Package: gamtree (via r-universe)

May 17, 2026

Title GAM-based Recursive Partitioning

Version 0.0.3

Description Recursively partitions the observations in a dataset, based on the parameters of a generalized additive model (GAM). Function splinetree implements partitioning based on parametric or non-penalized spline models. It offers faster computational speed and more flexibility for modeling multilevel and longitudinal data. Function gamtree implements partitioning based on semi-parametric or penalized spline models, a.k.a. smoothing splines. It requires less user involvement in correctly specifying the splines and knots, but has a heavier computational load and offers less flexibility in modeling multilevel and longitudinal data. Packages mgcv and splines are used for spline and model estimation, packages partykit and merDeriv are used for partitioning and derivative computations.

Depends R (>= 3.5.3), glmertree, lme4

Imports Formula, splines, mgcv, partykit, gamm4, merDeriv, nlme, grid

License GPL-2 | GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Suggests testthat

Config/pak/sysreqs cmake make

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

Date/Publication 2025-04-09 12:17:18 UTC

RemoteUrl https://github.com/marjoleinF/gamtree

RemoteRef HEAD

RemoteSha e13c3c8f24458ad05b64bc9e7e7339f7f7eb1368

Contents

`coef.gamtree` *Extract coefficients from a GAM tree.*

Description

`coef.gamtree` extracts fixed- or random-effects coefficients from a GAM tree.

Usage

```
## S3 method for class 'gamtree'
coef(object, which = "fixed", ...)
```

Arguments

`object`	an object of class `"gamtree"`.
`which`	character. Either `"fixed"` (default) or `"random"`, indicating that fixed- or random-effects coefficients should be returned, respectively.
`...`	further arguments to be passed to `fixef.merMod` or `ranef.merMod`.

eco *Example dataset of light-response curves*

Description

The data comprises light-response curves, which describe the relationship between photosynthetically active radiation (PAR) and photosynthetic rate (Pn). Observations are repeated measures on the same plants. Variable 'Specimen' provides an identifier for individual plants. Dataset was copied from https://stackoverflow.com/questions/37037445/using-mob-trees-partykit-package-with-nls-model For illustrating the funtionality of package gamtree, two variables were added: Specimen, which identifies individual plants, and noise an artificially generated continuous covariate that is pure (gaussian) noise.

Usage

```
data(eco)
```

Format

'eco' A data.frame with 628 observations of 5 variables:

Species categorical partitioning variable, indicator for specie (signal).

PAR continuous predictor variable for the node-specific model (signal).

Pn continuous response variable.

Specimen identification number of individual plants.

noise artifically generated continuous covariate (noise).

Source

https://stackoverflow.com/questions/37037445/using-mob-trees-partykit-package-with-nls-model

Examples

```
summary(eco)
```

fitted.gamm4 *Internal function for extracting fitted values from MOB-based GAM trees.*

Description

fitted.gamm4 extract fitted values from objects of class gamm4.

Usage

```
## S3 method for class 'gamm4'
fitted(object, ...)
```

Arguments

object	an object of class gamm4.
...	currently not used.

---

fixef.gamtree *Extract fixed-effects coefficients from a GAM tree.*

Description

fixef.gamtree extracts fixed-effects coefficients from a GAM tree.

Usage

```
## S3 method for class 'gamtree'
fixef(object, ...)
```

Arguments

object	an object of class "gamtree".
...	further arguments to be passed to fixef.merMod.

---

gamtree *Recursively partition a dataset based on penalized GAMs.*

Description

gamtree recursively partitions a dataset into subgroups with penalized GAMs, characterized by differences in the parameter estimates.

Usage

```
gamtree(
  formula,
  data,
  weights = NULL,
  REML = TRUE,
  method = "mob",
  cluster = NULL,
  offset = NULL,
  verbose = FALSE,
```

```
  parm = c(1, 2, 4),
  gam_ctrl = list(),
  tree_ctrl = list(),
  alt_formula = NULL,
  ...
)
```

Arguments

formula	specifies the model formula, consisting of three parts: the response variable followed by a tilde ('~'); the terms for the node-specific GAMs, followed by a vertical bar ('|') and the potential partitioning variables (separated by a '+'). The 'by' argument of function s may NOT be used in the node-specific GAM formulation. Refrain from using the dot ('.') to specify all remaining variables in data, this may yield unexpected results; make sure to specify each variable in the corresponding part of the model formula. See Examples.
data	data.frame containing the variables specified in formula.
weights	numeric vector of length nrow(data); optional case weights. A weight of 2, for example, is equivalent to having made exactly the same observation twice.
REML	logical, defaults to TRUE. Passed on to 'gamm4' and in turn 'lmer' (but not 'glmer') fitting routines to control whether REML or ML estimation is used.
method	character, one of "ctree" or "mob", indicates which partitioning algorithm should be used. See details below.
cluster	optional, a name refering to a colum of data, or a numeric or factor vector with a cluster ID to be employed for clustered covariances in the parameter stability tests. Most useful if method = "mob", for method = "ctree" probably less so as it may yield overly conservative splitting. This argument should be used when the partitioning variables are not measured on the individual observation level, but on a higher level. E.g., when the response variables consists of repeated measurements of the same respondents.
offset	numeric vector of length nrow(data). Supplies model offset for use in fitting. Note that this offset will always be completely ignored when predicting.
verbose	logical. Should progress be printed to the commande line in every iteration? If true, the iteration number, information on the splitting procedure, and the log-likelihood (with df) value of the fitted full mixed-effects gam model is printed.
parm	vector of one or more integers, indicating which parameters should be included in the parameter stability tests. The default c(1, 2, 4) includes the intercept, linear slope and error variance of the smoothing spline. The 3rd parameter is the variance of smooth term. It is excluded by default, because its inclusion yields too high power in many situations.
gam_ctrl	a list of fit control parameters to replace defaults returned by gam.control.
tree_ctrl	a list of one or more control parameters as accepted by mob_control (to be passed to function mob if method = "mob"), or ctree_control (to be passed to function ctree is method = "ctree"). Note: arguments xtype and ytype of mob_control are set to "data.frame", by default, this cannot be changed. Argument parm of mob_control will be overruled by the argument of the same name of the current function.

alt_formula	list with two elements, for specifying non-standard model formulae for GAM. E.g., the formula list required for use of the multinom family.
...	additional arguments to be passed to function gamm4.

Details

MOB is short for model-based recursive partitioning, ctree is short for conditional inference tree. MOB is based more strongly on parametric theory, thereby allowing for easy inclusion of clustering structures into the estimation procedure (see also argument cluster), yielding similar to a GEE-type approach for estimation of multilevel and longitudinal data structures. Yet, computation time for MOB is much larger than for ctree, which is mostly due to how it searches for the optimal splitting value, after the variable for splitting has been selected. ctree uses tests based on permutation theory, and thereby offers a less parametrically oriented approach. It is much faster than MOB, but does not provide a natural way of accounting for multilevel or longitudinal data structures.

Value

Returns an object of class "gamtree". This is a list, containing (amongst others) the GAM-based recursive partition (in $tree). The following methods are available to extract information from the fitted object: predict.gamtree, for obtaining predicted values for training and new observations; plot.gamtree for plotting the tree and variables' effects; coef.gamtree, fixef.gamtree and ranef.gamtree for extracting estimated coefficients. VarCorr.gamtree for extracting random-effects (co)variances, summary.gamtree for a summary of the fitted models.

See Also

predict.gamtree plot.gamtree coef.gamtree summary.gamtree

Examples

```
gt_m <- gamtree(Pn ~ s(PAR, k = 5L) | Species, data = eco, cluster = Specimen)
summary(gt_m)
gt_c <- gamtree(Pn ~ s(PAR, k = 5L) | Species, data = eco, method = "ctree")
summary(gt_c)
```

---

plot.gamtree *Plotting method for GAM trees*

Description

Takes a fitted GAM tree and plots the smooth functions fitted in each of the terminal nodes of the tree.

Usage

```
## S3 method for class 'gamtree'
plot(
  x,
  which = "both",
  ylim = "firstnode",
  treeplot_ctrl = list(),
  gamplot_ctrl = list(),
  ...
)
```

Arguments

x	object of class gamtree.
which	character. The default ("both") plots the tree structure, followed by the model fitted in the terminal nodes. Alternatively, "tree" will plot the tree structure, and "terms" will plot the smooth (and parametric) terms from the terminal-node-specific and global model. Note that the fitted curves in the tree do not convey a conditional function of the predictor on the x-axis (as plotted when "terms" is specified). They are a function of the predictor on the x-axis, as well as all other predictors in the model and could thus be referred to as 'marginal' fitted curves.
ylim	"firstplot" (default), NULL, or a numeric vector of length 2. Only used for plotting the terminal-node models (not the tree). Specifies how the limits of the y-axes of the terminal node plots should be chosen. The default ("firstnode") uses the observations in the first node to determine the limits of the y-axes for all plots. Alternatively, NULL will determine the limits of the y-axes separately for each plot. Alternatively, a numeric vector of length two may be specified, specifying the lower and upper limits of the y-axes.
treeplot_ctrl	list of (named) arguments to be passed to plot.party.
gamplot_ctrl	list of (named) arguments to be passed to plot.gam. Note that not all arguments of plot.gam are supported. .
...	further arguments, currently not used.

Warning

The plotted terms by default also represent confidence bands. These should be taken with a big grain of salt, because they do NOT account for the searching of the tree structure; they assume the tree structure was known in advance. They should be interpreted as overly optimistic and with caution.

Examples

```
gt <- gamtree(Pn ~ s(PAR, k = 5L) | Species, data = eco,
              cluster = Specimen)
plot(gt, which = "tree") # default is which = 'both'
plot(gt, which = "terms")
```

`plot.splinetree` *Plotting function for visualization of spline-based (g)lmertrees.*

Description

`plot.splinetree` takes a fitted `(g)lmertree` with splines and plots it. It is a wrapper for `plot.glmertree` and `plot.lmertree`, with critical adjustments for better visualization of spline models in the terminal nodes.

Usage

```
## S3 method for class 'splinetree'
plot(x, which = "all", fitted = "marginal", ...)
```

Arguments

`x`	fitted object of class `(g)lmertree` containing splines specified by `setup.spline` in the terminal node model.
`which`	character, "both", "tree" or "ranef". Other options are available, see `plot.glmertree` and `plot.lmertree`, but might be less helpful for spline models.
`fitted`	character, "marginal" (default), "combined" or "none". Specifies whether and how fitted values should be computed and visualized. See `plot.lmertree` or `plot.glmertree` for further detail.
`...`	additional arguments to be passed to `plot.lmertree` or `plot.glmertree`.

See Also

`setup.spline` `predict.splinetree` `plot.lmertree` `plot.glmertree`

Examples

```
st <- splinetree(Pn ~ ns(PAR, df = 5) | Specimen | Species, data = eco,
                 cluster = Specimen)
plot(st)
```

`predict.gamm4` *Internal function for extracting predictions from MOB-based GAM trees.*

Description

`predict.gamm4` extract predictions from objects of class gamm4.

Usage

```
## S3 method for class 'gamm4'
predict(object, newdata, ...)
```

Arguments

object	an object of class gamm4.
newdata	an optional data.frame in which to look for variables with which to predict. If omitted, the fitted values are used.
...	currently not used.

---

predict.gamtree *Get predictions from fitted GAM tree*

Description

Takes a fitted GAM tree (of class "gamtree") and returns predictions given a new set of values for the model covariates, or for the original covariate values used for fitting the GAM tree.

Usage

```
## S3 method for class 'gamtree'
predict(object, newdata = NULL, type = "link", ...)
```

Arguments

object	an object of class gamtree.
newdata	a data.frame containing the values of the model covariates for which predictions should be returned. The default (NULL) returns predictions for the original training data.
type	character vector of length 1, specifying the type of prediction to be returned. "response" (the default) returns values on the scale of the response variable. Alternatively, "link" (only available if method = "mob")returns values on the scale of the linear predictor; "node" returns an integer vector of node identifiers.
...	further arguments to be passed to predict.party.

Value

Returns a vector of predicted values.

predict.splinetree *Predict method for spline-based (g)lmertrees.*

Description

predict.splinetree computes predictions for a fitted (g)lmertree that is based on splines.

Usage

```
## S3 method for class 'splinetree'
predict(object, newdata, ...)
```

Arguments

object	fitted object of class (g)lmertree containing splines specified by setup.spline in the terminal node model.
newdata	data.frame with observations for which predictions should be computed.
...	additional arguments to be passed to predict.lmertree or predict.glmertree.

See Also

setup.spline predict.splinetree lmertree glmertree

Examples

```
st <- splinetree(Pn ~  ns(PAR, df = 5) | Specimen | Species, data = eco,
                cluster = Specimen)
predict(st, newdata = eco[1L:5L, ])
```

print.gamtree *Print method for a fitted GAM tree*

Description

Prints the local and/or global terms in a fitted GAM tree.

Usage

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

Arguments

x	object of class gamtree.
...	further arguments to be passed to print.modelparty or

Examples

```
gt <- gamtree(Pn ~ s(PAR, k = 5L) | Species, data = eco, cluster = Specimen)
gt ## or: print(gt)
```

`ranef.gamtree` *Extract random-effects coefficients from a GAM tree.*

Description

`ranef.gamtree` extracts random-effects coefficients from a GAM tree.

Usage

```
## S3 method for class 'gamtree'
ranef(object, ...)
```

Arguments

`object`	an object of class `"gamtree"`.
`...`	further arguments to be passed to `ranef.merMod`.

`setup.spline` *Set up splines bases for use with function (g)lmertree.*

Description

`setup.spline` takes a dataset and spline specification as input, and returns the dataset with spline bases added.

Usage

```
setup.spline(spline, data, ...)
```

Arguments

`spline`	character vector of length 1, describing the spline basis to be created. Currently, functions `ns` and `bs` are supported. See Examples.
`data`	a `data.frame` containing the variable referred to in `spline`.
`...`	additional arguments to be passed to the function specified in `spline`.

Value

a `data.frame` with a many rows and columns as `data`. The spline consists of `df` basis functions, but is contained in a single column named `spline`, followed by the name of the predictor variable specified.

See Also

plot.splinetree predict.splinetree lmertree glmertree

Examples

```
data <- setup.spline("ns(PAR, df = 3)", data = eco)
head(data)
matplot(x = data$PAR[order(data$PAR)],
        y = data$spline.PAR[order(data$PAR),], type = "l")
data <- setup.spline("bs(PAR, degree = 2, df = 4)", data = eco)
head(data)
matplot(x = data$PAR[order(data$PAR)],
        y = data$spline.PAR[order(data$PAR),], type = "l")
```

---

splinetree *Fit a (g)lmertree using spline-based partitioning.*

Description

splinetree is a wrapper for functions (g)lmertree to simplify fitting, visualizing and predicting spline-based trees.

Usage

```
splinetree(formula, data, family = "gaussian", ...)
```

Arguments

formula	A four-part function See Examples below, and lmertree or glmertree.
data	a data.frame containing the variable referred to in spline.
family	family specification for glmertree. See glm documentation for families.
...	additional arguments to be passed to function lmertree (default, i.e., family = "gaussian") or glmertree (family other than gaussian).

Value

A object of class"splinetree" and "lmertree" or "glmertree".

See Also

plot.splinetree predict.splinetree lmertree glmertree

Examples

```
 sp <- splinetree(Pn ~ ns(PAR, df = 5) | Specimen | Species, data = eco)
sp
```

summary.gamtree *Summary method for a fitted GAM tree*

Description

Prints a summary of the local and/or global terms in a fitted GAM tree.

Usage

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

Arguments

object	object of class gamtree.
...	further arguments to be passed to summary.gam.

Warning

The printed results by default also provide standard error and significance tests. These should be taken with a big grain of salt, because they do NOT account for the searching of the tree structure; they assume the tree structure was known in advance. They thus should be interpreted as overly optimistic and with caution.

Examples

```
## GAM tree without global terms:
gt <- gamtree(Pn ~ s(PAR, k = 5L) | Species, data = eco, cluster = Specimen)
summary(gt)
```

VarCorr.gamtree *Extract random-effects covariance matrices from a GAM tree.*

Description

VarCorr.gamtree extracts fixed-effects random-effects covariance matrices from the nodes of a GAM tree.

Usage

```
## S3 method for class 'gamtree'
VarCorr(x, sigma = 1, which = "terminal", ...)
```

Arguments

x	an object of class "gamtree".
sigma	an optional numeric value used as a multiplier for the standard deviations.
which	character. "terminal" (default) returns (co)variances for all terminal nodes, "inner" returns the (co)variances for all inner (splitting) nodes, "all" returns covariances for all nodes.
...	additional arguments to be passed to VarCorr.merMod.

Index