combination

library package

```
Warning: package 'pROC' was built under R version 4.3.3

Warning: package 'bartMachine' was built under R version 4.3.3

Warning: package 'randomForest' was built under R version 4.3.3

Warning: package 'missForest' was built under R version 4.3.3

Warning: package 'ggplot2' was built under R version 4.3.3

Warning: package 'dbarts' was built under R version 4.3.3

Warning: package 'BART' was built under R version 4.3.3
```

Combine all package

```
bart_package <- function(train_x,train_y,test_x,test_y,package,repeat_time,
                         num_trees,alpha,beta) {
  # define value
  results <- data.frame()
  time_BART <- c()
  mse_BART <- c()
  time_dbarts <- c()
  mse_dbarts <- c()
  time_BM <- c()
  mse_BM <- c()
```

```
time_SB <- c()
mse_SB <- c()
packageName <- ""
for(i in 1:repeat_time){
  for(package_name in package){
    if (package_name == "BART"){
      t <- system.time({fit <- pbart(x.train = train_x,
                                     y.train = train_y,
                                     x.test = test_x,
                                     ntree = num_trees,
                                     base = alpha,
                                     power = beta)})

      e_time <- as.numeric(t["elapsed"])
      predictions <- colMeans(fit$yhat.test)
      mse_score <- mean((test_y - predictions)^2)
      time_BART[i] <- e_time
      mse_BART[i] <- mse_score

    }

    if(package_name =="dbarts"){
      t <- system.time({bart_model <- bart(
        x.train = train_x,
        y.train = train_y,
        x.test = test_x,
        ntree = num_trees,
        power = beta,
        base = alpha
        )})

      e_time <- as.numeric(t["elapsed"])
      predictions <- colMeans(bart_model$yhat.test)
      mse_score <- mean((test_y - predictions)^2)
      time_dbarts[i] <- e_time
      mse_dbarts[i] <- mse_score

    }

    if(package_name=="bartMachine"){
      bart_model <- bartMachine(
        X = train_x,
```

```
        y = train_y,
        num_trees = num_trees,
        beta = beta,
        alpha = alpha

      )
      # The value of calculating the time required for modeling
      e_time <- bart_model$time_to_build
      predictions <- predict(bart_model,test_x,type = "prob")
      mse_score <- mean((test_y - predictions)^2)
      time_BM[i] <- e_time
      mse_BM[i] <- mse_score

    }

    if(package_name=="SoftBart"){
      t <- system.time({bart_model <- softbart(X = train_x, Y = train_y, X_test = test_x,
                          hypers = Hypers(train_x, train_y, num_tree = num_trees, gamma = a
                          opts = Opts(num_burn = 200, num_save = 1000, update_tau = TRUE))}
      #print(t)
      e_time <- as.numeric(t["elapsed"])
      #print(e_time)
      predictions <- bart_model$y_hat_test_mean
      mse_score <- mean((test_y - predictions)^2)
      time_SB[i] <- e_time
      mse_SB[i] <- mse_score
      #print(1)
    }
  }
}
for (package_name in package) {
  if (package_name == "BART"){
    time_mu <- mean(time_BART)
    time_sd <- sd(time_BART)
    mse_mu <- mean(mse_BART)
    mse_sd <- sd(mse_BART)
    new_row <- data.frame(MSE_mean = mse_mu, MSE_se = mse_sd,
                          running_time_mean = time_mu,running_time_sd = time_sd,
                          package_name = package_name)
    results <- rbind(results, new_row)
  }
  if (package_name == "dbarts"){
```

```
      time_mu <- mean(time_dbarts)
      time_sd <- sd(time_dbarts)
      mse_mu <- mean(mse_dbarts)
      mse_sd <- sd(mse_dbarts)
      new_row <- data.frame(MSE_mean = mse_mu, MSE_se = mse_sd,
                            running_time_mean = time_mu,running_time_sd = time_sd,
                            package_name = package_name)
      results <- rbind(results, new_row)
    }
    if (package_name == "bartMachine"){
      time_mu <- mean(time_BM)
      time_sd <- sd(time_BM)
      mse_mu <- mean(mse_BM)
      mse_sd <- sd(mse_BM)
      new_row <- data.frame(MSE_mean = mse_mu, MSE_se = mse_sd,
                            running_time_mean = time_mu,running_time_sd = time_sd,
                            package_name = package_name)
      results <- rbind(results, new_row)
    }
    if (package_name == "SoftBart"){
      time_mu <- mean(time_SB)
      time_sd <- sd(time_SB)
      mse_mu <- mean(mse_SB)
      mse_sd <- sd(mse_SB)
      new_row <- data.frame(MSE_mean = mse_mu, MSE_se = mse_sd,
                            running_time_mean = time_mu,running_time_sd = time_sd,
                            package_name = package_name)
      results <- rbind(results, new_row)
    }
  }
  return(results)
}
```

```
set.seed(316)
cancer <- read.csv("C:/Users/pyk/Desktop/nus/RA/project/imodels-data-master/data_cleaned/brea
x <- cancer[,-18]
y <- cancer[,18]

train_indices <- createDataPartition(y, p = 0.8, list = FALSE)
train_x <- x[train_indices, ]
test_x <- x[-train_indices, ]
train_y <- y[train_indices]
```

```r
test_y <- y[-train_indices]

p <- c("BART","dbarts","bartMachine","SoftBart")
```

```r
fit <- bart_package(train_x,train_y,test_x,test_y,package = p,repeat_time = 3,
                    num_trees = 20,alpha = 0.95,beta = 2)
```

```
*****Into main of pbart
*****Data:
data:n,p,np: 222, 17, 55
y1,yn: 1, 0
x1,x[n*p]: 3.000000, 1.000000
xp1,xp[np*p]: 4.000000, 0.000000
*****Number of Trees: 20
*****Number of Cut Points: 5 ... 1
*****burn and ndpost: 100, 1000
*****Prior:mybeta,alpha,tau: 2.000000,0.950000,0.335410
*****binaryOffset: -0.558392
*****Dirichlet:sparse,theta,omega,a,b,rho,augment: 0,0,1,0.5,1,17,0
*****nkeeptrain,nkeeptest,nkeeptreedraws: 1000,1000,1000
*****printevery: 100
*****skiptr,skipte,skiptreedraws: 1,1,1

MCMC
done 0 (out of 1100)
done 100 (out of 1100)
done 200 (out of 1100)
done 300 (out of 1100)
done 400 (out of 1100)
done 500 (out of 1100)
done 600 (out of 1100)
done 700 (out of 1100)
done 800 (out of 1100)
done 900 (out of 1100)
done 1000 (out of 1100)
time: 0s
check counts
trcnt,tecnt: 1000,1000

Running BART with binary y
```

```
number of trees: 20
number of chains: 1, number of threads 1
tree thinning rate: 1
Prior:
    k prior fixed to 2.000000
    power and base for tree prior: 2.000000 0.950000
    use quantiles for rule cut points: false
    proposal probabilities: birth/death 0.50, swap 0.10, change 0.40; birth 0.50
data:
    number of training observations: 222
    number of test observations: 55
    number of explanatory variables: 17

Cutoff rules c in x<=c vs x>c
Number of cutoffs: (var: number of possible c):
(1: 100) (2: 100) (3: 100) (4: 100) (5: 100)
(6: 100) (7: 100) (8: 100) (9: 100) (10: 100)
(11: 100) (12: 100) (13: 100) (14: 100) (15: 100)
(16: 100) (17: 100)
offsets:
    reg : 0.00 0.00 0.00 0.00 0.00
    test: 0.00 0.00 0.00 0.00 0.00
Running mcmc loop:
iteration: 100 (of 1000)
iteration: 200 (of 1000)
iteration: 300 (of 1000)
iteration: 400 (of 1000)
iteration: 500 (of 1000)
iteration: 600 (of 1000)
iteration: 700 (of 1000)
iteration: 800 (of 1000)
iteration: 900 (of 1000)
iteration: 1000 (of 1000)
total seconds in loop: 0.058036

Tree sizes, last iteration:
[1] 3 1 3 3 3 3 2 2 2 2 2 3 2 2 3 1 2 2
2 2

Variable Usage, last iteration (var:count):
(1: 0) (2: 0) (3: 3) (4: 2) (5: 3)
(6: 3) (7: 1) (8: 0) (9: 1) (10: 2)
(11: 2) (12: 0) (13: 3) (14: 0) (15: 2)
```

```
(16: 1) (17: 2)
DONE BART

bartMachine initializing with 20 trees...
bartMachine vars checked...
Warning: The response y is integer, bartMachine will run regression.
bartMachine java init...
bartMachine factors created...
bartMachine before preprocess...
bartMachine after preprocess... 17 total features...
bartMachine sigsq estimated...
bartMachine training data finalized...
Now building bartMachine for regression...
evaluating in sample data...done
[1] "Preprocessing data frame"
[1] "Using default grouping; if this is not desired, preprocess data frame manually using pr
Finishing warmup 100
Finishing warmup 200
Finishing save 100
Finishing save 200
Finishing save 300
Finishing save 400
Finishing save 500
Finishing save 600
Finishing save 700
Finishing save 800
Finishing save 900
Finishing save 1000
*****Into main of pbart
*****Data:
data:n,p,np: 222, 17, 55
y1,yn: 1, 0
x1,x[n*p]: 3.000000, 1.000000
xp1,xp[np*p]: 4.000000, 0.000000
*****Number of Trees: 20
*****Number of Cut Points: 5 ... 1
*****burn and ndpost: 100, 1000
*****Prior:mybeta,alpha,tau: 2.000000,0.950000,0.335410
*****binaryOffset: -0.558392
*****Dirichlet:sparse,theta,omega,a,b,rho,augment: 0,0,1,0.5,1,17,0
*****nkeeptrain,nkeeptest,nkeeptreedraws: 1000,1000,1000
*****printevery: 100
*****skiptr,skipte,skiptreedraws: 1,1,1
```

```
MCMC
done 0 (out of 1100)
done 100 (out of 1100)
done 200 (out of 1100)
done 300 (out of 1100)
done 400 (out of 1100)
done 500 (out of 1100)
done 600 (out of 1100)
done 700 (out of 1100)
done 800 (out of 1100)
done 900 (out of 1100)
done 1000 (out of 1100)
time: 0s
check counts
trcnt,tecnt: 1000,1000

Running BART with binary y

number of trees: 20
number of chains: 1, number of threads 1
tree thinning rate: 1
Prior:
    k prior fixed to 2.000000
    power and base for tree prior: 2.000000 0.950000
    use quantiles for rule cut points: false
    proposal probabilities: birth/death 0.50, swap 0.10, change 0.40; birth 0.50
data:
    number of training observations: 222
    number of test observations: 55
    number of explanatory variables: 17

Cutoff rules c in x<=c vs x>c
Number of cutoffs: (var: number of possible c):
(1: 100) (2: 100) (3: 100) (4: 100) (5: 100)
(6: 100) (7: 100) (8: 100) (9: 100) (10: 100)
(11: 100) (12: 100) (13: 100) (14: 100) (15: 100)
(16: 100) (17: 100)
offsets:
    reg : 0.00 0.00 0.00 0.00 0.00
    test: 0.00 0.00 0.00 0.00 0.00
Running mcmc loop:
iteration: 100 (of 1000)
```

```
iteration: 200 (of 1000)
iteration: 300 (of 1000)
iteration: 400 (of 1000)
iteration: 500 (of 1000)
iteration: 600 (of 1000)
iteration: 700 (of 1000)
iteration: 800 (of 1000)
iteration: 900 (of 1000)
iteration: 1000 (of 1000)
total seconds in loop: 0.058565

Tree sizes, last iteration:
[1] 2 3 2 1 3 2 3 3 2 2 2 3 3 2 3 3 2 3
2 2

Variable Usage, last iteration (var:count):
(1: 1) (2: 2) (3: 6) (4: 1) (5: 2)
(6: 1) (7: 2) (8: 1) (9: 1) (10: 3)
(11: 0) (12: 0) (13: 3) (14: 1) (15: 0)
(16: 1) (17: 3)
DONE BART

bartMachine initializing with 20 trees...
bartMachine vars checked...
Warning: The response y is integer, bartMachine will run regression.
bartMachine java init...
bartMachine factors created...
bartMachine before preprocess...
bartMachine after preprocess... 17 total features...
bartMachine sigsq estimated...
bartMachine training data finalized...
Now building bartMachine for regression...
evaluating in sample data...done
[1] "Preprocessing data frame"
[1] "Using default grouping; if this is not desired, preprocess data frame manually using pre
Finishing warmup 100
Finishing warmup 200
Finishing save 100
Finishing save 200
Finishing save 300
Finishing save 400
Finishing save 500
Finishing save 600
```

```
Finishing save 700
Finishing save 800
Finishing save 900
Finishing save 1000
*****Into main of pbart
*****Data:
data:n,p,np: 222, 17, 55
y1,yn: 1, 0
x1,x[n*p]: 3.000000, 1.000000
xp1,xp[np*p]: 4.000000, 0.000000
*****Number of Trees: 20
*****Number of Cut Points: 5 ... 1
*****burn and ndpost: 100, 1000
*****Prior:mybeta,alpha,tau: 2.000000,0.950000,0.335410
*****binaryOffset: -0.558392
*****Dirichlet:sparse,theta,omega,a,b,rho,augment: 0,0,1,0.5,1,17,0
*****nkeeptrain,nkeeptest,nkeeptreedraws: 1000,1000,1000
*****printevery: 100
*****skiptr,skipte,skiptreedraws: 1,1,1

MCMC
done 0 (out of 1100)
done 100 (out of 1100)
done 200 (out of 1100)
done 300 (out of 1100)
done 400 (out of 1100)
done 500 (out of 1100)
done 600 (out of 1100)
done 700 (out of 1100)
done 800 (out of 1100)
done 900 (out of 1100)
done 1000 (out of 1100)
time: 0s
check counts
trcnt,tecnt: 1000,1000

Running BART with binary y

number of trees: 20
number of chains: 1, number of threads 1
tree thinning rate: 1
Prior:
    k prior fixed to 2.000000
```

```
    power and base for tree prior: 2.000000 0.950000
    use quantiles for rule cut points: false
    proposal probabilities: birth/death 0.50, swap 0.10, change 0.40; birth 0.50
data:
    number of training observations: 222
    number of test observations: 55
    number of explanatory variables: 17

Cutoff rules c in x<=c vs x>c
Number of cutoffs: (var: number of possible c):
(1: 100) (2: 100) (3: 100) (4: 100) (5: 100)
(6: 100) (7: 100) (8: 100) (9: 100) (10: 100)
(11: 100) (12: 100) (13: 100) (14: 100) (15: 100)
(16: 100) (17: 100)
offsets:
    reg : 0.00 0.00 0.00 0.00 0.00
    test: 0.00 0.00 0.00 0.00 0.00
Running mcmc loop:
iteration: 100 (of 1000)
iteration: 200 (of 1000)
iteration: 300 (of 1000)
iteration: 400 (of 1000)
iteration: 500 (of 1000)
iteration: 600 (of 1000)
iteration: 700 (of 1000)
iteration: 800 (of 1000)
iteration: 900 (of 1000)
iteration: 1000 (of 1000)
total seconds in loop: 0.057012

Tree sizes, last iteration:
[1] 3 2 2 2 2 2 2 3 2 2 2 2 3 2 2 2 2 2
2 1

Variable Usage, last iteration (var:count):
(1: 1) (2: 4) (3: 3) (4: 3) (5: 0)
(6: 1) (7: 1) (8: 1) (9: 1) (10: 0)
(11: 0) (12: 1) (13: 1) (14: 2) (15: 0)
(16: 2) (17: 1)
DONE BART

bartMachine initializing with 20 trees...
bartMachine vars checked...
```

```
Warning: The response y is integer, bartMachine will run regression.
bartMachine java init...
bartMachine factors created...
bartMachine before preprocess...
bartMachine after preprocess... 17 total features...
bartMachine sigsq estimated...
bartMachine training data finalized...
Now building bartMachine for regression...
evaluating in sample data...done
[1] "Preprocessing data frame"
[1] "Using default grouping; if this is not desired, preprocess data frame manually using pre
Finishing warmup 100
Finishing warmup 200
Finishing save 100
Finishing save 200
Finishing save 300
Finishing save 400
Finishing save 500
Finishing save 600
Finishing save 700
Finishing save 800
Finishing save 900
Finishing save 1000
```

```
print(fit)
```

```
   MSE_mean       MSE_se running_time_mean running_time_sd package_name
1 1.3951936 0.0222138705        0.25000000      0.00000000         BART
2 1.3727706 0.0097309230        0.07666667      0.02081666       dbarts
3 0.1845267 0.0002833019        0.31948725      0.12082123  bartMachine
4 0.1931440 0.0011923028        4.32333333      0.08504901     SoftBart
```