Package 'dynfeature'

October 13, 2022

Type Package

Title Feature Importance for Dynamic Processes

Version 1.0.0

Description Calculating feature importance scores from trajectories using the random forests algorithm and more. Saelens and Cannoodt et al. (2019) <doi:10.1038/s41587-019-0071-9>.

License GPL-3

Encoding UTF-8

Imports dplyr, dynutils (>= 1.0.2), dynwrap (>= 1.0.0), purrr, magrittr, methods, ranger, reshape2, testthat, tidyr, tibble

Suggests caret, covr

RoxygenNote 7.1.1

NeedsCompilation no

Author Robrecht Cannoodt [aut, cre] (<https://orcid.org/0000-0003-3641-729X>, rcannood), Wouter Saelens [aut] (<https://orcid.org/0000-0002-7114-6248>, zouter)

Maintainer Robrecht Cannoodt <rcannood@gmail.com>

Repository CRAN

Date/Publication 2021-06-14 07:30:12 UTC

R topics documented:

calculate_branch_feature_importance
Calculating feature importances across trajectories

Description

Uses the feature importance measures of ranger or caret. calculate_overall_feature_importance calculates the importance for the whole trajectory, calculate_milestone_feature_importance calculates it for individual milestones (e.g. branching points)

Usage

```
calculate_branch_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_branching_point_feature_importance(
  trajectory,
  expression_source = "expression",
  milestones_oi = trajectory$milestone_ids,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_cell_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_milestone_feature_importance(
  trajectory,
  expression_source = "expression",
  milestones_oi = NULL,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)

calculate_overall_feature_importance(
  trajectory,
  expression_source = "expression",
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
```

```
)

calculate_waypoint_feature_importance(
  trajectory,
  expression_source = "expression",
  waypoints = NULL,
  fi_method = fi_ranger_rf_lite(),
  verbose = FALSE
)
```

Arguments

`trajectory`	A trajectory object containing expression values and a trajectory.
`expression_source`	The expression data matrix, with features as columns. • If a matrix is provided, it is used as is. • If a character is provided, `trajectory[[expression_source]]` should contain the matrix. • If a function is provided, that function will be called in order to obtain the expression (useful for lazy loading).
`fi_method`	A feature importance method. Default: `fi_ranger_rf_lite()`. Check `?fi_methods` for a full list of available feature importance methods.
`verbose`	Whether to print out extra information.
`milestones_oi`	The milestone(s) for which to calculate feature importance
`waypoints`	The waypoints, optional

Value

A data frame with two or more columns, `feature_id`, and `importance`. `feature_id` is a column in the trajectory expression matrix. Additional columns may be available depending on the function called.

Examples

```
library(dynwrap)
data(example_trajectory)

calculate_overall_feature_importance(example_trajectory)
```

`dynfeature` *Feature Importance for Dynamic Processes*

Description

Calculating feature importance scores from trajectories using the random forests algorithm.

`fi_ranger_rf_lite` *Feature Importance methods*

Description

Feature Importance methods

Usage

```
fi_ranger_rf_lite(
  num_trees = 2000,
  num_variables_per_split = 50,
  num_samples_per_tree = 250,
  min_node_size = 20,
  ...
)

fi_ranger_rf(...)

fi_caret(caret_method, ...)

fi_ranger_rf_tiny(
  num_trees = 100,
  num_variables_per_split = 50,
  num_samples_per_tree = 250,
  min_node_size = 20,
  ...
)
```

Arguments

`num_trees`	(fi_ranger_rf_lite) The number of trees to use
`num_variables_per_split`	(fi_ranger_rf_lite) The number of variables to sample per split
`num_samples_per_tree`	(fi_ranger_rf_lite) The number of samples to bootstrap per split
`min_node_size`	(fi_ranger_rf_lite) The minimum node size, no split will be made if the node size is less than this value.
`...`	Extra parameters to pass onto the underlying feature importance function.
`caret_method`	(fi_caret) Which caret method to use for feature importance.

Value

A list containing a helper function for calling a feature importance function.

Examples

```
library(dynwrap)
data(example_trajectory)

calculate_overall_feature_importance(example_trajectory, fi_method = fi_ranger_rf())
```

Index