Package 'LogicForest'

August 28, 2026

Title Logic Forest

Version 2.1.5

Depends R (>= 2.10)

Imports LogicReg, methods, survival, utils

Suggests data.table

Description Logic Forest is an ensemble machine learning method that identifies important and interpretable combinations of binary predictors using logic regression trees to model complex relationships with an outcome. Wolf, B.J., Slate, E.H., Hill, E.G. (2010) <doi:10.1093/bioinformatics/btq354>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Bethany Wolf [aut],
Melica Nikahd [ctb, cre],
Andrew Gothard [ctb],
Madison Hyer [ctb]

Maintainer Melica Nikahd <melica.nikahd@osumc.edu>

Repository CRAN

Date/Publication 2026-08-28 20:00:14 UTC

Contents

build.interactions 2
logforest 3
predict.logforest 5
print.LFprediction 6
print.logforest 7

Index **9**

`build.interactions` *Building Interactions*

Description

Builds interactions found from logic forest fit

Usage

```
build.interactions(
  fit,
  test.data,
  n_ints = NULL,
  remove_negated = FALSE,
  req_frequency = NULL
)
```

Arguments

`fit`	Fitted logic regression tree object containing outcome, model type, and logic tree information.
`test.data`	Any dataset that contains the variables to create the interactions
`n_ints`	Max number of interactions to build
`remove_negated`	Whether to build interactions that consist of only negated PIs (True/False)
`req_frequency`	Minimum frequency required to build interaction (0-1)

Details

This function creates the interactions in the data that are found via logic forest.

Value

A dataframe containing the the input dataframe and the interactions built from logic forest.

Author(s)

Andrew Gothard <andrew.gothard@osumc.edu>

References

Wolf BJ, Hill EG, Slate EH. Logic Forest: an ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*. 2010;26(17):2183–2189. doi:10.1093/bioinformatics/btq354

See Also

`logforest`

logforest *Logic Forest & Logic Survival Forest*

Description

Constructs an ensemble of logic regression models using bagging for classification or regression, and identifies important predictors and interactions. Logic Forest (LF) efficiently searches the space of logical combinations of binary variables using simulated annealing. It has been extended to support linear and survival regression.

Usage

```
logforest(
  resp.type,
  resp,
  resp.time = data.frame(X = rep(1, nrow(resp))),
  Xs,
  nBSXVars,
  anneal.params,
  nBS = 100,
  h = 0.5,
  norm = TRUE,
  numout = 5,
  nleaves,
  progressbar = TRUE
)
```

Arguments

resp.type	String indicating regression type: "bin" for classification, "lin" for linear regression, "exp_surv" for exponential time-to-event, and "cph_surv" for Cox proportional hazards.
resp	Numeric vector of response values (binary for classification/survival, continuous for linear regression). For time-to-event, indicates event/censoring status.
resp.time	Numeric vector of event/censoring times (used only for survival models).
Xs	Matrix or data frame of binary predictor variables (0/1 only).
nBSXVars	Integer. Number of predictors sampled for each tree (default is all predictors).
anneal.params	A list of parameters for simulated annealing (see logreg.anneal.control). Defaults: start = 1, end = -2, iter = 50000.
nBS	Number of trees to fit in the logic forest.
h	Numeric. Minimum proportion of trees predicting "1" required to classify an observation as "1" (used for classification).
norm	Logical. If FALSE, importance scores are not normalized.
numout	Integer. Number of predictors and interactions to report.
nleaves	Integer. Maximum number of leaves (end nodes) allowed per tree.
progressbar	Logical. If TRUE (default), displays a progress bar while fitting the forest.

Details

Logic Forest is designed to identify interactions between binary predictors without requiring their pre-specification. Using simulated annealing, it searches the space of all possible logical combinations (e.g., AND, OR, NOT) among predictors. Originally developed for binary outcomes in gene-environment interaction studies, it has since been extended to linear and time-to-event outcomes (Logic Survival Forest).

Value

A `logforest` object containing:

Predictor.frequency Frequency of each predictor across trees.

Predictor.importance Importance of each predictor.

PI.frequency Frequency of each interaction across trees.

PI.importance Importance of each interaction.

Note

Development of Logic Forest was supported by NIH/NCATS UL1RR029882. Logic Survival Forest development was supported by NIH/NIA R01AG082873.

Author(s)

Bethany J. Wolf <wolfb@musc.edu>
J. Madison Hyer <madison.hyer@osumc.edu>

References

Wolf BJ, Hill EG, Slate EH. (2010). Logic Forest: An ensemble classifier for discovering logical combinations of binary markers. *Bioinformatics*, 26(17):2183–2189. doi:10.1093/bioinformatics/btq354
Wolf BJ et al. (2012). LBoost: A boosting algorithm with application for epistasis discovery. *PLoS One*, 7(11):e47281. doi:10.1371/journal.pone.0047281
Hyer JM et al. (2019). Novel Machine Learning Approach to Identify Preoperative Risk Factors Associated With Super-Utilization of Medicare Expenditure Following Surgery. *JAMA Surg*, 154(11):1014–1021. doi:10.1001/jamasurg.2019.2979

See Also

`pimp.import`, `logreg.anneal.control`

Examples

```
## Not run:
set.seed(10051988)
N_c <- 50
N_r <- 200
init <- as.data.frame(matrix(0, nrow = N_r, ncol = N_c))
colnames(init) <- paste0("X", 1:N_c)
```

```
for(n in 1:N_c){
  p <- runif(1, min = 0.2, max = 0.6)
  init[,n] <- rbinom(N_r, 1, p)
}

X3X4int <- as.numeric(init$X3 == init$X4)
X5X6int <- as.numeric(init$X5 == init$X6)
y_p <- -2.5 + init$X1 + init$X2 + 2 * X3X4int + 2 * X5X6int
p <- 1 / (1 + exp(-y_p))
init$Y.bin <- rbinom(N_r, 1, p)

# Classification
LF.fit.bin <- logforest("bin", init$Y.bin, NULL, init[,1:N_c], nBS=10, nleaves=8, numout=10)
print(LF.fit.bin)

# Continuous
init$Y.cont <- rnorm(N_r, mean = 0) + init$X1 + init$X2 + 5 * X3X4int + 5 * X5X6int
LF.fit.lin <- logforest("lin", init$Y.cont, NULL, init[,1:N_c], nBS=10, nleaves=8, numout=10)
print(LF.fit.lin)

# Time-to-event
shape <- 1 - 0.05*init$X1 - 0.05*init$X2 - 0.2*init$X3*init$X4 - 0.2*init$X5*init$X6
scale <- 1.5 - 0.05*init$X1 - 0.05*init$X2 - 0.2*init$X3*init$X4 - 0.2*init$X5*init$X6
init$TIME_Y <- rgamma(N_r, shape = shape, scale = scale)
LF.fit.surv <- logforest("exp_surv", init$Y.bin, init$TIME_Y, init[,1:N_c],
  nBS=10, nleaves=8, numout=10)
print(LF.fit.surv)

## End(Not run)
```

`predict.logforest` *Predict Outcomes Using a Logic Forest Model*

Description

Computes predicted values for new observations or the out-of-bag (OOB) predictions for a logic forest model fitted using `logforest`.

Usage

```
## S3 method for class 'logforest'
predict(object, newdata, cutoff, newcens = NULL, ...)
```

Arguments

`object`	An object of class `"logforest"`.
`newdata`	A matrix or data frame of new predictor values. If omitted, predictions are made for the original data used to fit the model (OOB predictions).

cutoff	A numeric value between 0 and 1 specifying the minimum proportion of trees that must predict a class of 1 for the overall prediction to be 1. Ignored for non-classification models.
newcens	Optional censoring indicator for new observations when predicting from an exponential time-to-event model. If NULL (default), predictions are made without specifying censoring status.
...	Additional arguments (currently ignored).

Details

For classification models, predictions are determined based on the cutoff proportion. For regression or time-to-event models, the function returns predicted values and OOB statistics if newdata is not provided.

Value

An object of class "LFprediction" containing:

- LFprediction: numeric vector of predicted responses.
- proportion_one: numeric vector of the proportion of trees predicting class 1 (classification only).
- AllTrees: matrix or data frame with predicted values from each tree, the proportion of trees predicting 1, and the overall predicted class (classification), or predicted values for regression/time-to-event models.

Author(s)

Bethany Wolf <wolfb@musc.edu>

See Also

logforest

---

print.LFprediction *Print Method for Logic Forest Predictions*

Description

Displays predictions from a logic forest model, including the predicted classes and, for classification models, the proportion of trees predicting a class of one.

Usage

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

Arguments

x	An object of class "LFprediction".
...	Additional arguments (currently ignored).

Details

For classification models, this method prints the predicted classes for each observation and the proportion of trees in the logic forest that predict class 1. For linear regression models, it prints the predicted values and, if available, the out-of-bag mean squared error.

Value

No return value. This function is called for its side effects (printing).

Author(s)

Bethany Wolf <wolfb@musc.edu>

See Also

predict.logforest

---

print.logforest *Print Method for Logic Forest Models*

Description

Prints the most important predictors and interactions from a fitted logic forest model, along with their importance scores and frequency of occurrence.

Usage

```
## S3 method for class 'logforest'
print(x, sortby = "importance", ...)
```

Arguments

x	An object of class "logforest".
sortby	Character string specifying whether to sort the output by "importance" (default) or "frequency".
...	Additional arguments (currently ignored).

Details

This method displays a matrix of the top predictors and interactions from a logic forest model. If x$norm = TRUE, the variable importance scores are normalized such that the largest score is 1 and all other scores are scaled accordingly.

Value

No return value. This function is called for its side effect of printing.

Author(s)

Bethany Wolf <wolfb@musc.edu>

See Also

logforest

Index

build.interactions, 2

logforest, *2*, 3, *6*, *8*
logreg.anneal.control, *3*, *4*

pimp.import, *4*
predict.logforest, 5, *7*
print.LFprediction, 6
print.logforest, 7