

parameter search

load data

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

```
set.seed(316)
num_trees_grid <- c(20,50,100)
alpha_grid <- c(0.8,0.95,0.98)
beta_grid <- c(1.5,2,3)
k_folds <- 5 # Number of folds for cross-validation

folds <- cut(seq(1, nrow(cancer)), breaks=k_folds, labels=FALSE)

# Store results
results <- expand.grid(num_trees=num_trees_grid, alpha = alpha_grid,
                      beta= beta_grid,MSE=rep(NA, 1))

a <- 0
# Perform grid search
for(i in seq_along(num_trees_grid)) {
  num_trees <- num_trees_grid[i]
  for (k in seq_along(alpha_grid)) {
    base <- alpha_grid[k]
    for (b in seq_along(beta_grid)) {
      power <- beta_grid[b]
      mse_values <- numeric(k_folds)
      a <- a+1
      for(j in 1:k_folds) {
        test_indices <- which(folds == j)
```

```

train_indices <- setdiff(1:nrow(X), test_indices)

model <- dbarts::bart(X[train_indices, ], y[train_indices], X[test_indices, ], ntree=n
predictions <- colMeans(model$yhat.test)

mse_values[j] <- mean((predictions - y[test_indices])^2)

}
results$MSE[a] <- mean(mse_values)
}
}

# Print results

```

```
print(results)
```

	num_trees	alpha	beta	MSE
1	20	0.80	1.5	1.252947
2	50	0.80	1.5	1.253675
3	100	0.80	1.5	1.241940
4	20	0.95	1.5	1.314428
5	50	0.95	1.5	1.270600
6	100	0.95	1.5	1.252864
7	20	0.98	1.5	1.300624
8	50	0.98	1.5	1.279215
9	100	0.98	1.5	1.260024
10	20	0.80	2.0	1.262275
11	50	0.80	2.0	1.253862
12	100	0.80	2.0	1.239696
13	20	0.95	2.0	1.308172
14	50	0.95	2.0	1.283266
15	100	0.95	2.0	1.259743
16	20	0.98	2.0	1.301528
17	50	0.98	2.0	1.306474
18	100	0.98	2.0	1.274252
19	20	0.80	3.0	1.269992
20	50	0.80	3.0	1.261605
21	100	0.80	3.0	1.237408
22	20	0.95	3.0	1.317825
23	50	0.95	3.0	1.281176

24	100	0.95	3.0	1.257129
25	20	0.98	3.0	1.318714
26	50	0.98	3.0	1.298806
27	100	0.98	3.0	1.266316

```
value <- min(results$MSE)
position <- which(results$MSE == value)
print(results[position,])
```

	num_trees	alpha	beta	MSE
21	100	0.8	3	1.237408

```
ctyc <- read.csv("C:/Users/pyk/Desktop/nus/RA/project/imodels-data-master/data_cleaned/compa
x_ctyc <- ctyc[,-21]
y_ctyc <- ctyc[,21]
```

```
set.seed(316)
num_trees_grid <- c(50,100,200)
alpha_grid <- c(0.8,0.95,0.98)
beta_grid <- c(1.5,2,3)
k_folds <- 5 # Number of folds for cross-validation

folds <- cut(seq(1, nrow(ctyc)), breaks=k_folds, labels=FALSE)

# Store results
results <- expand.grid(num_trees=num_trees_grid, alpha = alpha_grid,
                      beta= beta_grid,MSE=rep(NA, 1))

a <- 0
# Perform grid search
for(i in seq_along(num_trees_grid)) {
  num_trees <- num_trees_grid[i]
  for (k in seq_along(alpha_grid)) {
    base <- alpha_grid[k]
    for (b in seq_along(beta_grid)) {
      power <- beta_grid[b]
      mse_values <- numeric(k_folds)
      a <- a+1
      for(j in 1:k_folds) {
        test_indices <- which(folds == j)
        train_indices <- setdiff(1:nrow(x_ctyc), test_indices)
```

```

    model <- dbarts::bart(x_ctyc[train_indices, ], y_ctyc[train_indices], x_ctyc[test_ind
    predictions <- colMeans(model$yhat.test)

    mse_values[j] <- mean((predictions - y_ctyc[test_indices])^2)

  }
  results$MSE[a] <- mean(mse_values)
}
}
}

```

```

# Print results
print(results)

```

	num_trees	alpha	beta	MSE
1	50	0.80	1.5	0.6447327
2	100	0.80	1.5	0.6431614
3	200	0.80	1.5	0.6425735
4	50	0.95	1.5	0.6489128
5	100	0.95	1.5	0.6444509
6	200	0.95	1.5	0.6429400
7	50	0.98	1.5	0.6458521
8	100	0.98	1.5	0.6452667
9	200	0.98	1.5	0.6427172
10	50	0.80	2.0	0.6421922
11	100	0.80	2.0	0.6400061
12	200	0.80	2.0	0.6389862
13	50	0.95	2.0	0.6440227
14	100	0.95	2.0	0.6431181
15	200	0.95	2.0	0.6393751
16	50	0.98	2.0	0.6449493
17	100	0.98	2.0	0.6426552
18	200	0.98	2.0	0.6417414
19	50	0.80	3.0	0.6385891
20	100	0.80	3.0	0.6382041
21	200	0.80	3.0	0.6357946
22	50	0.95	3.0	0.6423487
23	100	0.95	3.0	0.6390865
24	200	0.95	3.0	0.6372355
25	50	0.98	3.0	0.6432754

```
26      100  0.98  3.0 0.6415529
27      200  0.98  3.0 0.6376705
```

```
value <- min(results$MSE)
position <- which(results$MSE == value)
print(results[position,])
```

```
      num_trees alpha beta      MSE
21      200    0.8    3 0.6357946
```

```
heart <- read.csv("C:/Users/pyk/Desktop/nus/RA/project/imodels-data-master/data_cleaned/heart.csv")
x_h <- heart[,-16]
y_h <- heart[,16]
```

```
set.seed(316)
num_trees_grid <- c(50,100,20)
alpha_grid <- c(0.8,0.95,0.98)
beta_grid <- c(1.5,2,3)
k_folds <- 5 # Number of folds for cross-validation

folds <- cut(seq(1, nrow(heart)), breaks=k_folds, labels=FALSE)

# Store results
results <- expand.grid(num_trees=num_trees_grid, alpha = alpha_grid,
                      beta= beta_grid,MSE=rep(NA, 1))

a <- 0
# Perform grid search
for(i in seq_along(num_trees_grid)) {
  num_trees <- num_trees_grid[i]
  for (k in seq_along(alpha_grid)) {
    base <- alpha_grid[k]
    for (b in seq_along(beta_grid)) {
      power <- beta_grid[b]
      mse_values <- numeric(k_folds)
      a <- a+1
      for(j in 1:k_folds) {
        test_indices <- which(folds == j)
        train_indices <- setdiff(1:nrow(x_h), test_indices)

        model <- dbarts::bart(x_h[train_indices, ], y_h[train_indices], x_h[test_indices, ], n
```

```

        predictions <- colMeans(model$yhat.test)

        mse_values[j] <- mean((predictions - y_ctyc[test_indices])^2)
    }
    results$MSE[a] <- mean(mse_values)
}
}

# Print results

```

```
print(results)
```

	num_trees	alpha	beta	MSE
1	50	0.80	1.5	2.733050
2	100	0.80	1.5	2.748144
3	20	0.80	1.5	2.679370
4	50	0.95	1.5	2.918833
5	100	0.95	1.5	2.867490
6	20	0.95	1.5	2.800716
7	50	0.98	1.5	2.980651
8	100	0.98	1.5	2.880775
9	20	0.98	1.5	2.847538
10	50	0.80	2.0	2.717876
11	100	0.80	2.0	2.710765
12	20	0.80	2.0	2.669370
13	50	0.95	2.0	2.899393
14	100	0.95	2.0	2.848871
15	20	0.95	2.0	2.760285
16	50	0.98	2.0	2.951987
17	100	0.98	2.0	2.865779
18	20	0.98	2.0	2.840585
19	50	0.80	3.0	2.795444
20	100	0.80	3.0	2.759004
21	20	0.80	3.0	2.722035
22	50	0.95	3.0	2.907166
23	100	0.95	3.0	2.854623
24	20	0.95	3.0	2.850748
25	50	0.98	3.0	3.065787
26	100	0.98	3.0	2.869779

```
27          20  0.98  3.0 2.851192
```

```
value <- min(results$MSE)
position <- which(results$MSE == value)
print(results[position,])
```

```
  num_trees alpha beta    MSE
12         20  0.8    2 2.66937
```