BART

Fuyu Guo

2023-04-29

Load libraries and data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1      v purrr   1.0.1
## v tibble  3.2.1      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(parallel)
library(BART)
```

```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## Loading required package: nnet
## Loading required package: survival
##
## Attaching package: 'survival'
```

```
##
## The following object is masked from 'package:caret':
##
##     cluster
```

```
load("df_train_test.RData")
```

Split data into 10 CV-folds

```
set.seed(123)
folds <- createFolds(1:352, k = 10, list = TRUE, returnTrain = FALSE)
```

block for ntree = 10

```
ntree <- 10
true_type <- numeric(0)
prediction_type <- numeric(0)
for (i in 1:10) {
  X_train <- df_train[-folds[[i]],]
  Y_train <- population_train[-folds[[i]]]
  X_val <- df_train[folds[[i]],]
  Y_val <- population_train[folds[[i]]]

  Y_train_CEU <- ifelse(Y_train == "CEU", 1, 0)
  Y_train_FIN <- ifelse(Y_train == "FIN", 1, 0)
  Y_train_GBR <- ifelse(Y_train == "GBR", 1, 0)
  Y_train_IBS <- ifelse(Y_train == "IBS", 1, 0)
  Y_train_TSI <- ifelse(Y_train == "TSI", 1, 0)


  fit_CEU <- mc.gbart(X_train, Y_train_CEU, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_FIN <- mc.gbart(X_train, Y_train_FIN, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_GBR <- mc.gbart(X_train, Y_train_GBR, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_IBS <- mc.gbart(X_train, Y_train_IBS, x.test = X_val,
```

```
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_TSI <- mc.gbart(X_train, Y_train_TSI, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)
  prob_matrix <- matrix(NA, nrow = length(Y_val), ncol = 5)
  prob_matrix[,1] <- fit_CEU$prob.test.mean
  prob_matrix[,2] <- fit_FIN$prob.test.mean
  prob_matrix[,3] <- fit_GBR$prob.test.mean
  prob_matrix[,4] <- fit_IBS$prob.test.mean
  prob_matrix[,5] <- fit_TSI$prob.test.mean

  # select the column with the highest probability
  prob_vector <- apply(prob_matrix, 1, which.max)
  prediction <- case_when(prob_vector == 1 ~ "CEU",
                          prob_vector == 2 ~ "FIN",
                          prob_vector == 3 ~ "GBR",
                          prob_vector == 4 ~ "IBS",
                          prob_vector == 5 ~ "TSI")

  true_type <- c(true_type, Y_val)
  prediction_type <- c(prediction_type, prediction)
}


table(true_type, prediction_type)
```

```
##          prediction_type
## true_type CEU FIN GBR IBS TSI
##       CEU  26   2  26   6  12
##       FIN   0  63   0   0   0
##       GBR  19   1  37   2   4
##       IBS   7   0   5  53  14
##       TSI  10   0   6  13  46
```

```
mean(true_type == prediction_type)
```

```
## [1] 0.6392045
```

block for ntree = 50

```
ntree <- 50
true_type <- numeric(0)
prediction_type <- numeric(0)
for (i in 1:10) {
```

```
X_train <- df_train[-folds[[i]],]
Y_train <- population_train[-folds[[i]]]
X_val <- df_train[folds[[i]],]
Y_val <- population_train[folds[[i]]]

Y_train_CEU <- ifelse(Y_train == "CEU", 1, 0)
Y_train_FIN <- ifelse(Y_train == "FIN", 1, 0)
Y_train_GBR <- ifelse(Y_train == "GBR", 1, 0)
Y_train_IBS <- ifelse(Y_train == "IBS", 1, 0)
Y_train_TSI <- ifelse(Y_train == "TSI", 1, 0)


fit_CEU <- mc.gbart(X_train, Y_train_CEU, x.test = X_val,
                type = 'lbart', mc.cores = 4,
               ntree = ntree,
                nskip = 2000,
                ndpost = 1000)

fit_FIN <- mc.gbart(X_train, Y_train_FIN, x.test = X_val,
                type = 'lbart', mc.cores = 4,
               ntree = ntree,
                nskip = 2000,
                ndpost = 1000)

fit_GBR <- mc.gbart(X_train, Y_train_GBR, x.test = X_val,
                type = 'lbart', mc.cores = 4,
               ntree = ntree,
                nskip = 2000,
                ndpost = 1000)

fit_IBS <- mc.gbart(X_train, Y_train_IBS, x.test = X_val,
                type = 'lbart', mc.cores = 4,
               ntree = ntree,
                nskip = 2000,
                ndpost = 1000)

fit_TSI <- mc.gbart(X_train, Y_train_TSI, x.test = X_val,
                type = 'lbart', mc.cores = 4,
               ntree = ntree,
                nskip = 2000,
                ndpost = 1000)
prob_matrix <- matrix(NA, nrow = length(Y_val), ncol = 5)
prob_matrix[,1] <- fit_CEU$prob.test.mean
prob_matrix[,2] <- fit_FIN$prob.test.mean
prob_matrix[,3] <- fit_GBR$prob.test.mean
prob_matrix[,4] <- fit_IBS$prob.test.mean
prob_matrix[,5] <- fit_TSI$prob.test.mean

# select the column with the highest probability
prob_vector <- apply(prob_matrix, 1, which.max)
prediction <- case_when(prob_vector == 1 ~ "CEU",
                        prob_vector == 2 ~ "FIN",
                        prob_vector == 3 ~ "GBR",
```

```
                          prob_vector == 4 ~ "IBS",
                          prob_vector == 5 ~ "TSI")

  true_type <- c(true_type, Y_val)
  prediction_type <- c(prediction_type, prediction)
}
table(true_type, prediction_type)
```

```
##          prediction_type
## true_type CEU FIN GBR IBS TSI
##       CEU  31   1  29   7   4
##       FIN   0  63   0   0   0
##       GBR  23   0  34   3   3
##       IBS   9   0   4  56  10
##       TSI   3   0   5  16  51
```

```
mean(true_type == prediction_type)
```

```
## [1] 0.6676136
```

block for ntree = 100

```
ntree <- 100
true_type <- numeric(0)
prediction_type <- numeric(0)
for (i in 1:10) {
  X_train <- df_train[-folds[[i]],]
  Y_train <- population_train[-folds[[i]]]
  X_val <- df_train[folds[[i]],]
  Y_val <- population_train[folds[[i]]]

  Y_train_CEU <- ifelse(Y_train == "CEU", 1, 0)
  Y_train_FIN <- ifelse(Y_train == "FIN", 1, 0)
  Y_train_GBR <- ifelse(Y_train == "GBR", 1, 0)
  Y_train_IBS <- ifelse(Y_train == "IBS", 1, 0)
  Y_train_TSI <- ifelse(Y_train == "TSI", 1, 0)


  fit_CEU <- mc.gbart(X_train, Y_train_CEU, x.test = X_val,
                      type = 'lbart', mc.cores = 4,
                     ntree = ntree,
                      nskip = 2000,
                      ndpost = 1000)

  fit_FIN <- mc.gbart(X_train, Y_train_FIN, x.test = X_val,
                      type = 'lbart', mc.cores = 4,
                     ntree = ntree,
                      nskip = 2000,
                      ndpost = 1000)
```

```
  fit_GBR <- mc.gbart(X_train, Y_train_GBR, x.test = X_val,
                      type = 'lbart', mc.cores = 4,
                     ntree = ntree,
                      nskip = 2000,
                      ndpost = 1000)

  fit_IBS <- mc.gbart(X_train, Y_train_IBS, x.test = X_val,
                      type = 'lbart', mc.cores = 4,
                     ntree = ntree,
                      nskip = 2000,
                      ndpost = 1000)

  fit_TSI <- mc.gbart(X_train, Y_train_TSI, x.test = X_val,
                      type = 'lbart', mc.cores = 4,
                     ntree = ntree,
                      nskip = 2000,
                      ndpost = 1000)
  prob_matrix <- matrix(NA, nrow = length(Y_val), ncol = 5)
  prob_matrix[,1] <- fit_CEU$prob.test.mean
  prob_matrix[,2] <- fit_FIN$prob.test.mean
  prob_matrix[,3] <- fit_GBR$prob.test.mean
  prob_matrix[,4] <- fit_IBS$prob.test.mean
  prob_matrix[,5] <- fit_TSI$prob.test.mean

  # select the column with the highest probability
  prob_vector <- apply(prob_matrix, 1, which.max)
  prediction <- case_when(prob_vector == 1 ~ "CEU",
                          prob_vector == 2 ~ "FIN",
                          prob_vector == 3 ~ "GBR",
                          prob_vector == 4 ~ "IBS",
                          prob_vector == 5 ~ "TSI")

  true_type <- c(true_type, Y_val)
  prediction_type <- c(prediction_type, prediction)
}
table(true_type, prediction_type)
```

```
##          prediction_type
## true_type CEU FIN GBR IBS TSI
##       CEU  29   2  28   4   9
##       FIN   0  63   0   0   0
##       GBR  20   0  37   3   3
##       IBS   7   0   5  58   9
##       TSI   3   0   4  15  53
```

```
mean(true_type == prediction_type)
```

```
## [1] 0.6818182
```

block for ntree = 500

```
ntree <- 500
true_type <- numeric(0)
prediction_type <- numeric(0)
for (i in 1:10) {
  X_train <- df_train[-folds[[i]],]
  Y_train <- population_train[-folds[[i]]]
  X_val <- df_train[folds[[i]],]
  Y_val <- population_train[folds[[i]]]

  Y_train_CEU <- ifelse(Y_train == "CEU", 1, 0)
  Y_train_FIN <- ifelse(Y_train == "FIN", 1, 0)
  Y_train_GBR <- ifelse(Y_train == "GBR", 1, 0)
  Y_train_IBS <- ifelse(Y_train == "IBS", 1, 0)
  Y_train_TSI <- ifelse(Y_train == "TSI", 1, 0)


  fit_CEU <- mc.gbart(X_train, Y_train_CEU, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_FIN <- mc.gbart(X_train, Y_train_FIN, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_GBR <- mc.gbart(X_train, Y_train_GBR, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_IBS <- mc.gbart(X_train, Y_train_IBS, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_TSI <- mc.gbart(X_train, Y_train_TSI, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)
  prob_matrix <- matrix(NA, nrow = length(Y_val), ncol = 5)
  prob_matrix[,1] <- fit_CEU$prob.test.mean
  prob_matrix[,2] <- fit_FIN$prob.test.mean
  prob_matrix[,3] <- fit_GBR$prob.test.mean
  prob_matrix[,4] <- fit_IBS$prob.test.mean
  prob_matrix[,5] <- fit_TSI$prob.test.mean
```

```
  # select the column with the highest probability
  prob_vector <- apply(prob_matrix, 1, which.max)
  prediction <- case_when(prob_vector == 1 ~ "CEU",
                          prob_vector == 2 ~ "FIN",
                          prob_vector == 3 ~ "GBR",
                          prob_vector == 4 ~ "IBS",
                          prob_vector == 5 ~ "TSI")

  true_type <- c(true_type, Y_val)
  prediction_type <- c(prediction_type, prediction)
}
table(true_type, prediction_type)
```

```
##          prediction_type
## true_type CEU FIN GBR IBS TSI
##       CEU  33   2  23   7   7
##       FIN   0  63   0   0   0
##       GBR  20   0  38   4   1
##       IBS   6   0   7  55  11
##       TSI   1   0   5  17  52
```

```
mean(true_type == prediction_type)
```

```
## [1] 0.6846591
```

On test

```
ntree = 50
X_train <- df_train
X_val <- df_test
Y_val <- population_test

  Y_train_CEU <- ifelse(population_train == "CEU", 1, 0)
  Y_train_FIN <- ifelse(population_train == "FIN", 1, 0)
  Y_train_GBR <- ifelse(population_train == "GBR", 1, 0)
  Y_train_IBS <- ifelse(population_train == "IBS", 1, 0)
  Y_train_TSI <- ifelse(population_train == "TSI", 1, 0)


  fit_CEU <- mc.gbart(X_train, Y_train_CEU, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)

  fit_FIN <- mc.gbart(X_train, Y_train_FIN, x.test = X_val,
                 type = 'lbart', mc.cores = 4,
                ntree = ntree,
                 nskip = 2000,
                 ndpost = 1000)
```

```r
fit_GBR <- mc.gbart(X_train, Y_train_GBR, x.test = X_val,
                    type = 'lbart', mc.cores = 4,
                   ntree = ntree,
                    nskip = 2000,
                    ndpost = 1000)

fit_IBS <- mc.gbart(X_train, Y_train_IBS, x.test = X_val,
                    type = 'lbart', mc.cores = 4,
                   ntree = ntree,
                    nskip = 2000,
                    ndpost = 1000)

fit_TSI <- mc.gbart(X_train, Y_train_TSI, x.test = X_val,
                    type = 'lbart', mc.cores = 4,
                   ntree = ntree,
                    nskip = 2000,
                    ndpost = 1000)
prob_matrix <- matrix(NA, nrow = length(Y_val), ncol = 5)
prob_matrix[,1] <- fit_CEU$prob.test.mean
prob_matrix[,2] <- fit_FIN$prob.test.mean
prob_matrix[,3] <- fit_GBR$prob.test.mean
prob_matrix[,4] <- fit_IBS$prob.test.mean
prob_matrix[,5] <- fit_TSI$prob.test.mean

# select the column with the highest probability
prob_vector <- apply(prob_matrix, 1, which.max)
prediction <- case_when(prob_vector == 1 ~ "CEU",
                        prob_vector == 2 ~ "FIN",
                        prob_vector == 3 ~ "GBR",
                        prob_vector == 4 ~ "IBS",
                        prob_vector == 5 ~ "TSI")


table(Y_val, prediction, dnn = c("true", "prediction"))
```

```
##      prediction
## true  CEU FIN GBR IBS TSI
##   CEU  11   0  13   1   2
##   FIN   0  36   0   0   0
##   GBR   7   0  18   2   1
##   IBS   1   1   0  23   3
##   TSI   0   0   2   5  25
```

```r
print(paste("accuracy is:", mean(Y_val == prediction)))
```

```
## [1] "accuracy is: 0.748344370860927"
```