

Package: glmertree (via r-universe)

September 19, 2024

Title Generalized Linear Mixed Model Trees
Date 2024-09-19
Version 0.2-6
Description Recursive partitioning based on (generalized) linear mixed models (GLMMs) combining lmer()/glmer() from 'lme4' and lmtree()/glmmtree() from 'partykit'. The fitting algorithm is described in more detail in Fokkema, Smits, Zeileis, Hothorn & Kelderman (2018; <[DOI:10.3758/s13428-017-0971-x](https://doi.org/10.3758/s13428-017-0971-x)>). For detecting and modeling subgroups in growth curves with GLMM trees see Fokkema & Zeileis (2024; <[DOI:10.3758/s13428-024-02389-1](https://doi.org/10.3758/s13428-024-02389-1)>).
Depends R (>= 4.0.0), lme4, partykit (>= 1.0-4)
Imports graphics, stats, utils, Formula
Suggests vcd, lattice, betareg, glmmTMB, lmerTest
LazyData yes
License GPL-2 | GPL-3
Encoding UTF-8
Repository <https://zeileis.r-universe.dev>
RemoteUrl <https://github.com/r-forge/partykit>
RemoteRef HEAD
RemoteSha 575ca9f0bcd61c6c5ff75c8952766d96dad34a6

Contents

betamertree	2
coef.glmertree	5
cv.glmertree	6
DepressionDemo	8
glmertree	9
GrowthCurveDemo	13
MHserviceDemo	15
plot.glmertree	16

betamertree	<i>Beta Mixed-Effects Regression Trees</i>
-------------	--

Description

Model-based recursive partitioning based on mixed-effects beta regression.

Usage

```
betamertree(formula, data, family = NULL, weights = NULL, cluster = NULL,
  ranefstart = NULL, offset = NULL, REML = TRUE, joint = TRUE,
  abstol = 0.001, maxit = 100, dfsplit = TRUE, verbose = FALSE,
  plot = FALSE, glmmTMB.control = glmmTMB::glmmTMBControl(), ...)
```

Arguments

formula	formula specifying the response variable and a three-part right-hand-side describing the regressors, random effects, and partitioning variables, respectively. For details see below.
data	data.frame to be used for estimating the model tree.
family	currently not used. The default beta distribution parameterization of package betareg is used, see also <code>?glmmTMB::beta_family</code> .
weights	numeric. An optional numeric vector of weights. Can be a name of a column in data or a vector of length <code>nrow(data)</code> .
cluster	currently not used.
ranefstart	currently not used.
offset	optional numeric vector to be included in the linear predictor with a coefficient of one. Note that <code>offset</code> can be a name of a column in data or a numeric vector of length <code>nrow(data)</code> .
joint	currently not used. Fixed effects from the tree are always (re-)estimated jointly along with the random effects.
abstol	numeric. The convergence criterion used for estimation of the model. When the difference in log-likelihoods of the random-effects model from two consecutive iterations is smaller than <code>abstol</code> , estimation of the model tree has converged.
maxit	numeric. The maximum number of iterations to be performed in estimation of the model tree.
dfsplit	logical or numeric. <code>as.integer(dfsplit)</code> is the degrees of freedom per selected split employed when extracting the log-likelihood.
verbose	Should the log-likelihood value of the estimated random-effects model be printed for every iteration of the estimation?
plot	Should the tree be plotted at every iteration of the estimation? Note that selecting this option slows down execution of the function.

REML	logical scalar. Should the fixed-effects estimates be chosen to optimize the REML criterion (as opposed to the log-likelihood)? Will be passed to function <code>glmmTMB()</code> . See glmmTMB for details.
<code>glmmTMB.control</code>	list. An optional list with control parameters to be passed to <code>glmmTMB()</code> . See glmmTMBControl for details.
...	Additional arguments to be passed to <code>lmtree()</code> or <code>glmtree()</code> . See mob_control documentation for details.

Details

Function `betamertree` aims to learn a tree where each terminal node is associated with different fixed-effects regression coefficients, while adjusting for global random effects (such as a random intercept). It is a generalization of the ideas underlying function `glmertree`, to allow for detection of subgroups with different fixed-effects parameter estimates, keeping the random effects constant throughout the tree (i.e., random effects are estimated globally). The estimation algorithm iterates between (1) estimation of the tree given an offset of random effects, and (2) estimation of the random effects given the tree structure. See Fokkema et al. (2018) for a detailed description.

Where `glmertree` uses function `glmtree` from package `partykit` to find the subgroups, and function `glmer` from package `lme4` to estimate the mixed-effects model, `betamertree` uses function `betatree` from package `betareg` to find the subgroups, and function `glmmTMB` from package `glmmTMB` to estimate the mixed-effects model.

The code is experimental and will change in future versions.

Value

The function returns a list with the following objects:

<code>tree</code>	The final <code>betatree</code> .
<code>glmmTMB</code>	The final <code>glmmTMB</code> random-effects model.
<code>ranef</code>	The corresponding random effects of <code>glmmTMB</code> .
<code>varcorr</code>	The corresponding <code>VarCorr(glmmTMB)</code> .
<code>variance</code>	The corresponding <code>attr(VarCorr(glmmTMB), "sc")^2</code> .
<code>data</code>	The dataset specified with the <code>data</code> argument including added auxiliary variables <code>.ranef</code> and <code>.tree</code> from the last iteration.
<code>loglik</code>	The log-likelihood value of the last iteration.
<code>iterations</code>	The number of iterations used to estimate the <code>betamertree</code> .
<code>maxit</code>	The maximum number of iterations specified with the <code>maxit</code> argument.
<code>ranefstart</code>	The random effects used as an offset, as specified with the <code>ranefstart</code> argument.
<code>formula</code>	The formula as specified with the <code>formula</code> argument.
<code>randomformula</code>	The formula as specified with the <code>randomformula</code> argument.
<code>abstol</code>	The prespecified value for the change in log-likelihood to evaluate convergence, as specified with the <code>abstol</code> argument.

mob.control	A list containing control parameters passed to betatree(), as specified with
glmmTMB.control	A list containing control parameters passed to glmmTMB(), as specified in the control argument of function glmmTMB.
joint	Whether the fixed effects from the tree were (re-)estimated jointly along with the random effects, specified with the joint argument.

References

- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). “Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees”. *Behavior Research Methods*, **50**(5), 2016–2034. doi:[10.3758/s134280170971x](https://doi.org/10.3758/s134280170971x)
- Fokkema M, Edbrooke-Childs J, Wolpert M (2021). “Generalized Linear Mixed-Model (GLMM) Trees: A Flexible Decision-Tree Method for Multilevel and Longitudinal Data.” *Psychotherapy Research*, **31**(3), 329–341. doi:[10.1080/10503307.2020.1785037](https://doi.org/10.1080/10503307.2020.1785037)
- Fokkema M, Zeileis A (2024). “Subgroup Detection in Linear Growth Curve Models with Generalized Linear Mixed Model (GLMM) Trees.” *Behavior Research Methods*, **56**(7), 6759–6780. doi:[10.3758/s13428024023891](https://doi.org/10.3758/s13428024023891)
- Grün B, Kosmidis I, Zeileis A (2012). Extended Beta Regression in R: Shaken, Stirred, Mixed, and Partitioned. *Journal of Statistical Software*, **48**(11), 1–25. doi:[10.18637/jss.v048.i11](https://doi.org/10.18637/jss.v048.i11)

See Also

[glmmTMB](#), [betatree](#)

Examples

```
if (require("betareg") && require("glmmTMB")) {
  ## load example data
  data("ReadingSkills", package = "betareg")
  ## add random noise (not associated with reading scores)
  set.seed(1071)
  ReadingSkills$x1 <- rnorm(nrow(ReadingSkills))
  ReadingSkills$x2 <- runif(nrow(ReadingSkills))
  ReadingSkills$x3 <- factor(rnorm(nrow(ReadingSkills)) > 0)
  ReadingSkills$gr <- factor(rep(letters[1:5], length.out = nrow(ReadingSkills)))

  ## Fit beta mixed-effects regression tree
  betamer_form <- accuracy ~ iq | gr | dyslexia + x1 + x2 + x3
  bmertree <- betamertree(betamer_form, data = ReadingSkills, minsize = 10)
  VarCorr(bmertree)
  ranef(bmertree)
  fixef(bmertree)
  coef(bmertree)
  plot(bmertree)
  predict(bmertree, newdata = ReadingSkills[1:5,])
  predict(bmertree) ## see ?predict.glmmTMB for other arguments that can be passed
  residuals(bmertree) ## see ?residuals.glmmTMB for other arguments that can be passed
}
```

coef.glmertree	<i>Obtaining Fixed-Effects Coefficient Estimates of (Generalized) Linear Mixed Model Trees</i>
----------------	--

Description

coef and fixef methods for (g)lmertree objects.

Usage

```
## S3 method for class 'lmertree'
coef(object, which = "tree", drop = FALSE, ...)
## S3 method for class 'lmertree'
fixef(object, which = "tree", drop = FALSE, ...)
## S3 method for class 'glmertree'
coef(object, which = "tree", drop = FALSE, ...)
## S3 method for class 'glmertree'
fixef(object, which = "tree", drop = FALSE, ...)
```

Arguments

object	an object of class lmertree or glmertree.
which	character; "tree" (default) or "global". Specifies whether local (tree) or global fixed-effects estimates should be returned.
drop	logical. Only used when which = "tree"; delete the dimensions of the resulting array if it has only one level?
...	Additional arguments, currently not used.

Details

The code is still under development and might change in future versions.

Value

If type = "local", returns a matrix of estimated local fixed-effects coefficients, with a row for every terminal node and a column for every fixed effect. If type = "global", returns a numeric vector of estimated global fixed-effects coefficients.

References

Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). "Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees". *Behavior Research Methods*, **50**(5), 2016–2034. doi:10.3758/s134280170971x

Fokkema M, Zeileis A (2024). "Subgroup Detection in Linear Growth Curve Models with Generalized Linear Mixed Model (GLMM) Trees." *Behavior Research Methods*, **56**(7), 6759–6780. doi:10.3758/s13428024023891

See Also

[lmertree](#), [glmertree](#), [party-plot](#).

Examples

```
## load artificial example data
data("DepressionDemo", package = "glmertree")

## fit LMM tree with local fixed effects only
lt <- lmertree(depression ~ treatment + age | cluster | anxiety + duration,
  data = DepressionDemo)
coef(lt)

## fit LMM tree including both local and global fixed effect
lt <- lmertree(depression ~ treatment | (age + (1|cluster)) | anxiety + duration,
  data = DepressionDemo)
coef(lt, which = "tree") # default behaviour
coef(lt, which = "global")

## fit GLMM tree with local fixed effects only
gt <- glmertree(depression_bin ~ treatment | cluster |
  age + anxiety + duration, data = DepressionDemo)
coef(gt)

## fit GLMM tree including both local and global fixed effect
gt <- glmertree(depression_bin ~ treatment | (age + (1|cluster)) |
  anxiety + duration, data = DepressionDemo)
coef(gt, which = "tree") # default behaviour
coef(gt, which = "global")
```

cv.glmertree

Cross Validation of (Generalized) Linear Mixed Model Trees

Description

Performs cross-validation of a model-based recursive partition based on (generalized) linear mixed models. Using the tree or subgroup structure estimated from a training dataset, the full mixed-effects model parameters are re-estimated using a new set of test observations, providing valid computation of standard errors and valid inference. The approach is inspired by Athey & Imbens (2016), and "enables the construction of valid confidence intervals [...] whereby one sample is used to construct the partition and another to estimate [...] effects for each subpopulation."

Usage

```
cv.lmertree(tree, newdata, reference = NULL, omit.intercept = FALSE, ...)

cv.glmertree(tree, newdata, reference = NULL, omit.intercept = FALSE, ...)
```

Arguments

tree	An object of class <code>lmertree</code> or <code>glmertree</code> that was fitted on a set of training data.
newdata	A <code>data.frame</code> containing a new set of observations on the same variables that were used to fit <code>tree</code> .
reference	Numeric or character scalar, indicating the number of the terminal node of which the intercept should be taken as a reference for intercepts in all other nodes. If <code>NULL</code> , the default of taking the first terminal node's intercept as the reference category will be used. If the interest is in testing significance of differences between the different nodes intercepts, this can be overruled by specifying the number of the terminal node that should be used as the reference category.
omit.intercept	Logical scalar, indicating whether the intercept should be omitted from the model. The default (<code>FALSE</code>) includes the intercept of the first terminal node as the intercept and allows for significance testing of the differences between the first and the other terminal node's intercepts. Specifying <code>TRUE</code> will test the value of each terminal node's intercept against zero.
...	Not currently used.

Details

The approach is inspired by Athey & Imbens (2016), and "enables the construction of valid confidence intervals [...] whereby one sample is used to construct the partition and another to estimate [...] effects for each subpopulation."

Value

An object of with classes `lmertree` and `cv.lmertree`, or `glmertree` and `cv.glmertree`. It is the original (g)lmertree specified by argument `tree`, but the parametric model estimated based on the data specified by argument `newdata`. The default S3 methods for classes `lmertree` and `glmertree` can be used to inspect the results: `plot`, `predict`, `coef`, `fixef`, `ranef` and `VarCorr`. In addition, there is a dedicated summary method for classes `cv.lmertree` and `cv.glmertree`, which prints valid parameter estimates and standard errors, resulting from `summary.merMod`. For objects of class `cv.lmertree`, hypothesis tests (i.e., p-values) can be obtained by loading package `lmerTest` PRIOR to loading package(s) `glmertree` (and `lme4`), see examples.

References

- Athey S, Imbens G (2016). "Recursive Partitioning for Heterogeneous Causal Effects." *Proceedings of the National Academy of Sciences*, **113**(27), 7353–7360. doi:10.1073/pnas.1510489113
- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). "Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees". *Behavior Research Methods*, **50**(5), 2016–2034. doi:10.3758/s134280170971x
- Fokkema M, Edbrooke-Childs J, Wolpert M (2021). "Generalized Linear Mixed-Model (GLMM) Trees: A Flexible Decision-Tree Method for Multilevel and Longitudinal Data." *Psychotherapy Research*, **31**(3), 329–341. doi:10.1080/10503307.2020.1785037

Fokkema M, Zeileis A (2024). “Subgroup Detection in Linear Growth Curve Models with Generalized Linear Mixed Model (GLMM) Trees.” *Behavior Research Methods*, **56**(7), 6759–6780. [doi:10.3758/s13428024023891](https://doi.org/10.3758/s13428024023891)

See Also

[lmer](#), [glmer](#), [lmertree](#), [glmertree](#), [summary.merMod](#)

Examples

```
require("lmerTest") ## load BEFORE lme4 and glmertree to obtain hypothesis tests / p-values

## Create artificial training and test datasets
set.seed(42)
train <- sample(1:nrow(DepressionDemo), size = 200, replace = TRUE)
test <- sample(1:nrow(DepressionDemo), size = 200, replace = TRUE)

## Fit tree on training data
tree1 <- lmertree(depression ~ treatment | cluster | age + anxiety + duration,
                  data = DepressionDemo[train, ])

## Obtain honest estimates of parameters and standard errors using test data
tree2 <- cv.lmertree(tree1, newdata = DepressionDemo[test, ])
tree3 <- cv.lmertree(tree1, newdata = DepressionDemo[test, ],
                    reference = 7, omit.intercept = TRUE)

summary(tree2)
summary(tree3)

coef(tree1)
coef(tree2)
coef(tree3)

plot(tree1, which = "tree")
plot(tree2, which = "tree")
plot(tree3, which = "tree")

predict(tree1, newdata = DepressionDemo[1:5, ])
predict(tree2, newdata = DepressionDemo[1:5, ])
```

DepressionDemo

Artificial depression treatment dataset

Description

Simulated dataset of a randomized clinical trial (N = 150) to illustrate fitting of (G)LMM trees.

Usage

```
data("DepressionDemo")
```


Format

A data frame containing 150 observations on 6 variables:

depression numeric. Continuous treatment outcome variable (range: 3-16, M = 9.12, SD = 2.66).

treatment factor. Binary treatment variable.

cluster factor. Indicator for cluster with 10 levels.

age numeric. Continuous partitioning variable (range: 18-69, M = 45, SD = 9.56).

anxiety numeric. Continuous partitioning variable (range: 3-18, M = 10.26, SD = 3.05).

duration numeric. Continuous partitioning variable (range: 1-17, M = 6.97, SD = 2.90).

depression_bin factor. Binarized treatment outcome variable (0 = recovered, 1 = not recovered).

Details

The data were generated such that the duration and anxiety covariates characterized three subgroups with differences in treatment effects. The cluster variable was used to introduce a random intercept that should be accounted for. The treatment outcome is an index of depressive symptomatology.

See Also

[lmertree](#), [glmertree](#)

Examples

```
data("DepressionDemo", package = "glmertree")
summary(DepressionDemo)
lt <- lmertree(depression ~ treatment | cluster | anxiety + duration + age,
  data = DepressionDemo)
plot(lt)
gt <- glmertree(depression_bin ~ treatment | cluster | anxiety + duration + age,
  data = DepressionDemo)
plot(gt)
```

glmertree

(Generalized) Linear Mixed Model Trees

Description

Model-based recursive partitioning based on (generalized) linear mixed models.

Usage

```
lmertree(formula, data, weights = NULL, cluster = NULL,
  ranefstart = NULL, offset = NULL, joint = TRUE,
  abstol = 0.001, maxit = 100, dfsplit = TRUE, verbose = FALSE,
  plot = FALSE, REML = TRUE, lmer.control = lmerControl(), ...)

glmertree(formula, data, family = "binomial", weights = NULL,
  cluster = NULL, ranefstart = NULL, offset = NULL, joint = TRUE,
  abstol = 0.001, maxit = 100, dfsplit = TRUE, verbose = FALSE,
  plot = FALSE, nAGQ = 1L, glmer.control = glmerControl(), ...)
```

Arguments

formula	formula specifying the response variable and a three-part right-hand-side describing the regressors, random effects, and partitioning variables, respectively. For details see below.
data	data.frame to be used for estimating the model tree.
family	family specification for glmtree and glmer. See glm documentation for families.
weights	numeric. An optional numeric vector of weights. Can be a name of a column in data or a vector of length nrow(data).
cluster	optional vector of cluster IDs to be employed for clustered covariances in the parameter stability tests. Can be a name of a column in data or a vector of length nrow(data). If cluster = NULL (the default), observation-level covariances are employed in the parameter stability tests. If partitioning variables are measured on the cluster level, this can be accounted for by specifying the name of the cluster argument here, as a result cluster-level covariances will be employed in the parameter stability tests.
ranefstart	NULL (the default), TRUE, or a numeric vector of length nrow(data). Specifies the offset to be used in estimation of the first tree. NULL by default, yielding a zero offset initialization. If ranefstart = TRUE is specified, the random effects will be estimated first and the first tree will be grown using the random-effects predictions as an offset.
offset	optional numeric vector to be included in the linear predictor with a coefficient of one. Note that offset can be a name of a column in data or a numeric vector of length nrow(data).
joint	logical. Should the fixed effects from the tree be (re-)estimated jointly along with the random effects?
abstol	numeric. The convergence criterion used for estimation of the model. When the difference in log-likelihoods of the random-effects model from two consecutive iterations is smaller than abstol, estimation of the model tree has converged.
maxit	numeric. The maximum number of iterations to be performed in estimation of the model tree.
dfsplit	logical or numeric. as.integer(dfsplit) is the degrees of freedom per selected split employed when extracting the log-likelihood.

<code>verbose</code>	Should the log-likelihood value of the estimated random-effects model be printed for every iteration of the estimation?
<code>plot</code>	Should the tree be plotted at every iteration of the estimation? Note that selecting this option slows down execution of the function.
<code>REML</code>	logical scalar. Should the fixed-effects estimates be chosen to optimize the REML criterion (as opposed to the log-likelihood)? Will be passed to function <code>lmer()</code> . See lmer for details.
<code>nAGQ</code>	integer scalar. Specifies the number of points per axis for evaluating the adaptive Gauss-Hermite approximation to the log-likelihood, to be passed to function <code>glmer()</code> . See glmer for details.
<code>lmer.control, glmer.control</code>	list. An optional list with control parameters to be passed to <code>lmer()</code> or <code>glmer()</code> , respectively. See lmerControl for details.
<code>...</code>	Additional arguments to be passed to <code>lmtree()</code> or <code>glmtree()</code> . See mob_control documentation for details.

Details

(G)LMM trees learn a tree where each terminal node is associated with different fixed-effects regression coefficients while adjusting for global random effects (such as a random intercept). This allows for detection of subgroups with different fixed-effects parameter estimates, keeping the random effects constant throughout the tree (i.e., random effects are estimated globally). The estimation algorithm iterates between (1) estimation of the tree given an offset of random effects, and (2) estimation of the random effects given the tree structure. See Fokkema et al. (2018) for a detailed introduction.

To specify all variables in the model a formula such as $y \sim x1 + x2 \mid \text{random} \mid z1 + z2 + z3$ is used, where y is the response, $x1$ and $x2$ are the regressors in every node of the tree, `random` is the random effects, and $z1$ to $z3$ are the partitioning variables considered for growing the tree. If `random` is only a single variable such as `id` a random intercept with respect to `id` is used. Alternatively, it may be an explicit random-effects formula such as $(1 \mid \text{id})$ or a more complicated formula such as $((1 + \text{time}) \mid \text{id})$. (Note that in the latter two formulas, the brackets are necessary to protect the pipes in the random-effects formulation.)

In the random-effects model from step (2), two strategies are available: Either the fitted values from the tree can be supplied as an offset (`joint = FALSE`) so that only the random effects are estimated. Or the fixed effects are (re-)estimated along with the random effects using a nesting factor with nodes from the tree (`joint = TRUE`). In the former case, the estimation of each random-effects model is typically faster, but more iterations are required.

The code is still under development and might change in future versions.

Value

The function returns a list with the following objects:

<code>tree</code>	The final <code>lmtree</code> / <code>glmtree</code> .
<code>lmer</code>	The final <code>lmer</code> random-effects model.
<code>ranef</code>	The corresponding random effects of <code>lmer</code> .

<code>varcorr</code>	The corresponding <code>VarCorr(lmer)</code> .
<code>variance</code>	The corresponding <code>attr(VarCorr(lmer), "sc")^2</code> .
<code>data</code>	The dataset specified with the <code>data</code> argument including added auxiliary variables <code>.ranef</code> and <code>.tree</code> from the last iteration.
<code>loglik</code>	The log-likelihood value of the last iteration.
<code>iterations</code>	The number of iterations used to estimate the <code>lmer</code> tree.
<code>maxit</code>	The maximum number of iterations specified with the <code>maxit</code> argument.
<code>ranefstart</code>	The random effects used as an offset, as specified with the <code>ranefstart</code> argument.
<code>formula</code>	The formula as specified with the <code>formula</code> argument.
<code>randomformula</code>	The formula as specified with the <code>randomformula</code> argument.
<code>abstol</code>	The prespecified value for the change in log-likelihood to evaluate convergence, as specified with the <code>abstol</code> argument.
<code>mob.control</code>	A list containing control parameters passed to <code>lmtree()</code> , as specified with <code>...</code>
<code>lmer.control</code>	A list containing control parameters passed to <code>lmer()</code> , as specified in the <code>lmer.control</code> argument.
<code>joint</code>	Whether the fixed effects from the tree were (re-)estimated jointly along with the random effects, specified with the <code>joint</code> argument.

References

- Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). “Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees”. *Behavior Research Methods*, **50**(5), 2016–2034. doi:10.3758/s134280170971x
- Fokkema M, Edbrooke-Childs J, Wolpert M (2021). “Generalized Linear Mixed-Model (GLMM) Trees: A Flexible Decision-Tree Method for Multilevel and Longitudinal Data.” *Psychotherapy Research*, **31**(3), 329–341. doi:10.1080/10503307.2020.1785037
- Fokkema M, Zeileis A (2024). “Subgroup Detection in Linear Growth Curve Models with Generalized Linear Mixed Model (GLMM) Trees.” *Behavior Research Methods*, **56**(7), 6759–6780. doi:10.3758/s13428024023891

See Also

`plot.lmertree`, `plot.glmertree`, `cv.lmertree`, `cv.glmertree`, `GrowthCurveDemo`, `lmer`, `glmer`, `lmtree`, `glmtree`

Examples

```
## artificial example data
data("DepressionDemo", package = "glmertree")

## fit normal linear regression LMM tree for continuous outcome
lt <- lmertree(depression ~ treatment | cluster | age + anxiety + duration,
  data = DepressionDemo)
print(lt)
```

```

plot(lt, which = "all") # default behavior, may also be "tree" or "ranef"
coef(lt)
ranef(lt)
predict(lt, type = "response") # default behavior, may also be "node"
predict(lt, re.form = NA) # excludes random effects, see ?lme4::predict.merMod
residuals(lt)
VarCorr(lt) # see lme4::VarCorr

## fit logistic regression GLMM tree for binary outcome
gt <- glmertree(depression_bin ~ treatment | cluster | age + anxiety + duration,
  data = DepressionDemo)
print(gt)
plot(gt, which = "all") # default behavior, may also be "tree" or "ranef"
coef(gt)
ranef(gt)
predict(gt, type = "response") # default behavior, may also be "node" or "link"
predict(gt, re.form = NA) # excludes random effects, see ?lme4::predict.merMod
residuals(gt)
VarCorr(gt) # see lme4::VarCorr

## Alternative specification for binomial family: no. of successes and failures
DepressionDemo$failures <- as.numeric(DepressionDemo$depression_bin) - 1
DepressionDemo$successes <- 1 - DepressionDemo$failures
gt <- glmertree(cbind(failures, successes) ~ treatment | cluster | age + anxiety + duration,
  data = DepressionDemo, ytype = "matrix") ## see also ?partykit::mob_control

```

GrowthCurveDemo

Artificial dataset for partitioning of linear growth curve models

Description

Artificial dataset to illustrate fitting of LMM trees with growth curve models in the terminal nodes.

Usage

```
data("GrowthCurveDemo")
```

Format

A data frame containing 1250 repeated observations on 250 persons. x1 - x8 are time-invariant partitioning variables. Thus, they are measurements on the person (i.e., cluster) level, not on the individual observation level.

person numeric. Indicator linking repeated measurements to persons.

time factor. Indicator for timepoint.

y numeric. Response variable.

x1 numeric. Potential partitioning variable.

x2 numeric. Potential partitioning variable.
x3 numeric. Potential partitioning variable.
x4 numeric. Potential partitioning variable.
x5 numeric. Potential partitioning variable.
x6 numeric. Potential partitioning variable.
x7 numeric. Potential partitioning variable.
x8 numeric. Potential partitioning variable.

Details

Data were generated so that x1, x2 and x3 are true partitioning variables, x4 through x8 are noise variables. The (potential) partitioning variables are time invariant. Time-varying covariates can also be included in the model. For partitioning growth curves these should probably not be potential partitioning variables, as this could result in observations from the same person ending up in different terminal nodes. Thus, time-varying covariates are probably best included as predictors in the node-specific regression model. E.g.: $y \sim \text{time} + \text{timevarying_cov} \mid \text{person} \mid x1 + x2 + x3 + x4$.

References

Fokkema M & Zeileis A (2024). Subgroup detection in linear growth curve models with generalized linear mixed model (GLMM) trees. Behavior Research Methods. doi:10.3758/s13428024023891

See Also

[lmertree](#), [glmertree](#)

Examples

```
data("GrowthCurveDemo", package = "glmertree")
head(GrowthCurveDemo)

## Fit LMM tree with a random intercept w.r.t. person:
form <- y ~ time | person | x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8
lt.default <- lmertree(form, data = GrowthCurveDemo)
plot(lt.default, which = "tree") ## yields too large tree
VarCorr(lt.default)

## Account for measurement level of the partitioning variables:
lt.cluster <- lmertree(form, cluster = person, data = GrowthCurveDemo)
plot(lt.cluster, which = "tree") ## yields correct tree
plot(lt.cluster, which = "growth") ## plot individual growth curves not datapoints
coef(lt.cluster) ## node-specific fixed effects
VarCorr(lt.cluster) ## with smaller trees random effects explain more variance

## Fit LMM tree with random intercept and random slope of time w.r.t. person:
form.s <- y ~ time | (1 + time | person) | x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8
lt.s.cluster <- lmertree(form.s, cluster = person, data = GrowthCurveDemo)
plot(lt.s.cluster, which = "tree") ## same tree as before
```

```
coef(lt.cluster)
VarCorr(lt.s.cluster)
```

MHserviceDemo

Artificial mental-health service outcomes dataset

Description

Artificial dataset of treatment outcomes (N = 3739) of 13 mental-health services to illustrate fitting of (G)LMM trees with constant fits in terminal nodes.

Usage

```
data("MHserviceDemo")
```

Format

A data frame containing 3739 observations on 8 variables:

age numeric. Variable representing age in years (range: 4.8 - 23.6, M = 11.46).

impact numeric. Continuous variable representing severity of and impairment due to mental-health problems at baseline. Higher values indicate higher severity and impairment.

gender factor. Indicator for gender.

emotional factor. Indicator for presence of emotional disorder at baseline.

autism factor. Indicator for presence of autistic disorder at baseline.

conduct factor. Indicator for mental-health service provider.

cluster_id factor. Binarized treatment outcome variable (0 = recovered, 1 = not recovered).

outcome numeric. Variable representing treatment outcome as measured by a total mental-health difficulties score assessed about 6 months after baseline, corrected for the baseline assessment. Higher values indicate poorer outcome.

Details

Dataset was modelled after Edbrooke-Childs et al. (2017), who analyzed a sample of \$N = 3,739\$ young people who received treatment at one of 13 mental-health service providers in the UK. Note that the data were artificially generated and do not reflect actual empirical findings.

References

Fokkema M, Edbrooke-Childs J & Wolpert M (2021). "Generalized linear mixed-model (GLMM) trees: A flexible decision-tree method for multilevel and longitudinal data." *Psychotherapy Research*, 31(3), 329-341. doi:10.1080/10503307.2020.1785037

See Also

[lmer tree](#), [glmertree](#)

Examples

```
data("MHserviceDemo", package = "glmertree")
summary(MHserviceDemo)
lt <- lmertree(outcome ~ 1 | cluster_id | age + gender + emotional +
               autism + impact + conduct, data = MHserviceDemo)
plot(lt)

gt <- glmertree(factor(outcome > 0) ~ 1 | cluster_id | age + gender +
                 emotional + autism + impact + conduct,
                 data = MHserviceDemo, family = "binomial")
plot(gt)
```

plot.glmertree

Plotting (Generalized) Linear Mixed Model Trees

Description

plot method for (g)lmertree objects.

Usage

```
## S3 method for class 'lmertree'
plot(x, which = "all", nodesize_level = 1L,
     cluster = NULL, ask = TRUE, type = "extended",
     observed = TRUE, fitted = "combined", tp_args = list(),
     drop_terminal = TRUE, terminal_panel = NULL, dotplot_args = list(), ...)
## S3 method for class 'glmertree'
plot(x, which = "all", nodesize_level = 1L,
     cluster = NULL, ask = TRUE, type = "extended",
     observed = TRUE, fitted = "combined", tp_args = list(),
     drop_terminal = TRUE, terminal_panel = NULL, dotplot_args = list(), ...)
```

Arguments

x	an object of class lmertree or glmertree.
which	character; "all" (default), "tree", "random", "tree.coef" or "growth". Specifies whether the tree, random effects, or both should be plotted. "growth" should only be used in longitudinal models, it yields a tree with growth curves for each of the subjects in the tree nodes, instead of individual datapoints, and a thick red curve for the estimated node-specific fixed effects representing the average trajectory within the terminal node. "tree.coef" yields caterpillar plots of the estimated fixed-effects coefficients in every terminal node of the tree, but omits the tree structure (see Details).
nodesize_level	numeric. At which grouping level should sample size printed above each terminal node be computed? Defaults to 1, which is the lowest level of observation. If a value of 2 is specified, sample size at the cluster level will be printed above

	each terminal node. This only works if <code>x</code> (the <code>(g)lmertree</code>) was fitted using the <code>cluster</code> argument. Alternatively, a character vector of length one can be supplied, which gives the name of the grouping indicator in the <code>data.frame</code> used to fit <code>x</code> .
<code>cluster</code>	vector of cluster ids. Only used if <code>which = "growth"</code> . Need not be specified if clustered covariances were used for partitioning (i.e., argument <code>cluster</code> was specified). If <code>cluster</code> was not specified in the call to functions <code>(g)lmertree</code> , this argument should be specified for the plotting function to identify which individual observations belong to the same subject.
<code>ask</code>	logical. Should user be asked for input, before a new figure is drawn?
<code>type</code>	character; "extended" (default) or "simple". "extended" yields a plotted tree with observed data and/or fitted means plotted in the terminal nodes; "simple" yields a plotted tree with the value of fixed and/or random effects coefficients reported in the terminal nodes.
<code>observed</code>	logical. Should observed datapoints be plotted in the tree? Defaults to TRUE, FALSE is only supported for objects of class <code>lmertree</code> , not of class <code>glmertree</code> .
<code>fitted</code>	character. "combined" (default), "marginal" or "none". Specifies whether and how fitted values should be computed and visualized. Only used when predictor variables for the node-specific (G)LMs were specified. If "combined", fitted values will be computed, using observed values of the remaining (random and fixed-effects) predictor variables, which can yield very wiggly curves. If "marginal", fitted values will be calculated, fixing all remaining predictor variables (with random and/or fixed effects) at the observed sample mean (or majority class).
<code>tp_args</code>	list of arguments to be passed to panel generating function <code>node_glmertree</code> . See arguments <code>node_bivplot</code> in panelfunctions .
<code>drop_terminal</code>	logical. Should all terminal nodes be plotted at the bottom of the plot?
<code>terminal_panel</code>	an optional panel generating function to be passed to <code>plot.party()</code> , but will most likely be ignored. For passing arguments to the panel generating functions, use argument <code>tp_args</code> . For using a custom panel generating function, see Details .
<code>dotplot_args</code>	Optional list of additional arguments to be passed to dotplot . Only relevant when random- or fixed-effects plots are requested by specifying <code>which</code> as "ranef", "all", or "ranef".
<code>...</code>	Additional arguments to be passed to <code>plot.party()</code> . See party-plot documentation for details.

Details

If the node-specific model of the `(g)lmertree` object specified by argument `x` is an intercept-only model, observed data distributions will be plotted in the terminal nodes of the tree (using [node_barplot](#) (for categorical responses) or [node_boxplot](#) (for numerical responses)). Otherwise, fitted values will be plotted, in addition to observed datapoints, using a function taking similar arguments as [node_bivplot](#).

Exceptions:

If `fitted = "marginal"`, fitted values will be plotted by assuming the mean (continuous predictors) or mode (categorical predictors) for all predictor variables, except the variable on the x-axis of the current plot.

If `which = "growth"`, individual growth curves will be plotted as thin grey lines in the terminal nodes, while the node-specific fixed effect will be plotted on top of that as a thicker red curve.

If `which = "tree.coef"`, caterpillar plot(s) are created for the local (node-specific) fixed effects. These depict the estimated fixed-effects coefficients with 95% confidence intervals, but note that these CIs do not account for the searching of the tree structure and are therefore likely too narrow. There is currently no way to adjust CIs for searching of the tree structure, but the CIs can be useful to obtain an indication of the variability of the coefficient estimates, not for statistical significance testing.

If `which = "ranef"` or `"all"`, caterpillar plot(s) for the random effect(s) created, depicting the predicted random effects with 95% confidence intervals. See also [ranef](#) for more info. Note that the CIs do not account for the searching of the tree structure and may be too narrow.

If users want to specify custom panel generating functions, it might be best to not use the plotting method for (g)lmertrees. Instead, extract the (g)lmtree from the fitted (g)lmertree object (which is a list containing, amongst others, an element `$tree`). On this tree, most of the customization options from [party-plot](#) can then be applied.

The code is still under development and might change in future versions.

References

Fokkema M, Smits N, Zeileis A, Hothorn T, Kelderman H (2018). "Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees". *Behavior Research Methods*, **50**(5), 2016–2034. doi:10.3758/s134280170971x

Fokkema M, Zeileis A (2024). "Subgroup Detection in Linear Growth Curve Models with Generalized Linear Mixed Model (GLMM) Trees." *Behavior Research Methods*, **56**(7), 6759–6780. doi:10.3758/s13428024023891

See Also

[lmtree](#), [glmertree](#), [party-plot](#).

Examples

```
## load artificial example data
data("DepressionDemo", package = "glmertree")

## fit linear regression LMM tree for continuous outcome
lt <- lmtree(depression ~ treatment + age | cluster | anxiety + duration,
  data = DepressionDemo)
plot(lt)
plot(lt, type = "simple")
plot(lt, which = "tree", fitted = "combined")
plot(lt, which = "tree", fitted = "none")
plot(lt, which = "tree", observed = FALSE)
plot(lt, which = "tree.coef")
plot(lt, which = "ranef")
```

```
## fit logistic regression GLMM tree for binary outcome
gt <- glmertree(depression_bin ~ treatment + age | cluster |
  anxiety + duration, data = DepressionDemo)
plot(gt)
plot(gt, type = "simple")
plot(gt, which = "tree", fitted = "combined")
plot(gt, which = "tree", fitted = "none")
plot(gt, which = "tree.coef")
plot(gt, which = "ranef")
```

Index

- * **datasets**
 - DepressionDemo, 8
 - GrowthCurveDemo, 13
 - MHserviceDemo, 15
- * **hplot**
 - plot.glmertree, 16
- * **tree**
 - betamertree, 2
 - cv.glmertree, 6
 - glmertree, 9
- as.party.betamertree (betamertree), 2
- as.party.glmertree (glmertree), 9
- as.party.lmertree (glmertree), 9
- betamertree, 2
- betatree, 4
- coef.glmertree, 5
- coef.lmertree (coef.glmertree), 5
- cv.glmertree, 6, 12
- cv.lmertree, 12
- cv.lmertree (cv.glmertree), 6
- DepressionDemo, 8
- dotplot, 17
- fixef.glmertree (coef.glmertree), 5
- fixef.lmertree (coef.glmertree), 5
- glm, 10
- glmer, 8, 11, 12
- glmertree, 6, 8, 9, 9, 14, 15, 18
- glmmTMB, 3, 4
- glmmTMBControl, 3
- glmtree, 12
- GrowthCurveDemo, 12, 13
- lmer, 8, 11, 12
- lmerControl, 11
- lmertree, 6, 8, 9, 14, 15, 18
- lmertree (glmertree), 9
- lmtree, 12
- logLik.betamertree (betamertree), 2
- logLik.glmertree (glmertree), 9
- logLik.lmertree (glmertree), 9
- MHserviceDemo, 15
- mob_control, 3, 11
- model.frame.betamertree (betamertree), 2
- model.frame.glmertree (glmertree), 9
- model.frame.lmertree (glmertree), 9
- node_barplot, 17
- node_bivplot, 17
- node_boxplot, 17
- panelfunctions, 17
- plot.glmertree, 12, 16
- plot.lmertree, 12
- plot.lmertree (plot.glmertree), 16
- predict.betamertree (betamertree), 2
- predict.glmertree (glmertree), 9
- predict.lmertree (glmertree), 9
- print.betamertree (betamertree), 2
- print.glmertree (glmertree), 9
- print.lmertree (glmertree), 9
- ranef, 18
- ranef.betamertree (betamertree), 2
- ranef.glmertree (glmertree), 9
- ranef.lmertree (glmertree), 9
- resid.betamertree (betamertree), 2
- resid.glmertree (glmertree), 9
- resid.lmertree (glmertree), 9
- residuals.betamertree (betamertree), 2
- residuals.glmertree (glmertree), 9
- residuals.lmertree (glmertree), 9
- summary.merMod, 7, 8
- terms.betamertree (betamertree), 2

`terms.glmertree (glmertree)`, [9](#)
`terms.lmertree (glmertree)`, [9](#)

`VarCorr.betamertree (betamertree)`, [2](#)
`VarCorr.glmertree (glmertree)`, [9](#)
`VarCorr.lmertree (glmertree)`, [9](#)