analog of the feature distribution $D$. The axioms (see Section 2.3) were meant to guarantee that if the set function has certain properties (the antecedent), then the Shapley values must have certain properties (the consequent). For instance, the Dummy axiom says that if a function is insensitive to a feature (the antecedent), then the feature should have zero attributions (the consequent). However, there are two functions at work here: the function $f$ whose value we wish to attribute, and the set function $v$ used to compute Shapley values.

CES defines $v(S)$ using conditional expectations that depend both on the function $f$ and the distribution $D$. Consequently, even if the function $f$ satisfies certain properties, the consequent property of the axiom need not hold for $v$. This gives rise to counter-intuitive attributions. We give several such examples in this section. These do not imply that prior axiomatizations (cf. (Lundberg & Lee, 2017) or (Datta et al., 2016)) are incorrect. The various axioms still hold for the $v$ based on the conditional expectation (see Section 2.1.1). However, the axioms have no natural interpretation for this set function. In contrast, it is natural to seek interpret axioms as properties of the model function $f$.

In all our examples, each row of the table is a combination of feature values; the first column specifies the probability of this feature combination, the next two columns specify the feature values for two discrete, abstract features $x$ and $y$, and the remaining columns specify the value of the function for this feature combination. Feature and function values

used as explicand in the examples are in bold.

**Example 3.3** (Failure of Dummy)**.** *See the function $f_1$ in Table 1 modeled as a bivariate function of $x$ and $y$. For the explicand $x = 5, y = 5$, the CES attributions are $\frac{22.5}{2}$ each for $x$ and $y$ (a consequence of Remark 3.2). Therefore the variable $x$ gets a large attribution despite being dummy.*

| Probability | $x$ | $y$ | $f_1$=$y^2$ | $f_2 = x$ | $f_1 + f_2$ |
|---|---|---|---|---|---|
| $\epsilon$ | **5** | **5** | **25** | **5** | **30** |
| $\frac{1-\epsilon}{2}$ | 1 | 1 | 1 | 1 | 2 |
| $\frac{1-\epsilon}{2}$ | 1 | 2 | 4 | 1 | 5 |

*Table 1.* Example for: (a) Dummy, correlated variables can have large CES attributions. (b) CES attributions are not linear in the function.

The implication is this: Say, in the context of an analysis of fairness, we require that a certain feature play no role in the prediction model, and indeed, it does not. If we use CES, it may still be assigned significant attribution, leading us to incorrectly believe that the function is sensitive to the variable. In our diabetes prediction task, for the linear model (see Figure 1), we note that 7 of the 10 variables are dummy features. Despite this, CES assigns non-zero attributions to them. (In contrast, BShap assigns zero attributions to the dummy features.)

**Example 3.4** (Failure of Linearity)**.** *See the functions $f_1$ and $f_2$ in Table 1; model them as univariate functions of $y$ and $x$ respectively. Consider the explicand $x = 5, y = 5$. Then the CES for the variable $y$ with the function $f_1$ is the difference between the function value at the explicand ($25$) and the mean of the function ($2.5$), i.e., $22.5$, and that for the function $f_2$ is zero ($y$ does not appear in the function). Now consider the attribution of $y$ for the function $f_1 + f_2$; both variables get an attribution of $\frac{30-3.5}{2}$ (again, a consequence of Remark 3.2), which is not $22.5 + 0$.*

Here is an implication: Imagine, if we were computing attributions for an ensemble of trees. Recall that the prediction of the forest is a uniform average over the trees, i.e., it is linear in the prediction of the trees. Therefore, we would expect the attributions to also be linear. But this is not the case. We saw large failures in linearity for the diabetes prediction task even for a two tree ensemble with trees of depth two.

## 4. Baseline Shapley and its Properties

In this section, we discuss the properties of BShap and provide a proper axiomatic result for it. (Recall the definition of Baseline Shapley (BShap) from Section 2.2.) As discussed in the introduction, prior axiomatization results from the machine learning literature (e.g. (Datta et al., 2016; Štrumbelj & Kononenko, 2014; Lundberg & Lee, 2017)) did not cover the choice of function input to the Shapley value, and consequently there are a multiplicity of methods that yield different results.

### 4.1. BShap versus IG

The model explanation literature has largely built on top of the Shapley value from the binary cost-sharing literature. In this literature, players (features) are either present or absent. In contrast, machine learning tends to involve continuous features and deep learning involves only continuous features–even discrete/categorical are often turned into continuous features via embeddings. Therefore, it is worth connecting model explanation with the continuous cost-sharing literature.

We begin by defining cost-sharing formally.

A **cost-sharing problem** is an attribution problem with function $f$, explicand $x$ and baseline $x'$ such that the baseline $x' = 0$, the explicand $x$ is non-negative, and the function $f$ is non-decreasing in each variable, i.e., if two feature vectors $x^- \leq x^+$ (point-wise for every feature), then $f(x^-) \leq f(x^+)$.[3]

There are several extensions of the Shapley value to the continuous cost sharing literature (see (Friedman & Moulin, 1999) for details). Two of these extensions correspond to BShap and IG. BShap is a generalization of a classic cost-sharing method called Shapley-Shubik (cf. (Friedman & Moulin, 1999)), and IG is a generalization of a cost-sharing method called Aumann-Shapley (cf. (Aumann & Shapley, 1974)). One can get Shapley-Shubik from BShap and Aumann-Shapley from IG by setting the baseline $x'$ to zero; the explicand value $x_i$ corresponds to the demand of player $i$, the function $f$ corresponds to the cost incurred, and the attributions to the cost-shares.

It is relatively clear how Shapley-Shubik (BShap) is an extension of the binary Shapley value from its definition (see Section 2.2).

However it is less clear how Aumann-Shapley (IG) (Equation 5) is an extension of the binary Shapley value. IG traverses a single, smooth path between the baseline and the explicand, and aggregates the gradients along this path. Whereas the Shapley value takes an average over several discrete paths—in each step of a discrete path, a variable goes from being 'off' to 'on' in one shot. To establish the connection, notice that the IG path can be seen to be the internal diagonal of a $|N|$ dimensional hypercube, and in contrast, the Shapley value is an average over the extremal paths over the edges of this hypercube. Suppose we partition every feature $i$ into $m$ micro features, where each micro feature

[3] In (Friedman & Moulin, 1999), the function is defined to satisfy an additional property of being zero at $f(0)$; but this is only used to simplify the definition of the efficiency axiom (see Section 2.3 to not require the $f(0)$ term.

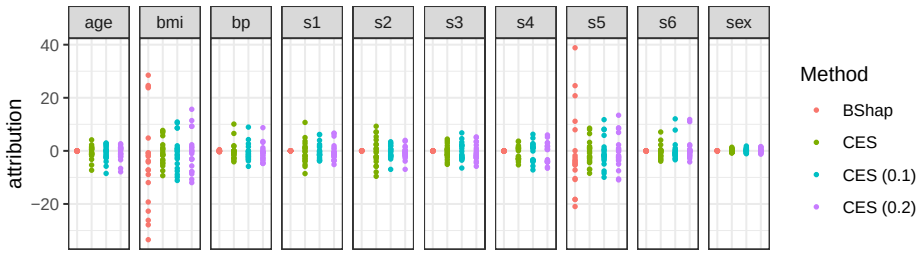


*Figure 1.* Attribution distribution across 20 explicands for four methods, BShap, CES, CES (smoothing 0.1), CES (smoothing 0.2).

represents a discrete change of the feature value of $\frac{x_i - x'_i}{m}$. And then we apply the Shapley value on these $N * m$ features. Notice that this is equivalent to creating a grid within the hypercube, and averaging over random, monotone walks from the baseline $x'$ to the explicand $x$ in this grid. As $m$ increases, the density of the random walks converges to the diagonal of the hypercube, and if the function $f$ is smooth, then running Shapley on these micro-features is equivalent to running IG on the original features.

Of course, in general, IG and BShap use different paths and hence give different attributions (see Example 4.6).

The standard axiomatization of the Shapley value (Shapley, 1953) only references (binary variants of) the first four axioms (Dummy, Efficiency, Linearity and Symmetry). However, in the continuous setting there are infinitely many methods that satisfy these four axioms. A uniqueness result requires further axioms, as we study in the next section.

### 4.2. Axiomatizing BShap (and IG)

In this section, we provide axiomatizations for BShap and IG by formally reducing model explanation to cost-sharing:

**Theorem 4.1** (Reducing Model Explanation to Cost-Sharing)**.** *Suppose there is an attribution method that satisfies Linearity and ASI. Then for every attribution problem with explicand $x$, baseline $x'$ and function $f$ (satisfying the minor technical condition that the derivatives are bounded) , then there exist two cost-sharing problems such that the resulting attributions for the attribution problem are the difference between cost-shares for the cost-sharing problems.*

*Proof.* Given an attribution problem $f, x, x'$, we progressively transform it into equivalent cost-sharing problems.

First, we transform the problem $f, x', x$ into a problem $f^n, x'^n, x^n$ such that the baseline $x'^n$ is the zero vector, and $x^n$ is non-negative. The proof is inductive. The base case is by definition: we define $f^0, x^0, x'^0$ to be $f, x, x'$. In step $i$ we transform $f^{i-1}, x^{i-1}, x'^{i-1}$ into $f^i, x^i, x'^i$ by transforming along feature $i$ using the transformation in the definition of ASI (see Section 2.3) using the $c = 1$ if $x_i$ is non-negative and $c = -1$ otherwise, and $d = -x'_i * c$. The proof for the inductive step: By ASI, the attribution for $f^{i-1}, x^{i-1}, x'^{i-1}$ should equal to that for $f^i, x^i, x'^i$, and we have set the baseline value for this feature to 0, and its explicand value is non-negative.

Next we express the function $f^n$, as the difference of two non-decreasing functions $f_1$ and $f_2$. Let $p$ denote the infimum of the partial derivative $\frac{\partial f^n(x)}{\partial x_i}$, where $x$ ranges over the domain of the function $f^n$, and $i$ ranges over all the variables. By the technical conditions, this infimum exists. If $p$ is zero or positive, then $f^n$ is itself non-decreasing—therefore set $f_1$ to $f^n$ and $f_2$ to the constant zero function. Otherwise, define $f_2$ to be the linear function $\sum_i -p * x_i$; notice that $-p$ is positive and so the function is non-decreasing. Set $f_1 = f^n + f_2$; by definition of $p$, $f_1$ is non-decreasing. By Linearity, the attributions for $f^n, x^n, x'^n$, (which we have already shown are equal to the attributions for $f, x, x'$) is the difference between the attributions of $f_1, x^n, x'^n$ and $f_2, x^n, x'^n$.

To complete the proof, notice that both $f_1, x^n, x'^n$ and $f_2, x^n, x'^n$ are cost-sharing problems. □

Both IG and BShap satisfy ASI and Linearity. Therefore **any** axiomatization that applies to Aumann-Shapley applies to IG and any axiomatization that applies to Shapley-Shubik applies to BShap. For instance, Corollary 1 from (Friedman & Moulin, 1999) reads:

**Theorem 4.2.** *Shapley-Shubik is the unique method that satisfies the Efficiency, Linearity, Dummy, Affine Scale Invariance (ASI), Demand Monotonicity (DM) and Symmetry (plus some technical conditions we exclude for clarity) for all cost-sharing problems.*

Therefore we have:

**Corollary 4.3.** *BShap is the unique method that satisfies the Linearity, Dummy, Affine Scale Invariance (ASI), Demand Monotonicity (DM), and Symmetry (plus minor technical conditions) for all attribution problems. (See Proof in Section 5.)*

Analogously, we have the following corollary of Theorem 3

from (Friedman & Moulin, 1999):

**Corollary 4.4.** *IG is the unique method that satisfies the Linearity, Dummy, Affine Scale Invariance (ASI), Proportionality, and Symmetry (plus minor technical conditions) for all attribution problems.*

**Remark 4.5** (Interpreting the Uniqueness Results)**.** *How should we interpret the axiomatic results? First, notice that we have two results Corollary 4.3 and 4.4. How do we use this to decide which of BShap or IG is superior? They differ in two axioms, Demand Monotonicity and Proportionality. If one property was clearly more desirable than the other, we would have a definitive answer. Unfortunately, this is not the case. Despite this, the axiomatic results adds to our understanding of the qualitative difference between the two methods. Remarks 4.6 and Remark 4.7 elaborate qualitative differences between the two approaches.*

*Second, both uniqueness results (Corollary 4.3 and 4.4) assume a fixed baseline $x'$. The attributions will still vary by choice of baseline and we only have uniqueness **up to** the choice of baseline. Remark 4.9 discusses practical implications of this.*

**Remark 4.6** (BShap versus IG)**.** *A simple example where the IG and BShap differ is the min of two variables $x_1$ and $x_2$. Suppose the baseline is $x'_1 = x'_2 = 0$, and the explicand is $x_1 = 5, x_2 = 1$. IG attributes the entire change in the function of $1$ to the arg min, i.e., $x_2$; this is intuitively reasonable if you see $x_2$ as the critical variable. BShap on the other hand assigns attributions of $2.5$ to the first variable and $-1.5$ to the second; this is intuitively reasonable if you think of $x_1$ as trying to increase the min and $x_2$ as trying to decrease it. It is not immediately clear which interpretation is obviously superior.*

*Let us now consider the cube of the sum of two variables $x_1$ and $x_2$. Suppose the baseline is $x'_1 = x'_2 = 0$, and the explicand is $x_1 = 5, x_2 = 1$. IG attributes $180$ to $x_1$ and $36$ to $x_2$, i.e., the attributions are in the ratio $5:1$, a consequence of the proportionality axiom. BShap attributes $170$ to $x_1$ and $46$ to $x_2$. The IG results appear a bit more principled. But there is a stronger consequence of the proportionality axiom: This axiom forces a smooth interpolation between the baseline and the explicand, i.e, the form of IG. For instance, a baseline of an entirely black image used in computer vision tasks results in intermediate inputs that are variants of the explicand with different intensities. In contrast, BShap is likely to construct more unrealistic inputs; in the vision examples, mixes of black pixels with the explicand pixels.*

**Remark 4.7** (Applicability of IG)**.** *IG requires that the score of the model to be differentiable with respect to the features. This is true for a deep learning models for computer vision that only use continuous features (pixels of the image). If the deep learning model uses discrete features, like words for a text model, or categorical variables, then IG can be applied by working with the embedding representation of the features. However, IG does not apply to models that are discontinuous and non-differentiable, for instance tree-ensembles. In this sense, BShap is more widely applicable.*

### 4.3. BShap fits in the CES framework

Thus far, we have studied the source of the difference between IG and BShap, both in terms of computation (they take different paths through a feature grid) and in terms of axioms (Corollary 4.3 versus Corollary 4.4). We now study the difference between BShap and CES.

First, we show that one can **posit** a distribution $D$ such that the BShap approach coincides with CES under the distribution $D$[4]. This shows that BShap fits in the framework of CES. In this sense, any difference in the properties of BS and CES($\hat{D}$) can be isolated to the choice of distribution on which to run CES.

**Lemma 4.8.** *For any explicand $x$ and baseline $x'$, there exists a feature distribution such that CES over that distribution results in attributions that are arbitrarily close to those produced by BShap. (See Proof in Section 5.)*

**Remark 4.9** (Explicit Comparison)**.** *Unlike CES($\hat{D}$), BShap does not depend on any distribution but requires an additional input (the baseline). We can use the baseline to model the explanation context. For instance, consider a model that makes loan decisions. If the applicant has been denied a loan, it is likely more useful to produce an explanation that only attributes to features that are in the applicant's power to change (e.g. getting a high school diploma as opposed to reducing their age). Such an explanation is achievable by selecting a baseline that coincides with the explicand on immutable features. In this sense the baseline parameter is useful flexibility. It makes the attributions germane to the decision or actions of the person consuming the explanation. However, this does pose an additional cognitive load to select the baseline and interpret the dependence of the explanation on the baseline. There are also situations where there is no compelling choice of one baseline over another. In this situation, RBShap presents an alternative.*

### 4.4. CES and RBShap

In Section 3, we saw that CES violates axioms like Dummy and Linearity. In contrast, Theorem 4.3 shows that BS satisfies these axioms. But since Lemma 4.8 shows that BS is a variant of CES, one could ask if there are other variants of CES that also satisfy some of the axioms.

[4](Lundberg & Lee, 2017) shows that CES reduces to BShap if the baseline is the feature means, **if** the features are independently distributed **and** the model is linear. Our reduction applies to non-linear models and baselines other than the feature means.

We first note that several axioms (Dummy, Linearity, Demand Monotonicity) also apply to RBShap, because these are satisfied by BShap and preserved by averaging the attributions over several baselines. (Symmetry requires that the distribution $D$ be symmetric in features over which the function $f$ is symmetric.) We now note that CES over an independent distribution $D$ is equal to RBShap where the baselines are drawn from distribution $D$. Therefore, these axioms also carry over to CES **if** the feature distribution $D$ is independent.

**Lemma 4.10.** *If the distribution $D$ is an independent distribution over the features, then RBShap and CES coincide. (See Proof in Section 5.)*

**Remark 4.11.** *If the function $f$ is linear and the distribution is independent, then, BShap with the baseline vector that has each feature set to its average across the data set, has the same attributions as CES (a consequence of Equation 9-12 from (Lundberg & Lee, 2017)) and hence RBShap (due to Lemma 4.10).*

**Remark 4.12.** *(Datta et al., 2016) runs CES over a contrived feature distribution $\Pi(D)$ that is a product of the marginal distributions of the input feature distribution $D$. By Lemma 4.10, CES over $\Pi(D)$ is equivalent to RBShap over $\Pi(D)$. One could ask if RBShap on the contrived feature distribution $\Pi(D)$ is equivalent to RBShap on the original feature distribution $D$. The following example proves that this is false, by showing that the sum of the attributions in the two cases differ: Suppose we are given two binary features $x_1$ and $x_2$, function $f(x_1, x_2) = x_1 * x_2$ and a distribution $D$ such that $P(0,0) = P(1,1) = 0.5$. Suppose that the explicand is $x_1 = 1, x_2 = 1$. Under $D$, $E[f(x)] = 0.5$. Under $D'$, $E[f(x)] = 0.25$. Recall that RBShap attributions sum to $f(x) - E[f(x)]$; this completes our counterexample.*

**Remark 4.13.** *The examples in Section 3.3, show that if the distribution is not independent, CES can fail Linearity, Dummy and Demand Monotonicity. However, it always satisfies Affine Scale Invariance (we omit the easy but technical proof).*

## 5. Missing Proofs

### 5.1. Proof of Lemma 3.1

*Proof.* Consider an $x^t$ that belongs to $T_{S'}$. We note that it also belongs to $T_S$. This is because an example that agrees with the explicand on a feature set $S'$ also agrees with the explicand on every feature set $S$ that is a subset of $S'$. □

### 5.2. Proof of Lemma 4.8

*Proof.* We construct the feature distribution for CES as follows: the distribution for feature $i$ has two points in its support, $x_i$ and $x'_i$, where $\Pr(x_i) = \epsilon$ and $\Pr(x'_i) = 1 - \epsilon$. It suffices to show that the set functions input to the Shapley value for the two approaches have values that are arbirarily close when $\epsilon \to 0$. The set function for CES is:

$$v(S) = \frac{\sum_{S' \subseteq N \setminus S} f(x_{S' \cup S}; x'_{N \setminus (S' \cup S)}) \epsilon^{|S'|} * (1-\epsilon)^{|N \setminus (S' \cup S)|}}{\sum_{S' \subseteq N \setminus S} \epsilon^{|S'|} * (1-\epsilon)^{|N \setminus (S' \cup S)|}} \quad (6)$$

Because $\epsilon \to 0$, the numerator is dominated by the term where $S'$ is empty, and the numerator tends to $f(x_S; x'_{N \setminus S}) * (1-\epsilon)^{|N \setminus S|}$. By an analogous argument, the denominator tends to $(1-\epsilon)^{|N \setminus S|}$. Dividing, we get the set function for BShap. □

### 5.3. Proof of Lemma 4.10

*Proof.* In the special case that the features follow an independent distribution we show that the set function $v(S)$ for RBShap and CES are the same for all sets $S$. Fix a set $S$ and consider $v(S)$ for RBShap:

$$\begin{aligned} v(S) &= E_{x' \sim D} f(x_S; x'_{N \setminus S}) && (7) \\ &= E_{x'_{N \setminus S}} f(x_S; x'_{N \setminus S}) && (8) \\ &= E_{x'_{N \setminus S}} [f(x_S; x'_{N \setminus S}) | x'_S = x_S] && (9) \\ &= E[f(x) | x_S] && (10) \end{aligned}$$

where 8 follows because the expression is dummy in $x'_S$; 9 is due to feature independence; and the final expression is the set function for CES. □

### 5.4. Proof of Corollary 4.3

*Proof.* It is easy to show that BShap satisfies all the axioms (we skip this part). Now consider the reverse direction, i.e., we would like to show that no other method satisfies these axioms. By Theorem 4.1, and because the attribution method satisfies Linearity and ASI, the attributions for an attribution problem are the difference between the cost-shares for two cost-sharing problems, and these cost-shares are uniquely determined by Theorem 4.2. □

## 6. A Concluding Remark

Estimating feature importance involves what-if analysis. One type of what-if analysis (e.g. BShap) performs interventions on the feature, while another (e.g. CES) marginalizes the feature over the training data. The former may construct out-of-distribution inputs, but regularization can ensure reasonable model behavior on these inputs. In contrast, the latter provides counter-intuitive explanations when the training data is sparse. The remedy to this common occurrence requires modeling the true feature distribution, a problem harder than original prediction problem.

## Acknowledgements

We thank Ankur Taly, Frederick Liu, Kedar Dhamdhere, and Scott Lundberg for helpful discussions, and the anonymous reviewers for feedback.

## References

Aas, K., Jullum, M., and Løland, A. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. **arXiv e-prints**, art. arXiv:1903.10464, Mar 2019.

Agarwal, A., Dhamdhere, K., and Sundararajan, M. A new interaction index inspired by the taylor series. **CoRR**, abs/1902.05622, 2019. URL http://arxiv.org/abs/1902.05622.

Agrawal, R. and Srikant, R. Fast algorithms for mining association rules. In **Proc. of 20th Intl. Conf. on VLDB**, pp. 487–499, 1994.

Aumann, R. J. and Shapley, L. S. **Values of Non-Atomic Games**. Princeton University Press, Princeton, NJ, 1974.

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, Los Alamitos, CA, USA, may 2016. IEEE Computer Society. doi: 10.1109/SP.2016.42. URL https://doi.ieeecomputersociety.org/10.1109/SP.2016.42.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. **Annals of Statistics**, 32:407–499, 2004.

Friedman, E. and Moulin, H. Three methods to share joint costs or surplus. **Journal of Economic Theory**, 87(2):275 – 312, 1999. ISSN 0022-0531. doi: https://doi.org/10.1006/jeth.1999.2534. URL http://www.sciencedirect.com/science/article/pii/S0022053199925346.

Grömping, U. Estimators of relative importance in linear regression based on variance decomposition. **The American Statistician**, 61(2):139–147, 2007. doi: 10.1198/000313007X188252. URL https://doi.org/10.1198/000313007X188252.

Janzing, D., Minorics, L., and Bloebaum, P. Feature relevance quantification in explainable ai: A causal problem. In Chiappa, S. and Calandra, R. (eds.), **Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics**, volume 108 of **Proceedings of Machine Learning Research**, pp. 2907–2916, Online, 26–28 Aug 2020. PMLR. URL http://proceedings.mlr.press/v108/janzing20a.html.

Lindeman, R., Merenda, P., and Gold, R. **Introduction to Bivariate and Multivariate Analysis**. Scott, Foresman, 1980. ISBN 9780673150998. URL https://books.google.com/books?id=-hfvAAAAMAAJ.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions. In **NIPS**, 2017.

Lundberg, S. M., Erion, G. G., and Lee, S. Consistent individualized feature attribution for tree ensembles. **CoRR**, abs/1802.03888, 2018. URL http://arxiv.org/abs/1802.03888.

Owen, A. Sobol' indices and shapley value. **SIAM/ASA Journal on Uncertainty Quantification**, 2(1):245–251, 2014. doi: 10.1137/130936233. URL https://doi.org/10.1137/130936233.

Owen, A. and Prieur, C. On shapley value for measuring importance of dependent inputs. **SIAM/ASA Journal on Uncertainty Quantification**, 5(1):986–1002, 2017. doi: 10.1137/16M1097717. URL https://doi.org/10.1137/16M1097717.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, 12:2825–2830, 2011.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should I trust you?": Explaining the predictions of any classifier. **CoRR**, abs/1602.04938, 2016. URL http://arxiv.org/abs/1602.04938.

Shapley, L. S. A value of n-person games. **Contributions to the Theory of Games**, pp. 307–317, 1953.

Štrumbelj, E., Kononenko, I., and Šikonja, M. R. Explaining instance classifications with interactions of subsets of feature values. **Data & Knowledge Engineering**, 68 (10):886–904, 2009.

Sun, Y. and Sundararajan, M. Axiomatic attribution for multilinear functions. **CoRR**, abs/1102.0989, 2011. URL http://arxiv.org/abs/1102.0989.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W. (eds.), **Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017**, volume 70 of **Proceedings of Machine Learning Research**, pp. 3319–3328. PMLR, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

Tibshirani, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, 58(1):267–288, 1996.

Štrumbelj, E. and Kononenko, I. Explaining prediction models and individual predictions with feature contributions. **Knowl. Inf. Syst.**, 41(3):647–665, December 2014. ISSN 0219-1377. doi: 10.1007/s10115-013-0679-x. URL http://dx.doi.org/10.1007/s10115-013-0679-x.