Shapley Values

David S. Rosenberg

Bloomberg ML EDU

December 1, 2021

Contents

1. Shapley Values
2. Shapley Values for Feature Importance
3. Shapley values for prediction functions

Shapley Values

Coalitional game[1]

- Suppose there is a game played by a team (or "coalition") of players.
- A **coalition game** is
  - a set N consisting of n "players" and
  - a function $v : 2^N \to \mathbb{R}$, with $v(\emptyset) = 0$, assigning a value to any subset of players.
- Think of N as a team. Maybe they're trying to solve a puzzle together...
  - Says how well a subset of the team would have done, cooperating on the puzzle.
- Suppose the whole team plays and gets value $v(N)$.
- How should that value be allocated to the individuals on the team?
- Is there a fair way to do it that reflects the contributions of each individual?

[1] Based on Shapley value article in Wikipedia [Wik20] and [MP08].

- Where we're headed is that we're going to apply this approach of "value allocation" to "coalitions" of feature "working together" to produce the final output.
- Of course, it's not really clear what it means to use a subset of features with a specific prediction function $f(x)$.
- Various approaches to this will give us different interpretations.

Solutions to coalition games

- Let $\mathcal{G}(N)$ denote the set of all coalition games on set N.
  - i.e. a game for every possible $v : 2^N \to \mathbb{R}$.
- A **solution** to the allocation problem on the set $\mathcal{G}(N)$ is a map $\Phi : \mathcal{G}(N) \to \mathbb{R}^n$
  - gives the allocation to each of n players for any game $v \in \mathcal{G}(N)$.
- Next we'll give a particular solution, the Shapley value solution.
- Then we'll give various properties that seem desirable for a solution.
- Finally, we'll state a theorem that says the Shapley value solution
  - is the unique solution satisfying these properties.

The Shapley value solution

- The **Shapley value solution** is $\Phi(v) = (\phi_i(v))_{i=1}^n$ where

$$\phi_i(v) = \sum_{S \subset (N - \{i\})} k_{|S|,n} \left(v(S \cup \{i\}) - v(S)\right),$$

  where $k_{s,n} = s!(n-s-1)!/n!$.
- In words, for any game $v \in \mathcal{G}(N)$, player i receives $\phi_i(v)$.
  - You can show that $\sum_{i=1}^n \phi_i(v) = v(N)$.
- Equivalently,

$$\phi_i(v) = \frac{1}{n!} \sum_R \left[v(P_i^R \cup \{i\}) - v\left(P_i^R\right) \right],$$

  - where sum ranges over all $n!$ permutations R of the players in N.
  - P_i^R is the set of players in N that precede i in order R.

- The second version can be explained by the "room parable" [MP08, p. 6]: Players enter a room one at a time to form the team of n players. Each player receives the marginal contribution of their presence (could be negative). If all orders of entering the room have the same probability, then $\phi_i(v)$ is the expected value of how much player i receives.

- Yet another way to write the Shapley value is as

$$\begin{aligned}\phi_i(v) &= \frac{1}{n}\sum_{s=0}^{n-1}\sum_{S\subset(N-\{i\})\text{ and }|S|=s}\binom{n-1}{s}^{-1}[v(S\cup\{i\})-v(S)]\\ &= \frac{1}{n}\sum_{s\text{:size of coalition}}\ \sum_{\text{coalition excluding } i \text{ of size } s}\frac{\text{marginal contribution of } i \text{ to the coalition}}{\text{number of coalitions of size } s \text{ excluding } i}.\end{aligned}$$

Efficiency and symmetry properties

- **Efficiency**: For any $v \in \mathcal{G}(N)$,

$$\sum_{i \in N} \phi_i(v) = v(N).$$

- **Symmetry**: For any $v \in \mathcal{G}(N)$, if players i and j are equivalent in the sense that

$$v(S \cup \{i\}) = v(S \cup \{j\})$$

  for every subset S of players that excludes i and j, then

$$\phi_i(v) = \phi_j(v).$$

- Also called "equal treatment of equals".

Linearity property

- **Linearity**: For any $v, w \in \mathcal{G}(N)$, we have

  $$\phi_i(v+w) = \phi_i(v) + \phi_i(w)$$

  for every player i in N. Also, for any $a \in \mathbb{R}$,

  $$\phi_i(av) = a\phi_i(v)$$

  for every player i in N.
- $v+w$ is the game resulting from summing the outcomes of v and w for each coalition.
- av is the game resulting from scaling up outcomes of v by the factor a
- (These can be useful for prediction functions that are linear combinations of other functions, such as gradient boosted regression trees.)

Null player property

- A player i is **null** in v if $v(S \cup \{i\}) = v(S)$ for all coalitions $S \subset N$.
- If player i is null in a game v, then $\phi_i(v) = 0$.

- (In the context of machine learning, this is called the Dummy property. Presumably because if we replace players by features, a dummy feature has no information, and we want it to get importance 0?)

Shapley value theorem (Shapley, 1953)

Theorem

The Shapley value solution $\Phi(v) = (\phi_i(v))_{i=1}^{n}$ *defined previously is the unique solution for* $\mathcal{G}(N)$ *that satisfies the*

- *efficiency, symmetry, linearity, and null properties.*

- Proof: See references.

Example: Shapley values for constant game

- Suppose $v(S) \equiv c$ for any coalition $S \subset N$, except $v(\emptyset) = 0$.
- Then for any $i, j \in N$, $S \subset (N - \{i, j\})$, we have

$$v(S \cup \{i\}) = v(S \cup \{j\}) = c,$$

  which implies $\phi_1(v) = \cdots = \phi_n(v)$ by the symmetry property.
- By the efficiency property,

$$\sum_{i \in N} \phi_i(v) = v(N) = c.$$

- Therefore, $\phi_1(v) = \cdots = \phi_n(v) = c/n$.

Example: game plus a constant

- Suppose we have a game $v(S)$ on N
  - with Shapley values $\phi_1(v),\ldots,\phi_n(v)$.
- Suppose we shift the rewards, so $v'(S) := v(S) + c$.
- What are the Shapely values for $v'(S)$?
- Let $w(S) \equiv c$ for $S \subset N$, except $w(\emptyset) = 0$.
- Then $v'(S) = v(S) + w(S)$ and by linearity,

$$\phi_i(v') = \phi_i(v+w) = \phi_i(v) + \phi_i(w) = \phi_i(v) + \frac{c}{n}.$$

- So if we shift by a constant, the shift is divided equally among the players.

Shapley Values for Feature Importance

Shapley values for features

- Shapley values are about *n*-player games.
- In particular, they are about set functions on a set of *n* elements.
- How can we connect this to the feature importance in machine learning?
- Easy part: each "player" is a feature.
- Hard part: what's the set function?
- We have a prediction function,
    - but it doesn't naturally apply to subsets of features.
- What if we start earlier:
    - building a model with a subset of features

Attribute R^2 to features

/Users/drosen/Dropbox/repos/mltopics/figures/interpretability/Screen Shot 2021

- An early application of Shapley values to machine learning [LC01].
- Applied Shapley values to allocate the R^2 performance measure to features
  - for linear regression, though we'll present the obvious generalization.
- Essentially the same approach was actually done much earlier,

Attribute model performance to features

- Let $R(f)$ be some performance measure of a prediction function f.
- Let $\mathcal{A} : \mathcal{D} \mapsto f$ represent a model training algorithm that
  - takes a training dataset $\mathcal{D}$ and
  - produces a prediction function f.
- Let $\{1, \ldots, d\}$ index the features available for a problem.
- Let $\mathcal{D}_S$ denote the dataset with just the features indexed by $S \subset \{1, \ldots, d\}$.
- Define the set function $v(S) := R(\mathcal{A}(\mathcal{D}_S))$ and $v(\emptyset) = 0$.
- For any subset of features, $v(S)$ gives
  - the performance of the model trained on just that subset of features.

Lipovetsky and Conklin (2001)

- In [LC01],
  - performance measure was R^2
  - model class was linear models.
- They used only 7 features, and linear models train quickly,
  - so computation wasn't an issue.
- Generally speaking, need to train 2^d models.
- Not practical in most machine learning settings.

Monte Carlo approach

- The Shapley values in our scenario are

$$\phi_i(v) = \frac{1}{d!}\sum_R \left[v(P_i^R \cup \{i\}) - v\left(P_i^R\right)\right],$$

  - where sum ranges over all $n!$ permutations R of the players in N.
  - P_i^R is the set of players in N that precede i in order R.
- We can approximate this by averaging a random sample of M permutations.
- This still requires training Md models, which may not be practical for large d.
- This whole approach is only realistic when d is small and training and evaluation are fast.

Connection to LOCO

- This approach is most related to LOCO from an earlier module.
- We're not saying anything about a particular prediction function.
- We're saying something about the importance of each feature
  - in a particular dataset,
  - for a particular model training procedure
- LOCO was about the effect of removing each single feature.
- Here we have something that seems a bit deeper,
  - though hard to say exactly what machine learning question it's answering.

Shapley values for prediction functions

Interpreting a prediction function

- Suppose we want to use Shapley values
  - to interpret a particular prediction function $f(x)$.
- It's not obvious what it means to evaluate f using a subset of features.
- This is not a standard operation in machine learning.
- Let's write x_S for the features corresponding to $S \subset \{1,\ldots,d\}$.
- Let's write x_C for the features corresponding to the complement $\{1,\ldots,d\} - S$.
- So if $f(x) = f(x_S, x_C)$, we need a definition for $f_S(x_S)$.

Two approaches to defining $f_S(x_S)$

- Two approaches, as described by [CJLL20, JMB19].
- **Conditional expectation** (or "**observational conditional expectation**")

$$f_S(x_S) \quad := \quad \mathbb{E}\left[f(x_S, X_C) \mid X_S = x_S\right].$$

- **Marginal expectation** (or "**interventional conditional expectation**")

$$\begin{aligned} f_S(x_S) \quad &:= \quad \mathbb{E}[f(x_S, X_C)] \\ &= \quad \mathbb{E}\left[f(x_S, X_C) \mid do(X_S = x_S)\right], \end{aligned}$$

  where the do-operator is beyond our scope, but see [JMB19].
- Conditional expectation keeps our evaluations $f(x_S, x_C)$ on the data manifold.
- Marginal expectation will potentially evaluate $f(x_S, x_C)$ off the data manifold,
  - when we have dependencies between x_S and x_C.

(Note: we discussed exactly these definitions of $f_X(x_S)$ in the feature importance module.)

Independent features

- Note that when X_S and X_C are independent, we have

$$\mathbb{E}\left[f(x_S, X_C) \mid X_S = x_S\right] = \mathbb{E}\left[f(x_S, X_C)\right].$$

- So the conditional expectation and marginal expectation methods are equivalent.
- In the literature, some works (e.g. [LL17, AJL21])
  - see conditional expectation as the preferred approach,
  - but use marginal expectation as an approximation (e.g. KernSHAP in [LL17]).
- Others argue that marginal expectations is what we wanted in the first place.
  - Since it gives a more interventinal interpretation.

Estimating $f_S(x_S)$

- We generally don't know the joint distribution of X,
  - so we can't exactly compute the expectations needed for $f_S(x_S)$ (either version).
- For the marginal version, we can use the same estimate as for partial dependency:

$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} f(x_S, x_{Ci}),$$

  where $(x_{C1}, \ldots, x_{Cn})$ are the n instantiations of x_C in a dataset $\mathcal{D}$.
- For consistency, we'll also define $\hat{f}_\emptyset = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$.
- For conditional expectation, this estimation is much more challenging.
  - In general, seems to require training 2^d regression models.
  - (TreeSHAP uses a special property of trees to kinda approximate this.)

Shapley values for prediction function

- Suppose we have an estimate $\hat{f}_S(x_S)$ for each $S \subset \{1,\ldots,d\}$.
- Then we can define the set function for our "game" on $\{1,\ldots,d\}$ as

$$\begin{aligned} v(S) &:= \hat{f}_S(x_S) \\ v(\emptyset) &:= 0. \end{aligned}$$

- Frequently it's defined as

$$\begin{aligned} v(S) &:= \hat{f}_S(x_S) - \hat{f}_\emptyset \\ v(\emptyset) &:= 0. \end{aligned}$$

- That way, Shapley values indicate how each feature
  - pulls the prediction away from the mean / "no information" prediction.

Linear case[2]

- Consider a linear regression model $f(x) = \beta_0 + \sum_{j=1}^{M} \beta_j x_j$.
- Consider the Shapley values based on marginal expectation,
  - or equivalently, based on conditional expectation with independent features.
- Let β_S be the vector of coefficients corresponding to X_S, and similarly for β_C.

$$\begin{aligned} f_S(x_S) &= \mathbb{E}[f(x_S, X_C)] \\ &= \mathbb{E}\left[\beta_0 + \beta_S^T x_S + \beta_C^T X_C\right] \\ &= \beta_0 + \beta_S^T x_S + \beta_C^T \mathbb{E}[X_C] \end{aligned}$$

- Let's define the set function as

$$\begin{aligned} v(S) &= f_S(x_S) - \mathbb{E}[f(X)] \\ v(\emptyset) &= 0 \end{aligned}$$

[2] Based on [AJL21, App B.1].

Shapley value linear case

- Let $C_i = C - \{i\}$. Then we have

$$\begin{aligned} v(S+\{i\}) - v(S) &= \beta_0 + \beta_S^T x_S + \beta_i x_i + \beta_{C_i}^T \mathbb{E}[X_{C_i}] - \mathbb{E}[f(X)] \\ &\quad - \left(\beta_0 + \beta_S^T x_S + \beta_i \mathbb{E}[X_i] + \beta_{C_i}^T \mathbb{E}[X_{C_i}] - \mathbb{E}[f(X)]\right) \\ &= \beta_i (x_i - \mathbb{E}[X_i]). \end{aligned}$$

- Note this is independent of S.
- Let's verify that this makes sense when $S = \emptyset$.
- Let C consist of all features except i:

$$\begin{aligned} v(\{i\}) - v(\emptyset) &= \beta_0 + \beta_i x_i + \beta_C^T \mathbb{E}[X_C] - \mathbb{E}[f(X)] \\ &= \beta_0 + \beta_i x_i + \beta_C^T \mathbb{E}[X_C] - \mathbb{E}\left[\beta_0 + \beta_i X_i + \beta_C^T X_C\right] \\ &= \beta_i (x_i - \mathbb{E}[X_i]). \end{aligned}$$

- Note that if we had defined $v(S) = f_S(x_S)$, without subtracting off $\mathbb{E}[f(X)]$, then

  $$v(S + \{i\}) - v(S) = \beta_i (x_i - \mathbb{E}[X_i])$$

  would only hold for $S \neq \emptyset$. Which would have complicated the subsequent calculations.
- Once we know the Shapley values for $v(S) = f_S(x_S) - \mathbb{E}[f(X)]$As we know, all the Shapley values would just increase by β_0/M if we used $v(S) = f_S(x_S)$.

Shapley value linear case

- So the Shapley value is

$$\begin{aligned}\phi_i(v) &= \sum_{S \subset (N - \{i\})} k_{|S|,n} \left(v\left(S \cup \{i\}\right) - v(S) \right) \\ &= \beta_i \left(x_i - \mathbb{E}\left[X_i\right] \right) \sum_{S \subset (N - \{i\})} k_{|S|,n} \\ &= \beta_i \left(x_i - \mathbb{E}\left[X_i\right] \right),\end{aligned}$$

where the last step we leave as an easy exercise.

References

Resources

- The most common citation for the proof of the Shapley value theorem is Shapley's paper [Sha53]. These slides provide a proof of the Shapley value theorem, and I think the first few sections of [MP08] are easier to read than Shapley's paper.
- [CJLL20] gives a comparison of using the marginal and the conditional forms of the set function.

References I

[AJL21] Kjersti Aas, Martin Jullum, and Anders Løland, *Explaining individual predictions when features are dependent: More accurate approximations to shapley values*, Artificial Intelligence **298** (2021), 103502.

[CJLL20] Hugh Chen, Joseph D. Janizek, Scott Lundberg, and Su-In Lee, *True to the model or true to the data?*, CoRR (2020).

[JMB19] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum, *Feature relevance quantification in explainable ai: a causal problem*, CoRR (2019).

[Kru87] William Kruskal, *Relative importance by averaging over orderings*, The American Statistician **41** (1987), no. 1, 6–10.

[LC01] Stan Lipovetsky and Michael Conklin, *Analysis of regression in game theory approach*, Applied Stochastic Models in Business and Industry **17** (2001), no. 4, 319–330.

References II

[LL17] Scott Lundberg and Su-In Lee, *A unified approach to interpreting model predictions*, 2017, pp. 4765–4774.

[MP08] Stefano Moretti and Fioravante Patrone, *Transversality of the shapley value*, TOP **16** (2008), no. 1, 1–41.

[Sha53] L. S. Shapley, *17. a value for n-person games*, Contributions to the Theory of Games (AM-28), Volume II, ch. 17, pp. 307–318, Princeton University Press, 1953.

[Wik20] Wikipedia contributors, *Shapley value — Wikipedia, the free encyclopedia*, 2020, [https://en.wikipedia.org/wiki/Shapley_value; accessed 26-April-2021].