Package 'mpbart'

February 7, 2016

Title Multinomial Probit Bayesian Additive Regression Trees

Version 0.2

Description Fits Multinomial Probit Bayesian Additive Regression Trees.

Depends R (>= 3.2.2), mlbench, bayesm, cvTools, mlogit

License GPL (>= 2)

LazyData true

RoxygenNote 5.0.1

NeedsCompilation yes

Author Bereket Kindo [aut, cre]

Maintainer Bereket Kindo <bpkindo@gmail.com>

Repository CRAN

Date/Publication 2016-02-07 12:39:20

R topics documented:

`mpbart` *Multinomial Probit Bayesian Additive Regression Trees*

Description

Multinomial probit modeling using Bayesian Additive Regression Trees,

Usage

```
mpbart(formula, train.data, test.data = NULL, base = NULL, varying = NULL,
  sep = ".", Prior = NULL, Mcmc = NULL, seedvalue = NULL)
```

Arguments

formula	response ~ choice speccific covariates | demographic covariates. If there are no, demographic variables use response ~ choice specific covariates| ~ 1. If there are no choice specific covariates, use response ~ 1 | demographic covariates
train.data	Training Data in wide format (for details on wide format, see documentation in R package **mlogit**),
test.data	Test Data in wide format, typically without the response,
base	base choice. Default is the highest class/choice,
varying	The indeces of the variables that are alternative specific,
sep	The seperator of the variable name and the alternative name in the choice specific covariates. For example a covariate name variabl1.choice1 indicates a separator of dot (.).
Prior	List of Priors for MPBART: e.g., Prior = list(nu=p+2, V= diag(p - 1), ntrees=200, kfac=2.0, pbd=1.0, pb=0.5 , beta = 2.0, alpha = 0.95, nc = 100, priorindep = FALSE, minobsnode = 10). The comonents of Prior are • nu
Mcmc	List of MCMC starting values, burn-in ...: e.g., list(sigma0 = diag(p - 1), keep = 1, burn = 100, ndraws = 1000, keep_sigma_draws=FALSE)
seedvalue	random seed value, default of 99 will be used if null,

Value

class_prob_train training data choice/class probabilities,

predicted_class_train training data predicted choices/classes,

class_prob_test test data choice/class probabilities,

predicted_class_test test data predicted choices/classes,

sigmasample posterior samples of the latent variable covariance matrix.

Examples

```
## Not run: library(mpbart)
set.seed(9)
data(Fishing)

table(Fishing$mode)
folds = cvFolds(n = nrow(Fishing), K = 10, R = 1,
                type = "random");
Fishing$fold = sample(folds$which)
Fishing$logincome = log(Fishing$income)

FishingTrain <- Fishing[Fishing$fold != 1,]
FishingTest <- Fishing[Fishing$fold == 1,]

burn <- 100
ndraws <- 200 # a higher number such as 1500 is better
```

```
p = 4
#'four choices
sigma0 <- diag(p-1)


Mcmc1 <- list(sigma0=sigma0, burn = burn, ndraws = ndraws)
Prior1 <- list( nu=p-1,
                V = .5*diag(p-1),
                ntrees = 5, # ntrees >= 50 is probably more appropriate
                kfac = 3.0,
                pbd = 1.0,
                pb = 0.5,
                alpha = 0.95,
                beta =  3.0,
                nc = 100,
                priorindep = FALSE,
                minobsnode = 10)



out <- mpbart(as.factor(mode) ~ price + catch | logincome,
    train.data =  FishingTrain,
    test.data =  FishingTest,
    base = 'boat',
    varying = 2:9,
    sep = ".",
    Prior = Prior1,
    Mcmc = Mcmc1,
    seedvalue = 99)

table(as.character(FishingTrain$mode), as.character(out$predicted_class_train))

table(as.character(FishingTest$mode), as.character(out$predicted_class_test))

test_err <- sum(as.character(FishingTest$mode) !=
 as.character(out$predicted_class_test))/length(FishingTest$mode)
cat("test error :", test_err )

# ############## Waveform recognition classification example
# set.seed(64)
# library(mpbart)
# p=3
# train_wave = mlbench.waveform(300)
# test_wave = mlbench.waveform(500)
# traindata = data.frame(train_wave$x, y = train_wave$classes)
# testdata = data.frame(test_wave$x, y = test_wave$classes)
#
#
# sigma0 = diag(p-1)
# burn = 100
# ndraws <- 200 # a higher number such as 1500 is better#'#
# Mcmc1=list(sigma0=sigma0, burn = burn, ndraws = ndraws)
# Prior1 = list(nu=p+2,
```

```
#               V=(p+2)*diag(p-1),
#               ntrees = 100,
#               kfac = 2.0,
#               pbd = 1.0,
#               pb = 0.5,
#               alpha = 0.99,
#               beta =  2.0,
#               nc = 200,
#               priorindep = FALSE)
#
#
#
# out <- mpbart(as.factor(y) ~ 1 | .,
#               train.data =  traindata,
#               test.data =  testdata,
#               base = NULL,
#               varying = NULL,
#               sep = NULL,
#               Prior = Prior1,
#               Mcmc = Mcmc1,
#               seedvalue = 99)
#
# # #The above output can alternatively be obtained via:
# # out <- mpbart(as.factor(y) ~ 1 | X1 + X2 + X3 + X4 + X5 + X6 +
# #                 X7 + X8 + X9 + X11 + X12 + X13 +
# #                 X14 + X15 + X16 + X17 + X18 + X19 +
# #                 X20 + X21,
# #               train.data =  traindata,
# #               test.data =  testdata,
# #               base = NULL,
# #               varying = NULL,
# #               sep = NULL,
# #               Prior = Prior1,
# #               Mcmc = Mcmc1,
# #               seedvalue = 99)
# #
# #
# # confusion matrix train
# table(traindata$y, out$predicted_class_train)
#table(traindata$y==out$predicted_class_train)/
#sum(table(traindata$y==out$predicted_class_train))
#
#
# #confusion matrix test
# table(testdata$y, out$predicted_class_test)
#
# test_err <- sum(testdata$y != out$predicted_class_test)/
#   sum(table(testdata$y == out$predicted_class_test))
#
# cat("test error :", test_err )
## Not run: END
```

rmpbart *Multinomial Probit Bayesian Additive Regression Trees*

Description

A function to implement multinomial probit regression via Bayesian Addition Regression Trees using partial marginal data augmentation.

Usage

```
rmpbart(x.train, y.train, x.test = NULL, Prior = NULL, Mcmc = NULL,
  seedvalue = NULL)
```

Arguments

x.train	Training data predictors.
y.train	Training data observed classes.
x.test	Test data predictors.
Prior	List of Priors for MPBART: e.g., Prior = list(nu=p+2, V= diag(p - 1), ntrees=200, kfac=2.0, pbd=1.0, pb=0.5 , beta = 2.0, alpha = 0.95, nc = 100, priorindep = 0, minobsnode = 10)
Mcmc	List of MCMC starting values, burn-in ...: e.g., list(sigma0 = diag(p - 1), keep = 1, burn = 100, ndraws = 1000, keep_sigma_draws=FALSE)
seedvalue	random seed value: e.g., seedvalue = 99

Examples

```
set.seed(64)
library(mpbart)
p=3
train_wave = mlbench.waveform(50)
test_wave = mlbench.waveform(100)
traindata = data.frame(train_wave$x, y = train_wave$classes)
testdata = data.frame(test_wave$x, y = test_wave$classes)

x.train = data.frame(train_wave$x)
x.test = data.frame(test_wave$x)

y.train = train_wave$classes

sigma0 = diag(p-1)
burn = 100
ndraws = 200 # a higher number >=1000 is more appropriate.

Mcmc1=list(sigma0=sigma0, burn = burn, ndraws = ndraws)
Prior1 = list(nu=p+2,
```

```
              V=(p+2)*diag(p-1),
              ntrees = 5, #typically 200 trees is good
              kfac = 2.0,
              pbd = 1.0,
              pb = 0.5,
              alpha = 0.99,
              beta =  2.0,
              nc = 200,
              priorindep = FALSE)



out = rmpbart(x.train = x.train, y.train = y.train, x.test = x.test,
            Prior = Prior1, Mcmc=Mcmc1, seedvalue = 99)

#confusion matrix train
table(y.train, out$predicted_class_train)
table(y.train==out$predicted_class_train)/sum(table(y.train==out$predicted_class_train))


#confusion matrix test
table(test_wave$classes, out$predicted_class_test)

test_err <- sum(test_wave$classes != out$predicted_class_test)/
    sum(table(test_wave$classes == out$predicted_class_test))

cat("test error :", test_err )
```

Index