The Journal of Open Source Software

shapr: An R package for explaining machine learning models with dependence-aware Shapley values

Nikolai Sellereite[1] **and Martin Jullum**[1]

1 Norwegian Computing Center

DOI: XX.XXXXX/joss.XXXXX

Software

- Review
- Repository
- Archive

Submitted:
Published:

Summary

A common task within machine learning is to train a model which is able to predict an unknown outcome (response variable) based on a set of known input variables/features. When using such models for real life applications, it is often crucial to understand why a certain set of features lead to exactly a specific prediction. Most machine learning models are however so complicated and hard to understand that they are often viewed as "black-boxes" producing output when provided some input.

Shapley values (Shapley (1953)) is a concept from cooperative game theory used to fairly distribute a joint payoff among the cooperating players. Štrumbelj & Kononenko (2010) and later Lundberg & Lee (2017) proposed to use the Shapley value framework to explain predictions by distributing the prediction value on the input features. Unfortunately, established methods and implementations for explaining predictions with Shapley values like Shapley Sampling Values (Štrumbelj & Kononenko (2014)), SHAP/Kernel SHAP (Lundberg & Lee (2017)), and to some extent TreeSHAP (Lundberg, Erion, & Lee (2018)), assume that the features are independent when approximating the Shapley values for prediction explanation. This `R`-package implements methodology proposed by Aas, Jullum, & Løland (2019) to explain predictions by accounting for the dependence between the features, resulting in significantly more accurate approximations to the Shapley values.

Implementation

The package implements a variant of the Kernel SHAP (Lundberg & Lee (2017)) methodology for efficiently dealing with the combinatorial problem related to Shapley value formula. The main methodological contribution of Aas et al. (2019) is three different methods to estimate certain conditional expectation quantities, referred to as the `empirical`, `gaussian` and `copula` approach. Additionaly, the user has the option of combining the three approaches. The implementation supports explanation of the following models natively: `stats::lm`, `stats::glm`, `ranger::ranger`, `mgcv::gam` and `xgboost::xgboost`. Moreover, the package supports explanation of custom models by supplying two simple additional class functions.

The user interface in the package has largely been adopted from the `R`-package `lime` (Pedersen & Benesty (2019)). The user first sets up the explainability framework for the model to explain with the `shapr` function. Then the output from `shapr` is provided to the `explain` function, along with the data to explain the prediction for and which method should be used to estimate the aforementioned conditional expectations.

The majority of the code is plain `R`, while the most time consuming parts of the code has been coded in `C++` through the `Rcpp` package for computational speed up. Our implementation has some minor methodological improvements to the Kernel SHAP methodology as implemented in the `Python` package `shap` (Lundberg (2019)). In addition to our package's ability to account for the feature dependence (which the `shap` package does not), basic tests suggest our implementation is about 3-4 times faster.

For more information about the underlying methodology of the package, we refer to the paper (Aas et al. (2019)). For getting started with the package, we refer to the package vignette and introductory examples available at the package's pkgdown site.

Acknowledgement

This work was supported by the Norwegian Research Council grant 237718 (Big Insight).

References

Aas, K., Jullum, M., & Løland, A. (2019). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*.

Lundberg, S. (2019). *Shap: A unified approach to explain the output of any machine learning model*. Retrieved from https://pypi.org/project/shap/

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (pp. 4765–4774).

Pedersen, T. L., & Benesty, M. (2019). *Lime: Local interpretable model-agnostic explanations*. Retrieved from https://CRAN.R-project.org/package=lime

Shapley, L. S. (1953). A Value for N-Person Games. *Contributions to the Theory of Games*, *2*, 307–317.

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, *11*(Jan), 1–18.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, *41*(3), 647–665.