combination

library package

Combine all package

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

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

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

```
print(fit)
```

```
   MSE_mean       MSE_se running_time_mean running_time_sd package_name
1 1.3951936 0.0222138705        0.26333333      0.01154701         BART
2 1.3727706 0.0097309230        0.07333333      0.01527525       dbarts
3 0.1850653 0.0007108292        0.32366776      0.10976634  bartMachine
4 0.1931440 0.0011923028        4.31666667      0.09073772     SoftBart
```

The mean reason causes different in MSE_mean is for bartMachine and SoftBart package, the predict function in it only provides the predicted value i.e. probability instead of posterior mean, one possible method to deal this is to try inverse sigmoid function to trace it back.