Package 'CLIQUE'

September 9, 2026

Title Conditional Local Importance by Quantile Expectations

Version 1.0.0

Description Provides methods for interpretable machine learning with an emphasis on local feature importance estimation. The package implements the CLIQUE framework for computing observation-level importance values. Additional tools are included for visualizing multi-class partial dependence relationships across predictors and response classes.

License GPL-3

URL https://github.com/KelvynBladen/CLIQUE

Depends R (>= 4.1.0)

Imports caret, dplyr, fastDummies, future, future.apply, stats, tidyr, ggplot2, ggh4x, randomForest, rlang, pdp

Suggests testthat (>= 3.0.0), datasets

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Kelvyn Bladen [aut, cre] (ORCID: <https://orcid.org/0000-0002-2028-5504>), Adele Cutler [aut], D. Richard Cutler [aut], Kevin R. Moon [aut]

Maintainer Kelvyn Bladen <kelvyn.bladen@usu.edu>

Repository CRAN

Date/Publication 2026-09-09 12:40:23 UTC

Contents

`clique` *clique*

Description

Implements the CLIQUE method for local variable importance estimation. The procedure fits a predictive model using cross-validation and evaluates changes in prediction error under controlled perturbations of each feature, enabling observation-level assessment of feature importance.

Usage

```
clique(
  x,
  y,
  formula,
  data,
  method = "rf",
  tuneGrid = NULL,
  folds = 5,
  parallel = TRUE,
  cores = 5,
  seed = 123,
  nsim = 25,
  quantile_grid = TRUE,
  class_loss = FALSE,
  loss_function = function(truth, predictions) abs(truth - predictions)
)
```

Arguments

`x`	x is an object where samples are in rows and features are in columns. This could be a data frame or matrix with column names.
`y`	A numeric or factor vector containing the response feature.
`formula`	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.

data	A data frame containing the variables in the model. By default the variables are taken from environment(formula).
method	A character string specifying the model to be passed to caret::train. Valid options correspond to models registered in caret. See caret::modelLookup() for available methods.
tuneGrid	A data frame with tuning parameters for the specified model. See caret::train for model-specific tuning structures.
folds	Number of folds used for cross-validation. Default is 5.
parallel	Logical; if TRUE, models are developed in parallel. Default is TRUE.
cores	Number of CPU cores to use when parallel = TRUE. Default is 5.
seed	Random seed for reproducibility. Default is 123. Set to NULL to disable.
nsim	Number of grid points used to replace each variable. Default is 25.
quantile_grid	Logical; if TRUE, replacement points are chosen using quantiles of the variable. If FALSE, a uniform grid is used. Default is TRUE.
class_loss	Logical; if FALSE simply computes individual errors based on classification accuracy. If TRUE, averages the loss_function across class level probabilities when computing individual errors, serving as a loss generalized version of a Brier Score. Only applicable for classification problems. Default is FALSE.
loss_function	A function for computing the loss/error between model predictions and the response. Argument is necessary for regression or if class_loss = TRUE. Default is Absolute Error.

Value

A list containing:

bestTune	Best hyper-parameter combination selected by caret.
results	A data frame of performance metrics for all hyper-parameter combinations.
finalModel	A fitted model trained using the optimal tuning parameters.
local_imp	A data frame of CLIQUE values (local variable importance measures).

See Also

clique_eval

Examples

```
v <- clique(x = iris[1:4], y = iris$Species,
            method = "rf", cores = 2)
v$local_imp
```

clique_eval *clique_eval*

Description

Computes CLIQUE local variable importance values using a pre-trained predictive model. The procedure evaluates changes in prediction error under controlled perturbations of each feature and can be applied to either the training data or an external test dataset. For model assessment and interpretation of generalization behavior, evaluation on an independent test dataset is recommended, as importance values computed on training observations may reflect overfitting or optimistic prediction performance. Consider using `clique` to evaluate training data through cross-validation.

Usage

```
clique_eval(
  model,
  x_train,
  y_train,
  x_test,
  y_test,
  formula,
  data_train,
  data_test,
  seed = 123,
  nsim = 25,
  quantile_grid = TRUE,
  class_loss = FALSE,
  loss_function = function(truth, predictions) abs(truth - predictions)
)
```

Arguments

`model`	A fitted predictive model with a compatible `predict()` method. Models generated by `caret::train` are recommended to ensure support.
`x_train`	A data frame object where samples are in rows and features are in columns. Used to construct the reference distribution for feature perturbations.
`y_train`	A numeric or factor vector containing the response feature corresponding to `x_train`.
`x_test`	Optional test data where samples are in rows and features are in columns. If omitted, CLIQUE values are computed for `x_train`.
`y_test`	Optional response vector corresponding to `x_test`. Must be provided for `x_test` to be evaluated. If omitted, `y_train` is used to evaluate `x_train`.
`formula`	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the variables used by the model.
`data_train`	A data frame containing the training variables in the model.

data_test	Optional test data frame for evaluating. If omitted, data_train is used.
seed	Random seed for reproducibility. Default is 123. Set to NULL to disable.
nsim	Number of grid points used to replace each variable. Default is 25.
quantile_grid	Logical; if TRUE, replacement points are chosen using quantiles of the variable. If FALSE, a uniform grid is used. Default is TRUE.
class_loss	Logical; if FALSE simply computes individual errors based on classification accuracy. If TRUE, averages the loss_function across class level probabilities when computing individual errors, serving as a loss-generalized version of a Brier Score. Only applicable for classification problems. Default is FALSE.
loss_function	A function for computing the loss/error between model predictions and the response. Argument is necessary for regression or if class_loss = TRUE. Default is Absolute Error.

Value

A data frame of CLIQUE values (local variable importance measures), with one row per evaluated observation and one column per feature. Larger values indicate greater local importance of the corresponding feature for the observation.

Note

When available, users are encouraged to provide an independent test dataset through x_test and y_test. If explanation of training observations is desired, users are encouraged to do so via the cross-validation framework found in clique. Computing CLIQUE values on raw training observations may yield importance estimates that reflect patterns learned from the training data, whereas test-set or cross-validation evaluation better characterizes feature importance for out-of-sample predictions.

See Also

clique

Examples

```
set.seed(123)
train_ind <- sample(1:150, 120)
test_ind <- (1:150)[!(1:150 %in% train_ind)]

model <- caret::train(
  x = iris[train_ind, 1:4], y = iris$Species[train_ind],
  method = "rf"
)

## get training data local importance
v_train <- clique_eval(
  model = model,
  x_train = iris[train_ind, 1:4],
  y_train = iris$Species[train_ind]
)
```

```
head(v_train)

## get test data local importance
v_test <- clique_eval(
  model = model,
  x_train = iris[train_ind, 1:4],
  y_train = iris$Species[train_ind],
  x_test = iris[test_ind, 1:4],
  y_test = iris$Species[test_ind]
)

head(v_test)
```

`concrete` *Concrete Compressive Strength Data Set*

Description

Concrete strength is very important in civil engineering and is a highly nonlinear function of age and ingredients. This dataset contains 1030 instances and there are 8 features relevant to the response: concrete strength. The description of the variables are given below.

Name – Data Type – Measurement

Usage

```
concrete
```

Format

A data frame with 1030 rows, 8 covariate variables, and 1 response variable.

Details

Cement – quantitative – kg in a m3 mixture

Slag (Blast Furnace Slag) – quantitative – kg in a m3 mixture

FlyAsh – quantitative – kg in a m3 mixture

Water – quantitative – kg in a m3 mixture

Superplast (Superplasticizer) – quantitative – kg in a m3 mixture

CoarseAgg (Coarse Aggregate) – quantitative – kg in a m3 mixture

FineAgg (Fine Aggregate) – quantitative – kg in a m3 mixture

Age – quantitative – Day (1~365)

Strength (Concrete compressive strength) – quantitative – MPa

Source

https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength

References

-Cheng Yeh, "Modeling of strength of high performance concrete using artificial neural networks," Cement and Concrete Research, Vol. 28, No. 12, pp. 1797-1808 (1998).

---

lichen *Lichen data from the Current Vegetation Survey*

Description

Data were collected between 1993 and 1999 as part of the Lichen Air Quality surveys on public lands in Oregon and southern Washington. Observations were obtained from 1-acre (0.4 ha) plots at Current Vegetation Survey (CVS) sites. Indicator variables denote the presences and absences of 7 lichen species. Data for each sampled plot include the topographic variables elevation, aspect, and slope; bioclimatic predictors including maximum, minimum, daily, and average temperatures, relative humidity precipitation, evapotranspiration, and vapor pressure; and vegetation variables including the average age of the dominant conifer and percent conifer cover. The data in lichenTest were collected from half-acre plots at CVS sites in the same geographical region and contains many of the same variables, including presences and absences for the 7 lichen species. As such, it is a good test dataset for predictive methods applied to the Lichen Air Quality data.

Usage

```
lichen
```

Format

A data frame with 840 observations and 40 variables. One variable is a location identifier, 7 (coded as 0 and 1) identify the presence or absence of a type of lichen species, and 32 are characteristics of the survey site where the data were collected.

There were 12 monthly values in the original data for each of the bioclimatic predictors. Principal components analyses suggested that for each of these predictors 2 principal components explained the vast majority (95.0%-99.5%) of the total variability. Based on these analyses, indices were created for each set of bioclimatic predictors. The variables with the suffix Ave in the variable name are the average of 12 monthly variables. The variables with the suffix Diff are contrasts between the sum of the April-September monthly values and the sum of the October-December and January-March monthly values, divided by 12. Roughly speaking, these are summer-to-winter contrasts.

The variables are summarized as follows:

LobaOreg Lobaria oregana (Absent = 0, Present = 1)

EvapoTransAve Average monthly potential evapotranspiration in mm

EvapoTransDiff Summer-to-winter difference in monthly potential evapotranspiration in mm

MoistIndexAve Average monthly moisture index in cm

MoistIndexDiff Summer-to-winter difference in monthly monthly moisture index in cm

PrecipAve Average monthly precipitation in cm

PrecipDiff Summer-to-winter difference in monthly precipitation in cm

RelHumidAve Average monthly relative humidity in percent

RelHumidDiff Summer-to-winter difference in monthly relative humidity in percent

PotGlobRadAve Average monthly potential global radiation in kJ

PotGlobRadDiff Summer-to-winter difference in monthly potential global radiation in kJ

AveTempAve Average monthly average temperature in degrees Celsius

AveTempDiff Summer-to-winter difference in monthly average temperature in degrees Celsius

MaxTempAve Average monthly maximum temperature in degrees Celsius

MaxTempDiff Summer-to-winter difference in monthly maximum temperature in degrees Celsius

MinTempAve Average monthly minimum temperature in degrees Celsius

MinTempDiff Summer-to-winter difference in monthly minimum temperature in degrees Celsius

DayTempAve Mean average daytime temperature in degrees Celsius

DayTempDiff Summer-to-winter difference in average daytime temperature in degrees Celsius

AmbVapPressAve Average monthly average ambient vapor pressure in Pa

AmbVapPressDiff Summer-to-winter difference in monthly average ambient vapor pressure in Pa

SatVapPressAve Average monthly average saturated vapor pressure in Pa

SatVapPressDiff Summer-to-winter difference in monthly average saturated vapor pressure in Pa

Aspect Aspect in degrees

TransAspect Transformed Aspect: TransAspect=(1-cos(Aspect))/2

Elevation Elevation in meters

Slope Percent slope

ReserveStatus Reserve Status (Reserve, Matrix)

StandAgeClass Stand Age Class (< 80 years, 80+ years)

ACONIF Average age of the dominant conifer in years

PctVegCov Percent vegetation cover

PctConifCov Percent conifer cover

PctBroadLeafCov Percent broadleaf cover

TreeBiomass Live tree (> 1inch DBH) biomass, above ground, dry weight

Source

Cutler, D. Richard., Thomas C. Edwards Jr., Karen H. Beard, Adele Cutler, Kyle T. Hess, Jacob Gibson, and Joshua J. Lawler. 2007. Random Forests for Classification in Ecology. Ecology 88(11): 2783-2792.

https://CRAN.R-project.org/package=EZtune/

long_mnist *The MNIST Digit Dataset pivoted longer*

Description

This dataset comes from 1797 8x8 images. Each image is of a hand-written digit. In order to utilize an 8x8 figure like this, we first transformed it into a feature vector with length 64. We then pivoted the data into 5 columns: the digit label (y), pixel identity (name), pivel value (value), pixel horizontal location (x1), pixel vertical location (x2).

Usage

```
long_mnist
```

Format

A data frame with 115008 rows and 5 variables.

Source

https://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html
https://archive.ics.uci.edu/dataset/81/pen+based+recognition+of+handwritten+digits

mnist *The MNIST Digit Dataset*

Description

This dataset is made up of 1797 8x8 images. Each image is of a hand-written digit. In order to utilize an 8x8 figure like this, we have transformed it into a feature vector with length 64.

Usage

```
mnist
```

Format

A data frame with 1797 rows and 65 variables.

Source

https://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html
https://archive.ics.uci.edu/dataset/81/pen+based+recognition+of+handwritten+digits

pdp_multiclass *Multi-class Partial Dependence Plots for Predictors*

Description

Computes partial dependence plots (PDPs) for multi-class classification models across one or more predictor variables and returns individual and combined visualizations.

For each predictor variable, PDPs are computed for each class of the response using `pdp::partial()` with class-specific probability estimates. The function produces:

- Class-specific PDP plots (color-coded and faceted)
- Combined small-multiple plots (color-coded and faceted) across predictor variables
- A unified dataset containing all PDP values

Usage

```
pdp_multiclass(object, pred.data, response, pred.vars, prob = TRUE, ...)
```

Arguments

`object`	A fitted multi-class classification model supporting `pdp::partial()`.
`pred.data`	A data frame containing the training data used for PDP estimation. Must include both predictors and the response variable.
`response`	A character string specifying the name of the response variable in `pred.data`. Associated response must be a factor for multi-class classification.
`pred.vars`	A character vector specifying one or more predictor variables for which partial dependence should be computed.
`prob`	Logical indicating whether or not partial dependence for classification problems should be returned on the probability scale, rather than the centered logit. If FALSE, the partial dependence function is on a scale similar to the logit. Default is TRUE.
`...`	Additional arguments passed to [pdp::partial()].

Details

This function assumes that the provided model supports class probability estimation and that `pdp::partial()` can be applied with the `which.class` argument. The PDPs are computed independently for each class and predictor.

Value

A named list containing:

data A data frame containing all PDP values across predictors and classes

<var>_color ggplot object showing PDPs colored by class for predictor <var>

<var>_facet ggplot object showing PDPs faceted by class for predictor <var>

combo_color Combined PDPs colored by class and faceted by predictor variable for comparison

combo_facet Combined PDPs with class-by-predictor faceting for comparison

Examples

```
rf <- randomForest::randomForest(Species ~ ., data = datasets::iris,
  probability = TRUE)

pd <- pdp_multiclass(
  object = rf,
  pred.data = datasets::iris,
  response = "Species",
  pred.vars = c("Petal.Length", "Petal.Width", "Sepal.Length", "Sepal.Width")
)

pd$combo_color
pd$combo_facet
```

Index