Package ‘glmertree’

October 13, 2022

Title Generalized Linear Mixed Model Trees

Date 2019-11-19

Version 0.2-0

Description Recursive partitioning based on (generalized) linear mixed models (GLMMs) combining lmer()/glmer() from 'lme4' and lmtree()/glmtree() from 'partykit'. The fitting algorithm is described in more detail in Fokkema, Smits, Zeileis, Hothorn & Kelderman (2018; <DOI:10.3758/s13428-017-0971-x>).

Depends R (>= 3.1.0), lme4, partykit (>= 1.0-4)

Imports graphics, stats, utils, Formula

Suggests vcd, lattice

LazyData yes

License GPL-2 | GPL-3

RoxygenNote 6.1.1

NeedsCompilation no

Author Marjolein Fokkema [aut, cre],
Achim Zeileis [aut]

Maintainer Marjolein Fokkema <`M.Fokkema@fsw.leidenuniv.nl`>

Repository CRAN

Date/Publication 2019-11-19 18:30:05 UTC

R topics documented:

`coef.glmertree` *Obtaining Fixed-Effects Coefficient Estimates of (Generalized) Linear Mixed Model Trees*

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, curretnly 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. `https://doi.org/10.3758/s13428-017-0971-x`

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")
```

`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 will likely yield spurious splits, which can be mitigated by specification of the cluster argument, which results in cluster-level covariances being 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 coeffcient of one. Note that `offset` can be a name of a column in `data` or a 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.
`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 fixef-effects estimates be chosen to optimize the REML criterion (as opposed to the log-likelihood)? Will be passed to funtion `lmer()`. See `lmer` for details.
`nAGQ`	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 `glmer()`. See `glmer` for details.

`lmer.control, glmer.control`	list. An optional list with control parameters to be passed to `lmer()` and `glmer()`, respectively. See `lmerControl` for details.
`...`	Additional arguments to be passed to `lmtree()` or `glmtree()`. 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 ~ x1 + x2 | random | 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 | id)` or a more complicated formula such as `((1+time) | 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:

`tree`	The final `lmtree`/`glmtree`.
`lmer`	The final `lmer` random-effects model.
`ranef`	The corresponding random effects of `lmer`.
`varcorr`	The corresponding `VarCorr(lmer)`.
`variance`	The corresponding `attr(VarCorr(lmer), "sc")^2`.
`data`	The dataset specified with the `data` argument including added auxiliary variables `.ranef` and `.tree` from the last iteration.
`loglik`	The log-likelihood value of the last iteration.
`iterations`	The number of iterations used to estimate the `lmertree`.
`maxit`	The maximum number of iterations specified with the `maxit` argument.
`ranefstart`	The random effects used as an offset, as specified with the `ranefstart` argument.

`formula`	The formula as specified with the `formula` argument.
`randomformula`	The formula as specified with the `randomformula` argument.
`abstol`	The prespecified value for the change in log-likelihood to evaluate convergence, as specified with the `abstol` argument.
`mob.control`	A list containing control parameters passed to `lmtree()`, as specified with ....
`lmer.control`	A list containing control parameters passed to `lmer()`, as specified in the `lmer.control` argument.
`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. `https://doi.org/10.3758/s13428-017-0971-x`

See Also

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

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
```

`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 ~ time + timevarying_cov | person | x1 + x2 + x3 + x4`.

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
VarCorr(lt.cluster) ## yields slightly larger ranef 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
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.

See Also

`lmertree`, `glmertree`

Examples

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

MHserviceDemo$outcome_bin <- factor(MHserviceDemo$outcome > 0)
gt <- glmertree(outcome_bin ~ 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", ask = TRUE, type = "extended",
    observed = TRUE, fitted = "combined", tp_args = list(),
    drop_terminal = TRUE, terminal_panel = NULL, ...)
## S3 method for class 'glmertree'
plot(x, which = "all", ask = TRUE, type = "extended",
    observed = TRUE, fitted = "combined", tp_args = list(),
    drop_terminal = TRUE, terminal_panel = NULL, ...)
```

Arguments

`x`	an object of class `lmertree` or `glmertree`.
`which`	character; `"all"` (default), `"tree"`, `"random"` or `"tree.coef"`. Specifies whether, tree, random effects, or both should be plotted. Alternatively, `"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).
`ask`	logical. Should user be asked for input, before a new figure is drawn?
`type`	character; `"extended"` (default) or `"simple"`. `type = "extended"` yields a plotted tree with observed data and/or fitted means plotted in the terminal nodes; `"simple"` yields a plottedtree with the value of fixed and/or random effects coefficients reported in the terminal nodes.
`observed`	logical. Should observed datapoints be plotted in the tree? Defaults to `TRUE`, `FALSE` is only supported for objects of class `lmertree`, not of class`glmertree`.
`fitted`	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 computed, based on the observed values of the remaining (random and fixed-effects) predictor variables, and their estimated effects. If `"marginal"`, fitted values will be calculated, keeping all remaining variables (with random and/or fixed effects) fixed at their (population and sample) means (or majority class).
`tp_args`	list of arguments to be passed to panel generating function `node_glmertree`. See arguments `node_bivplot` in `panelfunctions`.
`drop_terminal`	logical. Should all terminal nodes be plotted at the bottom of the plot?
`terminal_panel`	an optional panel function to be passed to `plot.party()`. See `party-plot` documentation for details.
`...`	Additional arguments to be passed to `plot.party()`. See `party-plot` documentation for details.

Details

The caterpillar plot(s) for the local (node-specific) fixed effects (created when `which = "tree.coef"`) depict the estimated fixed-effects coefficients with 95% confidence intervals, but 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.

The caterpillar plot(s) for the random effect (created if `which = "ranef"` or `"all"`) depict the predicted random effects with 95% confidence intervals. See also `ranef`.

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. https://doi.org/10.3758/s13428-017-0971-x

See Also

`lmertree`, `glmertree`, `party-plot`.

Examples

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

## fit linear regression LMM tree for continuous outcome
lt <- lmertree(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