

Package ‘LogicForest’

July 21, 2025

Title Logic Forest

Version 2.1.2

Depends R (>= 2.10)

Imports LogicReg, methods, survival

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](https://doi.org/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 2025-07-14 17:10:09 UTC

Contents

logforest	2
predict.logforest	4
print.LFprediction	5
print.logforest	5

Index	6
--------------	----------

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
)
```

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.

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	<i>Predict method for logic forest models</i>
-------------------	---

Description

Predicts outcomes for new observations using a fitted logic forest model.

Usage

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

Arguments

object	An object of class "logforest".
newdata	A data frame containing new observations to predict.
cutoff	A numeric value indicating the proportion of trees that must predict class 1 for an overall prediction of class 1.
...	Additional arguments (currently ignored).

Value

An object of class "LFprediction" containing the predicted outcomes.

print.LFprediction	<i>Print method for logic forest predictions</i>
--------------------	--

Description

Displays the results of a prediction from a logic forest model.

Usage

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

Arguments

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

Value

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

print.logforest	<i>Print method for logic forest models</i>
-----------------	---

Description

Prints the important predictors from a fitted logic forest model.

Usage

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

Arguments

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

Value

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

Index

logforest, [2](#)
logreg.anneal.control, [2](#), [3](#)

pimp.import, [3](#)
predict.logforest, [4](#)
print.LFprediction, [5](#)
print.logforest, [5](#)