Package 'btrm'

July 22, 2025

Type Package

Title Bayesian Treed Regression Model for Personalized Prediction and Precision Diagnostics

Version 0.2.0

Date 2025-05-26

Description Generalization of the Bayesian classification and regression tree (CART) model that partitions subjects into terminal nodes and tailors regression model to each terminal node.

License GPL (>= 2)

Depends R (>= 4.5.0), pROC, arm, stats, graphics, MASS

NeedsCompilation no

Author Yunro Chung [aut, cre] (ORCID: <https://orcid.org/0000-0001-9125-9277>), Yaliang Zhang [aut]

Maintainer Yunro Chung <yunro.chung@asu.edu>

Repository CRAN

Date/Publication 2025-05-26 22:30:02 UTC

Contents

`btrm` *Bayesian Treed Regression Model*

Description

The treed regression model generalizes the Bayesian classification and regression tree (CART) model by partitioning subjects into terminal nodes and tailoring simple regression model to each terminal node.

Usage

```
btrm(y,x,z,ynew,xnew,znew,sparse,nwarm,niter,minsample,base,power)
```

Arguments

y	Response vector. If a factor codied as 0 or 1, classification is assumed. Otherwise, regression is assumed.
x	Data.frame or matrix of predictors that is used to estimate a tree structure.
z	Data.frame or matrix of predictors that is used in terminal node specific ML models. See the description below about the difference between x and z.
ynew	Response vector for the test set corresponding to y (default ynew=NULL).
xnew	Data.frame or matrix for the test set corresponding to x (default xnew=NULL).
znew	Data.frame or matrix for the test set corresponding to z (default znew=NULL).
sparse	Whether to perform variable and machine learning model selections based on a sparse Dirichlet prior rather than simply uniform (default sparse=TRUE).
nwarm	Number of warm-up (default nwarm=1000).
niter	Number of iteration (defaut niter=1000).
minsample	The number of minimum sample size per each node, i.e., length(y)>min_sample if y is continuous and min(length(y==1),length(y==0))>min_sample (default min_sample=20).
base	Base parameter for tree prior (default base=0.95).
power	Power parameter for tree prior (default power=0.8).

Details

Ideally, there are two sets of predictors, x and z, e.g., demographic variables and biomarkers, where x is used to split trees, and z is assigned to each terminal node. However, if this is not possible, it allows us to use the same x and z in the btml function, e.g., btml(y=y, x=x, z=x, ...). For high-dimensional variables, increase nwarm=10000 and niter=10000, or more; and increase minsample.

Ideally, there are two sets of predictors, x and z, e.g., demographic variables and biomarkers, where x is used to split trees, and z is assigned to each terminal node. However, if this is not possible, it allows to use the same x and z in the btrm function, e.g., btrm(y=y, x=x, z=x, ...).

Regarding the node numbers, an internal node s has left and right child nodes 2*s and 2*s+1, respectively, where node 1 is a root node; nodes 2 and 3 are left and right child nodes of node 1; nodes 4 and 5 are left and right nodes of node 2; and so on.

Value

An object of class btrm, which is a list with the following components:

terminal	Node numbers in terminal nodes.
internal	Node numbers in internal nodes.
splitVariable	Variable (i.e., x[,u] if splitVariable[k]=u) used to split the internal node k.
cutoff	cutoff[k] is the cutoff value to split the internal node k.

marker	Marker (i.e., z[,v] if marker[t]=v) assigned to the terminal node t.
node.hat	Estimated node on the training set.
marker.hat	Estimated marker on the training set.
beta.hat	beta.hat[[t]] is estimated regression coefficients from the linear (or logistic) regression model at the terminal node t $\in$ terminal.
y.hat	Estimated y (or probability) on the training set if y is continuous (or binary).
mse	Training MSE.
bs	Training Brier Score.
roc	Training ROC curve.
auc	Training AUC.
y.hat.new	Estimated y (or probability) on the test set if y is continuous (or binary).
node.hat.new	Estimated node on the test set.
marker.hat.new	Estimated marker on the test set.
mse.new	Test MSE.
bs.new	Test Brier Score.
roc.new	Test ROC curve.
auc.new	Test AUC.

Author(s)

Yunro Chung [aut, cre], Yaliang Zhang [aut]

References

Yaliang Zhang and Yunro Chung, Bayesian treed model (in preperation)

Examples

```
set.seed(10)
###
#1. continuous y
###
n=200*2 #n=200 & 200 for training & test sets

x=matrix(rnorm(n*10),n,10) #10 predictors
z=matrix(rnorm(n*10),n,10) #10 biomarkers

xcut=median(x[,1])
subgr=1*(x[,1]<xcut)+2*(x[,1]>=xcut) #2 subgroups

lp=rep(NA,n)
for(i in 1:n)
  lp[i]=1+3*z[i,subgr[i]]
y=lp+rnorm(n,0,1)

idx.nex=sample(1:n,n*1/2,replace=FALSE)
```

```
ynew=y[idx.nex]
xnew=x[idx.nex,]
znew=z[idx.nex,]

y=y[-idx.nex]
x=x[-idx.nex,]
z=z[-idx.nex,]

fit1=btrm(y,x,z,ynew=ynew,xnew=xnew,znew=znew)
print(fit1$mse.new)
plot(fit1$y.hat.new~ynew,ylab="Predicted y",xlab="ynew")

###
#2. binary y
###
x=matrix(rnorm(n*10),n,10) #10 predictors
z=matrix(rnorm(n*10),n,10) #10 biomarkers

xcut=median(x[,1])
subgr=1*(x[,1]<xcut)+2*(x[,1]>=xcut) #2 subgroups

lp=rep(NA,n)
for(i in 1:n)
  lp[i]=1+3*z[i,subgr[i]]
prob=1/(1+exp(-lp))
y=rbinom(n,1,prob)
y=as.factor(y)

idx.nex=sample(1:n,n*1/2,replace=FALSE)
ynew=y[idx.nex]
xnew=x[idx.nex,]
znew=z[idx.nex,]

y=y[-idx.nex]
x=x[-idx.nex,]
z=z[-idx.nex,]

fit2=btrm(y,x,z,ynew=ynew,xnew=xnew,znew=znew)
print(fit2$auc.new)
plot(fit2$roc.new)
```

Index