Shapley Values for XAI

the good, the bad and the ugly

- Game Theory Preliminaries
- Shapley Values
- Shapley Values for Feature Importance
- Issues with Shapley Values for explanation
- Possible Solutions

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

$$\Gamma := (N, v) \tag{2}$$

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

$$\Gamma := (N, v) \tag{3}$$

Where:

- $N = \{1, 2, \ldots, n\}$ represents the finite set of *players*. It is also called the **grand coalition**.

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

$$\Gamma := (N, v) \tag{4}$$

Where:

- $N = \{1, 2, \ldots, n\}$ represents the finite set of *players*. It is also called the **grand coalition**.
- $v \colon 2^n \to \mathbb{R}$ is the characteristic function of the game and it satifies:

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

$$\Gamma := (N, v) \tag{5}$$

Where:

- $N = \{1, 2, \ldots, n\}$ represents the finite set of *players*. It is also called the **grand coalition**.
- $v: 2^n \to \mathbb{R}$ is the characteristic function of the game and it satifies:
  - $v(N) \geqslant \Sigma_{i \epsilon N} v(\{i\})$
  - $v(\emptyset) = 0$

Definitions

In **cooperative game theory**, we denote an n-person **coalitional game** game by:

$$\Gamma := (N, v) \tag{6}$$

Where:

- $N = \{1, 2, \ldots, n\}$ represents the finite set of *players*. It is also called the **grand coalition**.
- $v\colon 2^n \rightarrow \mathbb{R}$ is the characteristic function of the game and it satifies:
  - $v(N) \geqslant \Sigma_{i \epsilon N} v(\{i\})$
  - $v(\emptyset) = 0$

The value function represents how much collective payoff a set of players can gain by "cooperating" as a set.

Definitions

Given a game $\Gamma := (N, v)$, a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of real numbers, and a **coalition** $S \subseteq N$, we define:

Definitions

Given a game $\Gamma := (N, v)$, a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of real numbers, and a **coalition** $S \subseteq N$, we define:

$$\boldsymbol{x}(S) = \begin{cases} \sum_{i \in S} x_i & \text{if } S \neq \emptyset \\ 0 & \text{if } S = \emptyset \end{cases} \tag{8}$$

Definitions

Given a game $\Gamma := (N, v)$, a vector $\boldsymbol{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ of real numbers, and a **coalition** $S \subseteq N$, we define:

$$\boldsymbol{x}(S) = \begin{cases} \sum_{i \in S} x_i & \text{if } S \neq \emptyset \\ 0 & \text{if } S = \emptyset \end{cases} \tag{9}$$

Where:

- x_i represents the payoff of player i
- $\boldsymbol{x}(S)$ represents the payoff of coalition S

Definitions

Given a game $\Gamma := (N, v)$, a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ of real numbers, and a **coalition** $S \subseteq N$, we define:

$$\boldsymbol{x}(S) = \begin{cases} \sum_{i \in S} x_i & \text{if } S \neq \emptyset \\ 0 & \text{if } S = \emptyset \end{cases} \tag{10}$$

Where:

- x_i represents the payoff of player i
- $\boldsymbol{x}(S)$ represents the payoff of coalition S

$\boldsymbol{x}$ is called the **payoff vector** (**payoff** for short).

Definitions

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **efficient** if:

Definitions

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **efficient** if:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N)$$

Definitions

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **efficient** if:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N)$$

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **individually rational** if:

Definitions

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **efficient** if:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N)$$

- We say that the payoff $\boldsymbol{x} \in \mathbb{R}^n$ is **individually rational** if:

$$\forall i \in N, x_i \geqslant v(\{i\})$$

Example

Consider a group of n miners, who have discovered large bars of gold[1]:

[1] This example was taken from Wikipedia : https://en.wikipedia.org/wiki/Core_(game_theory)#Example_1:_Miners

Example

Consider a group of n miners, who have discovered large bars of gold[2]:

- The miners can only do a single trip to fetch the gold.

[2] This example was taken from Wikipedia : https://en.wikipedia.org/wiki/Core_(game_theory)#Example_1:_Miners

Example

Consider a group of n miners, who have discovered large bars of gold[3]:

- The miners can only do a single trip to fetch the gold.
- Two miners can carry one piece of gold together.

[3] This example was taken from Wikipedia : https://en.wikipedia.org/wiki/Core_(game_theory)#Example_1:_Miners

Example

Consider a group of n miners, who have discovered large bars of gold[4]:

- The miners can only do a single trip to fetch the gold.
- Two miners can carry one piece of gold together.
- The value function for coalition S is given by:

$$v(S) = \begin{cases} \frac{|S|}{2}, & \text{if } S \text{ is even.} \\ \frac{(|S|-1)}{2}, & \text{if } S \text{ is odd.} \end{cases}$$

[4] This example was taken from Wikipedia : https://en.wikipedia.org/wiki/Core_(game_theory)#Example_1:_Miners

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1\}) = 0$
- $v(\{1, 2\}) = 1$

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1\}) = 0$
- $v(\{1, 2\}) = 1$

One way to distribute the payoff in this situation is to simply split it evenly amongst the miners, i.e. $\boldsymbol{x} = \left(\frac{1}{2}, \frac{1}{2}\right)$.

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1\}) = 0$
- $v(\{1, 2\}) = 1$

One way to distribute the payoff in this situation is to simply split it evenly amongst the miners, i.e. $\boldsymbol{x} = \left(\frac{1}{2}, \frac{1}{2}\right)$.

If there were 3 miners instead, $S = \{1, 2, 3\}$:

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1\}) = 0$
- $v(\{1, 2\}) = 1$

One way to distribute the payoff in this situation is to simply split it evenly amongst the miners, i.e. $\boldsymbol{x} = (\frac{1}{2}, \frac{1}{2})$.

If there were 3 miners instead, $S = \{1, 2, 3\}$:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1, 2\}) = 1$
- $v(\{2, 3\}) = 1$
- $v(\{1\}) = 0$
- $v(\{3\}) = 0$
- $v(\{1, 3\}) = 1$
- $v(\{1, 2, 3\}) = 1$

Example

If there are 2 miners, i.e. $S = \{1, 2\}$, then the values of the different coalitions are:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1\}) = 0$
- $v(\{1, 2\}) = 1$

One way to distribute the payoff in this situation is to simply split it evenly amongst the miners, i.e. $\boldsymbol{x} = \left(\frac{1}{2}, \frac{1}{2}\right)$.

If there were 3 miners instead, $S = \{1, 2, 3\}$:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1, 2\}) = 1$
- $v(\{2, 3\}) = 1$
- $v(\{1\}) = 0$
- $v(\{3\}) = 0$
- $v(\{1, 3\}) = 1$
- $v(\{1, 2, 3\}) = 1$

How should we distribute the payoff?

Inessential Games

- A coalitional game is said to be **inessential** if:

$$\sum_{i=1}^{n} v(\{i\}) = v(N) \tag{11}$$

Inessential Games

- A coalitional game is said to be **inessential** if:

$$\sum_{i=1}^{n} v(\{i\}) = v(N) \tag{13}$$

- and **essential** if:

$$\sum_{i=1}^{n} v(\{i\}) < v(N) \tag{14}$$

Inessential Games

- A coalitional game is said to be **inessential** if:

$$\sum_{i=1}^{n} v(\{i\}) = v(N) \tag{15}$$

- and **essential** if:

$$\sum_{i=1}^{n} v(\{i\}) < v(N) \tag{16}$$

- From a game-theoretic viewpoint, inessential games are very simple. There is no tendency for the players to form coalitions.
- So the unique possible payoff is $x = (v(\{1\}), \ldots, v(\{n\}))$.

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \tag{19}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \tag{20}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \tag{21}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{22}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{23}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{24}$$

which can be interpreted as:

$$\varphi_i(v) =$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{25}$$

which can be interpreted as:

$$\varphi_i(v) = \frac{1}{\text{number of players}}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{26}$$

which can be interpreted as:

$$\varphi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } \boldsymbol{i}}$$

Definitions

Shapley values[7] are a solution concept in cooperative game theory that attempts to faily distribute payoffs. It is defined as follows:

$$\varphi_i(v) = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) \tag{27}$$

which can be interpreted as:

$$\varphi_i(v) = \frac{1}{\text{number of players}} \sum_{\text{coalitions excluding } \boldsymbol{i}} \frac{\text{marginal contribution of } \boldsymbol{i} \text{ to coalition}}{\text{number of coalitions excluding } \boldsymbol{i} \text{ of this size}}$$

Properties

Shapley values are the unique solution concept that satisfies the following 4 properties:

Properties

Shapley values are the unique solution concept that satisfies the following 4 properties:

- **Efficiency**: The payoff vector exactly splits the total value:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N) \tag{30}$$

Properties

Shapley values are the unique solution concept that satisfies the following 4 properties:

- **Efficiency**: The payoff vector exactly splits the total value:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N) \tag{32}$$

- **Symmetry**: If players i and j are equivalent in the sense that:

$$\forall S \subseteq N \setminus \{i, j\}, \quad v(S \cup \{i\}) = v(S \cup \{j\})$$

Properties

Shapley values are the unique solution concept that satisfies the following 4 properties:

- **Efficiency**: The payoff vector exactly splits the total value:

$$\boldsymbol{x}(N) = \sum_{i \in N} x_i = v(N) \tag{34}$$

- **Symmetry**: If players i and j are equivalent in the sense that:

$$\forall S \subseteq N \setminus \{i, j\}, \quad v(S \cup \{i\}) = v(S \cup \{j\})$$

  then:

$$\varphi_i(v) = \varphi_j(v) \tag{35}$$

Properties

- **Additivity**: For two coalitional games v and w:

$$\forall i \in N, \quad \varphi_i(v+w) = \varphi_i(v) + \varphi_i(w) \tag{36}$$

Properties

- **Additivity**: For two coalitional games v and w:

$$\forall i \in N, \quad \varphi_i(v+w) = \varphi_i(v) + \varphi_i(w) \tag{38}$$

- **Null Player**: For a single player i, if:

$$\forall S \subseteq N \setminus \{i\}, \quad v(S \cup \{i\}) = v(S)$$

Properties

- **Additivity**: For two coalitional games v and w:

$$\forall i \in N, \quad \varphi_i(v+w) = \varphi_i(v) + \varphi_i(w) \tag{40}$$

- **Null Player**: For a single player i, if:

$$\forall S \subseteq N \setminus \{i\}, \quad v(S \cup \{i\}) = v(S)$$

then:

$$\varphi_i(v) = 0 \tag{41}$$

Example (Continued)

Let's go back to our example with 3 miners, $S = \{1, 2, 3\}$:

Example (Continued)

Let's go back to our example with 3 miners, $S = \{1, 2, 3\}$:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1, 2\}) = 1$
- $v(\{2, 3\}) = 1$
- $v(\{1\}) = 0$
- $v(\{3\}) = 0$
- $v(\{1, 3\}) = 1$
- $v(\{1, 2, 3\}) = 1$

Example (Continued)

Let's go back to our example with 3 miners, $S = \{1, 2, 3\}$:

- $v(\emptyset) = 0$
- $v(\{2\}) = 0$
- $v(\{1, 2\}) = 1$
- $v(\{2, 3\}) = 1$
- $v(\{1\}) = 0$
- $v(\{3\}) = 0$
- $v(\{1, 3\}) = 1$
- $v(\{1, 2, 3\}) = 1$

The Shapley values are:

- $\varphi_1 = \frac{1}{3} \sum_{S \subseteq \{2,3\}} \binom{2}{|S|}^{-1} (v(S \cup \{1\}) - v(S)) = \frac{1}{3}$
- $\varphi_2 = \frac{1}{3}$
- $\varphi_3 = \frac{1}{3}$

Feature Importance as a Game

Given a model $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$ with features $\boldsymbol{x} = (x_1, x_{2,} \ldots, x_n)$, we can define a coalitional game for feature importance as follows:

Feature Importance as a Game

Given a model $f\colon \mathbb{R}^n \to \mathbb{R}$ with features $\boldsymbol{x} = (x_1, x_{2,} \ldots, x_n)$, we can define a coalitional game for feature importance as follows:

- The players are the features $N = \{1, 2, \ldots, n\}$
- The value function v is defined as some measure of the importance or influence of a subset of features on the model's predictions.

Feature Importance as a Game

Given a model $f\colon \mathbb{R}^n \to \mathbb{R}$ with features $\boldsymbol{x} = (x_1, x_{2,} \dots, x_n)$, we can define a coalitional game for feature importance as follows:

- The players are the features $N = \{1, 2, \dots, n\}$
- The value function v is defined as some measure of the importance or influence of a subset of features on the model's predictions.

For global feature importance, for a linear regression model we could define the value function to represent the R^2 of a linear model trained on a subset of features S[4].

Feature Importance as a Game

Given a model $f\colon \mathbb{R}^n \rightarrow \mathbb{R}$ with features $\boldsymbol{x} = (x_1, x_{2,} \ldots, x_n)$, we can define a coalitional game for feature importance as follows:

- The players are the features $N = \{1, 2, \ldots, n\}$
- The value function v is defined as some measure of the importance or influence of a subset of features on the model's predictions.

For global feature importance, for a linear regression model we could define the value function to represent the R^2 of a linear model trained on a subset of features S[4].

For local feature importance, many recently proposed define a value function that depends on a specific data instance x to explain how each feature contributes to the output of the function on this instance. The value of the grand coalition, in this setting, is the prediction of the model at x: $v_{f,x}(N) = f(x)$.

Feature Importance as a Game

Several methods have been proposed to apply the Shapley value to the problem of local feature importance. Each of which defines the value function differently and this also determines what happens to missing features i.e. features not in S:

Feature Importance as a Game

Several methods have been proposed to apply the Shapley value to the problem of local feature importance. Each of which defines the value function differently and this also determines what happens to missing features i.e. features not in S:

- It can be defined as the **conditional** expected model output on a data instance when only the features in S are known:

Feature Importance as a Game

Several methods have been proposed to apply the Shapley value to the problem of local feature importance. Each of which defines the value function differently and this also determines what happens to missing features i.e. features not in S:

- It can be defined as the **conditional** expected model output on a data instance when only the features in S are known:

$$v_{f,x}(S) = E_D[f(X)|X_S = x_s] \tag{46}$$

Feature Importance as a Game

Several methods have been proposed to apply the Shapley value to the problem of local feature importance. Each of which defines the value function differently and this also determines what happens to missing features i.e. features not in S:

- It can be defined as the **conditional** expected model output on a data instance when only the features in S are known:

$$v_{f,x}(S) = E_D[f(X)|X_S = x_s] \tag{48}$$

- It can also be defined as the **interventional** (**marginal**) expected model output on a data instance when features not in S are held fixed:

Feature Importance as a Game

Several methods have been proposed to apply the Shapley value to the problem of local feature importance. Each of which defines the value function differently and this also determines what happens to missing features i.e. features not in S:

- It can be defined as the **conditional** expected model output on a data instance when only the features in S are known:

$$v_{f,x}(S) = E_D[f(X)|X_S = x_s] \tag{50}$$

- It can also be defined as the **interventional** (**marginal**) expected model output on a data instance when features not in S are held fixed:

$$v_{f,x}(S) = E_D[f(x_S, X_{\bar{S}})] \tag{51}$$

Where $\bar{S}$ is the complement of S in N i.e. $S \cap \bar{S} = \emptyset$ and $S \cup \bar{S} = N$

SHAP

SHAP (SHapley Additive exPlanations)[5] is a unified approach for local feature importance.

It unifies six previous methods: LIME, DeepLift, Shapley regression values, Shapley sampling values, etc.

It uses the **conditional** expected model output on a data instance when only the features in S are known as value function:

$$v_{f,x}(S) = E[f(X)|X_S = x_s] \tag{52}$$

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

$$g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i x_i^{'} \tag{54}$$

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

$$g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i x_i^{'} \tag{55}$$

Where:

- $\varphi_i, i \in \{1, \ldots, M\}$ are the Shapley values and $\varphi_0 = E[f(X)]$

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

$$g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i x_i^{'} \tag{56}$$

Where:

- $\varphi_i, i \in \{1, \ldots, M\}$ are the Shapley values and $\varphi_0 = E[f(X)]$
- M is the total number of simplified input features.

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

$$g(x') = \varphi_0 + \sum_{i=1}^{M} \varphi_i x'_i \tag{57}$$

Where:

- $\varphi_i, i \in \{1, \ldots, M\}$ are the Shapley values and $\varphi_0 = E[f(X)]$
- M is the total number of simplified input features.
- $x' \in \{0,1\}^M$ is a boolean vector.

  A value of 1 in the simplified input features means that the corresponding feature value is "present" whereas a value of 0 means that it is "absent".

SHAP

It uses an additive explanation model g that is a linear function of **simplified input features**:

$$g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i x_i^{'} \tag{58}$$

Where:

- $\varphi_i, i \in \{1, \ldots, M\}$ are the Shapley values and $\varphi_0 = E[f(X)]$
- M is the total number of simplified input features.
- $x^{'} \in \{0,1\}^M$ is a boolean vector.

  A value of 1 in the simplified input features means that the corresponding feature value is "present" whereas a value of 0 means that it is "absent".

We denote by $h\colon \{0,1\}^M \rightarrow \mathbb{R}^n$ the mapping from simplified features to original features.

SHAP

SHAP

SHAP satisfies the following 3 properties:

SHAP

SHAP satisfies the following 3 properties:

- **Local Accuracy**: It is the same as efficiency:

$$f(x) = g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i \tag{61}$$

SHAP

SHAP satisfies the following 3 properties:

- **Local Accuracy**: It is the same as efficiency:

$$f(x) = g(x^{'}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i \tag{63}$$

- **Missingness**: If the simplified inputs represent feature presence, then missingness requires features missing in the original input to have no impact:

$$x_i^{'} = 0 \quad \Rightarrow \quad \varphi_i = 0 \tag{64}$$

In practice, this is only relevant for features that are constant.

SHAP

- **Consistency**: Consistency states that if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, that input's importance should increase or stay the same.

  Let $f_x(x^{'}) = f(h(x^{'}))$ and $x^{'}_{-i}$ denote setting $x^{'}_i = 0$. For any two models f and $f^{'}$, if:

  $$\forall x^{'} \in \{0,1\}^M, \quad f_x(x^{'}) - f_x(x^{'}_{-i}) \geqslant f^{'}_x(x^{'}) - f^{'}_x(x^{'}_{-i})$$

  then:

  $$\varphi_i(f, x) \geqslant \varphi_i(f^{'}, x) \tag{65}$$

Note

You may have noticed that, except for **local accuracy**, the properties listed for Shapley values are not the same as the ones listed for SHAP. That can be explained as follows:

Note

You may have noticed that, except for **local accuracy**, the properties listed for Shapley values are not the same as the ones listed for SHAP. That can be explained as follows:

- In 1985, Young[10] showed that the linearity and **null player** properties can be replaced by using a **monotonicity** property (Which is the same as the **consistency** property):

  For any two value functions v and w, if for all coalitions $S \subseteq N \setminus \{i\}$:

$$v(S \cup \{i\}) - v(S) \geqslant w(S \cup \{i\}) - w(S)$$

then:

$$\varphi_i(v) \geqslant \varphi_i(w)$$

Note

You may have noticed that, except for **local accuracy**, the properties listed for Shapley values are not the same as the ones listed for SHAP. That can be explained as follows:

- In 1985, Young[10] showed that the **linearity** and **null player** properties can be replaced by using a **monotonicity** property (Which is the same as the **consistency** property):

  For any two value functions v and w, if for all coalitions $S \subseteq N \setminus \{i\}$:

  $$v(S \cup \{i\}) - v(S) \geqslant w(S \cup \{i\}) - w(S)$$

  then:

  $$\varphi_i(v) \geqslant \varphi_i(w)$$

- In the supplementary material of the SHAP paper, the authors prove that the **symmetry** property is also implied by the **monotonicity** property.

Variants

The SHAP method has many variants, most of which are model-specific:

- Model-Agnostic Approximations:
  - KernelSHAP
  - Permutation SHAP
- Model-Specific Approximations:
  - Linear SHAP
  - Low-Order SHAP
  - Max SHAP
  - DeepSHAP
  - TreeSHAP

KernelSHAP

- The exact computation of SHAP values is challenging because it requires $O(2^n)$ evaluations of the value function. Therefore we need to make some simplifying assumptions in order to approximate it.

KernelSHAP

- The exact computation of SHAP values is challenging because it requires $O(2^n)$ evaluations of the value function. Therefore we need to make some simplifying assumptions in order to approximate it.
- KernalSHAP is a variant of SHAP that simplifies the computation by making 2 assumptions:
  - Feature independence
  - Model linearity

$$\begin{aligned} E[f(X)|X_S\!=\!x_S] &= E_{X_{\bar{S}}|X_S=x_S}[f(X)] \\ &\approx E_{X_{\bar{S}}}[f(X)] \\ &\approx f(x_S, E[X_{\bar{S}}]) \end{aligned}$$

KernelSHAP

- The exact computation of SHAP values is challenging because it requires $O(2^n)$ evaluations of the value function. Therefore we need to make some simplifying assumptions in order to approximate it.

- KernalSHAP is a variant of SHAP that simplifies the computation by making 2 assumptions:

  - Feature independence

  - Model linearity

$$\begin{aligned} E[f(X)|X_S = x_S] &= E_{X_{\bar{S}}|X_S = x_S}[f(X)] \\ &\approx E_{X_{\bar{S}}}[f(X)] \\ &\approx f(x_S, E[X_{\bar{S}}]) \end{aligned}$$

- This changes the conditional expectation into a marginal expectation.

KernelSHAP

It is based on Linear LIME[6] which is a method that fits a linear model around the data instance of interest by perturbing it and using a weighting kernel to give more weight to perturbed samples closer to it.

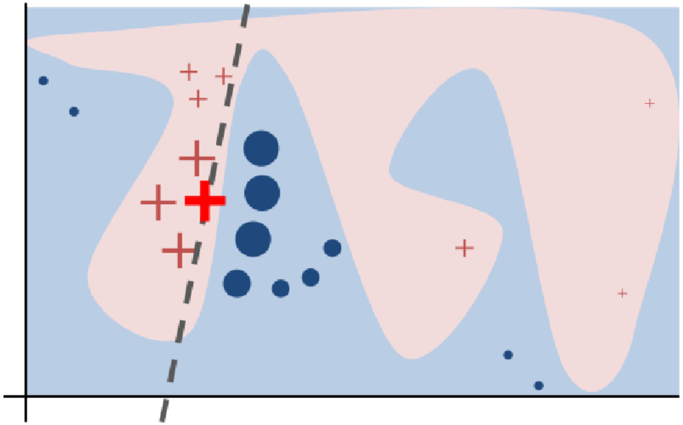


Figure 1. The black-box model's complex decision function (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using the model, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Conditional versus interventional distributions

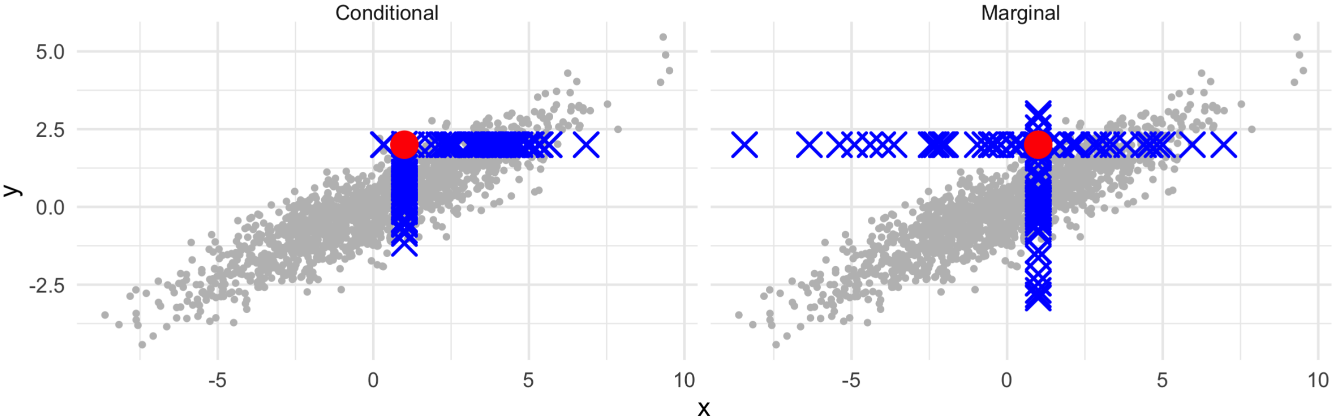


Figure 2. Samples that might be drawn to estimate $E[f(1,Y)]$ and $E[f(X,2)]$to explain $f(1,2)$ for some function f , given correlated Gaussian distributions for X and Y, depending on whether the expectation is taken over $X|Y=2$ and $Y|X=1$ (left) or X,Y (right)[2]

Issue with conditional distributions

- The exact computation of the Shapley value for a conditional value function would require knowledge of 2^n different multivariate distributions, and so a significant amount of approximation or modeling is necessary

Issue with conditional distributions

- The exact computation of the Shapley value for a conditional value function would require knowledge of 2^n different multivariate distributions, and so a significant amount of approximation or modeling is necessary
- Even if computational issues are resolved, there are additional inconsistencies introduced by the capacity of the Shapley value to attribute influence to an arbitrarily large feature set given a single function.

Issues with conditional distributions

- Consider the addition of a redundant variable x_3 to a dataset with two features, x_1 and x_2, so that $P(x_3 = x_2) = 1$.

Issues with conditional distributions

- Consider the addition of a redundant variable x_3 to a dataset with two features, x_1 and x_2, so that $P(x_3 = x_2) = 1$.
- Suppose a model $f(x_1, x_2, x_3)$ is trained on all three features. Intuitively, the features x_2 and x_3 should be equally informative to the model and so should have the same Shapley value under the conditional value function.

Issues with conditional distributions

- Consider the addition of a redundant variable x_3 to a dataset with two features, x_1 and x_2, so that $P(x_3 = x_2) = 1$.

- Suppose a model $f(x_1, x_2, x_3)$ is trained on all three features. Intuitively, the features x_2 and x_3 should be equally informative to the model and so should have the same Shapley value under the conditional value function.

- Now consider what would happen if we defined a new function $f'(x_1, x_2) = f(x_1, x_2, x_2)$t is effectively the same model for all in-distribution data points.

Issues with conditional distributions

- Consider the addition of a redundant variable x_3 to a dataset with two features, x_1 and x_2, so that $P(x_3 = x_2) = 1$.

- Suppose a model $f(x_1, x_2, x_3)$ is trained on all three features. Intuitively, the features x_2 and x_3 should be equally informative to the model and so should have the same Shapley value under the conditional value function.

- Now consider what would happen if we defined a new function $f^{'}(x_1, x_2) = f(x_1, x_2, x_2)$t is effectively the same model for all in-distribution data points.

- Yet if we choose to limit the scope of our explanation to two variables instead of three, the attribution for both x_1 and x_2 will come out to be different.

Issue with interventional distributions

- Methods which use an interventional value function fundamentally rely on evaluating a model on out-of-distribution samples.
- Consider, for example, a model trained on a data set with three features: x_1 and x_2, both $\mathcal{N}(0,1)$, and an engineered feature $x_3 = x_1 x_2$.
- To calculate $v_{f,x}(\{1,2\})$ for some $x = (x_1, x_2, x_3)$, we would have to estimate $E[f(x_1, x_2, X_3)]$ over some distribution for x_3 which does not depend on x_1 or x_2.
- Therefore we will almost certainly have to evaluate f on some sample $\{x_1, x_2, x_3^{'}\}$ which does not respect $x_3^{'} = x_1 x_2$ thus, it is well outside the domain of the actual data distribution.

Issues with the additivity property

- Consider for example applying it to a linear function with independent features:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

- The value function for the coalition $S = \{1\}$:

$$\begin{aligned} v_{f,x}(S) &= E_{X_{\bar{S}}|x_S}[f(x_S, X_{\bar{S}})] \\ &= E_{X_{\bar{S}}}[f(x_S, X_{\bar{S}})] \\ &= f(x_S, E[X_{\bar{S}}]) \\ &= \beta_0 + \beta_1 x_1 + \beta_2 E[X_2] \end{aligned}$$

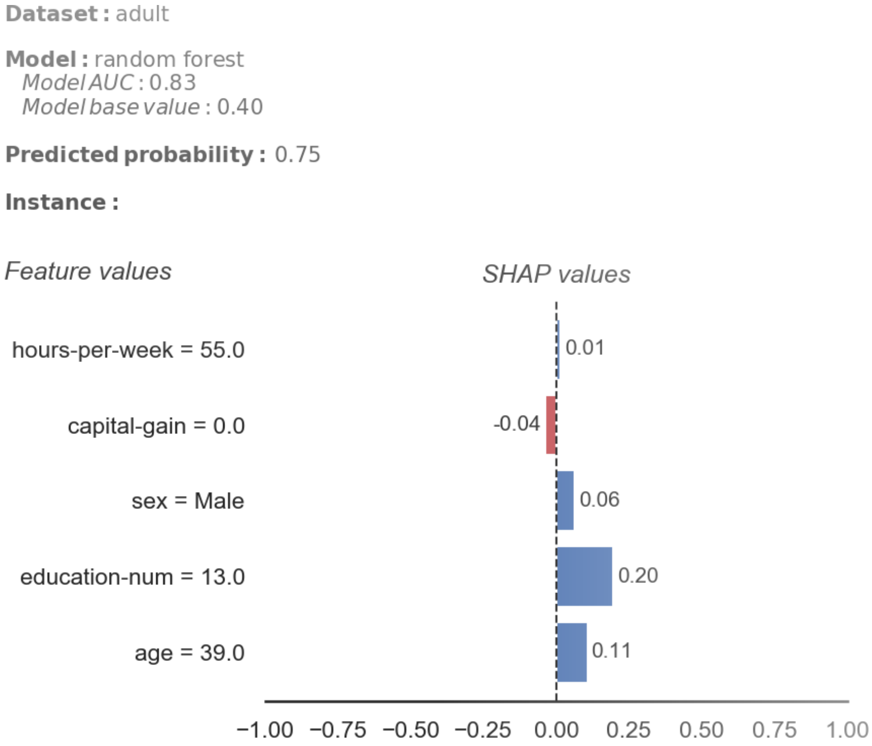
Issues with the additivity property

- The Shapley value intuitively aligns with what is considered important in an additive setting with independent features.
- What if instead the features were correlated or the function was non-linear?
- Then our intuition breaks and using methods with simplifying assumptions (e.g. KernelSHAP) could at best give us obviously wrong explanations and at worst a false sense of trust.

Human-centric issues

- Shapley value based feature attribution methods rely on mathematical correctness to justify their usefulness.
- Shapley value based methods do not explicitly attempt to provide guidance how a user might alter one's behavior in a desirable way. Further, observing that a certain feature carries a large influence over the model does not necessarily imply that changing that feature (even significantly) will change the outcome favorably.

Human-centric issues



Dataset : adult

Model : random forest
Model AUC : 0.83
Model base value : 0.40

Predicted probability : 0.75

Instance :

Figure 3. Example of an alert processing task in SHAP condition. In the NoSHAP condition, only the left part of the figure is shown[9].

Human-centric issues

- Weerts et al.[9] conducted a human-grounded evaluation to determine the utility of SHAP for domain experts who assess the correctness of predictions, such as in medical diagnosis and fraud detection.
- Real humans performed simplified alert processing tasks, with and without an explanation of the model's prediction.
- In contrast to common assumptions, they did not find a significant difference in alert processing performance between tasks for which a SHAP explanation was shown and tasks for which it was not shown.

Possible solutions

- **Shapley Residuals**[3]:

  Provides a method to compute a residual that act as a warning to practitioners against overestimating the degree to which Shapley-value-based explanations give them insight into a model.

- **Shapley on the Data Manifold**[1]:

  Provides two solutions to Shapley explainability that respect the data manifold.

  One solution, based on generative modelling, provides flexible access to data imputations; the other directly learns the Shapley value-function, providing performance and stability at the cost of flexibility.

Shapley residuals

- Shapley Residuals are vector-valued objects that capture a specific type of quantitative information lost by Shapley values.
- It is based on the concept of inessential games. It uses the degree to which a game is not inessential to provide insights into where Shapley values are not able to capture feature influence.
- When the residual of a feature exhibits a large norm, the associated Shapley value should be taken with skepticism: the resulting importance is not just due to the variable acting by itself.
- On the other hand, if a residual is small, most of the effect of the variable on the model is explainable by the variable acting independently.

Shapley residuals

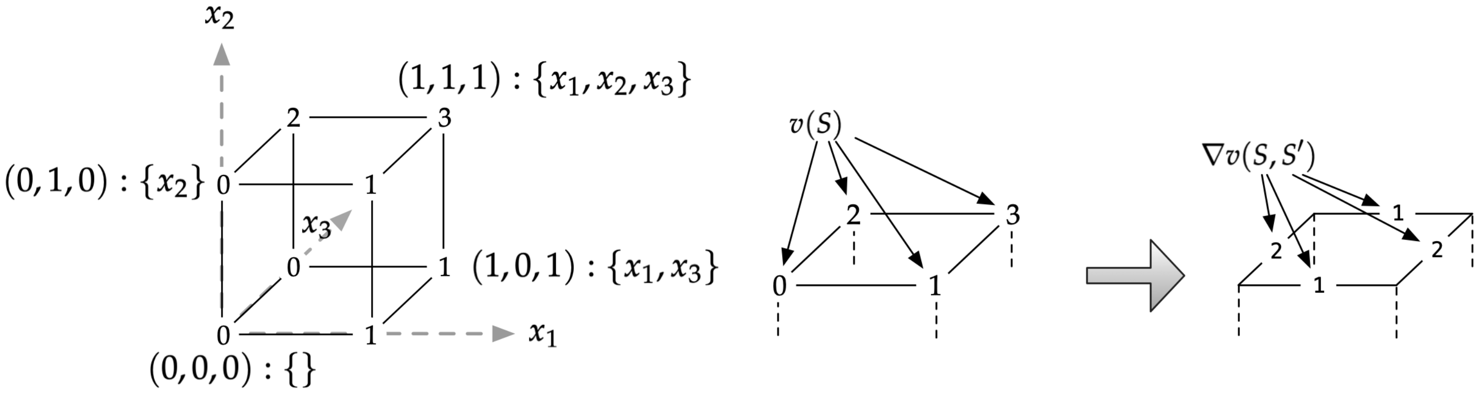
- Consider two models f_1 and f_2:

$$\begin{aligned} f_1(x_1, x_2, x_3) &= x_1 + x_2 + x_3 \\ f_2(x_1, x_2, x_3) &= x_1 + 2x_2x_3 \end{aligned}$$

- Suppose we use KernelSHAP to compute local feature importances for the output $f_1(1,1,1) = 3$ or $f_2(1,1,1) = 3$
- For both models, the Shapley values for the 3 features are all 1.
- Despite that, intervening by increasing the value of x_2 changes f_2 more than increasing the value of x_1; in f_1, this clearly does not happen.

Shapley residuals

- We start by visualizing the game as a function over the vertices of a n-dimensional hypercube
- Each coordinate corresponds to the presence or absence of a certain player, and each vertex corresponds to a subset of players.



(a) Graphical representation of v (b) Graphical representation of ∇v

Figure 4. Visualizing the game and gradient of the game corresponding to the example.

Shapley residuals

- We can think of the set N as the n-dimensional hypercube $G=(V=N,E)$ with each vertex labeled by a set S and edges between sets S and $S\cup\{i\}$.
- Let $\mathbb{R}^V$ be the space of functions from V to $\mathbb{R}$ and let $\mathbb{R}^E$ be the space of functions from E to $\mathbb{R}$. In particular, the game v is an element of $\mathbb{R}^V$.
- The differential operator $\nabla\colon \mathbb{R}^V \to \mathbb{R}^E$ is then defined as

$$\nabla v_i = \nabla v(S, S\cup\{i\}) = v(S\cup\{i\}) - v(S).$$

Essentially it is a discrete gradient operator on G, mapping functions on vertices to functions on edges.

Shapley residuals

- We will also define a partial gradient $\nabla_i \colon \mathbb{R}^V \to \mathbb{R}^E$:

$$\nabla_i v(S, S \cup \{j\}) = \begin{cases} v(S \cup \{j\}) - v(S), & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{66}$$

Intuitively, ∇_i evaluates a gradient for edges corresponding to the insertion of i, and takes the value 0 everywhere else. On the hypercube, only edges on the ith axis of $\nabla_i v$ will take a nonzero value.

Shapley residuals

- A game v is inessential if and only if for each i there exists $v_i \in \mathbb{R}^V$ such that $\nabla_i v = \nabla v_i$[8].

Shapley residuals

- A game v is inessential if and only if for each i there exists $v_i \in \mathbb{R}^V$ such that $\nabla_i v = \nabla v_i$[8].
- If v is not inessential, we cannot be sure to find v_i such that $\nabla_i v = \nabla v_i$, but we can find the "closest" one as the solution to the least squares problem:

$$\min_{x \in \mathbb{R}^V, x(\emptyset)=0} \|\nabla x - \nabla_i v\| \tag{68}$$

Shapley residuals

- A game v is inessential if and only if for each i there exists $v_i \in \mathbb{R}^V$ such that $\nabla_i v = \nabla v_i$[8].
- If v is not inessential, we cannot be sure to find v_i such that $\nabla_i v = \nabla v_i$, but we can find the "closest" one as the solution to the least squares problem:

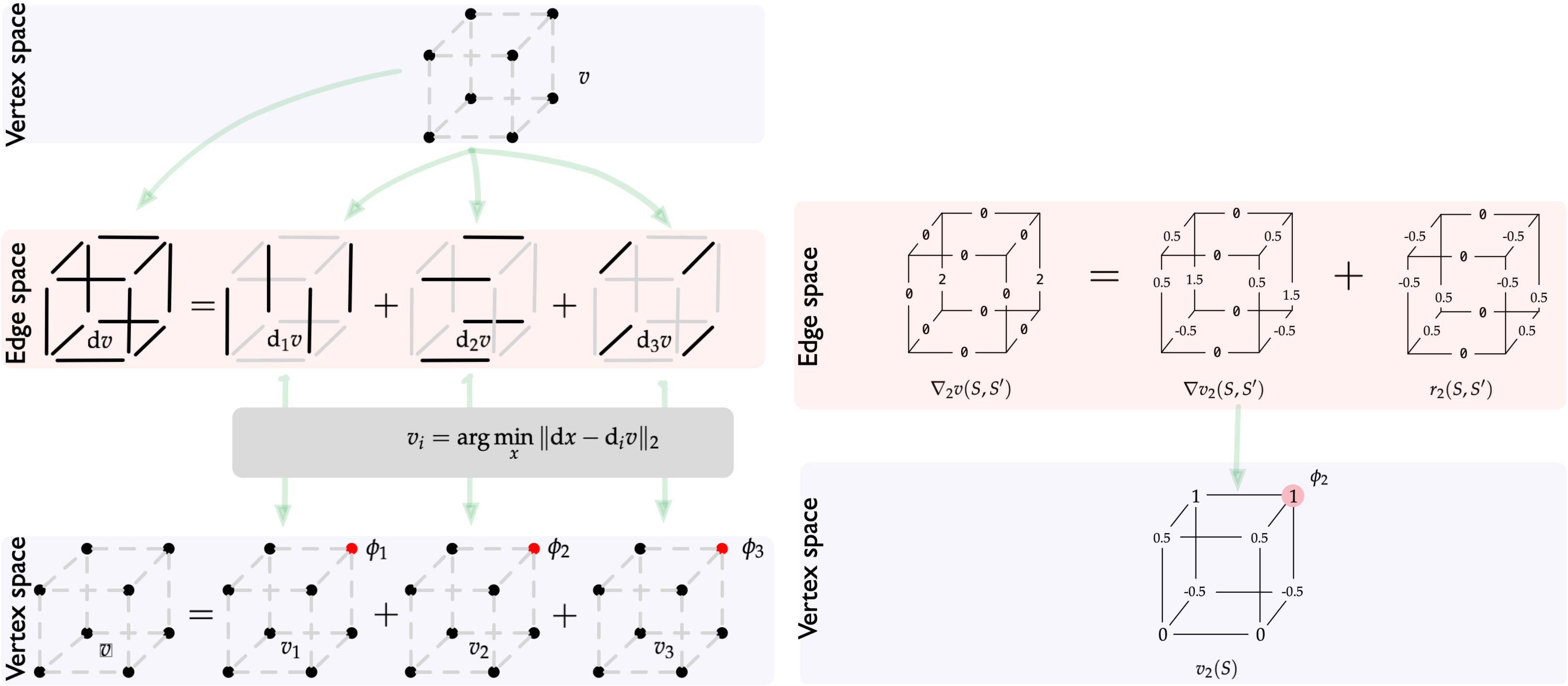
$$\min_{x \in \mathbb{R}^V, x(\emptyset)=0} \|\nabla x - \nabla_i v\| \tag{69}$$

- Given v_i, the game v can be decomposed as: $\sum v_i = v$
- And the shapley values are given by: $\varphi_i(v) = v_i([d])$

Shapley residuals

- We call $r_i = \nabla_i v - \nabla v_i$ the Shapley residual of player i.
- v is inessential iff $r_i = 0$ for each i
- The last statement allow us to interpert $\sum_{i \in N} \|r_i\|^2$ as the deviation from inessentiality of v.

Shapley residuals



(a) The decomposition of a game proposed by Stern and Tettenhorst

(b) The construction of Shapley residuals

Figure 5.

- The good:
  - Shapley values are a unique and fair method to distribute payoffs that satisfy certain properties.
  - SHAP provides a unified approach for local feature importances with desirable properties.
  - Shapley residuals provide a way to measure how much trust we should put into the provided feature importances.
- The bad and the ugly:
  - Tradeoff between computational complexity and correctness.
  - Conditional vs Interventional (Marginal) distributions.
  - Additivity property is only really needed for the uniqueness of the Shapley value.

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

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

[3] Indra Kumar, Carlos Scheidegger, Suresh Venkatasubramanian, and Sorelle Friedler. Shapley Residuals: Quantifying the limits of the Shapley value for explanations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 26598–26608. Curran Associates, Inc.

[4] Stan Lipovetsky and Michael Conklin. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.

[5] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

[6] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. 2016.

[7] Lloyd S. Shapley. A value for n-person games. 0:307–317.

[8] Ari Stern and Alexander Tettenhorst. Hodge decomposition and the shapley value of a cooperative game. *Games and Economic Behavior*, 113:186–198, 2019.

[9] Hilde JP Weerts, Werner van Ipenburg, and Mykola Pechenizkiy. A human-grounded evaluation of shap for alert processing. *ArXiv preprint arXiv:1907.03324*, 2019.

[10] H Peyton Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14(2):65–72, 1985.