Package: SelectBoost.gamlss (via r-universe)

June 1, 2026

Type Package

Title Stability-Selection via Correlated Resampling for 'GAMLSS' Models

Version 0.2.2

Date 2025-11-20

Depends R (>= 4.1.0)

Imports gamlss, Rcpp, rlang, SelectBoost, stats, utils

Suggests doParallel, foreach, future, future.apply, gamlss.data, gamlss.dist, ggplot2, glmnet, grpreg, knitr, knockoff, MASS, microbenchmark, nlme, parallel, pkgdown, pscl, rmarkdown, SGL, testthat (>= 3.0.0)

LinkingTo Rcpp, RcppArmadillo

Author Frederic Bertrand [cre, aut] (<https://orcid.org/0000-0002-0837-8281>)

Maintainer Frederic Bertrand <frederic.bertrand@lecnam.net>

Description Extends the 'SelectBoost' approach to Generalized Additive Models for Location, Scale and Shape (GAMLSS). Implements bootstrap stability-selection across parameter-specific formulas (mu, sigma, nu, tau) via gamlss::stepGAIC(). Includes optional standardization of predictors and helper functions for corrected AIC calculation. More details can be found in Bertrand and Maumy (2024) <https://hal.science/hal-05352041> that highlights correlation-aware resampling to improve variable selection for GAMLSS and quantile regression when predictors are numerous and highly correlated.

License GPL-3

Encoding UTF-8

Classification/MSC 62H11, 62J12, 62J99

LazyData true

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://fbertran.github.io/SelectBoost.gamlss/,
https://github.com/fbertran/SelectBoost.gamlss

BugReports https://github.com/fbertran/SelectBoost.gamlss/issues

Config/testthat/edition 3

SystemRequirements C++17

Config/pak/sysreqs
libglpk-dev make libmagick++-dev gsfonts texlive libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository https://fbertran.r-universe.dev

Date/Publication 2025-12-03 12:37:38 UTC

RemoteUrl https://github.com/fbertran/selectboost.gamlss

RemoteRef HEAD

RemoteSha 219797ff08a77bb67ea8d70b306196f0db36d85d

Contents

`.family_defaults` *Reasonable defaults*

Description

Adjust as needed per family docs

Usage

```
.family_defaults()
```

Value

List of list of default values for parameters for each supported distribution.

`.family_tolerance` *Per-family numeric tolerance for equality checks*

Description

Per-family numeric tolerance for equality checks

Usage

```
.family_tolerance()
```

Value

List of numerical values, the default tolerance, for each supported distribution.

.gen_family *Try to generate values for a family*

Description

Try to generate values for a family

Usage

```
.gen_family(fam, n)
```

Arguments

fam	Character scalar naming the family. This should correspond to a distribution available in the gamlss.dist package, as used when constructing r<family> generator functions (for example, NO'' or LOGNO").
n	Positive integer giving the number of observations to generate for the requested family.

Value

If successful, a umerical vector of n values randomly generated for the requested family. If the generator is not available or fails, then returns NULL.

AICc_gamlss *AICc for a gamlss fit*

Description

AICc for a gamlss fit

Usage

```
AICc_gamlss(object)
```

Arguments

object	a 'gamlss' object

Value

numeric AICc value

`autoboost_gamlss` *AutoBoost for GAMLSS (SelectBoost-style)*

Description

Runs a c0 grid, picks the c0 that maximizes total confidence, and returns the corresponding sb_gamlss fit.

Usage

```
autoboost_gamlss(
  formula,
  data,
  family,
  mu_scope,
  sigma_scope = NULL,
  nu_scope = NULL,
  tau_scope = NULL,
  base_sigma = ~1,
  base_nu = ~1,
  base_tau = ~1,
  c0_grid = seq(0.1, 0.9, by = 0.1),
  B = 60,
  sample_fraction = 0.7,
  pi_thr = 0.6,
  k = 2,
  direction = c("both", "forward", "backward"),
  pre_standardize = FALSE,
  trace = TRUE,
  progress = TRUE,
  use_groups = TRUE,
  corr_func = "cor",
  group_fun = SelectBoost::group_func_2,
  ...
)
```

Arguments

`formula`	Base formula for the location μ parameter (in the main model call).
`data`	Data frame.
`family`	A `gamlss.dist` family object (e.g., `gamlss.dist::NO()`).
`mu_scope`	Formula of candidate terms for μ.
`sigma_scope, nu_scope, tau_scope`	Formulas of candidate terms for σ, ν, τ.
`base_sigma, base_nu, base_tau`	Optional base (always-included) formulas for σ, ν, τ.

c0_grid	Numeric vector of c0 values.
B	Number of bootstrap subsamples for stability selection.
sample_fraction	Fraction of rows per subsample (e.g., 0.7).
pi_thr	Selection proportion threshold to define "stable" terms (e.g., 0.6).
k	Penalty weight for stepwise GAIC when engine = "stepGAIC" (default 2).
direction	Stepwise direction for stepGAIC ("both", "forward", "backward").
pre_standardize	Logical; standardize numeric predictors before penalized fits.
trace	Logical; print progress messages.
progress	Logical; show a progress bar in sequential runs.
use_groups	Logical; treat SelectBoost correlation groups during resampling.
corr_func	Correlation function passed to SelectBoost::boost.compcorrs.
group_fun	Grouping function passed to SelectBoost::boost.findgroups.
...	Passed to underlying engines (e.g., to gamlss::gamlss, glmnet, etc.).

Value

A SelectBoost_gamlss_grid with summary plots/tables.

---

boys7482 *Anthropometric measurements for Dutch boys (1997 reference)*

Description

Cross-sectional anthropometric records for 7,482 Dutch boys aged 0 to 21 years that were used to construct the 1997 Dutch growth references. The dataset stores standard auxological indicators together with puberty and regional classification variables.

Format

A data frame with 7,482 rows and 9 variables:

age Decimal age in years ranging from birth to 21.

hgt Standing height in centimetres.

wgt Body weight recorded in kilograms.

bmi Body mass index.

hc Head circumference in centimetres.

gen Ordered factor describing genital Tanner stage (G1–G5).

phb Ordered factor indicating pubic hair stage (P1–P6).

tv Testicular volume in millilitres.

reg Region of residence (north, east, west, south, city).

Details

The table combines the complete cross-sectional sample of Dutch boys aged 0–21 years that formed the basis of the Dutch 1997 growth references. Tanner stage variables are stored as ordered factors, while the region indicator is a nominal factor.

Source

Fredriks, A. M., van Buuren, S., Burgmeijer, R. J., Meulmeester, J. F., Beuker, R. J., Brugman, E., Roede, M. J., Verloove-Vanhorick, S. P., & Wit, J. M. (2000). Continuing positive secular growth change in The Netherlands 1955-1997. *Pediatric Research*, 47, 316-323.

Fredriks, A. M., van Buuren, S., Wit, J. M., & Verloove-Vanhorick, S. P. (2000). Body index measurements in 1996-97 compared with 1980. *Archives of Disease in Childhood*, 82, 107-112.

References

Stef van Buuren (2012).

Examples

```
data(boys7482)
str(boys7482)
```

---

check_fast_vs_generic *Numerical check: fast vs generic deviance log-likelihood*

Description

Computes both `loglik_gamlss_newdata_fast()` and `loglik_gamlss_newdata()` and reports the absolute difference. Useful for sanity checks.

Usage

```
check_fast_vs_generic(fit, newdata, tol = 1e-08)
```

Arguments

`fit`	A `gamlss` fit.
`newdata`	Data frame to evaluate on.
`tol`	Tolerance for pass/fail (default 1e-8).

Value

A list with fields: `ll_fast`, `ll_generic`, `abs_diff`, `pass`.

confidence_functionals *Confidence functionals from a c0 grid*

Description

Summarize selection proportions across c0 (SelectBoost threshold) into single-number confidence scores per term/parameter.

Usage

```
confidence_functionals(
  x,
  pi_thr = NULL,
  q = c(0.5, 0.8, 0.9),
  weight_fun = NULL,
  conservative = FALSE,
  B = NULL,
  method = c("trapezoid", "step")
)
```

Arguments

x	An object from sb_gamlss_c0_grid().
pi_thr	Stability threshold; defaults to x$pi_thr.
q	Numeric vector of quantiles to compute (in 0..1).
weight_fun	Optional function w(c0) for weighted AUSC; default uniform.
conservative	If TRUE, use Wilson lower confidence bounds for proportions.
B	Number of bootstraps (if not inferable when conservative = TRUE).
method	Integration method: "trapezoid" (default) or "step".

Value

A data.frame with per-term summaries, classed as "sb_confidence".

confidence_table *Compute SelectBoost-like confidence table across c0*

Description

Compute SelectBoost-like confidence table across c0

Usage

```
confidence_table(grid, pi_thr = NULL)
```

Arguments

grid	an object returned by sb_gamlss_c0_grid
pi_thr	optional override of the threshold (defaults to grid$pi_thr)

Value

data.frame with term, parameter, conf_index (mean positive excess), cover (fraction of c0 with prop>=thr)

---

cv_deviance_sb *K-fold deviance for an sb_gamlss configuration*

Description

K-fold deviance for an sb_gamlss configuration

Usage

```
cv_deviance_sb(K, build_fit, data)
```

Arguments

K	folds
build_fit	function(...) that returns an sb_gamlss object
data	data.frame used inside build_fit

Value

numeric: mean deviance across folds (-2 * mean loglik)

effect_plot *One-variable effect plot from an sb_gamlss (or gamlss) fit*

Description

Varies one variable and holds others at typical values (median/mode) to plot the predicted parameter curve (default: mu). Uses ggplot2 if available, otherwise base.

Usage

```
effect_plot(fit, var, data, what = "mu", grid = 100)

## S3 method for class 'effect_plot_failure'
print(x, ...)
```

Arguments

fit	sb_gamlss object (or gamlss)
var	character, name of the variable to vary
data	original data.frame used to fit
what	which parameter to predict ("mu","sigma","nu","tau")
grid	number of grid points for numeric variable
x	object returned by `effect_plot()` when prediction fails
...	unused

Value

a ggplot object if ggplot2 present; otherwise draws base plot and returns NULL

Invisibly returns `x`.

fast_vs_generic_ll *Compare fast vs generic deviance log-likelihood evaluation*

Description

Compare fast vs generic deviance log-likelihood evaluation

Usage

```
fast_vs_generic_ll(fit, newdata, reps = 100L, unit = "us")
```

Arguments

fit	A gamlss fit.
newdata	Data frame to evaluate on.
reps	Number of repetitions (default 100).
unit	microbenchmark unit (default "us").

Value

A data.frame with method, median, and relative speed.

---

fastboost_gamlss *FastBoost for GAMLSS (lightweight stability selection)*

Description

A faster variant with fewer bootstraps and smaller subsamples.

Usage

```
fastboost_gamlss(
  formula,
  data,
  family,
  mu_scope,
  sigma_scope = NULL,
  nu_scope = NULL,
  tau_scope = NULL,
  base_sigma = ~1,
  base_nu = ~1,
  base_tau = ~1,
  B = 30,
  sample_fraction = 0.6,
  pi_thr = 0.6,
  k = 2,
  direction = c("both", "forward", "backward"),
  pre_standardize = FALSE,
  use_groups = TRUE,
  c0 = 0.5,
  trace = TRUE,
  corr_func = "cor",
  group_fun = SelectBoost::group_func_2,
  ...
)
```

Arguments

formula	Base formula for the location μ parameter (in the main model call).
data	Data frame.
family	A gamlss.dist family object (e.g., gamlss.dist::NO()).
mu_scope	Formula of candidate terms for μ.
sigma_scope, nu_scope, tau_scope	Formulas of candidate terms for σ, ν, τ.
base_sigma, base_nu, base_tau	Optional base (always-included) formulas for σ, ν, τ.
B	Number of bootstrap subsamples for stability selection.
sample_fraction	Fraction of rows per subsample (e.g., 0.7).
pi_thr	Selection proportion threshold to define "stable" terms (e.g., 0.6).
k	Penalty weight for stepwise GAIC when engine = "stepGAIC" (default 2).
direction	Stepwise direction for stepGAIC ("both", "forward", "backward").
pre_standardize	Logical; standardize numeric predictors before penalized fits.
use_groups	Logical; treat SelectBoost correlation groups during resampling.
c0	SelectBoost meta-parameter controlling reweighting/thresholding (see vignette).
trace	Logical; print progress messages.
corr_func	Correlation function passed to SelectBoost::boost.compcorrs.
group_fun	Grouping function passed to SelectBoost::boost.findgroups.
...	Passed to underlying engines (e.g., to gamlss::gamlss, glmnet, etc.).

Details

Fast SelectBoost (single c0)

Value

An sb_gamlss fit at the given c0.

---

get_density_fun *Get a density function for a gamlss family*

Description

Get a density function for a gamlss family

Usage

```
get_density_fun(fit)
```

Arguments

fit	a gamlss fit (or family name)

Value

function(x, mu, sigma, nu, tau, log=FALSE)

---

knockoff_filter_mu *Knockoff filter for mu (approximate group control)*

Description

Knockoff filter for mu (approximate group control)

Usage

```
knockoff_filter_mu(data, response, mu_scope, fdr = 0.1, df_smooth = 6L)
```

Arguments

data	data.frame
response	response variable name
mu_scope	RHS-only term labels
fdr	target FDR level
df_smooth	df for smoother proxies (splines::bs)

Value

character vector of selected term names

---

knockoff_filter_param *Knockoff filter for sigma/nu/tau (approximate group control)*

Description

Knockoff filter for sigma/nu/tau (approximate group control)

Usage

```
knockoff_filter_param(data, scope, y_work, fdr = 0.1, df_smooth = 6L)
```

Arguments

data	data.frame
scope	RHS-only term labels
y_work	working response (numeric)
fdr	target FDR level
df_smooth	df for smoother proxies

Value

character vector of selected term names

---

loglik_gamlss_newdata *Log-likelihood (sum) on newdata given a gamlss fit*

Description

Log-likelihood (sum) on newdata given a gamlss fit

Usage

```
loglik_gamlss_newdata(fit, newdata)
```

Arguments

fit	gamlss object
newdata	data.frame

Value

numeric scalar: sum of log-likelihoods

---

plot.sb_confidence *Plot confidence functionals*

Description

Two-panel plot: (1) scatter of area_pos vs cover (size by rank), (2) barplot of top-N rank_score.

Usage

```
## S3 method for class 'sb_confidence'
plot(x, top = 15, label_top = 10, ...)
```

Arguments

x	An object from confidence_functionals().
top	Show top-N terms in the barplot (default 15).
label_top	Integer; number of points to label in the scatter (default 10).
...	Graphical parameters passed to plotting backend.

Value

An invisible copy of x.

---

plot.SelectBoost_gamlss

Plot selection proportions for a single sb_gamlss

Description

Plot selection proportions for a single sb_gamlss

Usage

```
## S3 method for class 'SelectBoost_gamlss'
plot(x, ...)
```

Arguments

x	A sb_gamlss object.
...	Graphical parameters.

Value

Invisibly returns x the plotted sb_gamlss object.

---

plot.SelectBoost_gamlss_grid

Plot summary for sb_gamlss_c0_grid

Description

Plot summary for sb_gamlss_c0_grid

Usage

```
## S3 method for class 'SelectBoost_gamlss_grid'
plot(x, top = 15, ...)
```

Arguments

x	A SelectBoost_gamlss_grid object.
top	Integer; how many top terms to show in the confidence barplot.
...	Ignored (reserved for future).

Value

An invisible copy of x.

---

plot_sb_gamlss *Plot selection frequencies for sb_gamlss*

Description

Plot selection frequencies for sb_gamlss

Usage

```
plot_sb_gamlss(x, top = Inf, ...)
```

Arguments

x	A sb_gamlss object
top	Show only the top N terms per-parameter (default all)
...	Graphical parameters.

Value

Invisibly returns x the plotted sb_gamlss object.

---

plot_stability_curves *Plot stability curves p(c0) for selected terms*

Description

Plot stability curves p(c0) for selected terms

Usage

```
plot_stability_curves(grid, terms, parameter = NULL, ncol = 2L)
```

Arguments

grid	An object from sb_gamlss_c0_grid().
terms	Character vector of term names to plot.
parameter	Optional parameter name ('mu','sigma','nu','tau'); if NULL, all.
ncol	Columns in the multi-panel layout.

Value

Invisibly returns grid the plotted object.

---

predict_params *Predict distribution parameters on newdata*

Description

Predict distribution parameters on newdata

Usage

```
predict_params(fit, newdata)
```

Arguments

fit	a gamlss fit
newdata	data.frame

Value

list with available components: mu, sigma, nu, tau

---

sb_gamlss *SelectBoost for GAMLSS (stability selection)*

Description

SelectBoost for GAMLSS (stability selection)

Usage

```
sb_gamlss(
  formula,
  data,
  family,
  mu_scope,
  sigma_scope = NULL,
  nu_scope = NULL,
  tau_scope = NULL,
  base_sigma = ~1,
  base_nu = ~1,
  base_tau = ~1,
  B = 100,
  sample_fraction = 0.7,
  pi_thr = 0.6,
  k = 2,
  direction = c("both", "forward", "backward"),
  pre_standardize = FALSE,
  use_groups = FALSE,
  c0 = 0.5,
  engine = c("stepGAIC", "glmnet", "grpreg", "sgl"),
  engine_sigma = NULL,
  engine_nu = NULL,
  engine_tau = NULL,
  grpreg_penalty = c("grLasso", "grMCP", "grSCAD"),
  sgl_alpha = 0.95,
  df_smooth = 6L,
  progress = TRUE,
  glmnet_alpha = 1,
  glmnet_family = c("gaussian", "binomial", "poisson"),
  parallel = c("none", "auto", "multisession", "multicore"),
  workers = NULL,
  trace = TRUE,
  corr_func = "cor",
  group_fun = SelectBoost::group_func_2,
  ...
)
```

Arguments

`formula`	Base formula for the location μ parameter (in the main model call).
`data`	Data frame.
`family`	A `gamlss.dist` family object (e.g., `gamlss.dist::NO()`).
`mu_scope`	Formula of candidate terms for μ.
`sigma_scope, nu_scope, tau_scope`	Formulas of candidate terms for σ, ν, τ.

base_sigma, base_nu, base_tau
: Optional base (always-included) formulas for σ, ν, τ.

B
: Number of bootstrap subsamples for stability selection.

sample_fraction
: Fraction of rows per subsample (e.g., 0.7).

pi_thr
: Selection proportion threshold to define "stable" terms (e.g., 0.6).

k
: Penalty weight for stepwise GAIC when engine = "stepGAIC" (default 2).

direction
: Stepwise direction for stepGAIC ("both", "forward", "backward").

pre_standardize
: Logical; standardize numeric predictors before penalized fits.

use_groups
: Logical; treat SelectBoost correlation groups during resampling.

c0
: SelectBoost meta-parameter controlling reweighting/thresholding (see vignette).

engine
: Engine for μ ("stepGAIC", "glmnet", "grpreg", "sgl").

engine_sigma, engine_nu, engine_tau
: Optional engines for σ, ν, τ.

grpreg_penalty
: Group penalty for grpreg ("grLasso", "grMCP", "grSCAD").

sgl_alpha
: Alpha for sparse group lasso.

df_smooth
: Degrees of freedom for proxy spline bases (pb()/cs() $\rightarrow$ splines::bs(df=df_smooth)) used only for grouped selection design.

progress
: Logical; show a progress bar in sequential runs.

glmnet_alpha
: Elastic-net mixing for glmnet (1 = lasso, 0 = ridge).

glmnet_family
: Family passed to glmnet-based selectors ("gaussian", "binomial", "poisson").

parallel
: Parallel mode ("none", "auto", "multisession", '"multicore").

workers
: Integer; number of workers if parallel.

trace
: Logical; print progress messages.

corr_func
: Correlation function passed to SelectBoost::boost.compcorrs.

group_fun
: Grouping function passed to SelectBoost::boost.findgroups.

...
: Passed to underlying engines (e.g., to gamlss::gamlss, glmnet, etc.).

Value

An object of class "sb_gamlss" with elements:

- final_fit: the final gamlss object.
- final_formula: list of formulas for mu/sigma/nu/tau.
- selection: data.frame of selection counts and proportions.
- B, sample_fraction, pi_thr, k.
- scaler: list with center, scale, vars, response.

Examples

```
set.seed(1)
dat <- data.frame(
  y = gamlss.dist::rNO(60, mu = 0),
  x1 = rnorm(60),
  x2 = rnorm(60),
  x3 = rnorm(60)
)
fit <- sb_gamlss(
  y ~ 1,
  data = dat,
  family = gamlss.dist::NO(),
  mu_scope = ~ x1 + x2 + gamlss::pb(x3),
  B = 8,
  pi_thr = 0.6,
  trace = FALSE
)
fit$final_formula
```

---

sb_gamlss_c0_grid *Stability curves over a c0 grid for sb_gamlss*

Description

Stability curves over a c0 grid for sb_gamlss

Usage

```
sb_gamlss_c0_grid(
  formula,
  data,
  family,
  mu_scope,
  sigma_scope = NULL,
  nu_scope = NULL,
  tau_scope = NULL,
  base_sigma = ~1,
  base_nu = ~1,
  base_tau = ~1,
  c0_grid = seq(0.1, 0.9, by = 0.1),
  B = 60,
  sample_fraction = 0.7,
  pi_thr = 0.6,
  k = 2,
  direction = c("both", "forward", "backward"),
  pre_standardize = FALSE,
  trace = TRUE,
```

```
  progress = TRUE,
  use_groups = TRUE,
  corr_func = "cor",
  group_fun = SelectBoost::group_func_2,
  ...
)
```

Arguments

formula	Base formula for the location μ parameter (in the main model call).
data	Data frame.
family	A gamlss.dist family object (e.g., gamlss.dist::NO()).
mu_scope	Formula of candidate terms for μ.
sigma_scope, nu_scope, tau_scope	Formulas of candidate terms for σ, ν, τ.
base_sigma, base_nu, base_tau	Optional base (always-included) formulas for σ, ν, τ.
c0_grid	Numeric vector of c0 thresholds in (0,1).
B	Number of bootstrap subsamples for stability selection.
sample_fraction	Fraction of rows per subsample (e.g., 0.7).
pi_thr	Selection proportion threshold to define "stable" terms (e.g., 0.6).
k	Penalty weight for stepwise GAIC when engine = "stepGAIC" (default 2).
direction	Stepwise direction for stepGAIC ("both", "forward", "backward").
pre_standardize	Logical; standardize numeric predictors before penalized fits.
trace	Logical; print progress messages.
progress	Logical; show a progress bar across c0_grid.
use_groups	Logical; treat SelectBoost correlation groups during resampling.
corr_func	Correlation function passed to SelectBoost::boost.compcorrs.
group_fun	Grouping function passed to SelectBoost::boost.findgroups.
...	Passed to underlying engines (e.g., to gamlss::gamlss, glmnet, etc.).

Value

An object of class "SelectBoost_gamlss_grid" containing

- results: named list of sb_gamlss fits, names are c0 values
- table: data.frame with parameter, term, count, prop, c0
- pi_thr: the threshold used

`SelectBoost_gamlss` *SelectBoost-style wrapper for GAMLSS*

Description

A thin wrapper around `sb_gamlss()` with SelectBoost-flavored arguments.

Usage

```
SelectBoost_gamlss(
  formula,
  data,
  family,
  mu_scope,
  sigma_scope = NULL,
  nu_scope = NULL,
  tau_scope = NULL,
  base_sigma = ~1,
  base_nu = ~1,
  base_tau = ~1,
  B = 100,
  sample_fraction = 0.7,
  pi_thr = 0.6,
  k = 2,
  direction = c("both", "forward", "backward"),
  pre_standardize = FALSE,
  use_groups = TRUE,
  c0 = 0.5,
  trace = TRUE,
  ...
)

## S3 method for class 'SelectBoost_gamlss'
summary(object, prop.level = 0.6, ...)

## S3 method for class 'summary.SelectBoost_gamlss'
plot(x, ...)
```

Arguments

`formula`	Base formula for the location μ parameter (in the main model call).
`data`	Data frame.
`family`	A `gamlss.dist` family object (e.g., `gamlss.dist::NO()`).
`mu_scope`	Formula of candidate terms for μ.
`sigma_scope, nu_scope, tau_scope`	Formulas of candidate terms for σ, ν, τ.

base_sigma, base_nu, base_tau
: Optional base (always-included) formulas for σ, ν, τ.

B
: Number of bootstrap subsamples for stability selection.

sample_fraction
: Fraction of rows per subsample (e.g., 0.7).

pi_thr
: Selection proportion threshold to define "stable" terms (e.g., 0.6).

k
: Penalty weight for stepwise GAIC when engine = "stepGAIC" (default 2).

direction
: Stepwise direction for stepGAIC ("both", "forward", "backward").

pre_standardize
: Logical; standardize numeric predictors before penalized fits.

use_groups
: Logical; enable SelectBoost grouping.

c0
: Correlation threshold for grouping (as in SelectBoost::group_func_2).

trace
: Logical; print progress messages.

...
: Not used.

object
: A asb_gamlss object.

prop.level
: A target proportion level.

x
: A summary of asb_gamlss object.

Value

An object of class c("SelectBoost_gamlss"), with slots similar to sb_gamlss.

A list with selection, threshold and confidence.

Invisibly returns x.

---

selection_table *Selection table accessor*

Description

Selection table accessor

Usage

```
selection_table(x)
```

Arguments

x
: A sb_gamlss object

Value

data.frame with parameter, term, count, prop

`tune_sb_gamlss` *Tune select engines/penalties via a small stability run*

Description

Evaluates a grid of configurations and picks the one maximizing a stability-based score, optionally penalized by complexity. Designed to be lightweight and robust.

Usage

```
tune_sb_gamlss(
  config_grid,
  base_args,
  score_lambda = 0,
  B_small = 30,
  metric = c("stability", "deviance"),
  K = 3,
  progress = TRUE
)
```

Arguments

`config_grid`	a list of named lists, each containing a subset of sb_gamlss args (e.g., list(engine="grpreg", engine_sigma="sgl", grpreg_penalty="grLasso", sgl_alpha=0.9))
`base_args`	a named list of arguments passed to `sb_gamlss()` common to all configs
`score_lambda`	Numeric; complexity penalty weight for stability metric.
`B_small`	number of bootstraps to use during tuning (defaults to 30)
`metric`	Character; `"stability"` or `"deviance"` (K-fold CV).
`K`	Integer; folds for deviance CV.
`progress`	Logical; show progress bar across configs.

Value

a list: best_config, scores (data.frame), and the fitted sb_gamlss object for the best config.

Index