Package 'rollshap'

May 21, 2026

Type Package

Title Rolling Shapley Values

Version 1.0.1

Description

Analytical computation of rolling and expanding Shapley values for time-series data. The 'rollshap' package decomposes the coefficient of determination (R-squared) of a linear regression into nonnegative contributions from each explanatory variable using the Shapley value from cooperative game theory (Shapley, 1953, <doi:10.1515/9781400881970-018>). For each window, the exact Shapley value is computed by fitting all subsets of the explanatory variables and averaging the marginal contribution to R-squared across all orderings, which returns an order-invariant attribution that sums to the full-model R-squared. Use cases include variable importance, factor attribution, and feature selection in time-series regression. The package supports rolling and expanding windows, weights, and handling of missing values via 'min_obs', 'complete_obs', and 'na_restore' arguments. The implementation uses the online and offline algorithms from the 'roll' package to compute rolling and expanding cross-products efficiently with parallelism across columns and windows provided by 'RcppParallel'.

License GPL (>= 2)

URL https://github.com/jasonjfoster/rollshap

BugReports https://github.com/jasonjfoster/rollshap/issues

Depends R (>= 3.0.2)

Imports Rcpp, RcppParallel

LinkingTo Rcpp, RcppArmadillo, RcppParallel, roll (>= 1.1.7)

SystemRequirements GNU make

Encoding UTF-8

Suggests covr, testthat, zoo, relaimpo, roll

Config/roxygen2/old_usage TRUE

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Jason Foster [aut, cre]

Maintainer Jason Foster <jason.j.foster@gmail.com>

Repository CRAN

Date/Publication 2026-05-21 09:10:02 UTC

Contents

`rollshap-package` *Rolling Shapley Values*

Description

Analytical computation of rolling and expanding Shapley values for time-series data. The 'rollshap' package decomposes the coefficient of determination (R-squared) of a linear regression into nonnegative contributions from each explanatory variable using the Shapley value from cooperative game theory (Shapley, 1953, <doi:10.1515/9781400881970-018>). For each window, the exact Shapley value is computed by fitting all subsets of the explanatory variables and averaging the marginal contribution to R-squared across all orderings, which returns an order-invariant attribution that sums to the full-model R-squared. Use cases include variable importance, factor attribution, and feature selection in time-series regression. The package supports rolling and expanding windows, weights, and handling of missing values via 'min_obs', 'complete_obs', and 'na_restore' arguments. The implementation uses the online and offline algorithms from the 'roll' package to compute rolling and expanding cross-products efficiently with parallelism across columns and windows provided by 'RcppParallel'.

Details

`rollshap` is a package that provides analytical computation of rolling Shapley values for time-series data.

Author(s)

Jason Foster [aut, cre]

roll_shap *Rolling Shapley Values*

Description

A function for computing the rolling and expanding Shapley values of time-series data.

Usage

```
roll_shap(x, y, width, weights = rep(1, width), intercept = TRUE,
  min_obs = width, complete_obs = TRUE, na_restore = FALSE,
  online = TRUE)
```

Arguments

x	vector or matrix. Rows are observations and columns are the independent variables.
y	vector or matrix. Rows are observations and columns are the dependent variables.
width	integer. Window size.
weights	vector. Weights for each observation within a window.
intercept	logical. Either TRUE to include or FALSE to remove the intercept.
min_obs	integer. Minimum number of observations required to have a value within a window, otherwise result is NA.
complete_obs	logical. If TRUE then rows containing any missing values are removed, if FALSE then pairwise is used.
na_restore	logical. Should missing values be restored?
online	logical. Process observations using an online algorithm.

Value

An object of the same class and dimension as x with the rolling and expanding Shapley values.

Examples

```
n <- 15
m <- 3
x <- matrix(rnorm(n * m), nrow = n, ncol = m)
y <- rnorm(n)
weights <- 0.9 ^ (n:1)

# rolling Shapley values with complete windows
roll_shap(x, y, width = 5)

# rolling Shapley values with partial windows
roll_shap(x, y, width = 5, min_obs = 1)
```

```
# expanding Shapley values with partial windows
roll_shap(x, y, width = n, min_obs = 1)

# expanding Shapley values with partial windows and weights
roll_shap(x, y, width = n, min_obs = 1, weights = weights)
```

Index