Package ‘metacart’

August 18, 2025

Type Package

Title Meta-CART: A Flexible Approach to Identify Moderators in Meta-Analysis

Version 3.0.4

Encoding UTF-8

Author Xinru Li [aut],
Elise Dusseldorp [aut, cph],
Kaihua Liu [ctb] (supported with the plot function),
Juan Claramunt [aut, cre],
Jacqueline Meulman [ctb]

Maintainer Juan Claramunt <`j.claramunt.gonzalez@fsw.leidenuniv.nl`>

Description Meta-CART integrates classification and regression trees (CART) into meta-analysis. Meta-CART is a flexible approach to identify interaction effects between moderators in meta-analysis. The method is described in Dusseldorp et al. (2014) <doi:10.1037/hea0000018> and Li et al. (2017) <doi:10.1111/bmsp.12088>.

LazyData TRUE

Depends R (>= 3.0.2), stats, ggplot2, gridExtra, methods, rpart

License GPL (>= 2)

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp

RoxygenNote 7.3.2

NeedsCompilation yes

Repository CRAN

Date/Publication 2025-08-18 09:00:02 UTC

Contents

`metacart-package` *R package for meta-CART*

Description

In meta-analysis, heterogeneity often exists between studies. To understand this heterogeneity, researchers search for study characteristics (i.e., potential moderators) that may account for the variance in study effect sizes. When multiple potential moderators are available (e.g., intervention characteristics), traditional meta-analysis methods often lack sufficient power to investigate interaction effects between moderators, especially high-order interactions. To solve this problem, meta-CART was proposed by integrating Classification and Regression Trees (CART) into meta-analysis. The method idenfities the interaction effects between influential moderators, partitions the studies into more homogeneous subgroups, and estimates summary effect size in each subgroup. The fixed effect or random effects assumption can be consistently taken into account in both tree-growing process and subgroup analysis.

Details

This method is suitable for identifying interaction effects between dichotomous, ordinal, continuous, and nominal moderators. The output of a `REmrt` object shows meta-CART analysis results based on the random effects model. And the output of a `FEmrt` object shows meta-CART analysis results based on the fixed effect model. The two objects display results for subgroup analysis including the Q-statistic and estimates for the subgroup effect sizes. Furthermore, the predict functions `predict.REmrt` and `predict.FEmrt` can be used to predict the effect size given the moderators. The plot functions `plot.REmrt` and `plot.FEmrt` show the interaction effects between identified moderators.

The core functions of the package are `FEmrt` and `REmrt`.

Author(s)

Maintainer: Xinru Li <x.li@math.leidenuniv.nl>; Contributors: Elise Dusseldorp, Kaihua Liu (supported with the plot function), Juan Claramunt Gonzalez, and Jacqueline Meulman.

References

Dusseldorp, E., van Genugten, L., van Buuren, S., Verheijden, M. W., & van Empelen, P. (2014). Combinations of techniques that effectively change health behavior: Evidence from meta-cart analysis. *Health Psychology, 33(12)*, 1530-1540. doi: 10.1037/hea0000018.

Li, X., Dusseldorp, E., & Meulman, J. J. (2017). Meta-CART: A tool to identify interactions between moderators in meta-analysis. *British Journal of Mathematical and Statistical Psychology, 70(1)*, 118-136. doi: 10.1111/bmsp.12088.

Li, X., Dusseldorp, E., Claramunt Gonzalez, J., Su, X., van Megen, J., & Meulman, J. J. (2025). Enhanced tree-based subgroup identification in meta-analysis. *To be published.*

Therneau, T., Atkinson, B., & Ripley, B. (2014) rpart: Recursive partitioning and regression trees. R package version, 4-1.

The articles of our own work can be found at `https://elisedusseldorp.nl/`

See Also

`FEmrt`, `REmrt`, `summary.FEmrt`,`summary.REmrt`, `plot.FEmrt`,`plot.REmrt`,`predict.FEmrt`,`predict.REmrt`

`BootCI` *This functions performs bootstrap to compute the confidence intervals for the subgroup effect size estimates. This function is only applicable to Random effects metaregression trees with 2 terminal nodes or more.*

Description

This functions performs bootstrap to compute the confidence intervals for the subgroup effect size estimates. This function is only applicable to Random effects metaregression trees with 2 terminal nodes or more.

Usage

```
BootCI(Metatree, nboot = 50)
```

Arguments

`Metatree`	fitted tree of class `REmrt`.
`nboot`	number of bootstrap samples.

Value

tree containing the input tree, the Bootstrap estimates for the effect sizes and standard errors, Bootstrap estimate for tau2, and the Bootstrap bias correction.

Examples

```
set.seed(12345)
data(dat.BCT2009)
library(Rcpp)
REtree <- REmrt(g ~ T1 + T2+ T4 +T25, vi = vi, data = dat.BCT2009, c.pruning = 0)
BootTree<-BootCI(REtree, nboot = 3)
summary(BootTree)
```

`dat.balanced` *A simulated meta-analytic data set with balanced pure interaction effects*

Description

Data simulated from a true model with pure interactions between two moderators: x1, x2. If x1 = 0 and x2 = 1 or x1 = 1 and x2 = 0, the true effect size is 0.50. Otherwise, the true effect size is 0.

Usage

```
data(dat.balanced)
```

Format

A data frame of 60 studies with 4 moderators

- efk: The effect size of each study expressed as Hedges' g
- vark: The sampling variance of the effect size
- x1 to x4: Four randomly generated moderators. x1, x2, and x4 are dichotomous variables, x3 is a continuous variable generated from uniform distribution.

`dat.BCT2009` *A subset of data from Michie et al. (2009)*

Description

The complete data consist of 101 studies reporting 122 interventions targeted at physical activity and healthy eating. In this subset of the data, the interventions that include at least one of the motivation-enhancing behaviour change techniques (BCTs) were selected (N = 106).

Usage

```
data(dat.BCT2009)
```

Format

A data frame of 106 interventions with five motivation-enhancing behavior change techniques (BCTs).

- study: The name of the intervention.
- g: The effect size of each intervention.
- vi: The sampling variance of the effect size.
- T1: Indicating whether the BCT1 "Provide information about behavior-health link" was used by the intervention. "1" for used and "0" for not used.
- T2: Indicating whether the BCT2 "Provide information on consequences" was used by the intervention. "1" for used and "0" for not used.
- T3: Indicating whether the BCT3 "Provide information about other's approval" was used by the intervention. "1" for used and "0" for not used.
- T4: Indicating whether the BCT4 "Prompt intention formation" was used by the intervention. "1" for used and "0" for not used.
- T25: Indicating whether the BCT25 " Motivational interviewing" was used by the intervention. "1" for used and "0" for not used.

Details

IMPORTANT: for questions about these data contact Juan Claramunt: j.claramunt.gonzalez@fsw.leidenuniv.nl.

References

If you use these data, please refer to: Michie, S., Abraham, C., Whittington, C., McAteer, J., & Gupta, S. (2009). Effective techniques in healthy eating and physical activity interventions: a meta-regression. *Health Psychology, 28(6)*, 690.

An application of (a preliminary version of) meta-CART to this data set is given in: Dusseldorp, E., Van Genugten, L., van Buuren, S., Verheijden, M. W., & van Empelen, P. (2014). Combinations of techniques that effectively change health behavior: Evidence from Meta-CART analysis. *Health Psychology, 33(12)*, 1530.

`FEmrt` *Fixed effect meta-tree*

Description

A function to fit fixed effect meta-trees to meta-analytic data. The model is assuming a fixed effect within subgroups and between subgroups. The tree growing process is equivalent to the approach described in Li et al. (2017) using fixed effect weights in the function **rpart()** developed by Therneau, Atkinson & Ripley (2014).

Usage

```
FEmrt(
  formula,
  data,
  vi,
  subset,
  c.pruning = 0.5,
  perm = NULL,
  sss = FALSE,
  lookahead = FALSE,
  cp = 1e-04,
  maxdepth = 10L,
  minsplit = 6,
  xval = 10,
  minbucket = 3,
  a = 50,
  alpha.endcut = 0.02,
  multi.start = T,
  n.starts = 3,
  ...
)
```

Arguments

`formula`	A formula, with an outcome variable (usually the effect size) and the potential moderator variables but no interaction terms.
`data`	A data frame of a meta-analytic data set, including the study effect sizes, sampling variance, and the potential moderators.
`vi`	sampling variance of the effect size.
`subset`	optional expression that selects only a subset of the rows of the data.
`c.pruning`	A non-negative scalar.The pruning parameter to prune the initial tree by the "c*standard-error" rule.
`perm`	the number of data sets to permute for the permutation test. If set as NULL, then perumuation test will not be performed
`sss`	boolean indicating whether the SSS strategy is used or not.
`lookahead`	a boolean argument indicating whether to apply the "look-ahead" strategy when fitting the tree
`cp`	complexity parameter as in rpart.
`maxdepth`	set the maximum depth of any node of the final tree, with the root node counted as depth 0
`minsplit`	the minimum number of observations that must exist in a node in order for a split to be attempted.
`xval`	number of cross-validations.
`minbucket`	the minimum number of observations in any terminal <leaf> node.

`a`	parameter used in the sss to determine the slope of the logistic function that replaces the indicator function.
`alpha.endcut`	parameter used in the splitting algorithm to avoid the endcut preference problem.
`multi.start`	boolean indicating whether multiple starts must be used
`n.starts`	number of multiple starts
`...`	Additional arguments passed to the tree growing algorithm based on **rpart**.

Value

If (a) moderator effect(s) is(are) detected, the function will return a `FEmrt` object including the following components:

tree: The pruned tree that represents the moderator effect(s) and interaction effect(s) between them.

n: The number of the studies in each subgroup

Qb: The between-subgroups Q-statistic

df: The degree of freedoms of the between-subgroups Q test

pval.Qb: The p-value of the between-subgroups Q test

Qw: The within-subgroup Q-statistic in each subgroup

g: The subgroup summary effect size, based on Hedges'g

se: The standard error of the subgroup summary effect size

zval: The test statistic of the subgroup summary effect size

pval: The p-value for the test statistics of the subgroup summary effect size

ci.lb: The lower bound of the confidence interval

ci.ub: The upper bound of the confidence interval

call: The matched call

If no moderator effect is detected, the function will return a `FEmrt` object including the following components:

n: The total number of the studies

Q: The Q-statistic of the heterogeneity test

df: The degrees of freedom of the heterogeneity test

pval.Q: The p-value of the heterogeneity test

g: The summary effect size for all studies

se: The standard error of the summary effect size

zval: The test statistic of the summary effect size

pval: The p-value of the test statistic of the summary effect size

ci.lb: The lower bound of the confidence interval for the summary effect size

ci.ub: The upper bound of the confidence interval for the summary effect size

formula: The formula provided as input.

call: The matched call

References

Dusseldorp, E., van Genugten, L., van Buuren, S., Verheijden, M. W., & van Empelen, P. (2014). Combinations of techniques that effectively change health behavior: Evidence from meta-cart analysis. *Health Psychology, 33(12)*, 1530-1540. doi: 10.1037/hea0000018.

Li, X., Dusseldorp, E., & Meulman, J. J. (2017). Meta-CART: A tool to identify interactions between moderators in meta-analysis. *British Journal of Mathematical and Statistical Psychology, 70(1)*, 118-136. doi: 10.1111/bmsp.12088.

Therneau, T., Atkinson, B., & Ripley, B. (2014) rpart: Recursive partitioning and regression trees. R package version, 4-1.

See Also

`summary.FEmrt`, `plot.FEmrt`, `rpart`,`rpart.control`

Examples

```
data(dat.BCT2009)
library(Rcpp)
FEtree <- FEmrt(g ~ T1 + T2+ T4 + T25, vi = vi, data = dat.BCT2009, c.pruning = 0.5)
print(FEtree)
summary(FEtree)
#plot(FEtree)
```

`plot.FEmrt` *Visualisation of a FE meta-tree*

Description

Plot function for a `FEmrt` object. The plot shows the result of `FEmrt`. The plot function uses the plot method from the package **ggplot2**

Usage

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

Arguments

`x`	A FEmrt object.
`...`	additional arguments to pass

Details

For categorical variables we recommend to use short names for levels to avoid overlapping labels at split points.

Value

A plot visualizing an FE meta-tree

`plot.REmrt` *Visualisation of a RE meta-tree*

Description

Plot function for a `REmrt` object. The plot shows the result of `REmrt`. The plot function uses the plot method from the package **ggplot2**

Usage

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

Arguments

`x`	A REmrt object.
`...`	Additional arguments to pass.

Details

For categorical variables we recommend to use short names for levels to avoid overlapping labels at split points.

Value

A plot visualizing an RE meta-tree

`plotPV` *Visualization of a initial RE meta-tree with frequency coloring*

Description

Plot function for a `REmrt` object. The plot shows the initial tree result of `REmrt` with the nodes being colored based on how often they appear as nodes in the amount of iterations given. The plot function uses the plot method from the package **ggplot2**

Usage

```
plotPV(x, iter = 10, c.pruning = 0, ...)
```

Arguments

`x`	A REmrt object.
`iter`	Amount of iterations for frequency check.
`c.pruning`	The pruning strictness value (used in the c * SD rule in the pruning process)
`...`	Additional arguments to pass.

Details

For categorical variables we recommend to use short names for levels to avoid overlapping labels at split points.

Value

A plot visualizing an RE meta-tree with node frequencies.

`predict.FEmrt` *Predictions from a fitted FE metacart object*

Description

Returns a data frame of predicted effect sizes and moderators from a fitted metacart object

Usage

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

Arguments

`object`	fitted model object of class "FEmrt".
`newdata`	data frame containing the values for which predictions are required.
`...`	Arguments that pass to other methods.

Value

A data frame containing the predicted effect size, the moderators, and the corresponding node labels in the fitted tree.

`predict.REmrt` *Predictions from a fitted RE metacart object*

Description

Returns a data frame of predicted effect sizes and moderators from a fitted metacart object

Usage

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

Arguments

`object`	fitted model object of class "REmrt".
`newdata`	data frame containing the values for which predictions are required.
`...`	Arguments that pass to other methods.

Value

A data frame containing the predicted effect size, the moderators, and the corresponding node labels in the fitted tree.

`print.FEmrt` *Print function for FEmrt*

Description

Print the results of a FEmrt object

Usage

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

Arguments

`x`	fitted tree of class `FEmrt`.
`...`	additional arguments to be passed.

Details

The function returns the objects concerning the analysis results.

Value

Printed output of a FE meta-tree

`print.REmrt` *Print function for REmrt*

Description

Print the results of a REmrt object

Usage

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

Arguments

`x`	fitted tree of class `FEmrt`.
`...`	additional arguments to be passed.

Details

The function returns the results (e.g., the value of the Q-between) after each split of the tree.

Value

Printed output of a RE meta-tree

`rcpp_hello_world` *Simple function using Rcpp*

Description

Simple function using Rcpp

Usage

```
rcpp_hello_world()
```

Examples

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

`REmrt` *Random effects meta-tree*

Description

A function to fit a random effects meta-tree

Usage

```
REmrt(
  formula,
  data,
  vi,
  c.pruning = 0,
  maxL = 5,
  minsplit = 6,
  cp = 1e-05,
  minbucket = 3,
  xval = 10,
  lookahead = TRUE,
  sss = TRUE,
  alpha.endcut = 0.02,
  a = 50,
  multi.start = TRUE,
  n.starts = 3,
  perm = 25,
  seed = NULL,
  ...
)
```

Arguments

`formula`	A formula, with a response variable (usually the effect size) and the potential moderator variables but no interaction terms.
`data`	A data frame of a meta-analytic data set, including the study effect sizes, sampling variance, and the potential moderators.
`vi`	sampling variance of the effect size.
`c.pruning`	A non-negative scalar.The pruning parameter to prune the initial tree by the "c*standard-error" rule.
`maxL`	the maximum number of splits
`minsplit`	the minimum number of studies in a parent node before splitting
`cp`	the stopping rule for the decrease of between-subgroups Q. Any split that does not decrease the between-subgroups Q is not attempted.
`minbucket`	the minimum number of the studies in a terminal node
`xval`	the number of folds to perform the cross-validation

`lookahead`	an argument indicating whether to apply the "look-ahead" strategy when fitting the tree
`sss`	boolean parameter indicating whether the SSS algorithm must be used.
`alpha.endcut`	parameter used in the splitting algorithm to avoid the endcut preference problem.
`a`	parameter used in the sss to determine the slope of the logistic function that replaces the indicator function.
`multi.start`	boolean indicating whether multiple starts must be used
`n.starts`	number of multiple starts
`perm`	the number of permuted data sets, if NULL then no permutation test is performed.
`seed`	Seed used to obtain replicable results. Default = NULL.
`...`	Additional arguments to be passed.

Value

If (a) moderator effect(s) is(are) detected, the function will return a list including the following objects:

tree: A data frame that represents the tree, with the Q-between and the residual heterogeneity (tau^2) after each split.

n: The number of the studies in each subgroup

moderators: the names of identified moderators

Qb: The between-subgroups Q-statistic

tau2: The estimate of the residual heterogeneity

df: The degrees of freedom of the between-subgroups Q test

pval.Qb: The p-value of the between-subgroups Q test

g: The subgroup summary effect size, based on Hedges'g

se: The standard error of subgroup summary effect size

zval: The test statistic of the subgroup summary effect size

pval: The p-value of the test statistic of the subgroup summary effect size

ci.lb: The lower bound of the confidence interval

ci.ub: The upper bound of the confidence interval

call: The matched call

cptable: The cross-validation table

data: the data set subgrouped by the fitted tree

If no moderator effect is detected, the function will return a list including the following objects:

n: The total number of the studies

Q: The Q-statistics for the heterogeneity test

df: The degree of freedoms of the heterogeneity test pval.Q: The p-value for the heterogeneity test

g: The summary effect size for all studies (i.e., the overall effect size)

se: The standard error of the summary effect size

zval: The test statistic of the summary effect size

pval: The p-value for the test statistic of the summary effect size

ci.lb: The lower bound of the confidence interval for the summary effect size

ci.ub: The upper bound of the confidence interval for the summary effect size

formula: The formula provided as input.

call: The matched call

initial.tree: The initial tree obtained before pruning.

See Also

`summary.REmrt`, `plot.REmrt`

Examples

```
#set.seed is required to obtain the same tree
#due to the use of a probabilistic algorithm for pruning
set.seed(12345)
data(dat.BCT2009)
library(Rcpp)
REtree <- REmrt(g ~ T1 + T2+ T4 +T25, vi = vi, data = dat.BCT2009, c.pruning = 0)
summary(REtree)
plot(REtree)

#You can obtain the non-pruned tree by calling the initial.tree output argument
REtree$initial.tree
```

`SimData` *A simulated meta-analytic data set*

Description

Data simuated from a true model with a three-way interaction between three moderators: m1, m2 and m3. If the values of the three moderators are all "B"s the true effect size will be 0.80. Otherwise, the true effect size is 0.

Usage

```
data(SimData)
```

Format

A data frame of 120 studies with 5 moderators

- efk: The effect size of each study expressed as Hedges' g
- vark: The sampling variance of the effect size
- m1 to m5: Five randomly generated moderators. m1 and m2 have two levels (A and B), whereas m3, m4 and m5 have three levels (A, B and C)

`summary.FEmrt` *Summary of the results of a FE meta-tree object*

Description

Summary of the results of a FE meta-tree object

Usage

```
## S3 method for class 'FEmrt'
summary(object, digits = 3, ...)
```

Arguments

`object`	fitted tree of class `FEmrt`.
`digits`	specified number of decimals in the printed results.
`...`	additional arguments to be passed.

Details

If no moderator effect is detected, the summary function will show the standard meta-analysis results. Otherwise, the summary function will show the subgroup meta-analysis results, with the significance test results for moderator effects, the split points of the moderators, and the estimated subgroup summary effect sizes.

Value

Summary of a FE meta-tree

`summary.REmrt` *Summary of the results of a RE meta-tree object*

Description

Summary of the results of a RE meta-tree object

Usage

```
## S3 method for class 'REmrt'
summary(object, digits = 3, ...)
```

Arguments

`object`	fitted tree of class `REmrt`.
`digits`	specified number of decimals in the printed results.
`...`	additional arguments to be passed.

Details

If no moderator effect is detected, the summary function will show the standard meta-analysis results. Otherwise, the summary function will show the subgroup meta-analysis results, with the significance test results for moderator effects, the split points of the moderators, and the estimated subgroup summary effect sizes.

Value

Summary of a RE meta-tree

Index