Package 'kernelshap'

August 12, 2022

Title Kernel SHAP

Version 0.1.0

Description Implementation of the model-agnostic Kernel SHAP algorithm by Ian Covert and Su-In Lee (2021) <http://proceedings.mlr.press/v130/covert21a>. Due to its iterative nature, standard errors of the SHAP values are provided and convergence is monitored. The package allows to work with any model that provides numeric predictions. Examples include linear regression, logistic regression (logit or probability scale), other generalized linear models, generalized additive models, and neural networks. The package plays well together with meta-learning packages like 'caret' or 'mlr3'. Visualizations can be done using the R package 'shapviz'.

License GPL (>= 2)

Depends R (>= 3.2.0)

Encoding UTF-8

RoxygenNote 7.1.2

Imports stats, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Michael Mayer [aut, cre]

Maintainer Michael Mayer <mayermichael79@gmail.com>

Repository CRAN

Date/Publication 2022-08-12 12:20:06 UTC

R topics documented:

`extractors` *Extractor Functions*

Description

Functions to extract SHAP values, feature values, standard errors etc. from a "kernelshap" object.

Usage

```
ks_shap_values(object, ...)

## S3 method for class 'kernelshap'
ks_shap_values(object, ...)

## Default S3 method:
ks_shap_values(object, ...)

ks_feature_values(object, ...)

## S3 method for class 'kernelshap'
ks_feature_values(object, ...)

## Default S3 method:
ks_feature_values(object, ...)

ks_baseline(object, ...)

## S3 method for class 'kernelshap'
ks_baseline(object, ...)

## Default S3 method:
ks_baseline(object, ...)

ks_standard_errors(object, ...)

## S3 method for class 'kernelshap'
ks_standard_errors(object, ...)

## Default S3 method:
ks_standard_errors(object, ...)

ks_n_iter(object, ...)

## S3 method for class 'kernelshap'
ks_n_iter(object, ...)
```

```
## Default S3 method:
ks_n_iter(object, ...)

ks_converged(object, ...)

## S3 method for class 'kernelshap'
ks_converged(object, ...)

## Default S3 method:
ks_converged(object, ...)
```

Arguments

`object`	Object to extract something.
`...`	Currently unused.

Value

The corresponding object is returned, i.e.,

- `ks_shap_values()` returns the matrix of SHAP values,
- `ks_feature_values()` the `data.frame` of feature values,
- `ks_baseline()` the numeric baseline value of the input,
- `ks_standard_errors()` the matrix of standard errors of SHAP values,
- `ks_converged()` returns the vector of convergence flags, and finally
- `ks_n_iter()` the number of iterations per row.

Examples

```
fit <- stats::lm(Sepal.Length ~ ., data = iris)
pred_fun <- function(X) stats::predict(fit, X)
s <- kernelshap(iris[1:2, -1], pred_fun = pred_fun, iris[-1])
ks_shap_values(s)
```

`is.kernelshap` *Check for kernelshap*

Description

Is object of class "kernelshap"?

Usage

```
is.kernelshap(object)
```

Arguments

object	An R object.

Value

Returns TRUE if object has "kernelshap" among its classes, and FALSE otherwise.

Examples

```
fit <- stats::lm(Sepal.Length ~ ., data = iris)
pred_fun <- function(X) stats::predict(fit, X)
s <- kernelshap(iris[1:2, -1], pred_fun = pred_fun, iris[-1])
is.kernelshap(s)
is.kernelshap("a")
```

---

kernelshap *Kernel SHAP*

Description

This function implements the model-agnostic Kernel SHAP algorithm explained in detail in Covert and Lee (2021). It is an iterative refinement of the original Kernel SHAP algorithm of Lundberg and Lee (2017). The algorithm is applied to each row in X. Due to its iterative nature, approximate standard errors of the resulting SHAP values are provided, and convergence is monitored. The data rows X to be explained and the background data bg_X should only contain feature columns required by the prediction function pred_fun. The latter is a function taking a data structure like X and bg_X and providing one numeric prediction per row. During each iteration, m subsets are evaluated until the worst standard error of the SHAP values is small enough relative to the range of the SHAP values. This exactly follows the logic used by Covert and Lee (2021).

Usage

```
kernelshap(
  X,
  pred_fun,
  bg_X,
  bg_w = NULL,
  paired_sampling = TRUE,
  m = "auto",
  tol = 0.01,
  max_iter = 250,
  verbose = TRUE,
  ...
)
```

Arguments

X	Matrix or data.frame containing the observations to be explained. Should only contain model features.
pred_fun	A function taking objects like X or bg_X as input and providing numeric predictions. Example: If "fit" denotes a logistic regression fitted via stats::glm, and SHAP values should be on the probability scale, then this argument is function(X) predict(fit, X, type = "response").
bg_X	Matrix or data.frame used as background data to calculate marginal expectations. Its column structure must be similar to X. This data should neither be too small nor too large (50-200 rows). A large background data slows down the calculations, while a small data set leads to imprecise SHAP values.
bg_w	Optional vector of case weights for each row of bg_X.
paired_sampling	Logical flag indicating whether to use paired sampling. The default is TRUE. This means that with every feature subset S, also its complement is evaluated, which leads to faster convergence.
m	Number of feature subsets S to be evaluated during one iteration. The default, "auto", equals trunc(20 * sqrt(ncol(X))). For the paired sampling strategy, 2m evaluations are done per iteration.
tol	Tolerance determining when to stop. The algorithm keeps iterating until max(sigma_n) / diff(range(beta_n)) < tol, where sigma_n are the standard errors and beta_n are the SHAP values of a given observation.
max_iter	If the stopping criterion (see tol) is not reached after max_iter iterations, then the algorithm stops.
verbose	Set to FALSE to suppress messages, warnings, and the progress bar.
...	Currently unused.

Value

An object of class "kernelshap" with the following components:

- S: Matrix with SHAP values.
- X: Same as parameter X.
- baseline: The average prediction on the background data.
- SE: Standard errors corresponding to S.
- n_iter: Number of iterations until convergence per row.
- converged: Logical vector indicating convergence per row.

References

1. Ian Covert and Su-In Lee. Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression. Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, PMLR 130:3457-3465, 2021.
2. Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems 30, 2017.

Examples

```
fit <- stats::lm(Sepal.Length ~ ., data = iris)
pred_fun <- function(X) stats::predict(fit, X)
s <- kernelshap(iris[1:2, -1], pred_fun = pred_fun, iris[-1])
s

# Matrix input works as well, and pred_fun may contain preprocessing steps.
fit <- stats::lm(Sepal.Length ~ ., data = iris[1:4])
pred_fun <- function(X) stats::predict(fit, as.data.frame(X))
X <- data.matrix(iris[2:4])
s <- kernelshap(X[1:3, ], pred_fun = pred_fun, X)
s
```

---

print.kernelshap *Prints "kernelshap" Object*

Description

Prints "kernelshap" Object

Usage

```
## S3 method for class 'kernelshap'
print(x, n = 2L, ...)
```

Arguments

x	An object of class "kernelshap".
n	Maximum number of rows of SHAP values, standard errors and feature values to print.
...	Further arguments passed from other methods.

Value

Invisibly, the input is returned.

See Also

kernelshap.

Examples

```
fit <- stats::lm(Sepal.Length ~ ., data = iris)
pred_fun <- function(X) stats::predict(fit, X)
s <- kernelshap(iris[1:3, -1], pred_fun = pred_fun, iris[-1])
s
```

Index