

# Package ‘evtree’

May 8, 2026

**Title** Evolutionary Learning of Globally Optimal Trees

**Version** 1.0-8

**Date** 2019-05-19

**Description** Commonly used classification and regression tree methods like the CART algorithm are recursive partitioning methods that build the model in a forward stepwise search. Although this approach is known to be an efficient heuristic, the results of recursive tree methods are only locally optimal, as splits are chosen to maximize homogeneity at the next step only. An alternative way to search over the parameter space of trees is to use global optimization methods like evolutionary algorithms. The 'evtree' package implements an evolutionary algorithm for learning globally optimal classification and regression trees in R. CPU and memory-intensive tasks are fully computed in C++ while the 'partykit' package is leveraged to represent the resulting trees in R, providing unified infrastructure for summaries, visualizations, and predictions.

**Depends** R (>= 3.3.0), partykit

**Suggests** Formula, kernlab, lattice, mlbench, multcomp, party, rpart, xtable

**LazyData** yes

**License** GPL-2 | GPL-3

**Author** Thomas Grubinger [aut, cre],  
Achim Zeileis [aut] (ORCID: <<https://orcid.org/0000-0003-0918-3766>>),  
Karl-Peter Pfeiffer [aut]

**Maintainer** Thomas Grubinger <ThomasGrubinger@gmail.com>

**Repository** CRAN

**Repository/R-Forge/Project** partykit

**Repository/R-Forge/Revision** 2594

**Repository/R-Forge/DateTimeStamp** 2019-05-26 18:21:12

**Date/Publication** 2019-05-26 19:40:04 UTC

**NeedsCompilation** yes

## Contents

|                               |           |
|-------------------------------|-----------|
| BBBClub . . . . .             | 2         |
| ContraceptiveChoice . . . . . | 4         |
| evtree . . . . .              | 5         |
| evtree.control . . . . .      | 6         |
| GermanCredit . . . . .        | 7         |
| MAGICGammaTelescope . . . . . | 9         |
| StatlogHeart . . . . .        | 11        |
| <b>Index</b>                  | <b>13</b> |

---

|         |                               |
|---------|-------------------------------|
| BBBClub | <i>Bookbinder's Book Club</i> |
|---------|-------------------------------|

---

### Description

Marketing case study about a (fictitious) American book club to whose customers a book about “The Art History of Florence” was advertised.

### Usage

```
data("BBBClub")
```

### Format

A data frame containing 1,300 observations on 11 variables.

**choice** factor. Did the customer buy the advertised book?

**gender** factor indicating gender.

**amount** total amount of money spent at the BBB Club.

**freq** number of books purchased at the BBB Club.

**last** number of months since the last purchase.

**first** number of months since the first purchase.

**child** number of children’s books purchased.

**youth** number of youth books purchased.

**cook** number of cookbooks purchased.

**diy** number of do-it-yourself books purchased.

**art** number of art books purchased.

## Details

The data is a marketing case study about a (fictitious) American book club. taken from the *Marketing Engineering* textbook of Lilien and Rangaswamy (2004). In this case study, a brochure of the book “The Art History of Florence” was sent to 20,000 customers and 1,806 of which bought the book. A subsample of 1,300 customers is provided in BBBClub for building a predictive model for choice.

The use of a cost matrix is suggested for this dataset. Classifying a customer that purchased the book as a non-buyer is worse (cost = 5), than it is to classify a customer that did not purchase the book as a buyer (cost = 1).

## Source

Complements to Lilien and Rangaswamy (2004).

## References

Lilien GL, Rangaswamy A (2004). *Marketing Engineering: Computer-Assisted Marketing Analysis and Planning*, 2nd edition. Victoria, BC: Trafford Publishing.

## Examples

```
## Not run:
## data, packages, random seed
data("BBBClub", package = "evtree")
library("rpart")
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)

## learn trees
ev <- evtree(choice ~ ., data = BBBClub, minbucket = 10, maxdepth = 2)
rp <- as.party(rpart(choice ~ ., data = BBBClub, minbucket = 10, model = TRUE))
ct <- ctree(choice ~ ., data = BBBClub, minbucket = 10, mincrit = 0.99)

## visualization
plot(ev)
plot(rp)
plot(ct)

## accuracy: misclassification rate
mc <- function(obj) 1 - mean(predict(obj) == BBBClub$choice)
c("evtree" = mc(ev), "rpart" = mc(rp), "ctree" = mc(ct))

## complexity: number of terminal nodes
c("evtree" = width(ev), "rpart" = width(rp), "ctree" = width(ct))

## compare structure of predictions
ftable(tab <- table(evtree = predict(ev), rpart = predict(rp),
  ctree = predict(ct), observed = BBBClub$choice))

## compare customer predictions only (absolute, proportion correct)
sapply(c("evtree", "rpart", "ctree"), function(nam) {
```

```

mt <- margin.table(tab, c(match(nam, names(dimnames(tab))), 4))
c(abs = as.vector(rowSums(mt))[2],
  rel = round(100 * prop.table(mt, 1)[2, 2], digits = 3))
})

## End(Not run)

```

---

ContraceptiveChoice    *Contraceptive Method Choice*

---

### Description

Data of married women who were either not pregnant or do not know if they were at the time of interview. The task is to predict the women's current contraceptive method choice (*no use, long-term methods, short-term methods*) based on her demographic and socio-economic characteristics.

### Usage

```
data("ContraceptiveChoice")
```

### Format

A data frame containing 1,437 observations on 10 variables.

**wifes\_age** wife's age in years.

**wifes\_education** ordered factor indicating the wife's education, with levels "low", "medium-low", "medium-high" and "high".

**husbands\_education** ordered factor indicating the wife's education, with levels "low", "medium-low", "medium-high" and "high".

**number\_of\_children** number of children.

**wifes\_religion** binary variable indicating the wife's religion, with levels "non-Islam" and "Islam".

**wife\_now\_working** binary variable indicating if the wife is working.

**husbands\_occupation** ordered factor indicating the husbands occupation, with levels "low", "medium-low", "medium-high" and "high".

**standard\_of\_living\_index** standard of living index with levels "low", "medium-low", "medium-high" and "high".

**media\_exposure** binary variable indicating media exposure, with levels "good" and "not good".

**contraceptive\_method\_used** factor variable indicating the contraceptive method used, with levels "no-use", "long-term" and "short-term".

### Source

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey and was created by Tjen-Sien Lim.

It has been taken from the UCI Repository Of Machine Learning Databases at

<http://archive.ics.uci.edu/ml/>.

## References

Lim, T.-S., Loh, W.-Y. & Shih, Y.-S. (1999). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. *Machine Learning*, 40(3), 203–228.

## Examples

```
data("ContraceptiveChoice")
summary(ContraceptiveChoice)
## Not run:
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)
contt <- evtree(contraceptive_method_used ~ . , data = ContraceptiveChoice)
contt
table(predict(contt), ContraceptiveChoice$contraceptive_method_used)
plot(contt)

## End(Not run)
```

---

 evtree

*Evolutionary Learning of Globally Optimal Trees*


---

## Description

Learning of globally optimal classification and regression trees by using evolutionary algorithms.

## Usage

```
evtree(formula, data, subset, na.action, weights,
       control = evtree.control(...), ...)
```

## Arguments

|                         |  |
|-------------------------|--|
| formula                 | a symbolic description of the model to be fit, no interactions should be used. |
| data, subset, na.action | arguments controlling formula processing via <a href="#">model.frame</a> .     |
| weights                 | optional integer vector of case weights.                                       |
| control                 | a list of control arguments specified via <a href="#">evtree.control</a> .     |
| ...                     | arguments passed to <a href="#">evtree.control</a> .                           |

## Details

Globally optimal classification and regression trees are learned by using evolutionary algorithm. Roughly, the algorithm works as follows. First, a set of trees is initialized with random split rules in the root nodes. Second, mutation and crossover operators are applied to modify the trees' structure and the tests that are applied in the internal nodes. After each modification step a survivor selection mechanism selects the best candidate models for the next iteration. In this evolutionary process the

mean quality of the population increases over time. The algorithm terminates when the quality of the best trees does not improve further, but not later than a maximum number of iterations specified by `niterations` in `evtree.control`.

More details on the algorithm are provided Grubinger et al. (2014) which is also provided as vignette("evtree", package = "evtree").

The resulting trees can be summarized and visualized by the `print.constparty`, and `plot.constparty` methods provided by the **partykit** package. Moreover, the `predict.party` method can be used to compute fitted responses, probabilities (for classification trees), and nodes.

## Value

An object of class `party`.

## References

Grubinger T, Zeileis A, Pfeiffer KP (2014). `evtree`: Evolutionary Learning of Globally Optimal Classification and Regression Trees in R. *Journal of Statistical Software*, **61**(1), 1-29. <http://www.jstatsoft.org/v61/i01/>

## Examples

```
## regression
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)
airq <- subset(airquality, !is.na(Ozone) & complete.cases(airquality))
ev_air <- evtree(Ozone ~ ., data = airq)
ev_air
plot(ev_air)
mean((airq$Ozone - predict(ev_air))^2)

## classification
## (note that different equivalent "perfect" splits for the setosa species
## in the iris data may be found on different architectures/systems)
ev_iris <- evtree(Species ~ ., data = iris)
## IGNORE_RDIFF_BEGIN
ev_iris
## IGNORE_RDIFF_END
plot(ev_iris)
table(predict(ev_iris), iris$Species)
1 - mean(predict(ev_iris) == iris$Species)
```

---

evtree.control

*Control for evtree*

---

## Description

Various parameters that control aspects of the `evtree` fit.

**Usage**

```
evtree.control(minbucket = 7L, minsplit = 20L, maxdepth = 9L,
  iterations = 10000L, ntrees = 100L, alpha = 1,
  operatorprob = list(pmutatemajor = 0.2, pmutateminor = 0.2,
    pcrossover = 0.2, psplit = 0.2, pprune = 0.2),
  seed = NULL, ...)
```

**Arguments**

|              |   |
|--------------|---|
| minbucket    | the minimum sum of weights in a terminal node.  |
| minsplit     | the minimum sum of weights in a node in order to be considered for splitting.   |
| maxdepth     | maximum depth of the tree. Note, that the memory requirements increase by the square of the maximum tree depth.   |
| iterations   | in case the run does not converge, it terminates after a specified number of iterations defined by iterations.  |
| ntrees       | the number of trees in the population.  |
| alpha        | regulates the complexity part of the cost function. Increasing values of alpha encourage decreasing tree sizes.   |
| operatorprob | list or vector of probabilities for the selection of variation operators. May also be specified partially in which case the default values are still used for the unspecified arguments. Always scaled to sum to 100 percent.   |
| seed         | an numeric seed to initialize the random number generator (for reproducibility). By default the seed is randomly drawn using <code>runif</code> in order to inherit the state of <code>.Random.seed</code> . If set to <code>seed = -1L</code> , the random number generator is initialized by the system time. |
| ...          | additional arguments.   |

**Value**

A list with the (potentially processed) control parameters.

---

GermanCredit

*Statlog German Credit*


---

**Description**

The dataset contains data of past credit applicants. The applicants are rated as *good* or *bad*. Models of this data can be used to determine if new applicants present a *good* or *bad* credit risk.

**Usage**

```
data("GermanCredit")
```

## Format

A data frame containing 1,000 observations on 21 variables.

**status** factor variable indicating the status of the existing checking account, with levels ... < 0 DM, 0 <= ... < 200 DM, ... >= 200 DM/salary for at least 1 year and no checking account.

**duration** duration in months.

**credit\_history** factor variable indicating credit history, with levels no credits taken/all credits paid back duly, all credits at this bank paid back duly, existing credits paid back duly till now, delay in paying off in the past and critical account/other credits existing.

**purpose** factor variable indicating the credit's purpose, with levels car (new), car (used), furniture/equipment, radio/television, domestic appliances, repairs, education, retraining, business and others.

**amount** credit amount.

**savings** factor. savings account/bonds, with levels ... < 100 DM, 100 <= ... < 500 DM, 500 <= ... < 1000 DM, ... >= 1000 DM and unknown/no savings account.

**employment\_duration** ordered factor indicating the duration of the current employment, with levels unemployed, ... < 1 year, 1 <= ... < 4 years, 4 <= ... < 7 years and ... >= 7 years.

**installment\_rate** installment rate in percentage of disposable income.

**personal\_status\_sex** factor variable indicating personal status and sex, with levels male:divorced/separated, female:divorced/separated/married, male:single, male:married/widowed and female:single.

**other\_debtors** factor. Other debtors, with levels none, co-applicant and guarantor.

**present\_residence** present residence since?

**property** factor variable indicating the client's highest valued property, with levels real estate, building society savings agreement/life insurance, car or other and unknown/no property.

**age** client's age.

**other\_installment\_plans** factor variable indicating other installment plans, with levels bank, stores and none.

**housing** factor variable indicating housing, with levels rent, own and for free.

**number\_credits** number of existing credits at this bank.

**job** factor indicating employment status, with levels unemployed/unskilled - non-resident, unskilled - resident, skilled employee/official and management/self-employed/highly qualified employee/officer.

**people\_liable** Number of people being liable to provide maintenance.

**telephone** binary variable indicating if the customer has a registered telephone number.

**foreign\_worker** binary variable indicating if the customer is a foreign worker.

**credit\_risk** binary variable indicating credit risk, with levels good and bad.

## Details

The use of a cost matrix is suggested for this dataset. It is worse to class a customer as good when they are bad (cost = 5), than it is to class a customer as bad when they are good (cost = 1).

## Source

The original data was provided by:

Professor Dr. Hans Hofmann, Institut fuer Statistik und Oekonometrie, Universitaet Hamburg, FB Wirtschaftswissenschaften, Von-Melle-Park 5, 2000 Hamburg 13

The dataset has been taken from the UCI Repository Of Machine Learning Databases at <http://archive.ics.uci.edu/ml/>.

## Examples

```
data("GermanCredit")
summary(GermanCredit)
## Not run:
gcw <- array(1, nrow(GermanCredit))
gcw[GermanCredit$credit_risk == "bad"] <- 5
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)
gct <- evtree(credit_risk ~ . , data = GermanCredit, weights = gcw)
gct
table(predict(gct), GermanCredit$credit_risk)
plot(gct)

## End(Not run)
```

---

MAGICGammaTelescope    *MAGIC Gamma Telescope*

---

## Description

The data was generated to simulate registration of high energy gamma particles in a Major Atmospheric Gamma-Ray Imaging Cherenkov (MAGIC) Gamma Telescope. The task is to distinguish *gamma rays* (signal) from *hadronic showers* (background).

## Usage

```
data("MAGICGammaTelescope")
```

## Format

A data frame containing 19,020 observations on 11 variables.

**fLength** major axis of ellipse [mm].

**fWidth** minor axis of ellipse [mm].

**fSize** 10-log of sum of content of all pixels [in #phot].

**fConc** ratio of sum of two highest pixels over fSize [ratio].

**fConc1** ratio of highest pixel over fSize [ratio].

**fAsym** distance from highest pixel to center, projected onto major axis [mm].

**fM3Long** 3rd root of third moment along major axis [mm].  
**fM3Trans** 3rd root of third moment along minor axis [mm].  
**fAlpha** angle of major axis with vector to origin [deg].  
**fDist** distance from origin to center of ellipse [mm].  
**class** binary variable class, with levels gamma (signal) and hadron (background).

### Details

Classifying a background event as signal is worse than classifying a signal event as background. For a meaningful comparison of different classifiers the use of an ROC curve with thresholds 0.01, 0.02, 0.05, 0.1, 0.2 is suggested.

### Source

The original data was provided by:

R. K. Bock, Major Atmospheric Gamma Imaging Cherenkov Telescope project (MAGIC), rkb '@' mail.cern.ch, <https://magic.mppmu.mpg.de/>

and was donated by:

P. Savicky, Institute of Computer Science, AS of CR, Czech Republic, savicky '@' cs.cas.cz

The dataset has been taken from the UCI Repository Of Machine Learning Databases at

<http://archive.ics.uci.edu/ml/>.

### References

Bock, R.K., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jirina, M., Klaschka, J., Kotrc, E., Savicky, P., Towers, S., Vaicilius, A., Wittek W. (2004). Methods for Multidimensional event Classification: a Case Study Using Images From a Cherenkov Gamma-Ray Telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(1), 511–528.

P. Savicky, E. Kotrc (2004). Experimental Study of Leaf Confidences for Random Forest. In *Proceedings of COMPSTAT*, pp. 1767–1774. Physica Verlag, Heidelberg, Germany.

J. Dvorak, P. Savicky (2007). Softening Splits in Decision Trees Using Simulated Annealing. In *Proceedings of the 8th International Conference on Adaptive and Natural Computing Algorithms, Part I*, pp. 721–729, Springer-Verlag, New-York.

### Examples

```
data("MAGICGammaTelescope")
summary(MAGICGammaTelescope)
## Not run:
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)
mgtt <- evtree(class ~ . , data = MAGICGammaTelescope)
mgtt
table(predict(mgtt), MAGICGammaTelescope$class)
plot(mgtt)

## End(Not run)
```

---

StatlogHeart

*Statlog Heart*

---

### Description

Models of this data predict the *absence* or *presence* of heart disease.

### Usage

```
data("StatlogHeart")
```

### Format

A data frame containing 270 observations on 14 variables.

**age** age in years.

**sex** binary variable indicating sex.

**chest\_pain\_type** factor variable indicating the chest pain type, with levels typical angina, atypical angina, non-anginal pain and asymptomatic.

**resting\_blood\_pressure** resting blood pressure.

**serum\_cholesterol** serum cholesterol in mg/dl.

**fasting\_blood\_sugar** binary variable indicating if fasting blood sugar > 120 mg/dl.

**resting\_electrocardiographic\_results** factor variable indicating resting electrocardiographic results, with levels 0: normal, 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) and 2: showing probable or definite left ventricular hypertrophy by Estes' criteria.

**maximum\_heart\_rate** the maximum heart rate achieved.

**exercise\_induced\_angina** binary variable indicating the presence of exercise induced angina.

**oldpeak** oldpeak = ST depression induced by exercise relative to rest.

**slope\_of\_the\_peak** ordered factor variable describing the slope of the peak exercise ST segment, with levels upsloping, flat and downsloping.

**major\_vessels** number of major vessels colored by flouroscopy.

**thal** factor variable thal, with levels normal, fixed defect and reversible defect.

**heart\_disease** binary variable indicating the presence or absence of heart disease.

### Details

The use of a cost matrix is suggested for this dataset. It is worse to class patients with heart disease as patients without heart disease (cost = 5), than it is to class patients without heart disease as having heart disease (cost = 1).

### Source

The dataset has been taken from the UCI Repository Of Machine Learning Databases at <http://archive.ics.uci.edu/ml/>.

**Examples**

```
data("StatlogHeart")
summary(StatlogHeart)
shw <- array(1, nrow(StatlogHeart))
shw[StatlogHeart$heart_disease == "presence"] <- 5
suppressWarnings(RNGversion("3.5.0"))
set.seed(1090)
sht <- evtree(heart_disease ~ . , data = StatlogHeart, weights = shw)
sht
table(predict(sht), StatlogHeart$heart_disease)
plot(sht)
```

# Index

## \* datasets

BBBClub, [2](#)  
ContraceptiveChoice, [4](#)  
GermanCredit, [7](#)  
MAGICGammaTelescope, [9](#)  
StatlogHeart, [11](#)

## \* misc

evtree.control, [6](#)

## \* tree

evtree, [5](#)  
.Random.seed, [7](#)

BBBClub, [2](#)

ContraceptiveChoice, [4](#)

evtree, [5](#)  
evtree-package (evtree), [5](#)  
evtree.control, [5](#), [6](#), [6](#)

GermanCredit, [7](#)

MAGICGammaTelescope, [9](#)  
model.frame, [5](#)

party, [6](#)  
plot.constparty, [6](#)  
predict.party, [6](#)  
print.constparty, [6](#)

runif, [7](#)

StatlogHeart, [11](#)