SHAFF: Fast and consistent SHapley eFfects via random Forests
GATSBII

Clément Bénard[1]**, Sébastien Da Veiga**[2]

[1] Thales Research & Technologies, [2] ENSAI, CREST

January 2025

Shapley effects

- Originally defined in game theory (Shapley, 1953)
- Attribute the value produced by a joint team to its individual members

Shapley effects

- Originally defined in game theory (Shapley, 1953)
- Attribute the value produced by a joint team to its individual members
- Difference of produced value between a subset of the team and the same subteam with an additional member (averaged over all possible subteams).

Figure: Illustration of Shapley effects (Lopez, 2021)

Shapley effects

Adapted by Owen (2014) to variable importance in supervised machine learning:

- member of the team = input variable
- value function = explained output variance

Shapley effects

Adapted by Owen (2014) to variable importance in supervised machine learning:

- member of the team = input variable
- value function = explained output variance

Regression setting

- input vector $\mathbf{X} = (X^{(1)}, \dots, X^{(p)}) \in \mathbb{R}^p$
- output $Y \in \mathbb{R}$
- dataset $\mathcal{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$, where $(\mathbf{X}_i, Y_i) \sim \mathbb{P}_{\mathbf{X},Y}$.

Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\ldots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p

Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \dots, p\}$

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^\star(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U\cup\{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
   Literature: Monte-Carlo methods
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \dots, p\}$

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\dots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
   Literature: Monte-Carlo methods
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \dots, p\}$
   Literature: strong approximation of the conditional distributions

Shapley effects

Formally, the Shapley effect of the j-th variable is defined by

$$Sh^{\star}(X^{(j)}) = \sum_{U \subset \{1,\ldots,p\} \setminus \{j\}} \frac{1}{p} \binom{p-1}{|U|}^{-1} \frac{\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U \cup \{j\})}]] - \mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]}{\mathbb{V}[Y]}.$$

Main property: equitably allocate contributions due to dependence and interactions across input variables

Two obstacles arise to estimate Shapley effects:

1. the computational complexity is exponential with the dimension p
   Literature: Monte-Carlo methods
2. $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ requires a fast and accurate estimate for all variable subsets $U \subset \{1, \ldots, p\}$
   Literature: strong approximation of the conditional distributions

Objective: use random forests to improve these two features.

Random Forests

Random forests

- learning algorithm introduced by Breiman (2001)
- state-of-the-art on a wide range of problems
- ensemble method: aggregation of a large number of weak learners
- weak learner: randomized CART tree

CART tree

CART tree

- piecewise constant estimate
- construction: recursive partition of the input space

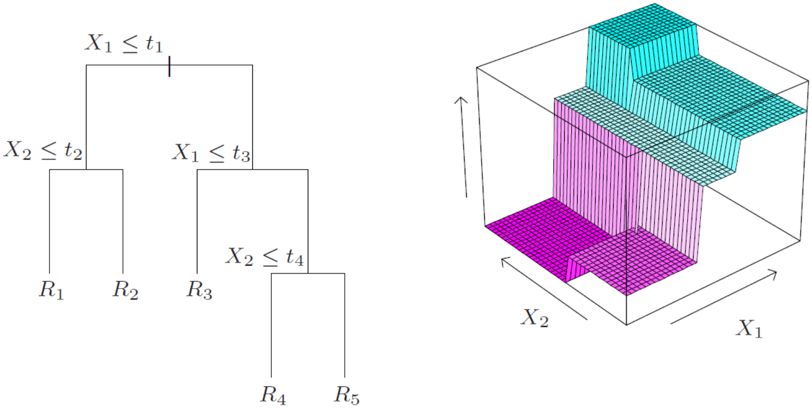
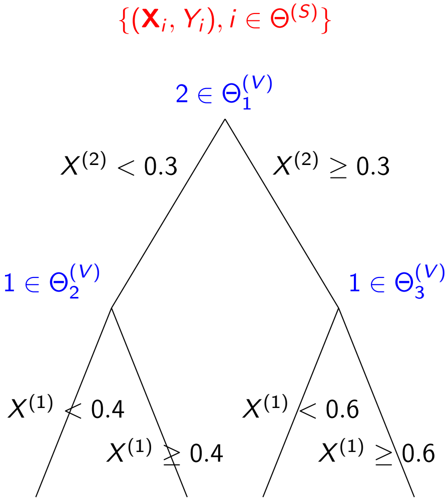


Figure: Example of a decision tree and the associated estimated function for $p = 2$ (Friedman et al., 2001).

Randomized CART Tree

Double randomization:
$\Theta = (\Theta^{(S)}, \Theta^{(V)})$

- data resampling: $\Theta^{(S)}$
- tree optimization: $\Theta^{(V)}$



1. Introduction

2. SHAFF Algorithm
   - Algorithm
   - Convergence
   - Experiments

SHAFF proceeds in three steps:

1. sample many subsets U (typically a few hundreds) using importance sampling

SHAFF: SHApley efFects via random Forests

SHAFF proceeds in three steps:

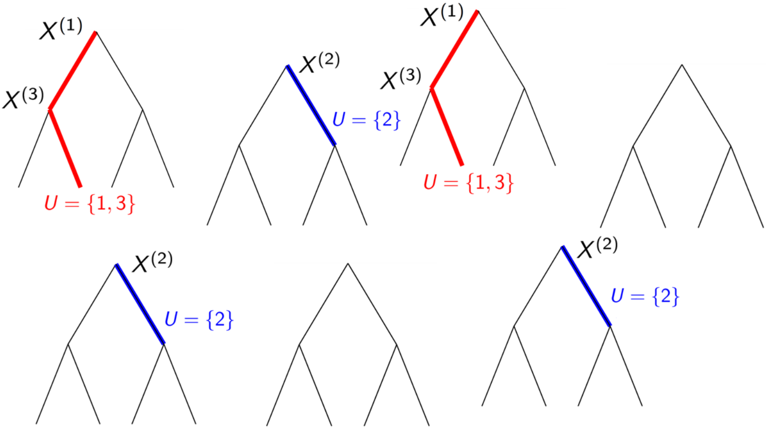
1. sample many subsets U (typically a few hundreds) using importance sampling
2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm

SHAFF proceeds in three steps:

1. sample many subsets U (typically a few hundreds) using importance sampling
2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm
3. solve a weighted linear regression problem to recover Shapley effects (Lundberg and Lee, 2017)

SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest



SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest
2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm for all selected U and their complementary sets $\{1,\ldots,p\} \setminus U$: $\hat{v}_{M,n}(U)$

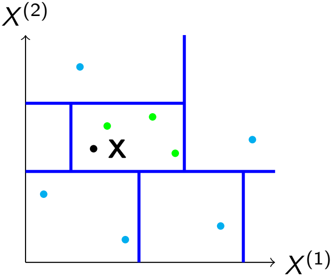


Figure: Partition of $[0,1]^2$ by a random tree (left side) projected on the subspace span by $\mathbf{X}^{(U)} = X^{(1)}$ (right side), for $p = 2$ and $U = \{1\}$.

SHAFF proceeds in three steps:

1. sample many subsets U, typically a few hundreds, based on their occurrence frequency $\hat{p}_{M,n}(U)$ in the random forest
2. estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$ with the projected forest algorithm for all selected U and their complementary sets $\{1, \ldots, p\} \setminus U$: $\hat{v}_{M,n}(U)$
3. solve a weighted linear regression problem to recover Shapley effects $\hat{\mathrm{Sh}}_{M_n,n}$ by minimizing in β

$$\ell_{M,n}(\beta) = \frac{1}{K} \sum_{U \in \mathcal{U}_{n,K}} \frac{w(U)}{\hat{p}_{M,n}(U)} (\hat{v}_{M,n}(U) - \beta^T I(U))^2,$$

where $w(U) = \frac{p-1}{\binom{p}{|U|}|U|(p-|U|)}$ and $I(U)$ is the binary vector of dimension p where the j-th component takes the value 1 if $j \in U$ and 0 otherwise.

Assumptions

(A1)

The response $Y \in \mathbb{R}$ follows

$$Y = m(\boldsymbol{X}) + \varepsilon$$

where

- $\boldsymbol{X} = (X^{(1)}, \ldots, X^{(p)}) \in [0, 1]^p$
- $\boldsymbol{X}$ *admits a density* f *such that* $c_1 < f(\boldsymbol{x}) < c_2$*, with constants* $c_1, c_2 > 0$
- m *is continuous*
- *the noise* ε *is sub-Gaussian and centered*

Assumptions

(A2): the random forest algorithm is slightly modified to converge

Assumptions

(A2): the random forest algorithm is slightly modified to converge

(A2)

- *A node split is constrained to generate child nodes with at least a small fraction* $\gamma > 0$ *of the parent node observations.*
- *The split selection is slightly modified: at each tree node, the number* $mtry$ *of covariates drawn to optimize the split is set to* $mtry = 1$ *with a small probability* $\delta > 0$. *Otherwise, with probability* $1 - \delta$, *the default value of* $mtry$ *is used.*

Assumptions

(A2): the random forest algorithm is slightly modified to converge

(A2)

- *A node split is constrained to generate child nodes with at least a small fraction* $\gamma > 0$ *of the parent node observations.*
- *The split selection is slightly modified: at each tree node, the number* $\mathtt{mtry}$ *of covariates drawn to optimize the split is set to* $\mathtt{mtry} = 1$ *with a small probability* $\delta > 0$*. Otherwise, with probability* $1 - \delta$*, the default value of* $\mathtt{mtry}$ *is used.*

(A3): tree partition is not too complex with respect to n

Assumptions

(A2): the random forest algorithm is slightly modified to converge

(A2)

- *A node split is constrained to generate child nodes with at least a small fraction* $\gamma > 0$ *of the parent node observations.*
- *The split selection is slightly modified: at each tree node, the number* $\mathtt{mtry}$ *of covariates drawn to optimize the split is set to* $\mathtt{mtry} = 1$ *with a small probability* $\delta > 0$*. Otherwise, with probability* $1 - \delta$*, the default value of* $\mathtt{mtry}$ *is used.*

(A3): tree partition is not too complex with respect to n

(A3)

The asymptotic regime of a_n*, the size of the subsampling without replacement, and the number of terminal leaves* t_n *is such that* $a_n \leq n - 2$, $a_n/n < 1 - \kappa$ *for a fixed* $\kappa > 0$, $\lim\limits_{n\to\infty} a_n = \infty$, $\lim\limits_{n\to\infty} t_n = \infty$, *and* $\lim\limits_{n\to\infty} 2^{t_n} \frac{(\log(a_n))^9}{a_n} = 0$.

Assumptions

(A4): The number of trees and the number of Monte-Carlo sampling grows with n

(A4)

The number of Monte-Carlo sampling K_n and the number of trees M_n grow with n, such that $M_n \longrightarrow \infty$ and $n.M_n/K_n \longrightarrow 0$.

SHAFF consistency

Theorem

*If Assumptions (A1), (A2), (A3), and (A4) are satisfied, then **SHAFF** is consistent, that is*

$$\hat{\mathrm{Sh}}_{M_n,n} \xrightarrow{p} \mathrm{Sh}^{\star}.$$

- valid when inputs are dependent
- most other Shapley algorithms are inconsistent (except brute force approaches)

Proof

Lemma

If Assumptions (A2) and (A3) are satisfied, for all $U \subset \{1, \ldots, p\}$, we have

$$\mathbb{P}\big(\hat{p}_{M_n,n}(U) > 0\big) \longrightarrow 1.$$

Lemma

If Assumptions (A1) and (A2) are satisfied, the PRF is consistent, that is, for all $M \in \mathbb{N}^\star$ and $U \subset \{1, \ldots, p\}$,

$$\hat{v}_{M,n}(U) \xrightarrow{p} \mathbb{V}[\mathbb{E}[Y|\boldsymbol{X}^{(U)}]]/\mathbb{V}[Y] \stackrel{\text{def}}{=} v^\star(U).$$

Proof

Lemma

If Assumptions (A1), (A2), and (A3) are satisfied, we have

$$\ell_{M,n}(\beta) \xrightarrow{p} \mathbb{E}[(v^\star(Z) - \beta^T I(Z))^2] \stackrel{\text{def}}{=} \ell^\star(\beta),$$

where Z is a discrete random variable such that

- $Z \subset \{1, \ldots, p\}$
- *for* $U \subset \{1, \ldots, p\}, \mathbb{P}(Z = U) = w(U)$.

Competitors

1. Williamson and Feng (2020)
    - Monte-Carlo sample of the variable subsets U
    - brute force retraining of the forest for each U
2. SAGE (Covert et al., 2020)
    - Monte-Carlo sample of the variable subsets U (using permutations)
    - sample from conditional distributions assuming variable independence
    - only use forest predictions to estimate $\mathbb{V}[\mathbb{E}[Y|\mathbf{X}^{(U)}]]$

Experiment 1: a linear case

- correlated centered Gaussian input vector of dimension 11
- linear model: $Y = \beta^T \mathbf{X} + \varepsilon$
- $\mathbb{V}[\varepsilon] = 0.05 \times \mathbb{V}[Y]$
- $X^{(2)}$ are appended to the data as $X^{(12)}$ and $X^{(13)}$
- two dummy Gaussian variables $X^{(14)}$ and $X^{(15)}$ are also added.

Experiment 1: a linear case

- correlated centered Gaussian input vector of dimension 11
- linear model: $Y = \beta^T \mathbf{X} + \varepsilon$
- $\mathbb{V}[\varepsilon] = 0.05 \times \mathbb{V}[Y]$
- $X^{(2)}$ are appended to the data as $X^{(12)}$ and $X^{(13)}$
- two dummy Gaussian variables $X^{(14)}$ and $X^{(15)}$ are also added.

Algorithm	Experiment 1
SHAFF	0.25
Williamson	0.64
SAGE	0.33

Table: Cumulative Absolute Error of SHAFF versus State-of-the-art Shapley Algorithms.

Experiment 1: a linear case

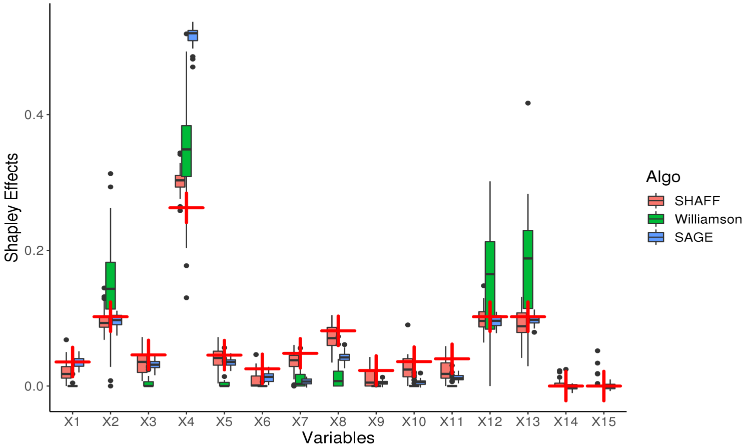


Figure: Shapley effects for a linear case. Red crosses are the theoretical Shapley effects.

Experiment 1b: high dimension

Extension to $p = 100$ with noisy variables.

Algorithm	Experiment 1a	Experiment 1b
SHAFF	0.25	0.80
Williamson	0.64	1.17
SAGE	0.33	1.16

Table: Cumulative Absolute Error of SHAFF versus State-of-the-art Shapley Algorithms.

Experiement 2: high-order interactions.

- Correlated centered Gaussian input vector of dimension 10
- 5 noisy Gaussian variables are also added
- $\mathbb{V}[\varepsilon] = 0.05 \times \mathbb{V}[Y]$

$$
\begin{aligned}
Y =& 3\sqrt{3} \times X^{(1)} X^{(2)} \mathbb{1}_{X^{(3)}>0} + \sqrt{3} \times X^{(4)} X^{(5)} \mathbb{1}_{X^{(3)}<0} \\
&+ 3 \times X^{(6)} X^{(7)} \mathbb{1}_{X^{(8)}>0} + X^{(9)} X^{(10)} \mathbb{1}_{X^{(8)}<0} + \varepsilon,
\end{aligned}
$$

Experiement 2: high-order interactions.

- Correlated centered Gaussian input vector of dimension 10
- 5 noisy Gaussian variables are also added
- $\mathbb{V}[\varepsilon] = 0.05 \times \mathbb{V}[Y]$

$$
\begin{aligned}
Y =& 3\sqrt{3} \times X^{(1)} X^{(2)} \mathbb{1}_{X^{(3)}>0} + \sqrt{3} \times X^{(4)} X^{(5)} \mathbb{1}_{X^{(3)}<0} \\
&+ 3 \times X^{(6)} X^{(7)} \mathbb{1}_{X^{(8)}>0} + X^{(9)} X^{(10)} \mathbb{1}_{X^{(8)}<0} + \varepsilon,
\end{aligned}
$$

Algorithm	Experiment 2
SHAFF	0.15
Williamson	0.24
SAGE	0.18

Table: Cumulative Absolute Error of SHAFF versus State-of-the-art Shapley Algorithms.

Conclusion

- SHAFF: consistent Shapley effect estimate for random forests
- Bénard, C., Biau, G., Da Veiga, S., & Scornet, E. (2022, May). SHAFF: Fast and consistent SHApley eFfect estimates via random Forests. In International Conference on Artificial Intelligence and Statistics (pp. 5563-5582). PMLR.
- R/C++ package `shaff`
  (available online: `https://gitlab.com/drti/shaff`)

Questions ?

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

I. Covert, S. Lundberg, and S.-I. Lee. Understanding global feature contributions through additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics New York, 2001.

F. Lopez. Shap: Shapley additive explanations, 2021. URL `https://towardsdatascience.com/shap-shapley-additive-explanations-5a2a271ed9c3`.

S.M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, New York, 2017.

A.B. Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2:245–251, 2014.

L.S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2:307–317, 1953.

B. Williamson and J. Feng. Efficient nonparametric statistical inference on population feature importance using shapley values. In *International Conference on Machine Learning*, pages 10282–10291. PMLR, 2020.