Package ‘bmabart’

April 29, 2025

Type Package

Title Bayesian Mediation Analysis Using BART

Version 1.0

Date 2025-04-28

Depends R (>= 2.14.1), BART, survival, gplots

Imports lattice, methods

Description
Used for Bayesian mediation analysis based on Bayesian additive Regression Trees (BART). The analysis method is described in Yu and Li (2025) ``Mediation Analysis with Bayesian Additive Regression Trees'', submitted for publication.

License GPL (>= 2)

URL `https://www.r-project.org`,
`https://publichealth.lsuhsc.edu/Faculty_pages/qyu/index.html`

RoxygenNote 7.3.2

NeedsCompilation no

Author Qingzhao Yu [aut, cre],
Bin Li [aut]

Maintainer Qingzhao Yu <`qyu@lsuhsc.edu`>

Repository CRAN

Date/Publication 2025-04-29 08:50:11 UTC

Contents

`bmabart-package` *Bayesian Mediation Analysis Using Bayesian Additive Regression Trees*

Description

Used for Bayesian mediation analysis based on Bayesian additive Regression Trees (BART). The analysis method is described in Yu and Li (2025) "Mediation Analysis with Bayesian Additive Regression Trees", submitted for publication.

Details

Build BARTs using the R package and perform the Bayesian Mediation Analysis.

Author(s)

Qingzhao Yu and Bin Li

Maintainer: Qingzhao Yu <qyu@lsuhsc.edu>

References

Yu, Q., and Li, B. (2025) <doi:>. "Mediation Analysis with Bayesian Additive Regression Trees," submitted.

`bma.bart` *Bayesian Mediation Analysis Using Bayesian Additive Regression Trees*

Description

Build BARTs using the R package and perform the Bayesian Mediation Analysis.

Usage

```
bma.bart(pred, m, y, refy = rep(NA, ncol(data.frame(y))),
         predref = rep(NA, ncol(data.frame(pred))), deltap = NA,
         deltam = NA, mref = rep(NA, ncol(data.frame(m))), cova = NULL,
         cova.ref = list(), mcov = NULL, mcov.ref = list(), mclist = NULL,
         complete = FALSE, ntree = 200L, numcut = 100L, ndpost = 1000L,
         nskip = 100L, keepevery = 1L, nkeeptrain = ndpost, nkeeptest = ndpost,
         nkeeptestmean = ndpost, nkeeptreedraws = ndpost, printevery = 100L,
         seed = sample(1:1e+06, 1))
```

Arguments

`pred`	The vector/matrix of the exposure/predict variable(s).
`m`	The dataframe of all potential mediators
`y`	The vector/matrix of the outcome(s).
`refy`	The reference groups of y when the corresponding outcome is binary or categorical.
`predref`	The reference groups of pred when the corresponding outcome is binary or categorical.
`deltap`	A vector of the length of the number of exposures. The difference in pred when calculate the changing rate by pred. If not set, the difference is 1 for categorical predictor and one tenth of the standard deviaiton of the predictor if continuous.
`deltam`	A vector of the length of the number of mediators. The ith item is the difference in the ith mediator when calculate the changing rate by each mediator. If not set, the difference is 1 for categorical mediators and one tenth of the standard deviaiton of the mediator if continuous.
`mref`	The reference groups of mediators when the corresponding mediator is binary or categorical.
`cova`	The covariance data for y.
`cova.ref`	The reference group for the binary or categorical covariates in cova.
`mcov`	The covariance data for mediators.
`mcov.ref`	The reference group if the mcovs are categorical or binary.
`mclist`	If mclist is null but not mcov, mcov is applied to all mediators. If both mcov and mclist are not NULL, the first item of mclist lists all mediators that are using different mcov, the following items gives the mcov for the mediators in order, NA if no mcov to be used. e.g. mclist=list(c(1,2,4),l1=1,l2=NA,l4=c(1,3)), mediator 1, m[,1], use mcov[,1], 2 uses no covariates, 4 uses mcov[,c(1,3)], all other mediators use all. Can also replace variable names with column numbers in the mclist.
`complete`	complete=TRUE if only completed cases are used in analysis.
`ntree`	As in the BART package, the number of trees in the sum.
`numcut`	See the BART package. The number of possible values of c (see usequants). If a single number if given, this is used for all variables. Otherwise a vector with length equal to ncol(x.train) is required, where the ith element gives the number of c used for the ith variable in x.train. If usequants is false, numcut equally spaced cutoffs are used covering the range of values in the corresponding column of x.train. If usequants is true, then min(numcut, the number of unique values in the corresponding columns of x.train - 1) c values are used.
`ndpost`	As in the BART package, the number of posterior draws returned.
`nskip`	As in the BART package, number of MCMC iterations to be treated as burn in.
`keepevery`	As in the BART package, every keepevery draw is kept to be returned to the user.
`nkeeptrain`	As in the BART package, number of MCMC iterations to be returned for train data.

`nkeeptest`	As in the BART package, number of MCMC iterations to be returned for test data.
`nkeeptestmean`	As in the BART package, number of MCMC iterations to be returned for test mean.
`nkeeptreedraws`	As in the BART package, number of MCMC iterations to be returned for tree draws.
`printevery`	As in the BART package, as the MCMC runs, a message is printed every printevery draws.
`seed`	A seed number to keep the results repeatable.

Details

Please refer to the reference for the details of model fitting and inferences of mediation effects.

Value

`aieX`	posterior samples of average indirect effects using method X. method 2 to show the results from the partial differences, method 3 to show the results from the G-computation, and method 4 for G-computation with non-parametric method (binary exposures only).
`adeX`	posterior samples of average direct effects using method X.
`ateX`	posterior samples of average total effects using method X.
`ieX`, `deX`, `teX`	posterior samples of indirect effects, direct effects and total effects using method X.
`apart.ie`	posterior samples of the a-part:changing rate of mediators with pred, using method 2.
`bpart.ie`	posterior samples of the b-part:changing rate of outcomes with mediators, using method 2.
`data0`	the output from data_org.
`y.type`	the type of outcomes.
`y.model`	the BART model of outcomes.
`m.models`	the BART model of each mediator.
`DIC`	the estimated DIC,deviances, D_bar, Var_D, and p_D.

Note

data_org is run automatically in bma.bart. No need to run it separately.

Author(s)

Qingzhao Yu and Bin Li

References

Yu, Q., and Li, B. (2025) <doi:>. "Mediation Analysis with Bayesian Additive Regression Trees," submitted.

Examples

```
data(weight_behavior)
#nubmer of mcmc iterations are set to 3 to reduce time. Need to bring it up to reasonable times.
#binary predictor
try0= bma.bart(pred=weight_behavior[,3], m=weight_behavior[,c(2,4:14)],
               y=weight_behavior[,15], refy = 0, predref = "F",nskip=0,ndpost=2)
summary(try0)

#add covariate for mediators
try1= bma.bart(pred=weight_behavior[,3], m=weight_behavior[,c(2,4:13)],
               mcov=weight_behavior[,14], mclist=append(list(var=1:10),rep(NA,10)),
               #"sweater" is used as a cov for "excercises" only
               y=weight_behavior[,15], refy = 0, predref = "F",nskip=0,ndpost=2)
summary(try1)
summary(try1,trim=0)
#multiple prdictor
try2= bma.bart(pred=weight_behavior[,4], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,15], refy = 0, predref = "OTHER",nskip=0,ndpost=2)
summary(try2)
try3= bma.bart(pred=weight_behavior[,c(1,4)], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,15], refy = 0, predref = "OTHER",nskip=0,ndpost=2)
summary(try3)
#continuous y
try4= bma.bart(pred=weight_behavior[,4], m=weight_behavior[,c(2:3,5)],
               y=weight_behavior[,1], refy = 0, predref = "OTHER",nskip=0,ndpost=2)
summary(try4)
#categorical y
try5= bma.bart(pred=weight_behavior[,1], m=weight_behavior[,c(2:3,5)],
               y=weight_behavior[,4], refy = "",nskip=0,ndpost=2)
summary(try5)
#add covariates for y and for mediators
try6= bma.bart(pred=weight_behavior[,4], m=weight_behavior[,c(5:12)],
               cova=weight_behavior[,2:3], mcov=weight_behavior[,13:14],
               mclist=c(list(var=1:7),rep(NA,6),list(1)),
               y=weight_behavior[,1], refy = 0, predref = "OTHER",nskip=0,ndpost=2)
#cova and mcov needs to be binarized and numerized
summary(try6)

##Surv class outcome (survival analysis)
data(cgd1)       #a dataset in the survival package
x=cgd1[,c(4:5,7:12)]
pred=cgd1[,6]
status<-ifelse(is.na(cgd1$etime1),0,1)
y=Surv(cgd1$futime,status)
#for continuous predictor
try7= bma.bart(pred=pred,m=x,y=y,nskip=0,ndpost=3)
#summary(try7)
```

---

cgd1 *cgd1 Data Set*

Description

This database was obtained from the survival package containing a time-to-event data.

Usage

```
data(weight_behavior)
```

Format

The data set contains many variables.

Examples

```
data(cgd1)
names(cgd1)
```

`data_org` *Prepare Variables for Bayesian Mediation Analysis with BART*

Description

Read in exposure, mediators, outcome, and covariates, and transform them into formats fit for BART fitting.

Usage

```
data_org(pred, m, y, refy = rep(NA, ncol(data.frame(y))),
         predref = rep(NA, ncol(data.frame(pred))), deltap = NA,
         deltam = NA, mref = rep(NA, ncol(data.frame(m))), cova = NULL,
         cova.ref = list(), mcov = NULL, mcov.ref = list(), mclist = NULL,
         complete = FALSE)
```

Arguments

`pred`	The vector/matrix of the exposure/predict variable(s).
`m`	The dataframe of all potential mediators
`y`	The vector/matrix of the outcome(s).
`refy`	The reference groups of y when the corresponding outcome is binary or categorical.
`predref`	The reference groups of pred when the corresponding outcome is binary or categorical.
`deltap`	A vector of the length of the number of exposures. The difference in pred when calculate the changing rate by pred. If not set, the difference is 1 for categorical predictor and one tenth of the standard deviaiton of the predictor if continuous.

`deltam`	A vector of the length of the number of mediators. The ith item is the difference in the ith mediator when calculate the changing rate by each mediator. If not set, the difference is 1 for categorical mediators and one tenth of the standard deviaiton of the mediator if continuous.
`mref`	The reference groups of mediators when the corresponding mediator is binary or categorical.
`cova`	The covariance data for y.
`cova.ref`	The reference group for the binary or categorical covariates in cova.
`mcov`	The covariance data for mediators.
`mcov.ref`	The reference group if the mcovs are categorical or binary.
`mclist`	If mclist is null but not mcov, mcov is applied to all mediators. If both mcov and mclist are not NULL, the first item of mclist lists all mediators that are using different mcov, the following items gives the mcov for the mediators in order, NA if no mcov to be used. e.g. mclist=list(c(1,2,4),l1=1,l2=NA,l4=c(1,3)), mediator 1, m[,1], use mcov[,1], 2 uses no covariates, 4 uses mcov[,c(1,3)], all other mediators use all. Can also replace variable names with column numbers in the mclist.
`complete`	complete=TRUE if only completed cases are used in analysis.

Details

The function helps organize input data into formats readible to the BART package for building BART. It also recoganize the type of the response variable(s), so that different functions and methods will be used for the mediation effect inferences.

Value

Return the cleaned up dataset and organized by types, which is ready for the Bayesian Mediation Analysis.

`N`	The total number of observations.
`y_type`	The format of the response variable(s): 1 for continuous, 2 binary, 3 categorical, and 4 time-to-event. It is the same length as the number of outcomes.
`y`	The original y with observations of missing data removed, if complete=T.
`y1`	The outcome variables where binary or categorical variables are replaced with dummy design matrix.
`cova`	The covariates for y, where binary or categorical variables are replaced with dummy design matrix.
`npred`	The number of predictors/exposures, where a categorical exposure of k levels has k-1 dummy predictors.
`nm`	The number of original mediators, ncol(m).
`mcov`	Reformated mcov.
`mind`	If mcov is not NULL, mind is a matrix of (# of mediator)*ncol(mcov), cell (i,j) is the indicator of whether the jth column of mcov should be used for mediator i in m1.

`pred1`	The original pred with observations of missing data removed, if complete=T.
`pred2`	The pred1 with all categorical or binary variables are turned into dummis.
`binpred1`	The column numbers of binary predictors in pred1.
`binpred2`	The column numbers of binary predictors in pred2.
`catpred1`	The column numbers of categorical predictors in pred1.
`catpred2`	The column numbers of categorical predictors in pred2.
`contpred1`	The column numbers of continuous predictors in pred1.
`contpred2`	The column numbers of continuous predictors in pred2.
`m1`	The original m with observations of missing data removed, if complete=T.
`m2`	The m1 with all categorical or binary variables are turned into dummis.
`m3.1`	The m2 with all continuous variables minus a deltam[i]/2, where i is the ith mediator.
`m3.2`	The m2 with all continuous variables add a deltam[i]/2, where i is the ith mediator.
`p1`	The number of continuous mediators.
`p2`	The number of binary mediators.
`p3`	The number of categorical mediators.
`binm1`	The column number of binary mediators in m1.
`binm2`	The column number of binary mediators in m2.
`catm1`	The column number of categorical mediators in m1.
`catm2`	A matrix with the number of rows the number of categorical meidators by the order of catm1. Each row has the start (first column) and end (second column) column numbers of the categorical variable's design matrix in m2.
`contm1`	The column number of continuous mediators in m1.
`contm2`	The column number of continuous mediators in m2.
`deltap`	A vector of the length of the number of exposures. The difference in pred when calculate the changing rate by pred. If not input, the difference is 1 for categorical predictor and one tenth of the standard deviaiton of the predictor if continuous.
`deltam`	A vector of the length of the number of mediators. The ith item is the difference in the ith mediator when calculate the changing rate by each mediator. If not set, the difference is 1 for categorical mediators and one tenth of the standard deviaiton of the mediator if continuous.

Note

data_org is run within bma.bart function. Users do not have to run data_org separately.

Author(s)

Qingzhao Yu and Bin Li

References

Yu, Q., and Li, B. (2025) <doi:>. "Mediation Analysis with Bayesian Additive Regression Trees," submitted.

Examples

```
data("weight_behavior")
#binary predictor
try0= data_org(pred=weight_behavior[,3], m=weight_behavior[,c(2,4:14)],
               y=weight_behavior[,15], refy = 0, predref = "F")
#add covariate for mediators
try1= data_org(pred=weight_behavior[,3], m=weight_behavior[,c(2,4:13)],
               mcov=weight_behavior[,14], mclist=append(list(var=1:10),rep(NA,10)),
               #"sweater" is used as a cov for "excercises" only
               y=weight_behavior[,15], refy = 0, predref = "F")  #,complete=T
#multiple prdictor
try2= data_org(pred=weight_behavior[,4], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,15], refy = 0, predref = "OTHER")
try3= data_org(pred=weight_behavior[,c(1,4)], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,15], refy = 0, predref = "OTHER")
#continuous y
try4= data_org(pred=weight_behavior[,4], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,1], refy = 0, predref = "OTHER")
#categorical y
try5= data_org(pred=weight_behavior[,1], m=weight_behavior[,c(2:3,5:14)],
               y=weight_behavior[,4], refy = "", predref = "OTHER")
#add covariates for y and for mediators
try6= data_org(pred=weight_behavior[,4], m=weight_behavior[,c(5:12)],
               cova=weight_behavior[,2:3],mcov=weight_behavior[,13:14],
               mclist=c(list(var=1:7),rep(NA,6),list(1)),
               y=weight_behavior[,1], refy = 0, predref = "OTHER")
#time-to-event outcome
data(cgd1)       #a dataset in the survival package
x=cgd1[,c(4:5,7:12)]
pred=cgd1[,6]
status<-ifelse(is.na(cgd1$etime1),0,1)
y=Surv(cgd1$futime,status)
#for continuous predictor
try7<-data_org(pred=pred,m=x,y=y)
```

`print.summary.bma.bart`

Print the summary results for bma.bart object.

Description

Print and plot the inference results.

Usage

```
## S3 method for class 'summary.bma.bart'
print(x, ..., digit = x$digit, method = x$method, RE = x$RE)
```

Arguments

`x`	the summary.bma.bart object from the summary function.
`...`	other arguments passed to the print function.
`digit`	the number of decimal digits to keep.
`method`	method=2 to show the results from the partial differences, method=3 to show the results from the G-computation, and method=4 for G-computation with non-parametric method (binary exposures only).
`RE`	If ture, print the relative effects.

Value

No return value, called for side effects.

Author(s)

Qingzhao Yu and Bin Li

References

Yu, Q., and Li, B. (2025) <doi:>. "Mediation Analysis with Bayesian Additive Regression Trees," submitted.

See Also

"`bma.bart`" for examples.

`summary.bma.bart` *Summary of a bma.bart object*

Description

The bma.bart object is from the bma.bart function. The summary function is to calculate the estimates, standard deviation and credible sets of the mediation effects and relative effects.

Usage

```
## S3 method for class 'bma.bart'
summary(object, ..., plot = TRUE, RE = TRUE,
                 quant = c(0.025, 0.25, 0.5, 0.75, 0.975),
                 digit = 4, method = 3, trim = 0.05)
```

Arguments

`object`	a bma.bart object created by bma.bart.
`...`	other arguments passed to the print function.
`plot`	default is TRUE, if ture, draw a barplot of the mediation effects with credible sets.
`RE`	default is FALSE, if ture, show the inferences on relative mediation effects.
`quant`	show the quantiles defined by quant of the posterior distributions of mediation effects.
`digit`	the number of decimal digits to keep.
`method`	method=2 to show the results from the partial differences, method=3 to show the results from the G-computation, and method=4 for G-computation with non-parametric method (binary exposures only).
`trim`	the percentage of trims to calcuate the trimed average mediation effects. By default, trim=0.5.

Details

Show the posterior distribution of the estimated mediation effects.

Value

`resultX`	the mediation effect estimates using method X.
`resultX.re`	the relative effect estimates using method X.

Author(s)

Bin Li and Qingzhao Yu

References

Yu, Q., and Li, B. (2025) <doi:>. "Mediation Analysis with Bayesian Additive Regression Trees," submitted.

See Also

"`bma.bart`" for examples.

`weight_behavior` *Weight_Behavior Data Set*

Description

This database was obtained from the Louisiana State University Health Sciences Center, New Orleans, by Dr. Richard Scribner. He explored the relationship between BMI and kids behavior through a survey at children, teachers and parents in Grenada in 2014. This data set includes 691 observations and 15 variables.

Usage

```
data(weight_behavior)
```

Format

The data set contains the following variables:

bmi - body mass index, calculated by weight(kg)/height(cm)^2, numeric

age - children's age in years at the time of survey, numeric

sex - sex of the children, factor

race - race of the children, factor

numpeople - number of people in family, numeric

car - the number of cars in family, numeric

gotosch - the method used to go to school, factor

snack - eat snack or not in a day, binary

tvhours - number of hours watching TV per week, numeric

cmpthours - number of hours using computer per week, numeric

cellhours - number of hours playing with cell phones per week, numeric

sports - join in a sport team or not, 1: yes; and 2: no

exercises - number of hours of exercises per week, numeric

sweat - number of hours of sweating activities per week, numeric

overweigh - the child is overweighed or not, binary

Examples

```
data(weight_behavior)
names(weight_behavior)
```

Index