The Journal of Open Source Software

# Gala: A Python package for galactic dynamics

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

**1** Norwegian Computing Center

**DOI:** XX.XXXXX/joss.XXXXX

**Software**

- Review
- Repository
- Archive

**Submitted:**
**Published:**



## Summary

The most common task of 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 that prediction. However, explaining predictions from complex, or seemingly simple, machine learning models is a practical and ethical question, as well as a legal issue. Can I trust the model? Is it biased? Can I explain it to others? We want to explain individual predictions from a complex machine learning model by learning simple, interpretable explanations.

Shapley values is the only prediction explanation framework with a solid theoretical foundation (Lundberg & Lee (2017)). Unless the true distribution of the features are known, and there are less than say 10-15 features, these Shapley values needs to be estimated/approximated. Popular methods 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 may lead to very inaccurate Shapley values, and consequently wrong interpretations of the predictions. Aas, Jullum, & Løland (2019) extends and improves the Kernel SHAP method of Lundberg & Lee (2017) to account for the dependence between the features, resulting in significantly more accurate approximations to the Shapley values. See the paper for details.

## The Kernel SHAP Method

Assume a predictive model $f(\boldsymbol{x})$ for a response value $y$ with features $\boldsymbol{x} \in \mathbb{R}^M$, trained on a training set, and that we want to explain the predictions for new sets of features. This may be done using ideas from cooperative game theory, letting a single prediction take the place of the game being played and the features the place of the players. Letting $N$ denote the set of all $M$ players, and $S \subseteq N$ be a subset of $|S|$ players, the "contribution" function $v(S)$ describes the total expected sum of payoffs the members of $S$ can obtain by cooperation. The Shapley value (Shapley (1953)) is one way to distribute the total gains to the players, assuming that they all collaborate. The amount that player $i$ gets is then

$$\phi_i(v) = \phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!}(v(S \cup \{i\}) - v(S)),$$

that is, a weighted mean over all subsets $S$ of players not containing player $i$. Lundberg & Lee (2017) define the contribution function for a certain subset $S$ of these features $\boldsymbol{x}_S$ as $v(S) = \mathrm{E}[f(\boldsymbol{x})|\boldsymbol{x}_S]$, the expected output of the predictive model conditional on the feature values of the subset. Lundberg & Lee (2017) names this type of Shapley values

SHAP (SHapley Additive exPlanation) values. Since the conditional expectations can be written as

$$E[f(\boldsymbol{x})|\boldsymbol{x}_s = \boldsymbol{x}_S^*] = E[f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S)|\boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*)\, p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*) d\boldsymbol{x}_{\bar{S}},$$

the conditional distributions $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ are needed to compute the contributions. The Kernel SHAP method of Lundberg & Lee (2017) assumes feature independence, so that $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*) = p(\boldsymbol{x}_{\bar{S}})$. If samples $\boldsymbol{x}_{\bar{S}}^k, k = 1, \ldots, K$, from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ are available, the conditional expectation in above can be approximated by

$$v_{\text{KerSHAP}}(S) = \frac{1}{K}\sum_{k=1}^{K} f(\boldsymbol{x}_{\bar{S}}^k, \boldsymbol{x}_S^*).$$

In Kernel SHAP, $\boldsymbol{x}_{\bar{S}}^k, k = 1, \ldots, K$ are sampled from the $\bar{S}$-part of the training data, *independently* of $\boldsymbol{x}_S$. This is motivated by using the training set as the empirical distribution of $\boldsymbol{x}_{\bar{S}}$, and assuming that $\boldsymbol{x}_{\bar{S}}$ is independent of $\boldsymbol{x}_S = \boldsymbol{x}_S^*$. Due to the independence assumption, if the features in a given model are highly dependent, the Kernel SHAP method may give a completely wrong answer. This can be avoided by estimating the conditional distribution $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ directly and generating samples from this distribution. With this small change, the contributions and Shapley values may then be approximated as in the ordinary Kernel SHAP framework. Aas et al. (2019) propose three different approaches for estimating the conditional probabilities. The methods may also be combined, such that e.g. one method is used when conditioning on a small number of features, while another method is used otherwise.

## Multivariate Gaussian Distribution Approach

The first approach arises from the assumption that the feature vector $\boldsymbol{x}$ stems from a multivariate Gaussian distribution with some mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Under this assumption, the conditional distribution $p(\boldsymbol{x}_{\bar{\mathcal{S}}}|\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}}^*)$ is also multivariate Gaussian $\mathrm{N}_{|\bar{\mathcal{S}}|}(\boldsymbol{\mu}_{\bar{\mathcal{S}}|\mathcal{S}}, \boldsymbol{\Sigma}_{\bar{\mathcal{S}}|\mathcal{S}})$, with analytical expressions for the conditional mean vector $\boldsymbol{\mu}_{\bar{\mathcal{S}}|\mathcal{S}}$ and covariance matrix $\boldsymbol{\Sigma}_{\bar{\mathcal{S}}|\mathcal{S}}$, see Aas et al. (2019) for details. Hence, instead of sampling from the marginal empirical distribution of $\boldsymbol{x}_{\bar{S}}$ approximated by the training data, we can sample from the Gaussian conditional distribution, which is fitted using the training data. Using the resulting samples $\boldsymbol{x}_{\bar{S}}^k, k = 1, \ldots, K$, the conditional expectations be approximated as in the Kernel SHAP.

## Gaussian Copula Approach

If the features are far from multivariate Gaussian, an alternative approach is to instead represent the marginals by their empirical distributions, and model the dependence structure by a Gaussian copula. Assuming a Gaussian copula, we may convert the marginals of the training data to Gaussian features using their empirical distributions, and then fit a multivariate Gaussian distribution to these.

To produce samples from the conditional distribution $p(\boldsymbol{x}_{\bar{\mathcal{S}}}|\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}}^*)$, we convert the marginals of $\boldsymbol{x}_{\mathcal{S}}$ to Gaussians, sample from the conditional Gaussian distribution as above, and convert the marginals of the samples back to the original distribution. Those samples are then used to approximate the sample from the resulting multivariate Gaussian conditional distribution. While other copulas may be used, the Gaussian copula has the benefit that we may use the analytical expressions for the conditionals $\boldsymbol{\mu}_{\bar{\mathcal{S}}|\mathcal{S}}$ and $\boldsymbol{\Sigma}_{\bar{\mathcal{S}}|\mathcal{S}}$.

Finally, we may convert the marginals back to their original distribution, and use the resulting samples to approximate the conditional expectations as in the Kernel SHAP.

### Empirical Conditional Distribution Approach

If both the dependence structure and the marginal distributions of $\boldsymbol{x}$ are very far from the Gaussian, neither of the two aforementioned methods will work very well. Few methods exists for the non-parametric estimation of conditional densities, and the classic kernel estimator (Rosenblatt (1956)) for non-parametric density estimation suffers greatly from the curse of dimensionality and does not provide a way to generate samples from the estimated distribution. For such situations, Aas et al. (2019) propose an empirical conditional approach to sample approximately from $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}^*_S)$. The idea is to compute weights $w_{\mathcal{S}}(\boldsymbol{x}^*, \boldsymbol{x}^i),\ i = 1, ..., n_{\text{train}}$ for all training instances based on their Mahalanobis distances (in the $S$ subset only) to the instance $\boldsymbol{x}^*$ to be explained. Instead of sampling from this weighted (conditional) empirical distribution, Aas et al. (2019) suggests a more efficient variant, using only the $K$ instances with the largest weights:

$$v_{\text{condKerSHAP}}(\mathcal{S}) = \frac{\sum_{k=1}^{K} w_{\mathcal{S}}(\boldsymbol{x}^*, \boldsymbol{x}^{[k]}) f(\boldsymbol{x}^{[k]}_{\mathcal{S}}, \boldsymbol{x}^*_{\mathcal{S}})}{\sum_{k=1}^{K} w_{\mathcal{S}}(\boldsymbol{x}^*, \boldsymbol{x}^{[k]})},$$

The number of samples $K$ to be used in the approximate prediction can for instance be chosen such that the $K$ largest weights accounts for a fraction $\eta$, for example 0.9, of the total weight. If $K$ exceeds a certain limit, for instance $5,000$, it might be set to that limit. A bandwidth parameter $\sigma$ used to scale the weights, must also be specified. This choice may be viewed as a bias-variance trade-off. A small $\sigma$ puts most of the weight to a few of the closest training observations and thereby gives low bias, but high variance. When $\sigma \to \infty$, this method converges to the original Kernel SHAP assuming feature independence. Typically, when the features are highly dependent, a small $\sigma$ is typically needed such that the bias does not dominate. Aas et al. (2019) show that a proper criterion for selecting $\sigma$ is a small-sample-size corrected version of the AIC known as AICc. As calculation of it is computationally intensive, an approximate version of the selection criterion is also suggested. Details on this is found in Aas et al. (2019).

## 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).

Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, *27*, 832–837.

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

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