Shapley Value Review

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Shapley Value: Introduction

- Divide some Value among members of a society
- A set of axioms about a fair distribution
- Shapley Vaue: A unique solution to satisfy all these 'fairness' axioms
- main idea: members should receive payments proportional to their 'marginal' contributions

Shapley Value: Introduction

- A Coalition Game of N players
- players: $\{1, \ldots, N\}$
- Coalition $S \subseteq N$: subset of players
- The set of all players N: Grand Coalition
- A value/payoff function: A value $V(S) : 2^N \rightarrow R$ for each coalition S
- Assuming $V(S)$ is *superadditive* : For all disjoint subsetd A,B $V(A \bigcup B) = V(A) + V(B)$[1]
- how to divide the payoff or value among N?

[1] no player has an incentive to play alone

Axioms: Fair Distribution

Shapley value is the unique value that satisfies this:

- **Efficiency**: The sum of the Shapley values of all agents equals the value of the grand coalition, $\Sigma_{i=1}^{N} \psi_i(V) = V(N)$
- **dummy player**: i is a dummy player if the amount that i contributes to any coalition is exactly the amount that i is able to achieve alone. For any V, if i is a dummy player then $\psi_i(N, V) = V(\{i\})$
- **Symmetry**: i and j are interchangeable if they always contribute the same amount to every coalition of the other agents. for all S that contains neither i nor j, $V(S \bigcup \{i\}) = V(S \bigcup \{j\})$.
- **Additivity**: For any two v_1 and v_2, we have for any player i that $\psi_i(N, v_1 + v_2) = \psi_i(N, v_1) + \psi_i(N, v_2)$, where the game $(N, v_1 + v_2)$ is defined by $(v_1 + v_2)(S) = v_1(S) + v_2(S)$ for every coalition S.

Unique Solution: Shapley Value

Shapley Value of feature i, for value function V:

$$\psi_i(V) = \frac{1}{N!} \sum_R V(P_i^R \bigcup \{i\}) - V(P_i^R) \tag{1}$$

R is the set of all permutations of N features, P_i^R the set of players in N which precede i in the order R

Why permutation? even though players are sets

Shapley value, has a nice interpretation in terms of expected marginal contribution. It is calculated by considering all the possible orders of arrival of the players into a room and giving each player his marginal contribution. Say $v\{1\} = 10$, $v\{2\} = 12$, $v\{1,2\} = 23$. There are two possible orders of arrival: (1) first 1 then 2, and (2) first 2 then 1.

Probability	Order of arrival	1's marginal contribution	2's marginal contribution
$\frac{1}{2}$	first 1 then 2	10	13
$\frac{1}{2}$	first 2 then 1	11	12

Solution: Shapley Value

Revised to account for repeated calculation of Shapley Value of feature i, for value function V:

$$\psi_i(V) = \frac{1}{N!} \sum_{S \subseteq N/\{i\}} (|S|!)(|N| - |S| - 1)! V(S \bigcup \{i\}) - V(S) \qquad (2)$$

Example: Taxi Fare for three people

3 players share a taxi. Here are the costs for each individual journey: - Player 1: 6 - Player 2: 12 - Player 3: 42 How much should each individual contribute?

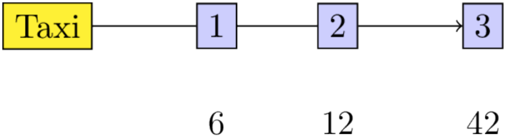

Figure 1: A taxi journey.

$$v(C) = \begin{cases} 6, & \text{if } C = \{1\} \\ 12, & \text{if } C = \{2\} \\ 42, & \text{if } C = \{3\} \\ 12, & \text{if } C = \{1,2\} \\ 42, & \text{if } C = \{1,3\} \\ 42, & \text{if } C = \{2,3\} \\ 42, & \text{if } C = \{1,2,3\} \end{cases}$$

Example: Taxi Fare for three people

calculate on board [2]

[2] ANSWER: fair way of sharing the taxi fare is for player 1 to pay 2, player 2 to pay 5 and player 3 to pay 35

Shapley Value for Interpretability

- Need $V(S)$ for each subset
- usually defined as feature subset values form a coalition which causes a change in the classifier's prediction.
- retraining the classifier for each $S \subseteq N$, so the method would no longer be independent of the learning algorithm and we would also require the training set that the original classifier was trained on.

Shapley Values for feature attributions: Computational Challenge

- Defining the value function V
- To be able to compute the Shapley for explanation, define the contribution function $v(\mathcal{S})$ for a certain subset $\mathcal{S}$.
- This function should resemble the value of $f(\boldsymbol{x}^*)$ when we only know the value of the subset $\mathcal{S}$ of these features.
- use the expected output of the predictive model, conditional on the feature values $\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}^*_{\mathcal{S}}$ of this subset:

$$v(\mathcal{S}) = \mathrm{E}[f(\boldsymbol{x})|\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}^*_{\mathcal{S}}]. \tag{3}$$

- or use a baseline value (all zeros or mean etc)

Making Shapley Faster: Monte Carlo Sampling[K+10]

Shapley Value of each feature for an instance x:

$$\psi_i(V) = \frac{1}{N!|A|} \sum_{R} \sum_{y \in A} V(x, y, P_i^R \bigcup \{i\}) - V(x, y, P_i^R) \tag{4}$$

Complexity

- Sample over all permutations and feature values(discrete case): $O(2^N|A|)$, where N is the number of features and $|A|$ is the possible values of features in total set A(exponential again)

Previous Ways to make shapley faster

- sampling
- kernelSHAP
- L Shapley C shapley
- Most can't work on image/sequence data with too many features

Kernel SHAP[LL17]

$$f(\boldsymbol{x}^*) = \phi_0 + \sum_{j=1}^{M} \phi_j^*,$$

where $\phi_0 = \mathrm{E}[f(\boldsymbol{x})]$ and ϕ_j^* is the ϕ_j for the prediction $\boldsymbol{x} = \boldsymbol{x}^*$. That is, the Shapley values explain the difference between the prediction $y^* = f(\boldsymbol{x}^*)$ and the global average prediction. In its simplest form, the WLS problem can be stated as the problem of minimizing

$$\sum_{\mathcal{S} \subseteq \mathcal{M}} (v(\mathcal{S}) - (\phi_0 + \sum_{j \in \mathcal{S}} \phi_j))^2 k(M, \mathcal{S}), \tag{5}$$

with respect to $\phi_0, \ldots, \phi_M$, where $k(M, \mathcal{S}) = (M-1)/(\binom{M}{|\mathcal{S}|} |\mathcal{S}| (M - |\mathcal{S}|))$, are denoted the Shapley kernel weights.

KernelSHAP

Then (5) may be rewritten to

$$(\boldsymbol{v} - \boldsymbol{Z}\phi)^T \boldsymbol{W}(\boldsymbol{v} - \boldsymbol{Z}\phi), \tag{6}$$

for which the solution is

$$\phi = \left(\boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{Z}\right)^{-1} \boldsymbol{Z}^T \boldsymbol{W} \boldsymbol{v}. \tag{7}$$

- $\boldsymbol{Z}$: $2^M \times (M+1)$ binary matrix representing all possible combinations of inclusion/exclusion of the M features[3]
- $\boldsymbol{v}$: vector containing $v(\mathcal{S})$,
- $\boldsymbol{W}$: $2^M \times 2^M$ diagonal matrix containing $k(M, |\mathcal{S}|)$, [4]

[3] where the first column is 1 for every row, while entry $j+1$ of row l is 1 if feature j is included in combination l, and 0 otherwise.

[4] where $\mathcal{S}$ in both cases resembles the feature combinations of the corresponding row in $\boldsymbol{Z}$.

Kernel SHAP Sampling

- When the model contains more than a few features M, computing the rhs : computationally expensive.
- The Shapley kernel weights have very different sizes, meaning that the majority of the subsets $\mathcal{S}$, that is, the rows in $\boldsymbol{Z}$, contributes very little to the Shapley value.
- Hence, assuming that we have an proper approximation for the elements in $\boldsymbol{v}$, a consistent approximation may be obtained by sampling (with replacement) a subset $\mathcal{D}$ of $\mathcal{M}$ from a probability distribution following the Shapley weighting kernel, and using only those rows $\boldsymbol{Z}_{\mathcal{D}}$ of $\boldsymbol{Z}$ and elements $\boldsymbol{v}_{\mathcal{D}}$ of $\boldsymbol{v}$ in the computation. As the Shapley kernel weights are used in the sampling, the sampled subsets are weighted equally in the new least squares problem.

L Shapley C shapley[CSWJ18]

- Do away with all possible feature values: use some reference x_0
- Choose 'good' subsets based on graph structure
- LShapley:

$$\hat{\phi}_x^k(i) = \frac{1}{|\mathcal{N}ki|} \sum_{\substack{T \ni i \\ T \subseteq \mathcal{N}ki}} \frac{1}{\binom{|\mathcal{N}ki|-1}{|T|-1}} m_x(T, i). \tag{8}$$

- CShapley:

$$\tilde{\phi}_x^k(i) = \sum_{U \in \mathcal{C}_k(i)} \frac{2}{(|U|+2)(|U|+1)|U|} m_x(U, i), \tag{9}$$

LS Tree: Model Interpretation When the Data Are Linguistic [CJ19]

f denotes the function that maps an input sentence $x = (x_1, \ldots, x_d)$ to the log probability score of a selected class.
Let $2^{[d]}$ denote the powerset of $[d] := \{1, 2, \ldots, d\}$. The parse tree maps the sentence to a collection of subsets, denoted as $\mathcal{P}t \subset 2^{[d]}$, where each subset $S \in \mathcal{P}$ contains the indices of words corresponding to one node in the parse tree.

$$\min_{\psi \in \mathbb{R}^d} \sum_{S \in \mathcal{P}} [v(S) - \sum_{i \in S} \psi_i]^2, \tag{10}$$

where component ψ_i of the optimal ψ is the importance score of word with index i.

Data Shapley[GZ19]

Algorithm 1 Truncated Monte Carlo Shapley

Input: Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathscr{A}$, performance score V
Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$

Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$
while Convergence criteria not met **do**
  $t \leftarrow t + 1$
  π^t: Random permutation of train data points
  $v_0^t \leftarrow V(\emptyset, \mathscr{A})$
  for $j \in \{1, \ldots, n\}$ **do**
    if $|V(D) - v_{j-1}^t| <$ Performance Tolerance **then**
      $v_j^t = v_{j-1}^t$
    else
      $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathscr{A})$
    end if
    $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t} \phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$
  end for
end while

Other: Temporal Shapley Value

- takes into account which algorithm came first
- Paper Quantifying Algorithmic Improvements over Time [KFM+18]

References I

- Jianbo Chen and Michael I Jordan, *Ls-tree: Model interpretation when the data are linguistic*, arXiv preprint arXiv:1902.04187 (2019).
- Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan, *L-shapley and c-shapley: Efficient model interpretation for structured data*, arXiv preprint arXiv:1808.02610 (2018).
- Amirata Ghorbani and James Zou, *Data shapley: Equitable valuation of data for machine learning*, arXiv preprint arXiv:1904.02868 (2019).
- Igor Kononenko et al., *An efficient explanation of individual classifications using game theory*, Journal of Machine Learning Research **11** (2010), no. Jan, 1–18.
- Lars Kotthoff, Alexandre Fréchette, Tomasz P Michalak, Talal Rahwan, Holger H Hoos, and Kevin Leyton-Brown, *Quantifying algorithmic improvements over time.*, IJCAI, 2018, pp. 5165–5171.

References II

Scott M Lundberg and Su-In Lee, *A unified approach to interpreting model predictions*, Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.