7-1-2022

Why Shapley Value and Its Variants Are Useful in Machine Learning (and in Other Applications)

Laxman Bokati
The University of Texas at El Paso, lbokati@miners.utep.edu

Olga Kosheleva
The University of Texas at El Paso, olgak@utep.edu

Vladik Kreinovich
The University of Texas at El Paso, vladik@utep.edu

Nguye Ngoc Thach
Banking University Ho Chi Minh City, ajeb@buh.edu.vn

Comments:
Technical Report: UTEP-CS-22-89

Why Shapley Value and Its Variants Are Useful in Machine Learning (and in Other Applications)

Laxman Bokati, Olga Kosheleva, Vladik Kreinovich, and Nguyen Ngoc Thach

Abstract Shapley value – a useful way to allocate gains in cooperative games – has been very successful in machine learning (and in other applications beyond cooperative games). This success is somewhat puzzling, since the usual derivation of the Shapley value is based on requirements like additivity that are natural in cooperative games and but not in machine learning. In this paper, we provide a new simple derivation of the Shapley value, a derivation that does not use game-specific requirements like additivity and is, thus, applicable in the machine learning case as well.

1 Formulation of the Problem

Before we start analyzing this problem, let us recall what is Shapley value; see, e.g., [2].

How to distribute gain between the agents: the problem for which Shapley value was invented. In a cooperating scenario, we have n collaborating agents $1, \ldots, n$.

Laxman Bokati
Computational Science Program, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: lbokati@miners.utep.edu

Olga Kosheleva
Department of Teacher Education, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: olgak@utep.edu

Vladik Kreinovich
Computational Science Program, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

Nguyen Ngoc Thach
Institute for Research Science and Banking Technology
Banking University Ho Chi Minh City, 39 Ham Nghi Street, District 1, Ho Chi Minh City
post code 71010, Vietnam, e-mail: thachnn@buh.edu.vn

We assume that for each set $S \subseteq N \stackrel{\text{def}}{=} \{1, \ldots, n\}$ of agents, we know the largest gain $v(S)$ that agents from this set can get with guarantee is they act together. The best strategy is for everyone to get together and get the gain $v(N)$.

The question is: how to divide this resulting gain $v(N)$ between the agents?

Let us reformulate this problem in more precise terms. In mathematical terms, what we need is a function φ that assigns:

- to each function $v : 2^N \to \mathbb{R}_0^+$ from the set 2^N of all subsets of N to the set $\mathbb{R}_0^+$ of all non-negative real numbers,
- an n-dimensional vector $(\varphi_1(v), \ldots, \varphi_n(v))$ of non-0negative values v_i for which

$$\varphi_1(v) + \ldots + \varphi_n(v) = v(N).$$

Natural requirements. A solution to the above problem was proposed in 1951 by Lloyd S. Shapley [5] – who received the 2012 Nobel Prize in Economics for this discovery. Shapley considered the following two natural requirements.

First requirement: fairness. The first requirement is fairness.

Namely, if in some situation, agents i and j contribute equally, i.e., if we have $v(S) = v(\pi_{i \leftrightarrow j}(S))$ for all sets S, where $\pi_{i \leftrightarrow j}$ is a permutation that swaps i and j and leaves all other elements intact, then these two agents should get the exact same amount:

$$\varphi_i(v) = \varphi_j(v).$$

Second requirement: additivity. The second requirement is additivity.

Namely, if we have two different independent situations with the same set of agents:

- one situation characterized by a function u, and
- another situation characterized by a function v.

Then the overall amount that each agent i gets in both situations is $\varphi_i(u) + \varphi_i(v)$.

Alternatively, we can consider these two situations as a single situation, with $w(S) = u(S) + v(S)$ for all S. It is reasonable to require that since we did not change anything by simply considering the two situation as one, the overall gain of each player in this new situation should be the same, i.e., we should have

$$\varphi_i(w) = \varphi_i(u + v) = \varphi_i(u) + \varphi_i(v).$$

Resulting formula. Shapley showed that these two requirements uniquely determine the function φ_i as

$$\varphi_i(v) = \sum_{S:\ i \notin S} \frac{|S|! \cdot (n - |S| - 1)!}{n!} \cdot (v(S \cup \{i\}) - v(S)),$$

where $|S|$ denotes the number of elements in the set S.

Shapley value is easy to compute. When n is small, we can simply use the above formula. For large n, we can use the equivalent description of the Shapley value $\varphi_i(v)$ in terms of permutations $\pi : N \to N$ of the set N.

Namely, each permutation sorts the elements of the set N as $\pi(1) < \pi(2) < \ldots$ We can then add these elements one by one, and for the case when we add the agent i, we can compute the difference between the new and the previous value of v. The expected value of this difference over random permutations is exactly the Shapley value.

It is easy to simulate a random permutation:

- as $\pi(1)$, we select each of elements $1, \ldots, n$ with the same probability $1/n$;
- then, as $\pi(2)$, we select each of the $n - 1$ remaining elements with the same probability $1/(n - 1)$, etc.

Thus, by using such Monte-Carlo simulations, we can estimate the Shapley value as accurately as possible.

Variants of the Shapley value. In some applications, it is useful to use variants of the Shapley value, of the type

$$\phi_i(v) = \sum_{S:\ i \notin S} a(|S|) \cdot (v(S \cup \{i\}) - v(S)),$$

where the function $a(|S|)$ is such that

$$\sum_{S:\ i \notin S} a(|S|) = 1,$$

i.e., equivalently, that

$$\sum_{k=0}^{n-1} \frac{(n-1)!}{(n-1-k)! \cdot k!} \cdot a(k) = 1.$$

Successful use of Shapley value in machine learning. Lately, the Shapley value has been successfully used in machine learning and in other applications, to describe the importance of different inputs; see, e.g., [1, 3, 4]. In this case:

- instead of agents, we gave inputs, and
- instead of a gain $v(S)$, we have a different characteristic – e.g., classification efficiency – corresponding to the case when we only use inputs from the set S.

This success is somewhat puzzling. The usual derivation of the Shapley value is based on additivity. However, for classification efficiency, adding two efficiencies makes no sense.

We therefore need a different explanation for the empirical success of Shapley value in these applications.

What we do in this paper. In this paper, we provide a simple alternative derivation of Shapley value and its variants, a derivation that does not use additivity and can, therefore, explain the success of Shapley value and its variants in machine learning applications.

2 A New Derivation of Shapley Value (and Its Variants)

Main idea. We want to come up with a value φ_i that describes how much adding an input i improves the desired result – e.g., improves the classification efficiency. In other words, we want this value to describe the difference $v(S \cup \{i\}) - v(S)$ between:

- the result $v(S \cup \{i\})$ obtained by adding i and
- the result $v(S)$ that we get without adding the input i.

So, we want to have to make sure that for each set S that does not contain the input i, this difference is close to the desired value φ_i:

$$v(S \cup \{i\}) - v(S) \approx \varphi_i. \tag{1}$$

From the idea to the exact formulation of the problem. In mathematical terms, we have several equations (1) for determining a single unknown φ_i. In other words, we have an over-determined system of linear equations. In data processing, a usual way to deal with such systems is to the use the Least Squares approach (see, e.g., [6]), i.e., to find the value φ_i for which the following sum attains its smallest possible value:

$$\sum_{S:\ i \notin S} \frac{((v(S \cup \{i\}) - v(S)) - \varphi_i)^2}{\sigma^2(S)} \tag{2}$$

where the coefficients $\sigma^2(S)$ describe the weight that we assign to each equation (1).

Requiring permutation-invariance. A priori, there is usually no reason to believe that some inputs and more important than others. Thus, it makes sense to require – as in the original derivation of the Shapley value – that the weights $\sigma^2(S)$ should not depend on which exactly inputs are included in the set S. These weights should be permutation-invariant – and thus, they should depend only on the size $|S|$ of the corresponding set S: $\sigma^2(S) = b(|S|)$ for some function $b : \{0, \ldots, n-1\} \to \mathbb{R}_0^+$.

Thus, we arrive at the need to minimize the following expression:

$$\sum_{S:\ i \notin S} \frac{((v(S \cup \{i\}) - v(S)) - \varphi_i)^2}{b(|S|)}. \tag{3}$$

Solving the resulting optimization problem leads exactly to Shapley value and its variants. In general, to find the value of the input that minimizes a given expression, we can differentiate this expression and equate the derivative to 0.

Differentiating the expression (3) with respect to the unknown φ_i and equating the derivative to 0, we conclude that

$$2 \cdot \sum_{S:\ i \notin S} \frac{\varphi_i - (v(S \cup \{i\}) - v(S))}{b(|S|)} = 0, \tag{4}$$

i.e., equivalently, that

$$\varphi_i \cdot \sum_{S:\ i \notin S} \frac{1}{b(|S|)} = \sum_{S:\ i \notin S} \frac{1}{b(|S|)} \cdot (v(S \cup \{i\}) - v(S)). \tag{5}$$

Thus, we conclude that

$$\varphi_i = \sum_{S:\ i \notin S} a(|S|) \cdot (v(S \cup \{i\}) - v(S)), \tag{6}$$

where we denoted

$$a(k) = \frac{\dfrac{1}{b(k)}}{\displaystyle\sum_{S:\ i \notin S} \frac{1}{b(|S|)}}. \tag{7}$$

This is exactly the above formula for the variants of Shapley value.

Vice versa, each variant of the Shapley value corresponding to the values $a(k)$ can be obtained this way: it is sufficient to take

$$b(k) = \frac{1}{a(k)}.$$

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

It was also supported by the program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and by a grant from the Hungarian National Research, Development and Innovation Office (NRDI).

References

1. A. Ghorbani and J. Zou, "Data Shapley: equitable valuation of data for machine learning", *Proceedings of the 36-th International Conference on Machine Learning*, Long Beach, California, June 9–15, 2019.
2. R. D. Luce and R. Raiffa, *Games and Decisions: Introduction and Critical Survey*, Dover, New York, 1989.
3. L. Merrick and A. Taly, *The Explanation Game: Explaining Machine Learning Models Using Shapley Values*, arXiv:1909.08128v3, 2020.
4. B. Rozemberczki, L. Watson, P. Bayer, ´ H.-T. Yang, O. Kiss, S. Nilsson, and R. Sarkar, *The Shapley Value in Machine Learning*, arXiv:2202.05594v2, 2022.
5. L. S. Shapley, *Notes on the n-Person Game – II: The Value of an n-Person Game*, RAND Corporation Technical Reports, Santa Monica, California, 1951.
6. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.