Variational Characterization of Shapley Values

David S. Rosenberg

NYU: CDS

May 5, 2021

Contents

Recap of 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.
- Suppose the whole team plays and gets value $v(N)$.
- Show 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 the Shapley value article in Wikipedia [Wik20] and [MP08].

- Where we're headed here 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 feature 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 $v \in \mathcal{G}(N)$.
- 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 $k_{s,n} = s!\,(n-s-1)!/n!$.

The Shapley value solution is special

The Shapley value solution is the unique solution
with the following properties:

- **Efficiency**: For any $v \in \mathcal{G}(N)$, $\sum_{i \in N} \phi_i(v) = v(N)$.
- **Symmetry**: For any $v \in \mathcal{G}(N)$, $\forall i, j \in N$, if $v(S \cup \{i\}) = v(S \cup \{j\})$ for every subset S of players that excludes i and j, then $\phi_i(v) = \phi_j(v)$.
- **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.
- **Null**: 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 $\phi_i(v) = 0$.
- That's all very nice... but doesn't give me much intuition on what the values are.
- Shapley values are the sum of exponentially many terms, with mysterious weights.

Reformulation of Shapley values

A set function from Shapley values

- Shapley values are defined for a given set function $v(S)$.
- Turns out, we can use the Shapley values to define a new set function:

$$w(S) := \sum_{i \in S} \phi_i(v),$$

  with the convention that an empty sum is 0.
- Is there a relation between $w(S)$ and $v(S)$?
- At the extremes, they agree:
  - $w(\emptyset) = v(\emptyset) = 0$ by construction of w.
  - $w(N) = v(N)$ by the efficiency property of Shapley values.
- Does $w(S)$ approximate $v(S)$ in general?
- If so, can we derive Shapley values as the solution to an optimization problem?

Fitting a set function

- Let's define a new set function on $S \subset N$:

$$w(S) := \sum_{i \in S} w_i,$$

  for some $w \in \mathbb{R}^n$, with the convention that an empty sum is 0.
- Consider the following objective function on $w \in \mathbb{R}^n$:

$$J(w) \quad = \quad \sum_{S \subset N} [w(S) - v(S)]^2 \, q(|S|),$$

  where $q : \{1, \ldots, n-1\} \to \mathbb{R}$ is an arbitrary **weight function** (but not identically 0).
- This is a weighted least squares fit of $w(S)$ to $v(S)$.
- Note that the weight corresponding to S depends only on the size of S.
- We've intentionally left $q(0)$ and $q(n)$ undefined, as they won't matter...

Generalized Shapley values

Theorem ([CGKR88, Thm 3])

The $w \in \mathbb{R}^n$ that minimizes

$$J(w) \quad = \quad \sum_{S \subset N} [w(S) - v(S)]^2 \, q(|S|),$$

subject to the constraint that $w(N) = v(N)$ is

$$w_i = \frac{v(N)}{n} + \frac{1}{\beta} \left(\sum_{S \subset N : i \in S} v(S) q(|S|) - \frac{1}{n} \sum_{i=1}^{n} \sum_{S \subset N : i \in S} v(S) q(|S|) \right),$$

where $\beta = \sum_{s=1}^{n-1} q(s) \binom{n-2}{s-1}$, provided $\beta \neq 0$.

- The w_i's are called **generalized Shapley values**.

- Note that $q(0)$ and $q(n)$ can take any values in the objective function without affecting the results since
  - $w(\emptyset) = v(\emptyset) = 0$ by construction of $w(S)$ and
  - $w(N) = v(N)$ by the constraint.
- The notation in the paper is different, but here we've translated it to our notation.

A quadratic optimization for Shapley values

- If we take $q(s) = c\binom{n-2}{s-1}^{-1}$, for any $c \neq 0$, then

$$w_i \quad = \quad \sum_{S \subset (N - \{i\})} k_{|S|,n} [v(S \cup \{i\}) - v(S)],$$

  where $k_{s,n} = \frac{1}{s+1}\binom{n}{s+1}^{-1} = s!\,(n-s-1)!/n!$ [CGKR88, Thm 4].
- That is, w_i are the Shapley values!
- So for the right choice of weight function $q(s)$, we can find all n Shapley values by minimizing
  - the quadratic objective $J(w) = \sum_{S \subset N} [w(S) - v(S)]^2 \, q(|S|)$,
  - subject to the constraint $w(N) = v(N)$.
- Issue: The objective function has 2^n terms.

Theorem ([CGKR88, Thm 4])

If we choose $q(s) = c\binom{n-2}{s-1}^{-1}$ *for any* $c \neq 0$*, then the solution to the constrained optimization problem stated above is*

$$\begin{aligned} w_i &= \frac{1}{n} \sum_{S \subset N : i \in S} \binom{n-1}{|S|-1}^{-1} [v(S) - v(S - \{i\})] \\ &= \sum_{S \subset (N - \{i\})} k_{|S|,n} [v(S \cup \{i\}) - v(S)], \end{aligned}$$

where $k_{s,n} = \frac{1}{s+1}\binom{n}{s+1}^{-1} = \frac{1}{n}\binom{n-1}{s-1}^{-1} = s!\,(n-s-1)!/n!$*. And so* w_i *are the Shapley values.*

- The notation in [CGKR88, Thm 4] is different, but here we've translated it to our notation. There are many equivalent formulations of Shapley values, so we've tried to give a variety here. The last one matches our original definition.
- With a few lines of algebra and taking $c = \frac{1}{n}$, one can show this result is equivalent to [LL17, Thm 2].

Finding the Shapley values

Dropping the constraint

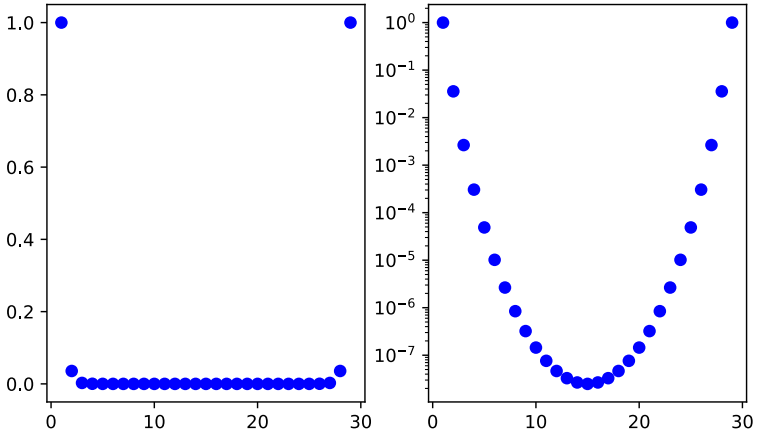
- The generalized Shapley value objective is

$$J(w) \quad = \quad \sum_{S \subset N} [w(S) - v(S)]^2 \, q(|S|).$$

- We have the constraint that $w(N) = \sum_{i=1}^{n} w_i = v(N)$.
- An easy way to enforce the equality constraint is to eliminate a variable.
- Let's eliminate a variable by setting $w_n = v(N) - \sum_{i=1}^{n-1} w_i$.
- With this substitution, we no longer need an explicit constraint that $w(N) = v(N)$.
- We can take $q(0) = q(n) = 0$, without affecting the result.

The weights for the Shapley kernel

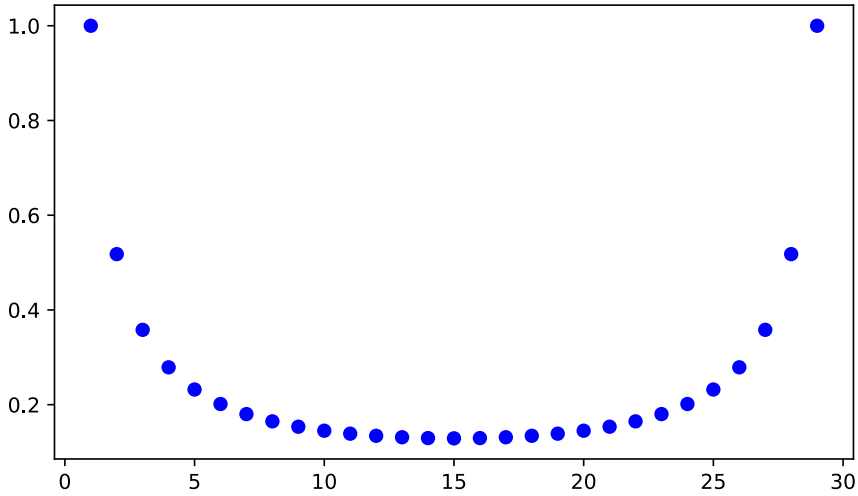
- For $n = 30$, let $q(s) = \binom{n-2}{s-1}^{-1}$ be the Shapley weight function.



- This is a plot of the weight function $q(s)$ on $\{1, 2, \ldots, 29\}$. As noted, we can take $q(0) = q(30) = 0$ in our optimization.
- At first it seems like almost all the weight of the objective function is concentrated on the smallest and largest sets.
- This is true, but it's not immediate from these plots. Remember that there are also far more subsets of intermediate sizes than extreme sizes.

Total weight by subset size

- For $n = 30$, let $w(s) = \binom{n}{s} q(s)$ be the total weight for subsets of each size.



- This is a plot of the total weight for all subsets of each size s, on $\{1, 2, \ldots, 29\}$.
- That is, we're plotting $w(s) = \binom{n}{s} q(s) = \binom{n}{s} \binom{n-2}{s-1}^{-1}$

$$w(s) = \binom{n}{s}\binom{n-2}{s-1}^{-1} = \frac{n!}{s!(n-s)!}\frac{(s-1)!(n-s-1)!}{(n-2)!} = \frac{n(n-1)}{s(n-s)}$$

- So most of the weight is on the small and large sets, but there is nontrivial weight throughout the range..

Approximate the objective function

- The objective function has 2^n terms,
  - which quickly becomes too large to handle exactly.
- We can approximate the objective function in many different ways.
- In SHAP, they use a combination of direct computation and random sampling.

Hybrid approach

- We can compute

$$J_2(w) := \sum_{S:|S|\in\{1,2,(n-2),n-1)\}} [w(S) - v(S)]^2 \, q(|S|).$$

- And then use random sampling to estimate the rest:

$$J_{-2}(w) := \sum_{S:3\leqslant|S|\leqslant(n-3)} [w(S) - v(S)]^2 \, q(|S|).$$

- How?

Using random sampling

- Define a probability mass function on $\{S : 3 \leqslant |S| \leqslant (n-3)\}$,
  - such that $p(S) = kq(|S|)$ for some $k > 0$.
- Then

$$\begin{aligned} J_{-2}(w) &= \frac{1}{k} \sum_{S:3\leqslant|S|\leqslant(n-3)} [w(S) - v(S)]^2 \, p(S) \\ &= \frac{1}{k} \mathbb{E}_{S\sim p} [w(S) - v(S)]^2 \end{aligned}$$

- We can now sample $S_1, \ldots, S_r$ from p to get a Monte Carlo estimate of $J_{-2}(w)$.
- (This is just one approach to get the idea – not exactly how SHAP does it.)

Optimizing our approximate objective

- Our final objective function would be

$$J(w) = \sum_{S:|S|\in\{1,2,(n-2),n-1)\}} [w(S) - v(S)]^2 \, q(|S|) + \frac{1}{kr} \sum_{i=1}^{r} [w(S_r) - v(S_r)]^2 .$$

- Now we have a quadratic objective of manageable size.
- We can use a standard optimization method.
- Or we can differentiate, set to zero, and end up with a **weighted least squares problem**.
- This is the idea of **Kernel SHAP**.
- For the exact implementation details, see SHAP code. The details aren't in [LL17].

References

Resources

- The ideas in the reformulation of Shapley values and generalized Shapley values are from [CGKR88]. However, I found Yuchen Pei's blog post to be very helpful in understanding the proofs in the paper by Charnes et al [CGKR88].
- The approach to finding the Shapley values is based on the implementation of Kernel SHAP in the SHAP GitHub repo. Although this is not a fully faithful representation of what they do, it should capture the mathematical ideas involved. I also looked at the description of Kernel SHAP in [AJL21, p. 5], but their representation is even less faithful to the actual implementation, as best as I can tell.

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.

[CGKR88] A. Charnes, B. Golany, M. Keane, and J. Rousseau, *Extremal principle solutions of games in characteristic function form: Core, chebychev and shapley value generalizations*, pp. 123–133, Springer Netherlands, Dordrecht, 1988.

[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.

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