

Package ‘figsr’

September 1, 2026

Title Fast Interpretable Greedy-Tree Sums for Tree Ensembles

Version 0.1.1

Description Flexible, interpretable machine learning algorithm for additive tree sums (FIGS). Fits a sum of shallow classification and regression trees (CART) by greedily minimizing residual impurity, growing a new tree or deepening an existing one at each step, whichever reduces the residuals most. Supports regression and two-class classification, variable importance, bootstrap ensembling and seamless integration with 'parsnip' and 'tidymodels' workflows. The method is described in Tan et al. (2023) <[doi:10.1073/pnas.2310151122](https://doi.org/10.1073/pnas.2310151122)>.

License MIT + file LICENSE

URL <https://github.com/bonijoao/figsr>

BugReports <https://github.com/bonijoao/figsr/issues>

Depends R (>= 4.1)

Imports dials, graphics, parsnip, rlang, stats, tibble

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

NeedsCompilation no

Author João Paulo Assis Bonifácio [aut, cre, cph] (ORCID: <<https://orcid.org/0009-0009-2335-8391>>),
Geraldo Magela da Cruz Pereira [aut, cph] (ORCID: <<https://orcid.org/0000-0001-6280-4870>>),
Pedro Mambelli Fernandes [aut, cph] (ORCID: <<https://orcid.org/0009-0001-1695-279X>>),
João Vitor Andrade Alves de Souza [aut, cph] (ORCID: <<https://orcid.org/0009-0006-2780-318X>>)

Maintainer João Paulo Assis Bonifácio <jpab.27@hotmail.com>
Repository CRAN
Date/Publication 2026-08-31 22:30:16 UTC

Contents

bagging_figs	2
figs	3
figsr_importance	5
figs_tree	6
fit_figs	7
max_splits	8
max_trees	8
plot.figsr_fit	9
predict.bagging_figs_fit	10
predict.figsr_fit	10
predict_figs	11
print.figsr_fit	12
summary.figsr_fit	12
update.figs_tree	13
Index	14

bagging_figs	<i>Bagging-FIGS: Ensemble of FIGS Models</i>
--------------	--

Description

bagging_figs() fits an ensemble of FIGS models using bootstrap resampling to increase stability and predictive performance on noisy datasets.

Usage

```
bagging_figs(  
  formula,  
  data,  
  n_estimators = 10,  
  max_splits = 6,  
  min_n = 5,  
  mode = "regression",  
  ...  
)
```

Arguments

formula	A formula.
data	A data frame.
n_estimators	Integer. Number of bootstrap FIGS models to train. Default is 10.
max_splits	Integer. Maximum splits per individual FIGS model. Default is 6.
min_n	Integer. Minimum node size. Default is 5.
mode	Character. "regression" or "classification"; classification supports two-class outcomes only.
...	Additional arguments passed to <code>figs()</code> .

Value

An object of class `bagging_figs_fit`.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(50), x2 = rnorm(50), y = rnorm(50))
bag_fit <- bagging_figs(y ~ x1 + x2, data = df, n_estimators = 3)
```

figs

Fit Fast Interpretable Greedy-Tree Sums (FIGS)

Description

`figs()` fits a Fast Interpretable Greedy-Tree Sums model for regression or binary classification tasks.

Usage

```
figs(
  formula,
  data,
  max_splits = 10,
  max_trees = NULL,
  min_n = 5,
  mode = "regression",
  subset = NULL,
  na.action = stats::na.omit,
  ...
)
```

Arguments

<code>formula</code>	A formula specifying outcome and predictor variables.
<code>data</code>	A data frame containing training data.
<code>max_splits</code>	Integer. Maximum total number of splits across all trees in the sum. Default is 10.
<code>max_trees</code>	Integer. Maximum number of trees allowed in the sum. Default is NULL (unconstrained up to <code>max_splits</code>).
<code>min_n</code>	Integer. Minimum number of observations required in a node to split. Default is 5.
<code>mode</code>	Character. Either "regression" or "classification". Only two-class outcomes are supported in classification mode. Default is "regression".
<code>subset</code>	An optional vector selecting the rows of data to fit on, passed to <code>stats::model.frame()</code> .
<code>na.action</code>	A function describing what to do with missing values, passed to <code>stats::model.frame()</code> . Defaults to <code>stats::na.omit()</code> .
<code>...</code>	Additional arguments, currently ignored. Case weights are not supported and passing weights raises an error.

Details

Trees are grown one split at a time. At each step every leaf of every existing tree, plus the root of a possible new tree, competes for the split that most reduces the residual sum of squares; the single best candidate is taken and the residuals are recomputed against the whole sum. Growth stops at `max_splits` splits or when no candidate improves the fit.

Classification is limited to two classes. A factor outcome with more than two levels raises an error. The binary case is fitted on the 0/1 encoding of the outcome using the same sum-of-squares criterion, so the sum of the leaf values already estimates the probability of the second level; it is clamped to the unit interval at prediction time rather than passed through a link function.

Predictors are used exactly as supplied: numeric features are split at midpoints between sorted unique values, capped at 30 sample quantiles, and factor features are split by enumerating subsets of their levels, which is skipped above 10 levels. A factor level that was not seen in training raises an error at prediction time, as it does for the other model-frame based fitting functions in `stats`. Rows with a missing value in a predictor are dropped at fit time by `stats::model.frame()` and raise an error at prediction time.

Value

An object of class `figsr_fit` containing fitted tree structures, predictions, and metadata.

Examples

```
set.seed(123)
df <- data.frame(
  x1 = rnorm(100),
  x2 = rnorm(100),
  y = rnorm(100)
)
```

```
fit <- figs(y ~ x1 + x2, data = df, max_splits = 4)
print(fit)
```

figsr_importance	<i>Calculate Variable Importance for a figsr_fit Model</i>
------------------	--

Description

`figsr_importance()` computes the total reduction in residual impurity (Sum of Squares gain) attributable to each predictor feature across all trees in the fitted FIGS model.

Usage

```
figsr_importance(object, relative = TRUE)
```

Arguments

<code>object</code>	A fitted <code>figsr_fit</code> model object.
<code>relative</code>	Logical. If TRUE (default), the importance column is rescaled to percentages of the total gain; otherwise it repeats the raw gain.

Details

Each split node stores the reduction in residual sum of squares it achieved at the moment it was chosen. Importance for a feature is the sum of those gains over every split made on that feature, across all trees in the sum. Features never selected receive a gain of zero.

Value

A tibble with one row per predictor and columns feature, gain (raw sum-of-squares reduction) and importance, sorted by decreasing importance.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(50), x2 = rnorm(50))
df$y <- 2 * (df$x1 > 0) + rnorm(50, sd = 0.2)
fit <- figs(y ~ x1 + x2, data = df, max_splits = 3)
figsr_importance(fit)
```

figs_tree*FIGS Model Specification for Parsnip*

Description

`figs_tree()` defines a Fast Interpretable Greedy-Tree Sums model for use with the `parsnip` and `tidymodels` ecosystem.

Usage

```
figs_tree(
  mode = "regression",
  engine = "figsr",
  max_splits = NULL,
  max_trees = NULL,
  min_n = NULL
)
```

Arguments

<code>mode</code>	A single character string for the prediction outcome mode: "regression" or "classification". Classification supports two-class outcomes only; a factor with more than two levels raises an error at fit time.
<code>engine</code>	A single character string for the computational engine. Only "figsr" is available.
<code>max_splits</code>	An integer for the maximum total splits across all trees. NULL (the default) leaves the engine default of 10 in place.
<code>max_trees</code>	An integer for the maximum number of trees. NULL (the default) leaves the number of trees unconstrained.
<code>min_n</code>	An integer for the minimum number of data points in a node to split. NULL (the default) leaves the engine default of 5 in place.

Value

A `parsnip` model specification object.

Examples

```
library(parsnip)
spec <- figs_tree(max_splits = 8) |>
  set_engine("figsr") |>
  set_mode("regression")
spec
```

Description

`fit_figs()` is the function `parsnip` calls when fitting a `figs_tree()` specification with the "figsr" engine. It accepts either the formula interface or the x/y interface and forwards to `figs()`. Users normally call `figs()` or `fit()` instead.

Usage

```
fit_figs(
  formula = NULL,
  data = NULL,
  x = NULL,
  y = NULL,
  max_splits = 10,
  max_trees = NULL,
  min_n = 5,
  mode = "regression",
  ...
)
```

Arguments

<code>formula</code>	A formula specifying outcome and predictors.
<code>data</code>	A data frame containing the training data.
<code>x</code>	A data frame or matrix of predictors.
<code>y</code>	An outcome vector.
<code>max_splits</code>	Integer. Maximum total number of splits across all trees.
<code>max_trees</code>	Integer. Maximum number of trees in the sum.
<code>min_n</code>	Integer. Minimum number of observations in a node to split.
<code>mode</code>	Character. Either "regression" or "classification"; classification supports two-class outcomes only.
<code>...</code>	Additional arguments passed to <code>figs()</code> .

Value

An object of class `figsr_fit`.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(60), x2 = rnorm(60))
df$y <- 2 * (df$x1 > 0) + rnorm(60, sd = 0.2)
fit_figs(x = df[, c("x1", "x2")], y = df$y, max_splits = 3)
```

max_splits	<i>Dials Parameter for Maximum Splits</i>
------------	---

Description

Dials Parameter for Maximum Splits

Usage

```
max_splits(range = c(2L, 20L), trans = NULL)
```

Arguments

range	A numeric vector of length 2 for lower and upper limits.
trans	A trans object.

Value

A dials parameter object.

Examples

```
max_splits()  
max_splits(range = c(2L, 40L))
```

max_trees	<i>Dials Parameter for Maximum Trees</i>
-----------	--

Description

Dials Parameter for Maximum Trees

Usage

```
max_trees(range = c(1L, 10L), trans = NULL)
```

Arguments

range	A numeric vector of length 2 for lower and upper limits.
trans	A trans object.

Value

A dials parameter object.

Examples

```
max_trees()
max_trees(range = c(1L, 5L))
```

plot.figsr_fit	<i>Plot Method for figsr_fit Models</i>
----------------	---

Description

plot.figsr_fit() visualizes individual decision trees comprising the Fast Interpretable Greedy-Tree Sums (FIGS) model using 100% Base R graphics (zero external package dependencies).

Usage

```
## S3 method for class 'figsr_fit'
plot(x, tree_idx = NULL, style = c("scientific", "modern", "classic"), ...)
```

Arguments

x	A fitted figsr_fit model object.
tree_idx	Optional integer vector specifying indices of trees to plot. Default is NULL (plots all trees).
style	Character string specifying the visual style: "scientific" (default, scientific/editorial notation), "modern" (rounded pastel boxes), or "classic" (high-contrast minimalist statistical style).
...	Additional arguments, currently ignored.

Value

Invisible NULL.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(50), x2 = rnorm(50), y = rnorm(50))
fit <- figs(y ~ x1 + x2, data = df, max_splits = 3)
plot(fit)
plot(fit, style = "scientific")
plot(fit, tree_idx = 1)
```

```
predict.bagging_figs_fit
```

Predict Method for bagging_figs_fit Models

Description

Predict Method for bagging_figs_fit Models

Usage

```
## S3 method for class 'bagging_figs_fit'
predict(object, new_data, type = NULL, ...)
```

Arguments

object	A fitted bagging_figs_fit object.
new_data	A data frame of predictor values.
type	Character. For classification, either "class" (the default) or "prob"; ignored for regression, which always returns .pred.
...	Additional arguments.

Value

A tibble containing averaged ensemble predictions.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(60), x2 = rnorm(60))
df$y <- 2 * (df$x1 > 0) + rnorm(60, sd = 0.2)
bag_fit <- bagging_figs(y ~ x1 + x2, data = df, n_estimators = 3)
predict(bag_fit, new_data = df)
```

```
predict.figsr_fit
```

Predict Method for figsr_fit Models

Description

Predict Method for figsr_fit Models

Usage

```
## S3 method for class 'figsr_fit'
predict(object, new_data, type = NULL, ...)
```

Arguments

object	A fitted figsr_fit object.
new_data	A data frame of new predictor observations.
type	Character. Either "numeric", "class", or "prob". Default depends on model mode.
...	Additional arguments.

Value

A tibble with predictions in standardized tidymodels format.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(50), y = rnorm(50))
fit <- figs(y ~ x1, data = df)
predict(fit, new_data = df)
```

predict_figs	<i>Internal / Parsnip Predict Bridge Function for figsr_fit</i>
--------------	---

Description

Internal / Parsnip Predict Bridge Function for figsr_fit

Usage

```
predict_figs(object, new_data, type = "numeric", ...)
```

Arguments

object	A fitted figsr_fit object.
new_data	A data frame of new predictor observations.
type	Character. "numeric", "class", or "prob".
...	Additional arguments.

Value

A tibble of predictions.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(60))
df$y <- 2 * (df$x1 > 0) + rnorm(60, sd = 0.2)
fit <- figs(y ~ x1, data = df, max_splits = 3)
predict_figs(fit, new_data = df, type = "numeric")
```

print.figsr_fit	<i>Print Method for figsr_fit Models</i>
-----------------	--

Description

Print Method for figsr_fit Models

Usage

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

Arguments

x	A figsr_fit object.
...	Additional arguments, currently ignored.

Value

x, invisibly. Called for the side effect of printing a summary of the fitted tree sum to the console.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(60), x2 = rnorm(60))
df$y <- 2 * (df$x1 > 0) + rnorm(60, sd = 0.2)
print(figs(y ~ x1 + x2, data = df, max_splits = 3))
```

summary.figsr_fit	<i>Summary Method for figsr_fit Models</i>
-------------------	--

Description

summary.figsr_fit() prints human-readable decision rules for each tree in the sum.

Usage

```
## S3 method for class 'figsr_fit'
summary(object, ...)
```

Arguments

object	A figsr_fit object.
...	Additional arguments, currently ignored.

Value

object, invisibly. Called for the side effect of printing the IF-THEN decision rules of each tree to the console.

Examples

```
set.seed(42)
df <- data.frame(x1 = rnorm(60), x2 = rnorm(60))
df$y <- 2 * (df$x1 > 0) + rnorm(60, sd = 0.2)
summary(figs(y ~ x1 + x2, data = df, max_splits = 3))
```

update.figs_tree	<i>Update a FIGS Model Specification</i>
------------------	--

Description

update() changes the arguments of a `figs_tree()` specification in place, the way tidymodels users expect of any model specification.

Usage

```
## S3 method for class 'figs_tree'
update(
  object,
  max_splits = NULL,
  max_trees = NULL,
  min_n = NULL,
  fresh = FALSE,
  ...
)
```

Arguments

object	A <code>figs_tree()</code> specification.
max_splits	An integer for the maximum total splits across all trees.
max_trees	An integer for the maximum number of trees.
min_n	An integer for the minimum number of data points in a node to split.
fresh	Logical. Should the arguments be replaced rather than merged?
...	Not used.

Value

An updated `figs_tree()` specification.

Examples

```
spec <- figs_tree(max_splits = 4)
update(spec, max_splits = 8)
```

Index

`bagging_figs`, [2](#)

`figs`, [3](#)

`figs()`, [3](#), [7](#)

`figs_tree`, [6](#)

`figs_tree()`, [7](#), [13](#)

`figsr_importance`, [5](#)

`fit_figs`, [7](#)

`max_splits`, [8](#)

`max_trees`, [8](#)

`plot.figsr_fit`, [9](#)

`predict.bagging_figs_fit`, [10](#)

`predict.figsr_fit`, [10](#)

`predict_figs`, [11](#)

`print.figsr_fit`, [12](#)

`stats::model.frame()`, [4](#)

`stats::na.omit()`, [4](#)

`summary.figsr_fit`, [12](#)

`update.figs_tree`, [13](#)