Package 'ExactTree'

August 7, 2026

Type Package

Title Exact Tree

Version 0.1.1

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

Description

Grows optimally global trees based on the algorithm defined in ``An exact dynamic programming algorithm for regression and classification trees'' (2011). It is possible to obtain both classification and regression trees depending on the measurement level of the outcome variable. The algorithm is based on the dynamic programming principle and guarantees that the resulting tree is optimal with respect to the chosen impurity measure. The package also includes a function to visualize the resulting trees, a function that summarizes the tree with its splitting information and leaf information, and a predict function that provides estimates for a new dataset given a model fit.

Depends R (>= 3.0.2), partykit, pracma, stats, grid, utils, graphics, formula.tools

Imports gridtext, DescTools, methods, rpart

License GPL-3

Encoding UTF-8

Author Juan Claramunt Gonzalez [aut, cre, cph], Bart Jan van Os [aut], Elise Dusseldorp [aut]

NeedsCompilation yes

RoxygenNote 7.3.2

Repository CRAN

Date/Publication 2026-08-07 10:30:14 UTC

Contents

as.party.ETree 2
ETree 2
ETree.control 4
plot.ETree 6

predict.ETree 7
prune.ETree 8
SelectVar 9
summary.ETree 10

Index **12**

`as.party.ETree` *Transformation of a Exact Tree object to party object*

Description

Transformation of a Exact Tree object to party object

Usage

```
## S3 method for class 'ETree'
as.party(obj, nodeID = 1L, ...)
```

Arguments

`obj`	tree of class `ETree`.
`nodeID`	Node identification.
`...`	additional arguments to be passed.

Value

object transformed to a `constparty` object.

`ETree` *Main function of the package. It performs the Exact Tree method.*

Description

Main function of the package. It performs the Exact Tree method.

Usage

```
ETree(
  formula = NULL,
  data = NULL,
  map = NULL,
  original = NULL,
  round = NULL,
  discretize = NULL,
  selV = NULL,
  control = NULL,
  verbose = TRUE
)
```

Arguments

formula	a description of the model to be fit. The format is Y ~ X1 + ... + Xn, where the variable before ~ represents the dependent variable and the variables after the ~ are the independent variables.
data	Dataset to be analyzed. Note: If data contains ordinal variables, transform them to numerical before using this function. Otherwise, they will be treated as nominal variables.
map	[ori vars] sets for each variable whether the tree will depict thresholds based upon original values (var number), or recoded values (0).
original	[TRUE/FALSE] sets for each variable whether the tree will depict thresholds based upon original values (true), or recoded values (false).
round	[factors] autorecoding based upon rounded categories; the categories will be round after multiplying with the factor, and devided by the factor after rounding
discretize	[ncat] optimal discretization while minimizing SS within a group, indicate with a number how many categories;set ncat=0 for each variable that is to be left unaltered
selV	list containing the output obtained using SelectVar. Use this list instead of the previous inputs if you want to perform SelectVars outside ETree.
control	a list with control parameters as returned by ETree.control.
verbose	logical, if TRUE, prints information about the progress of the algorithm.

Details

The function results in a global tree (for given maxsize and maxdepth) and transforms the output to obtain summary information and plot the tree.

Value

Returns the following 6 elements:

h	contains the objective function value (fit) of the best tree.
Tree	contains the largest Tree table.
hAll	(OPTIONAL) contains the objective function values for all trees, if more than one is requested
TAll	(OPTIONAL) contains Tree tables for all trees, if more than one is requested.
Transf_Trees	contains an object of class ExactTree that can be used in plot.ETree and summary.ETree
CVOutput	Cross validation results for all the requested trees.

See Also

summary.ETree, SelectVar, plot.ETree,ETree.control, predict.ETree,prune.ETree

Examples

```
data(iris)
# Fit an Exact Tree model
controlEtree <- ETree.control(measure=0, maxsize = 4, maxdepth = 3,
minbucket = 5, ncv=5, alltreesizes = FALSE)
ETree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
control= controlEtree, data = iris)
```

`ETree.control` *Control Parameters for ETree Algorithm*

Description

Various parameters that control aspects of the "ETree" algorithm.

Usage

```
ETree.control(
  measure = NULL,
  maxsize = 4,
  maxdepth = 2,
  minbucket = 10,
  minheterogeneity = 0.05,
  ncv = 10,
  cvVector = NULL,
  heterogeneityonly = NULL,
  BoundH = NULL,
  branchandbound = NULL,
  alltreesizes = FALSE,
  sortedpsplits = NULL,
  lookaheadheuristic = NULL
)
```

Arguments

`measure`	0 = minimize residual Sum of Squares (continuous Y) 1 = minimize Misclassification Rate (discrete Y) 3 = minimize the Risk Sum P(A)*R(A) (discrete Y) Measure = scalar or Measure=c(3,c(Prior,LossM))
`maxsize`	if null: only maxdepth restrictions are applied if defined: the maximum number of terminal nodes for the best tree
`maxdepth`	The maximum number of layers in a tree, NOT INCLUDING THE LAYER 0. maxdepth=0 means a tree with just one node (the root node), maxdepth=1 means one split and two terminal nodes, etc.

`minbucket` defines the minimum number of observations in a terminal node if the number of observations in a node is equal to this minimum the node becomes a possible terminal node; if the number of observations in a node is smaller, the node is set illegal and disregarded.

`minheterogeneity` defines the situation when a node A is considered Very Homogeneous and is not allowed to be split further, i.e. the node A becomes a possible terminal node (|A| = number of objects in node)

`ncv` specify N for N-fold cross-validation if NCV = 0, no N-fold cross-validation if NCV<0, abs(N)-fold cross-validation ONLY (no tree estimation) if NULL (default), NCV=0

`cvVector` <vector> process N-fold cross-validation according to the classes defined by <vector> vector is assumed to assign each observational unit to one of N-classes, number 1,...,N

`heterogeneityonly` (TRUE,FALSE) Algorithm will only give Heterogeneity/Impurity not a tree table => setting to false will give treetable as well

`BoundH` optional bound for best tree value, for BranchandBound

`branchandbound` (TRUE,FALSE) Algorithm will use branch and bound rules to speed up => for maxsize tree only, h only => make sure you specify boundH

`alltreesizes` (FALSE,TRUE) algorithm will compute multiple trees of size<=size restriction

`sortedpsplits` (TRUE,FALSE) Algorithm will try to speed up use a sorting logic. => suitable for continuous predictors with many categories

`lookaheadheuristic` Specify treedepth=the number of levels the heuristic will lookahead. When growing a tree with lookahead search (heurist approach), this specifies the maximum depth with respect to which any local split is optimized. When setting this depth = 1, this ammounts to convential tree growing.

Value

A list containing the options for the function ETree.

See Also

ETree

Examples

```
ETree.control(measure=0, maxsize = 6, maxdepth = 4, minbucket = 5, ncv=10, alltreesizes = TRUE)
```

plot.ETree *Visualisation of a Exact Tree*

Description

Visualisation of a Exact Tree

Usage

```
## S3 method for class 'ETree'
plot(x, TerminalNodes = NULL, digits = 2, ...)
```

Arguments

x	transformed tree of class ExactTree.
TerminalNodes	Number of terminal nodes of the tree to plot.
digits	specified number of decimal places of the splitpoints in the graph (default is 2).
...	additional arguments to be passed.

Value

A plot of the tree with the specified number of terminal nodes.

References

Torsten Hothorn and Achim Zeileis (2013). partykit: A Toolkit for Recursive Partytioning. R package version 0.1-5.

See Also

ETree,summary.ETree

Examples

```
data(iris)
# Fit an Exact Tree model
controlEtree <- ETree.control(measure=0, maxsize = 4, maxdepth = 3,
minbucket = 5, ncv=5, alltreesizes = FALSE)
tree<-ETree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
control= controlEtree, data = iris)
plot(tree, TerminalNodes=4)
```

predict.ETree *Predictions for new data with a ETree object*

Description

Predicts for (new) subjects the outcome variable based on a fitted ETree object.

Usage

```
## S3 method for class 'ETree'
predict(object, newdata = NULL, type = "pred", depth = NULL, ...)
```

Arguments

object	an object of the class "ETree".
newdata	a data frame with data on new subjects for whom predictions should be made. The data frame should contain at least the variables used in the splits of the fitted tree. It is not necessary to include the treatment variable.
type	character string denoting the type of predicted object to be returned. The default is set to type="pred": a vector with predicted treatment subgroup classes per subject is returned. If set to "matrix", a matrix is returned with the leaf and corresponding node of the tree to which a subject is assigned.
depth	If alltreesizes was set to TRUE in ETree, you need to specify the depth of the tree you want to use in the predict function. This parameter should be equal to the number of the tree you want to use in the TAll output of ETree. If NULL, the largest tree is used.
...	optional additional arguments.

Value

One of the following objects is returned depending on output type specified in the function:

If type="pred": vector of the predicted outcome for every individual in the data set. Returns NA for subjects with missing values on one or more of the splitting variables.

If type="matrix": a matrix with predicted locations of subjects within the fitted tree. The leaf numbers are in the first column and the corresponding node numbers in the second column. Returns NA for subjects with missing values on one or more of the splitting variables.

If type="prob": a matrix with probabilities.

See Also

ETree, ETree.control

Examples

```
data(iris)
trainingIris<-iris[1:100,]
# Fit an Exact Tree model
controlEtree <- ETree.control(measure=0, maxsize = 4, maxdepth = 3,
minbucket = 5, ncv=5, alltreesizes = FALSE)
tree<-ETree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
control= controlEtree, data = trainingIris)
testIris<-iris[101:150,]
predictions<-predict(tree, newdata=testIris, type="pred", depth=1)
```

`prune.ETree` *Pruning of a Exact Tree*

Description

Determines the optimally pruned size of the tree by applying the one standard error rule to the results from the bias-corrected bootstrap procedure.

Usage

```
## S3 method for class 'ETree'
prune(tree, pp = 1, ...)
```

Arguments

`tree`	fitted tree of the class `ETree`.
`pp`	pruning parameter, the constant (c) to be used in the $c*$standard error rule. The default value is 1.
`...`	optional additional arguments.

Details

The one standard error rule for `ETrees` uses the estimates of the bias-corrected criterion value (C) and its standard error for each value of L (= maximum number of leaves). The optimally pruned tree corresponds to the smallest tree with a bias-corrected C higher or equal to the maximum bias-corrected C minus its standard error.

Value

Returns an object of class `ETree`. The number of leaves of this object is equal to the optimally pruned size of the tree.

SelectVar *This function preprocess the data for the Optimal Trees function.*

Description

This function preprocess the data for the Optimal Trees function.

Usage

```
SelectVar(
  formula = NULL,
  data,
  Names = NULL,
  YSelected = NULL,
  XSelected = NULL,
  map = NULL,
  original = NULL,
  round = NULL,
  discretize = NULL,
  verbose = TRUE
)
```

Arguments

formula	a description of the model to be fit. The format is Y1 + ... + Yn ~ X1 + ... + Xn, where the variable before ~ represents the dependent variables and the variables after the ~ are the independent variables.
data	Dataset to be analyzed. If data contains ordinal variables, order the factor levels before using this function. Otherwise, the factors will be ordered alphabetically and the results will not be correct.
Names	names of the variables in the data. If empty, it uses the column names of Data
YSelected	index corresponding to the first Selected column.
XSelected	Indices corresponding to the selected columns that will form X
map	[ori vars] sets for each variable whether the tree will depict thresholds based upon original values (var number), or recoded values (0).
original	[TRUE/FALSE] sets for each variable whether the tree will depict thresholds based upon original values (true), or recoded values (false).
round	[factors] autorecoding based upon rounded categories; the categories will be round after multiplying with the factor, and divided by the factor after rounding. Set round=0 for variables that are not rounded (e.g. discretized variables).
discretize	[ncat] optimal discretization while minimizing SS within a group, set ncat=0 for each variable that is to be left unaltered
verbose	logical, if TRUE, prints information about the progress of the algorithm.

Details

The function selects the Y and X variables according to the inputs and also returns Desc, a list containing variable cutting points and variable names. These 3 elements are required by the OptimalTrees function.

Value

Returns the following 3 elements:

`Y`	dataset containing the Y variables.
`X`	dataset containing the X variables.
`Desc`	List containing the variable names and their corresponding cutting points.

See Also

`summary.ETree`, `ETree`, `plot.ETree`, `predict.ETree`

Examples

```
data(mtcars)
dataSelection<-SelectVar( mpg ~ cyl + hp + wt, data = mtcars, discretize= c(0, 10, 10))
```

---

summary.ETree *Summarizing Exact Trees Information*

Description

Summary method for an object of class `ETree`.

Usage

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

Arguments

`object`	a `ETree` object. This can be the output of `ETree`.
`TerminalNodes`	Number of terminal nodes of the tree to summarize.
`digits`	specified number of decimal places (default is 3).
`...`	optional additional arguments.

Details

This function is a method for the generic function summary for class `ETree`. It extracts the following essential components from a `ETree` object: 1) Split information; 2) Leaf information, and 3) CV information.

Value

prints a summarized version of the ETree output.

Examples

```
data(iris)
# Fit an Exact Tree model
controlEtree <- ETree.control(measure=0, maxsize = 4, maxdepth = 3,
minbucket = 5, ncv=5, alltreesizes = FALSE)
tree<-ETree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
control= controlEtree, data = iris)
summary(tree, TerminalNodes=4)
```

Index

∗ **as.party**
as.party.ETree, 2
plot.ETree, 6
∗ **exact**
ETree, 2
SelectVar, 9
∗ **plot**
plot.ETree, 6
∗ **summary**
summary.ETree, 10
∗ **tree**
ETree, 2
prune.ETree, 8
SelectVar, 9

as.party.ETree, 2

ETree, 2, *5–7*, *10*
ETree.control, *3*, 4, *7*

plot.ETree, *3*, 6, *10*
predict.ETree, *3*, 7, *10*
prune.ETree, *3*, 8

SelectVar, *3*, 9
summary.ETree, *3*, *6*, *10*, 10