The Shapley Value in Machine Learning

Citation for published version:

Digital Object Identifier (DOI):
10.24963/ijcai.2022/778

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 31st International Joint Conference on Artificial Intelligence, IJCAI-ECAI 2022

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
The Shapley Value in Machine Learning

Benedek Rozemberczki¹, Lauren Watson², Péter Bayer³, Hao-Tsung Yang², Oliver Kiss⁴, Sebastian Nilsson¹ and Rik Sarkar²

¹Research Data & Analytics, Research & Development IT, AstraZeneca
²The University of Edinburgh, School of Informatics
³Toulouse School of Economics & Institute for Advanced Study in Toulouse
⁴Central European University, Department of Economics and Business
benedek.rozemberczki@astrazeneca.com

Abstract

Over the last few years, the Shapley value, a solution concept from cooperative game theory, has found numerous applications in machine learning. In this paper, we first discuss fundamental concepts of cooperative game theory and axiomatic properties of the Shapley value. Then we give an overview of the most important applications of the Shapley value in machine learning: feature selection, explainability, multi-agent reinforcement learning, ensemble pruning, and data valuation. We examine the most crucial limitations of the Shapley value and point out directions for future research.

1 Introduction

Measuring importance and the attribution of various gains is a central problem in many practical aspects of machine learning such as explainability [Lundberg et al., 2017], feature selection [Cohen et al., 2007], data valuation [Ghorbani et al., 2019], ensemble pruning [Rozemberczki et al., 2021] and federated learning [Wang et al., 2020; Fan et al., 2021]. For example, one might ask: What is the importance of a certain feature in the decisions of a machine learning model? How much is an individual data point worth? Which models are the most valuable in an ensemble? These questions have been addressed in different domains using specific approaches. Interestingly, there is also a general and unified approach to these questions as a solution to a transferable utility (TU) cooperative game. In contrast with other approaches, solution concepts of TU games are theoretically motivated with axiomatic properties. The best known solution of this type is the Shapley value [Shapley, 1953] characterized by several desiderata that include fairness, symmetry, and efficiency [Chalkiadakis et al., 2011].

In the TU setting, a cooperative game consists of: a player set and a scalar-valued characteristic function that defines the value of coalitions (subsets of players). In such a game, the Shapley value offers a rigorous and intuitive way to distribute the collective value (e.g., the revenue, profit, or cost) of the team across individuals. To apply this idea to machine learning, we need to define two components: the player set and the characteristic function. In a machine learning setting players may be represented by a set of input features, reinforcement learning agents, data points, models in an ensemble, or data silos. The characteristic function can then describe the goodness of fit for a model, reward in reinforcement learning, financial gain on instance level predictions, or out-of-sample model performance. We provide an example about model valuation in an ensemble [Rozemberczki et al., 2021] in Figure 1.

![Figure 1: The Shapley value can be used to solve cooperative games.](image)

Figure 1: The Shapley value can be used to solve cooperative games. An ensemble game is a machine learning application for it – models in an ensemble are players (red, blue, and green robots) and the financial gain of the predictions is the payoff (coins) for each possible coalition (rectangles). The Shapley value can distribute the gain of the grand coalition (right bottom corner) among models.

Present work. We introduce basic definitions of cooperative games and present the Shapley value, a solution concept that can allocate gains in these games to individual players. We discuss its properties and emphasize why these are important in machine learning. We overview applications of the Shapley value in machine learning: feature selection, data valuation, explainability, reinforcement learning, and model valuation. Finally, we discuss the limitations of the Shapley value and point out future directions.

2 Background

This section introduces cooperative games and the Shapley value followed by its properties. We also provide an illustrative running example for our definitions.

2.1 Cooperative games and the Shapley value

Definition 1. Player set and coalitions. Let $\mathcal{N} = \{1, \ldots, n\}$ be the finite set of players. We call each non-empty subset $S \subseteq \mathcal{N}$ a coalition and $\mathcal{N}$ itself the grand coalition.

Definition 2. Cooperative game. A TU game is defined by the pair $(\mathcal{N}, v)$ where $v : 2^\mathcal{N} \rightarrow \mathbb{R}$ is a mapping called the
characteristic function or the coalition function of the game assigning a real number to each coalition and satisfying \( v(\emptyset) = 0 \).

**Example 1.** Let us consider a 3-player cooperative game where \( N = \{1, 2, 3\} \). The characteristic function defines the payoff for each coalition. Let these payoffs be given as:

\[
\begin{align*}
v(\emptyset) &= 0; & v(\{1\}) &= 7; & v(\{2\}) &= 11; & v(\{3\}) &= 14; \\
v(\{1, 2\}) &= 18; & v(\{1, 3\}) &= 21; & v(\{2, 3\}) &= 23; & v(\{1, 2, 3\}) &= 25.
\end{align*}
\]

**Definition 3.** Set of feasible payoff vectors. Let us define \( \mathcal{Z}(N, v) = \{ z \in \mathbb{R}^N \mid \sum_{i \in N} z_i \leq v(N) \} \) the set of feasible payoff vectors for the cooperative game \((N, v)\).

**Definition 4.** Solution concept and solution vector. Solution concept \( \Phi \) is a mapping associating a subset \( \Phi(N, v) \subseteq \mathcal{Z}(N, v) \) to every TU game \((N, v)\). A solution vector \( \phi(N, v) \in \mathbb{R}^N \) to the cooperative game \((N, v)\) satisfies solution concept \( \Phi \) if \( \phi(N, v) \in \Phi(N, v) \). Solution concept \( \Phi \) is single-valued if for every \((N, v)\) the set \( \Phi(N, v) \) is a singleton.

A solution concept defines an allocation principle through which rewards can be given to the individual players. The sum of these rewards cannot exceed the value of the grand coalition \( v(N) \). Solution vectors are specific allocations satisfying the principles of the solution concept.

**Definition 5.** Permutations of the player set. Let \( \Pi(N) \) be the set of all permutations defined on \( N \), a specific permutation is written as \( \pi \in \Pi(N) \) and \( \pi(i) \) is the position of player \( i \in N \) in permutation \( \pi \).

**Definition 6.** Predecessor set. Let the set of predecessors of player \( i \in N \) in permutation \( \pi \) be:

\[
P_\pi^i = \{ j \in N \mid \pi(j) < \pi(i) \}.
\]

Let us imagine that the permutation of the players in our illustrative game is \( \pi = (3, 2, 1) \). Under this permutation the predecessor set of the 1\textsuperscript{st} player is \( P_\pi^1 = \{3, 2\} \), that of the 2\textsuperscript{nd} player is \( P_\pi^2 = \{3\} \) and \( P_\pi^3 = \emptyset \).

**Definition 7.** Shapley value. The Shapley value [Shapley, 1953] is a single-valued solution concept for cooperative games. The \( i \)\textsuperscript{th} component of the single solution vector satisfying this solution concept for any cooperative game \((N, v)\) is given by Equation 1.

\[
\phi_i^{Sh} = \frac{1}{|\Pi(N)|} \sum_{\pi \in \Pi(N)} [v(P_\pi^i \cup \{i\}) - v(P_\pi^i)]
\]

The Shapley value of a player is the average marginal contribution of the player to the value of the predecessor set over every possible permutation of the player set. Table 1 contains manual calculations of the players' marginal contributions to each permutation and their Shapley values in Example 1.

### 2.2 Properties of the Shapley value

We define the solution concept properties that characterize the Shapley value and emphasize their relevance and meaning in a feature selection game. In this game input features are players, coalitions are subsets of features and the payoff is a scalar valued goodness of fit for a machine learning model using these input features.

<table>
<thead>
<tr>
<th>Permutation</th>
<th>Player 1</th>
<th>Player 2</th>
<th>Player 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 2, 3)</td>
<td>7</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>(1, 3, 2)</td>
<td>7</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>(2, 1, 3)</td>
<td>7</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>(2, 3, 1)</td>
<td>2</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>(3, 1, 2)</td>
<td>7</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>(3, 2, 1)</td>
<td>2</td>
<td>9</td>
<td>14</td>
</tr>
</tbody>
</table>

**Table 1:** The permutations of the player set, marginal contributions of the players in each permutation and the Shapley values.

**Definition 8.** Null player. Player \( i \) is called a null player if

\[
v(S \cup \{i\}) = v(S) \forall S \subseteq N \setminus \{i\}.
\]

A solution concept \( \Phi \) satisfies the null player property if for every game \((N, v)\), every \( \phi(N, v) \in \Phi(N, v) \), and every null player \( i \) it holds that \( \phi(N, v)_i = 0 \).

In the feature selection game a solution concept with the null player property assigns zero value to those features that never increase the goodness of fit when added to the feature set.

**Definition 9.** Efficiency. A solution concept \( \Phi \) is efficient or Pareto optimal if for every game \((N, v)\) and every solution vector \( \phi(N, v) \in \Phi(N, v) \) it holds that \( \sum_{i \in N} \phi(N, v)_i = v(N) \).

Consider the goodness of fit of the model trained by using the whole set of input features. The importance measures assigned to individual features by an efficient solution concept sum to this goodness of fit. This allows for quantifying the contribution of individual features to the whole performance of the trained model.

**Definition 10.** Symmetry. Two players \( i \) and \( j \) are symmetric if

\[
v(S \cup \{i\}) = v(S \cup \{j\}) \forall S \subseteq N \setminus \{i, j\}.
\]

A solution concept \( \Phi \) satisfies symmetry if for all \((N, v)\) for all \( \phi(N, v) \in \Phi(N, v) \) and all symmetric players \( i, j \in N \) it holds that \( \phi(N, v)_i = \phi(N, v)_j \).

The symmetry property implies that if two features have the same marginal contribution to the goodness of fit when added to any possible coalition then the importance of the two features is the same. This property is essentially a fair treatment of the input features and results in identical features receiving the same importance score.

**Definition 11.** Linearity. A single-valued solution concept \( \Phi \) satisfies linearity if for any two games \((N, v)\) and \((N, w)\), and for the solution vector of the TU game given by \((N, v + w)\) it holds that

\[
\phi(N, v + w)_i = \phi(N, v)_i + \phi(N, w)_i, \quad \forall i \in N.
\]

Let us imagine a binary classifier and two sets of data points – on both of these datasets, we can define feature selection games with binary cross entropy-based payoffs. The Shapley values of input features in the feature selection game calculated on the pooled dataset would be the same as adding together the Shapley values calculated from the two datasets separately.

These four properties together characterize the Shapley value.

**Theorem 1** [Shapley, 1953]. A single-valued solution concept satisfies the null player, efficiency, symmetry, and linearity properties if and only if it is the Shapley value.
3 Approximations of the Shapley Value

Shapley value computation requires an exponential number of characteristic function evaluations, resulting in exponential time complexity. This is prohibitive in a machine learning context when each evaluation can correspond to training a machine learning model. For this reason, machine learning applications use a variety of Shapley value approximation methods we discuss in this section. In the following discussion \( \hat{\phi}^S_i \) denotes an approximated Shapley value for player \( i \in \mathcal{N} \).

3.1 Monte Carlo Permutation Sampling

Monte Carlo permutation sampling for the general class of cooperative games was first proposed by Castro et al. [2009] to approximate the Shapley value in linear time.

```
Data: (\mathcal{N}, \nu) - Cooperative TU game.
\text{k} - Number of sampled permutations.

Result: \( \hat{\phi}^S_i \) - Approximated Shapley value \( \forall i \in \mathcal{N} \).
1. \( \hat{\phi}^S_i = 0 \); \( \forall i \in \mathcal{N} \)
2. \text{for} \ (1, \ldots, \text{k}) \text{ do}
   3. \( \pi \leftarrow \text{Uniform Sample}(\Pi(\mathcal{N})) \)
   4. \text{for} \ i \in \mathcal{N} \text{ do}
      5. \( \pi_i \leftarrow \{ j \in \mathcal{N} \mid \pi(j) < \pi(i) \} \)
      6. \( \hat{\phi}^S_i \leftarrow \hat{\phi}^S_i + \frac{\nu(\pi \cup \{i\} - \nu(\mathcal{P}_i))}{\text{k}} \)
   7. \text{end}
8. \text{end}
```

Algorithm 1: Monte Carlo permutation sampling approximation of the Shapley value.

As shown in Algorithm 1, the method performs a sampling-based approximation. At each iteration, a random element from the permutations of the player set is drawn. The marginal contributions of the players in the sample permutation are scaled down by the number of samples (which is equivalent to taking an average) and added to the approximated Shapley values from the previous iteration. Castro et al. [2009] provide asymptotic error bounds for this approximation algorithm via the central limit theorem when the variance of the marginal contributions is known. Maleki et al. [2013] extended the analysis of this sampling approach by providing error bounds when either the variance or the range of the marginal contributions is known via Chebyshev’s and Hoeffding’s inequalities. Their bounds hold for a finite number of samples in contrast to the previous asymptotic bounds.

Stratified Sampling for Variance Reduction

In addition to extending the analysis of Monte Carlo estimation, Maleki et al. [2013] demonstrate how to improve the Shapley Value approximation when sampling can be stratified by dividing the permutations of the player set into homogeneous, non-overlapping sub-populations. In particular, they show that if the set of permutations can be grouped into strata with similar marginal gains for players, then the approximation will be more precise. Following this, Castro et al. [2017] explored stratified sampling approaches using strata defined by the set of all marginal contributions when the player is in a specific position within the coalition. Burgess et al. [2021] propose stratified sampling approaches designed to minimize the uncertainty of the estimate via a stratified empirical Bernstein bound.

Other Variance Reduction Techniques

Following the stratified approaches of Maleki et al.; Castro et al.; Burgess et al. [2013; 2017; 2021], Illés et al. [2019] propose an alternative variance reduction technique for the sample mean. Instead of generating a random sequence of samples, they instead generate a sequence of ergodic but not independent samples, taking advantage of negative correlation to reduce the sample variance. Mitchell et al. [2021] show that other Monte Carlo variance reduction techniques can also be applied to this problem, such as antithetic sampling [Lomeli et al., 2019; Rubinstein et al., 2016]. A simple form of antithetic sampling uses both a randomly sampled permutation and its reverse. Finally, Touati et al. [2021] introduce a Bayesian Monte Carlo approach to Shapley value calculation, showing that Shapley value estimation can be improved by using Bayesian methods to approximate the Shapley value.

3.2 Multilinear Extension

By inducing a probability distribution over the subsets \( S \) where \( \mathcal{E}_i \) is a random subset that does not include player \( i \) and each player is included in a subset with probability \( q \), Owen [1972] demonstrated that the sum over subsets in Definition 7 can also be represented as an integral \( \int_0^1 e_i(q) dq \) where \( e_i(q) = \mathbb{E}[v(\mathcal{E}_i \cup i) - v(\mathcal{E}_i)] \). Sampling over \( q \) therefore provides an approximation method – the multilinear extension. For example, Mitchell et al. [2021] uses the trapezoid rule to sample \( q \) at fixed intervals while Okhrati et al. [2021] proposes incorporating antithetic sampling as a variance reduction technique.

3.3 Linear Regression Approximation

In their seminal work Lundberg et al. [2017] apply Shapley values to feature importance and explainability (SHAP values), demonstrating that Shapley values for TU games can be approximated by solving a weighted least squares optimization problem. Their main insight is the computation of Shapley values by approximately solving the following optimization problem:

\[
\begin{align*}
  w_S &= \frac{|\mathcal{N}| - 1}{(|S|!) |\mathcal{N}| - |S|} \\
  \min_{\hat{\phi}^S_0, \ldots, \hat{\phi}^S_n} \sum_{S \subseteq \mathcal{N}} w_S \left( \hat{\phi}^S_0 + \sum_{i \in S} \hat{\phi}^S_i - \nu(S) \right) \\
  & \text{s.t. } \hat{\phi}^S_0 = \nu(\emptyset), \quad \hat{\phi}^S_0 + \sum_{i \in \mathcal{N}} \hat{\phi}^S_i = \nu(\mathcal{N}).
\end{align*}
\]

The definition of weights in Equation (2) and the objective function in Equation (3) implies the evaluation of \( \nu(S) \) for 2\(^n\) coalitions. To address this Lundberg et al. [2017] propose approximating this problem subsampling the coalitions. Note that \( w_S \) is higher when coalitions are large or small. Covert et al. [2021] extend the study of this method, finding that while SHAP is a consistent estimator, it is not an unbiased estimator. By proposing and analyzing a variation of this method that is unbiased, they conclude that while there is a small bias incurred by SHAP it has a significantly lower variance than the corresponding unbiased estimator. Covert et al. [2021] then propose a variance reduction method for SHAP, improving convergence speed by a magnitude through sampling coalitions in pairs with each selected alongside its complement.
4 Machine Learning and the Shapley Value

Our discussion about applications of the Shapley value in the machine learning domain focuses on the formulation of the cooperative games, definition of the player set and payoffs, Shapley value approximation technique used, and the time complexity of the approximation. We summarized the most important application areas with this information in Table 2 and grouped the relevant works by the problem solved.

4.1 Feature Selection

The feature selection game treats input features of a machine learning model as players and model performance as the payoff [Guyon et al., 2003; Fryer et al., 2021]. The Shapley values of features quantify how much individual features contribute to the model’s performance on a set of data points.

Definition 12. Feature selection game. Let the player set be \( N = \{1, \ldots, n\} \), for \( S \subseteq N \) the train and test feature vector sets are \( X^\text{Train}_S = \{x^\text{Train}_i | i \in S\} \) and \( X^\text{Test}_S = \{x^\text{Test}_i | i \in S\} \). Let \( f_S(\cdot) \) be a machine learning model trained using \( X^\text{Train}_S \) as input, then the payoff is \( v(S) = g(y, \hat{y}_S) \) where \( g(\cdot) \) is a goodness of fit metric, and \( \hat{y}_S = f_S(X^\text{Train}_S) \) are the ground truth and predicted targets.

Shapley values, and close relatives such as the Banzhaf index [Banzhaf III, 1964], have been studied as a measure of feature importance in various contexts [Cohen et al., 2007; Sun et al., 2012; Williamson et al., 2020; Tripathi et al., 2020]. Using these importance estimates, features can be ranked and selected or removed accordingly. This approach has been applied to various tasks such as vocabulary selection in natural language processing [Patel et al., 2021] and feature selection in human action recognition [Guha et al., 2021].

4.2 Data valuation

In the data valuation game training set data points are players and the payoff is defined by the goodness of fit achieved by a model on the test data. Computing the Shapley value of players in a data valuation game measures how much data points contribute to the performance of the model.

Definition 13. Data valuation game. Let the player set be \( N = \{(x_i, y_i) | 1 \leq i \leq n\} \) where \( x_i \) is the input feature vector and \( y_i \) is the target. Given the coalition \( S \subseteq N \) let \( f_S(\cdot) \) be a learning model trained on \( S \). Let us denote the test set feature vectors and targets as \( X^\text{Test} \) and \( Y^\text{Test} \), given \( f_S(\cdot) \) the set of predicted labels is defined as \( \hat{Y} = \{f_S(x)|x \in X^\text{Test}\} \). Then the payoff of a model trained on the data points \( S \subseteq N \) is \( v(S) = g(Y^\text{Test}, \hat{Y}) \) where \( g(\cdot) \) is a goodness of fit metric.

The Shapley value is not the only method for data valuation – earlier works used function utilization [Koh et al., 2017; Sharchilev et al., 2018], leave-one-out testing [Cook, 1977] and core sets [Dasgupta et al., 2009]. However, these methods fall short when there are fairness requirements from the data valuation technique [Jia et al., 2019; Ghorbani et al., 2019; Kwon et al., 2021b]. Ghorbani proposed a framework of utilizing Shapley value in a data-sharing system [Ghorbani et al., 2019]; Jia et al. [2019] advanced this work with more efficient algorithms to approximate the Shapley value for data valuation. The distributional Shapley value has been discussed by Ghorbani et al. [2020a] who argued that keeping privacy is hard during Shapley value computation. Their method calculates the Shapley value over a distribution which solves problems such as lack of privacy. The computation time of this can be reduced as Kwon et al. [2021a] point out with approximation methods optimized for specific machine learning models.

4.3 Federated learning

A federated learning scenario can be seen as a cooperative game by modeling the data owners as players who cooperate to train a high-quality machine learning model [Liu et al., 2021].

Definition 14. Federated learning game. In this game players are a set of labeled dataset owners \( N = \{(X_i, Y_i) | 1 \leq i \leq n\} \) where \( X_i \) and \( Y_i \) are the feature and label sets owned by the \( i \)th silo. Let \( (X, Y) \) be a labeled test set, \( S \subseteq N \) a coalition of data silos, \( f_S(\cdot) \) a machine learning model trained on \( S \), and \( \hat{Y}_S \) the labels predicted by \( f_S(\cdot) \) on \( X \). The payoff of a coalition \( S \subseteq N \) is \( v(S) = g(Y, \hat{Y}_S) \) where \( g(\cdot) \) is a goodness of fit metric.

The system described by Liu et al. [2021] uses Monte Carlo sampling to approximate the Shapley value of data coming from the data silos in linear time. Given the potentially overlapping nature of the datasets, the use of configuration games could be an interesting future direction [Albizuri et al., 2006].

4.4 Explainable machine learning

In explainable machine learning the Shapley value is used to measure the contributions of input features to the output of a machine learning model at the instance level. Given a specific data point, the goal is to decompose the model prediction and assign Shapley values to individual features of the instance. There are universal solutions to this challenge that are model agnostic and designed for deep learning [Chen et al., 2018; Ancona et al., 2019], classification trees [Lundberg et al., 2017], and graphical models [Liu et al., 2020; Singal et al., 2021].

Universal explainability

A cooperative game for universal explainability is completely model agnostic; the only requirement is that a scalar-valued output can be generated by the model such as the probability of a class label being assigned to an instance.

Definition 15. Universal explainability game. Let us denote the machine learning model of interest with \( f(\cdot) \) and let the player set be the feature values of a single data instance: \( N = \{x_i | 1 \leq i \leq n\} \). The payoff of a coalition \( S \subseteq N \) in this game is the scalar valued prediction \( v(S) = \tilde{y}_S = f(S) \) calculated from the subset of feature values.

Calculating the Shapley value in a game like this offers a complete decomposition of the prediction because the efficiency axiom holds. The Shapley values of feature values are explanatory attributions to the input features and missing input feature values are imputed with a reference value such as the mean computed from multiple instances [Lundberg et al., 2017; Covert et al., 2021]. The pioneering Shapley value-based universal explanation method SHAP [Lundberg et al., 2017] proposes a linear time approximation of the Shapley values which we discussed in Section 3. This approximation has shortcomings and implicit assumptions about the features which are
Table 2: An application area, payoff definition, Shapley value approximation technique, and computation time (the player set is noted by $\mathcal{N}$) based comparison of research works. Specific applications of the Shapley value are grouped together and ordered chronologically.

<table>
<thead>
<tr>
<th>Application</th>
<th>Reference</th>
<th>Payoff</th>
<th>Approximation</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Selection</td>
<td>[Cohen et al., 2007]</td>
<td>Validation loss</td>
<td>Exact</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Sun et al., 2012]</td>
<td>Mutual information</td>
<td>Exact</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Williamson et al., 2020]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Tripathi et al., 2020]</td>
<td>Training loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Patel et al., 2021]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Ghara et al., 2021]</td>
<td>Validation loss</td>
<td>Exact</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Data Valuation</td>
<td>[Jia et al., 2019]</td>
<td>Validation loss</td>
<td>Restricted Monte Carlo sampling</td>
<td>$O(\sqrt{</td>
</tr>
<tr>
<td></td>
<td>[Ghorbani et al., 2019]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Shum et al., 2021]</td>
<td>Validation loss</td>
<td>Exact</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Deutch et al., 2021]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Kwon et al., 2021a]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Kwon et al., 2021b]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Federated Learning</td>
<td>[Liu et al., 2021]</td>
<td>Validation loss</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Universal Explainability</td>
<td>[Lundberg et al., 2017]</td>
<td>Interaction attribution</td>
<td>Integrated gradients</td>
<td>$O(</td>
</tr>
<tr>
<td></td>
<td>[Sundararajan et al., 2020a]</td>
<td>Interaction attribution</td>
<td>Integrated gradients</td>
<td>$O(</td>
</tr>
<tr>
<td>Explainability of Deep Learning</td>
<td>[Covert et al., 2021]</td>
<td>Attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Explainability of Graphical Models</td>
<td>[Chen et al., 2018]</td>
<td>Attribution</td>
<td>Restricted Monte Carlo sampling</td>
<td>$O(2^{</td>
</tr>
<tr>
<td></td>
<td>[Ancona et al., 2019]</td>
<td>Neuron attribution</td>
<td>Voting game</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Ghorbani et al., 2020b]</td>
<td>Neuron attribution</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Zhang et al., 2021]</td>
<td>Interaction Attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Explainability in Graph Machine Learning</td>
<td>[Liu et al., 2020]</td>
<td>Attribution</td>
<td>Exact</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Heskes et al., 2020]</td>
<td>Causal Attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Wang et al., 2021b]</td>
<td>Causal Attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Singal et al., 2021]</td>
<td>Causal Attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Multi-agent Reinforcement Learning</td>
<td>[Yuan et al., 2021]</td>
<td>Edge level attribution</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Duval et al., 2021]</td>
<td>Edge level attribution</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Yuan et al., 2021]</td>
<td>Edge level attribution</td>
<td>Linear regression</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Wang et al., 2021a]</td>
<td>Global reward</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td></td>
<td>[Li et al., 2021]</td>
<td>Global reward</td>
<td>Monte Carlo sampling</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
<tr>
<td>Model Valuation in Ensembles</td>
<td>[Rozemberczki et al., 2021]</td>
<td>Predictive performance</td>
<td>Voting game</td>
<td>$O(\mathcal{N}^2)$</td>
</tr>
</tbody>
</table>

addressed by newer Shapley value-based explanation techniques. For example, in [Frye et al., 2020a] the input features are not necessarily independent, [Frye et al., 2020b] restricts the permutations based on known causal relationships, and in [Covert et al., 2021] the proposed technique improves the convergence guarantees of the approximation. Several methods generalize SHAP beyond feature values to give attributions to first-order feature interactions [Sundararajan et al., 2020b; Sundararajan et al., 2020a]. However, this requires that the player set is redefined to include feature interaction values.

Deep learning
In neuron explainability games neurons are players and attributions to the neurons are payoffs. The primary goal of Shapley value-based explanations in deep learning is to solve these games and compute attributions to individual neurons and filters [Ghorbani et al., 2020b; Ancona et al., 2019].

Definition 16. Neuron explainability game. Let us consider $f_{IN}(\cdot)$ the encoder layer of a neural network and $x$ the input feature vector to the encoder. In the neuron explainability game the player set is $\mathcal{N} = f_{IN}(x) = \{h_1, \ldots, h_n\}$ - each player corresponds to the output of a neuron in the final layer of the encoder. The payoff of coalition $S \subseteq \mathcal{N}$ is defined as the predicted output $v(S) = \bar{y}_S = f_{OUT}(S)$ where $f_{OUT}(\cdot)$ is the head layer of the neural network.

In practical terms, the payoffs are the output of the neural network obtained by masking out certain neurons. Using the Shapley values obtained in these games the value of individual neurons can be quantified. At the same time, some deep learning specific Shapley value-based explanation techniques have designs and goals that are aligned with the games described in universal explainability. These methods exploit the structure of the input data [Chen et al., 2018] or the nature of feature interactions [Zhang et al., 2021] to provide efficient computations of attributions.

Graphical models
Compared to universal explanations the graphical model-specific techniques restrict the admissible set of player set permutations considered in the attribution process. These restrictions are defined based on known causal relations and permutations are generated by various search strategies on the graph describing the probabilistic model [Heskes et al., 2020; Liu et al., 2020; Singal et al., 2021]. Methods are differentiated from each other by how restrictions are defined and how permutations are restricted.

Relational machine learning
In the relational machine learning domain the Shapley value is used to create edge importance attributions of instance-level explanations [Duval et al., 2021; Yuan et al., 2021]. Essentially the Shapley value in these games measures the average marginal change in the outcome variable as one adds a specific edge to the edge set in all of the possible edges set permutations. It is worth noting that the edge explanation and attribution techniques proposed could be generalized to provide node attributions.

Definition 17. Relational explainability game. Let us define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ are the vertex and edge sets. Given the relational machine learning model $f(\cdot)$, node feature matrix $X$, node $v \in \mathcal{V}$, the payoff of coalition $S \subseteq \mathcal{V}$ in
4.5 Multi-agent reinforcement learning

Global reward multi-agent reinforcement learning problems can be modeled as TU games [Wang et al., 2021a; Li et al., 2021] by defining the player set as the set of agents and the payoff of coalitions as a global reward. The Shapley value allows an axiomatic decomposition of the global reward achieved by the agents in these games and the fair attribution of credit assignments to each of the participating agents.

4.6 Model valuation in ensembles

The Shapley value can be used to assess the contributions of machine learning models to a composite model in ensemble games. In these games, players are models in an ensemble and payoffs are decided by whether prediction mad by the model are correct.

Definition 18. Ensemble game. Let us consider a single target - feature instance denoted by \((y, x)\). The player set in ensemble games is defined as a set of machine learning models \(N = \{ f_i(\cdot) | 1 \leq i \leq n \}\) that operate on the feature set. The predicted target output by the ensemble \(S \subseteq N\) is defined as \(\hat{y}_S = \hat{f}(S, x)\) where \(\hat{f}(\cdot)\) is a prediction aggregation function. The payoffs \(S\) is \(v(S) = g(y, \hat{y}_S)\) where \(g(\cdot)\) is a goodness of fit metric.

The ensemble games described by [Rozenberczki et al., 2021] are formulated as a special subclass of voting games. This allows the use of precise game-specific approximation [Fatima et al., 2008] techniques and because of this the Shapley value estimates are obtained in quadratic time and have a tight approximation error. The games themselves are model agnostic concerning the player set – ensembles can be formed by heterogeneous types of machine learning models that operate on the same inputs.

5 Discussion

The Shapley value has a wide-reaching impact in machine learning, but it has limitations and certain extensions of the Shapley value could have important applications in machine learning.

5.1 Limitations

Computation time

Computing the Shapley value for each player naively in a TU game takes factorial time. In some machine learning application areas such as multi-agent reinforcement learning and federated learning where the number of players is small, this is not an issue. However, in large scale data valuation [Kwon et al., 2021a; Kwon et al., 2021b], explainability [Lundberg et al., 2017], and feature selection [Patel et al., 2021] settings the exact calculation of the Shapley value is not tractable. In Sections 3 and 4 we discussed approximation techniques proposed to make Shapley value computation possible. In some cases, asymptotic properties of these Shapley value approximation techniques are not well understood – see for example [Chen et al., 2018].

Interpretability

By definition, the Shapley values are the average marginal contributions of players to the payoff of the grand coalition computed from all permutations [Shapley, 1953]. Theoretical interpretations like this one are not intuitive and not useful for non-game theory experts. This means that translating the meaning of Shapley values obtained in many application areas to actions is troublesome [Kumar et al., 2020]. For example in a data valuation scenario: is a data point with a twice as large Shapley value as another one twice as valuable? Answering a question like this requires a definition of the cooperative game that is interpretable.

Axioms do not hold under approximations

As we discussed most applications of the Shapley value in machine learning use approximations. The fact that under these approximations the desired axiomatic properties of the Shapley value do not hold is often overlooked [Sundararajan et al., 2020b]. This is problematic because most works argue for the use of Shapley value based on these axioms. In our view, this is the greatest unresolved issue in the applications of the Shapley value.

5.2 Future Research Directions

Hierarchy of the coalition structure

The Shapley value has a constrained version called Owen value [Owen, 1977] in which only permutations satisfying conditions defined by a coalition structure - a partition of the player set - are considered. The calculation of the Owen value is identical to that of the Shapley value, with the exception that only those permutations are taken into account where the players in any of the subsets of the coalition structure follow each other. In several real-world data and feature valuation scenarios even more complex hierarchies of the coalition, the structure could be useful. Having a nested hierarchy imposes restrictions on the admissible permutations of the players and changes player valuation. Games with such nested hierarchies are called level structure games in game theory. [Winter, 1989] presents the Winter value a solution concept to level structure games - such games are yet to receive attention in the machine learning literature.

Overlapping coalition structure

Traditionally, it is assumed that players in a coalition structure are allocated in disjoint partitions of the grand coalition. Allowing players to belong to overlapping coalitions in configuration games [Albizuri et al., 2006] could have several applications in machine learning. For example in a data-sharing - feature selection scenario multiple data owners might have access to the same features - a feature can belong to overlapping coalitions.

Solution concepts beyond the Shapley value

The Shapley value is a specific solution concept of cooperative game theory with intuitive axiomatic properties (Section 2). At the same time it has limitations with respect to computation constraints and interpretability (Sections 3 and 5). Cooperative game theory offers other solution concepts such as the core, nucleolus, stable set, and kernel with their own axiomatizations. For example, the core has been used for model explainability and feature selection [Yan et al., 2021]. Research into the potential applications of these solution concepts is lacking.

6 Conclusion

In this survey we discussed the Shapley value, examined its axiomatic characterizations and the most frequently used Shapley value approximation approaches. We defined and reviewed its
uses in machine learning, highlighted issues with the Shapley value and potential new application and research areas in machine learning.

Acknowledgements
This research was supported by REPHRAIN: The National Research Centre on Privacy, Harm Reduction and Adversarial Influence Online (UKRI grant: EP/V011189/1). The authors would like to thank Anton Tsitsulin for feedback throughout the preparation of this manuscript.

References


[Kwon et al., 2021a] Yongchan Kwon, Manuel A Rivas, and James Zou. Efficient Computation and Analysis of Distributional Shapley...


