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 relies on the Kernel SHAP (Lundberg & Lee (2017)) methodology for efficiently dealing with combinatorial problems related to Shapley values.

Different methods (Gauss, copula nad empirical), user flexiboloty to choose method tailored for specific need, but with good default values.

Which models handled natively – support from custom models

Style adopted from the lime R package

Rcpp for speed up of some functions.

Faster than the KernelSHAP implemented in SHAP Python package.

Acknowledgement

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

Notes (do be deleted)

Mention and refer to the SHAP Python package

Manually transform this document to .md file when we are done to allow for automatic online compilation

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. 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).

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.