Package ‘sgboost’

February 17, 2025

Title Sparse-Group Boosting

Version 0.2.0

Description

Sparse-group boosting to be used in conjunction with the 'mboost' for modeling grouped data. Applicable to all sparse-group lasso type problems where within-group and between-group sparsity is desired.
Interprets and visualizes individual variables and groups.

Imports dplyr, mboost, stringr, rlang, tibble, ggplot2, ggforce

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

URL https://github.com/FabianObster/sgboost

BugReports https://github.com/FabianObster/sgboost/issues

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Author Fabian Obster [aut, cre, cph] (<https://orcid.org/0000-0002-6951-9869>)

Maintainer Fabian Obster <fabian.obster@unibw.de>

Repository CRAN

Date/Publication 2025-02-17 20:00:02 UTC

Contents

balance *Balances selection frequencies for unequal groups*

Description

Returns optimal degrees of freedom for group boosting to achieve more balanced variables selection. Groups should be defined through `group_df`. Each base_learner

Usage

```
balance(
  df = NULL,
  group_df = NULL,
  blearner = "bols",
  outcome_name = "y",
  group_name = "group_name",
  var_name = "var_name",
  n_reps = 3000,
  iterations = 15,
  nu = 0.5,
  red_fact = 0.9,
  min_weights = 0.01,
  max_weights = 0.99,
  intercept = TRUE,
  verbose = F
)
```

Arguments

`df`	data.frame to be analyzed
`group_df`	input data.frame containing variable names with group structure. All variables in `df` to used in the analysis must be present in this data.frame.
`blearner`	Type of baselearner. Default is `'bols'`.
`outcome_name`	String indicating the name of dependent variable. Default is `"y"`
`group_name`	Name of column in group_df indicating the group structure of the variables. Default is `"group_name`.
`var_name`	Name of column in group_df containing the variable names to be used as predictors. Default is `"var_name"`. should not contain categorical variables with more than two categories, as they are then treated as a group only.

n_reps	Number of samples to be drawn in each iteration
iterations	Number of iterations performed in the algorithm. Default is "20"
nu	Learning rate as the step size to move away from the current estimate. Default is 0.5
red_fact	Factor by which the learning rate is reduced if the algorithm overshoots, meaning the loss increases. Default is 0.9
min_weights	The minimum weight size to be used. Default is 0.01
max_weights	The maximum weight size to be used. Default is 0.99
intercept	Logical, should intercept be used?
verbose	Logical, should iteration be printed?

Value

Character containing the formula to be passed to `mboost::mboost()` yielding the sparse-group boosting for a given value mixing parameter `alpha`.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- create_formula(alpha = 0.3, group_df = group_df)
sgb_model <- mboost(formula = sgb_formula, data = df)
summary(sgb_model)
```

create_formula *Create a sparse-group boosting formula*

Description

Creates a `mboost` formula that allows to fit a sparse-group boosting model based on boosted Ridge Regression with mixing parameter `alpha`. The formula consists of a group baselearner part with degrees of freedom 1-`alpha` and individual baselearners with degrees of freedom `alpha`. Groups should be defined through `group_df`. The corresponding modeling data should not contain categorical variables with more than two categories, as they are then treated as a group only.

Usage

```
create_formula(
  alpha = 0.3,
  group_df = NULL,
  blearner = "bols",
  outcome_name = "y",
  group_name = "group_name",
  var_name = "var_name",
  group_weights = "group_weights",
  intercept = FALSE
)
```

Arguments

`alpha` Numeric mixing parameter. For alpha = 0 only group baselearners and for alpha = 1 only individual baselearners are defined.

`group_df` input data.frame containing variable names with group structure.

`blearner` Type of baselearner. Default is `'bols'`.

`outcome_name` String indicating the name of dependent variable. Default is `"y"`

`group_name` Name of column in group_df indicating the group structure of the variables. Default is `"group_name`.

`var_name` Name of column in group_df containing the variable names to be used as predictors. Default is `"var_name"`. should not contain categorical variables with more than two categories, as they are then treated as a group only.

`group_weights` Optional name of the column in group_df indication the group weights.

`intercept` Logical, should intercept be used?

Value

Character containing the formula to be passed to `mboost::mboost()` yielding the sparse-group boosting for a given value mixing parameter `alpha`.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- create_formula(alpha = 0.3, group_df = group_df)
sgb_model <- mboost(formula = sgb_formula, data = df)
summary(sgb_model)
```

get_coef *Aggregated and raw coefficients in a sparse group boosting model*

Description

Computes the aggregated coefficients from group and individual baselearners. Also returns the raw coefficients associated with each baselearner.

Usage

```
get_coef(sgb_model)
```

Arguments

sgb_model	Model of type mboost to compute the coefficients for.

Details

in a sparse group boosting models a variable in a dataset can be selected as an individual variable or as a group. Therefore there can be two associated effect sizes for the same variable. This function aggregates both and returns it in a data.frame.

Value

List of data.frames containing the a data.frame '$raw' with the variable and the raw (Regression) coefficients and the data.frame '$aggregated' with the aggregated (Regression) coefficients.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- create_formula(alpha = 0.3, group_df = group_df)
sgb_model <- mboost(formula = sgb_formula, data = df)
sgb_coef <- get_coef(sgb_model)
```

get_coef_path *Path of aggregated and raw coefficients in a sparse-group boosting model*

Description

Computes the aggregated coefficients from group and individual baselearners for each boosting iteration.

Usage

```
get_coef_path(sgb_model)
```

Arguments

`sgb_model`	Model of type `mboost` to compute the coefficient path for .

Details

in a sparse-group boosting models a variable in a dataset can be selected as an individual variable or as a group. Therefore there can be two associated effect sizes for the same variable. This function aggregates both and returns it in a data.frame for each boosting iteration

Value

List of data.frames containing the a data.frame `$raw` with the variable and the raw (Regression) coefficients and the data.frame `$aggregated` with the aggregated (Regression) coefficients.

See Also

`get_coef()`

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
```

```
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- create_formula(alpha = 0.3, group_df = group_df)
sgb_model <- mboost(formula = sgb_formula, data = df)
sgb_coef_path <- get_coef_path(sgb_model)
```

get_varimp *Variable importance of a sparse-group boosting model*

Description

Variable importance is computed as relative reduction of loss-function attributed to each predictor (groups and individual variables). Returns a list of two data.frames. The first contains the variable importance of a sparse-group model in a data.frame for each predictor. The second one contains the aggregated relative importance of all groups vs. individual variables.

Usage

```
get_varimp(sgb_model)
```

Arguments

`sgb_model`	Model of type `mboost` to compute the variable importance for.

Value

List of two data.frames. `$raw` contains the name of the variables, group structure and variable importance on both group and individual variable basis. `$group_importance` contains the the aggregated relative importance of all group baselearners and of all individual variables.

See Also

`mboost::varimp()` which this function uses.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- as.formula(create_formula(alpha = 0.3, group_df = group_df))
sgb_model <- mboost(formula = sgb_formula, data = df)
sgb_varimp <- get_varimp(sgb_model)
```

plot_effects *Visualizing a sparse-group boosting model*

Description

Radar or scatter/lineplot visualizing the effects sizes relative to the variable importance in a sparse-group boosting model. Works also for a regular mboost model.

Usage

```
plot_effects(
  sgb_model,
  plot_type = "radar",
  prop = 0,
  n_predictors = 30,
  max_char_length = 5,
  base_size = 8
)
```

Arguments

sgb_model Model of type mboost to be used.

plot_type String indicating the type of visualization to use. 'radar' refers to a radar plot using polar coordinates. Here the angle is relative to the cumulative relative importance of predictors and the radius is proportional to the effect size. "clock" does the same as "radar" but uses clock coordinates instead of polar coordinates. "scatter" uses the effect size as y-coordinate and the cumulative relative importance as x-axis in a classical Scatter plot.

prop Numeric value indicating the minimal importance a predictor/baselearner has to have to be plotted. Default value is zero, meaning all predictors are plotted. By increasing prop the number of plotted variables can be reduced. One can also use n_predictors for limiting the number of variables to be plotted directly.

`n_predictors` The maximum number of predictors to be plotted. Default is 30. Alternative to `prop`.

`max_char_length`
The maximum character length of a predictor to be printed. Default is 5. For long variable names one may adjust this number.

`base_size` The `base_size` argument to be passed to the `ggplot2` theme ggplot2::theme_classic to be used to control the overall size of the figure. Default value is 8.

Value

`ggplot2` object mapping the effect sizes and variable importance.

See Also

`get_coef()`, `get_varimp()` which this function uses.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- as.formula(create_formula(alpha = 0.3, group_df = group_df))
sgb_model <- mboost(formula = sgb_formula, data = df)
plot_effects(sgb_model)
```

`plot_path` *Coefficient path of a sparse-group boosting model*

Description

Shows how the effect sizes change throughout the boosting iterations in a sparse-group boosting model. Works also for a regular mboost models. Color indicates the selection of group or individual variables within a boosting iteration.

Usage

```
plot_path(sgb_model, max_char_length = 5, base_size = 8)
```

Arguments

`sgb_model` Model of type `mboost` to be used.

`max_char_length` The maximum character length of a predictor to be printed. Default is 5. For long variable names one may adjust this number.

`base_size` The `base_size` argument to be passed to the `ggplot2` theme ggplot2::theme_bw to be used to control the overall size of the figure. Default value is 8.

Value

`ggplot2` object mapping the effect sizes and variable importance.

See Also

`get_coef_path()` which this function uses.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- as.formula(create_formula(alpha = 0.4, group_df = group_df))
sgb_model <- mboost(formula = sgb_formula, data = df)
plot_path(sgb_model)
```

plot_varimp *Variable importance bar plot of a sparse group boosting model*

Description

Visualizes the variable importance of a sparse-group boosting model. Color indicates if a predictor is an individual variable or a group.

Usage

```
plot_varimp(
  sgb_model,
  prop = 0,
  n_predictors = 30,
  max_char_length = 15,
  base_size = 8
)
```

Arguments

sgb_model — Model of type mboost to plot the variable importance.

prop — Numeric value indicating the minimal importance a predictor/baselearner has to have. Default value is zero, meaning all predictors are plotted. By increasing prop the number of plotted variables can be reduced. One can also use 'n_predictors' for limiting the number of variables to be plotted directly.

n_predictors — The maximum number of predictors to be plotted. Default is 30. Alternative to 'prop'.

max_char_length — The maximum character length of a predictor to be printed. Default is 15. For larger groups or long variable names one may adjust this number to differentiate variables from groups.

base_size — The base_size argument to be passed to the ggplot2 theme ggplot2::theme_bw to be used to control the overall size of the figure. Default value is 8.

Details

Note that aggregated group and individual variable importance printed in the legend is based only on the plotted variables and not on all variables that were selected in the sparse-group boosting model.

Value

object of type ggplot2.

See Also

get_varimp which this function uses.

Examples

```
library(mboost)
library(dplyr)
set.seed(1)
df <- data.frame(
  x1 = rnorm(100), x2 = rnorm(100), x3 = rnorm(100),
  x4 = rnorm(100), x5 = runif(100)
)
df <- df %>%
  mutate_all(function(x) {
    as.numeric(scale(x))
  })
df$y <- df$x1 + df$x4 + df$x5
group_df <- data.frame(
  group_name = c(1, 1, 1, 2, 2),
  var_name = c("x1", "x2", "x3", "x4", "x5")
)

sgb_formula <- as.formula(create_formula(alpha = 0.3, group_df = group_df))
sgb_model <- mboost(formula = sgb_formula, data = df)
sgb_varimp <- plot_varimp(sgb_model)
```

Index