STATS 790
Final Project

Dean Hansen - 400027416

28 April, 2023

Contents

Introduction

In this project, we code the Random Forest algorithm invented by Leo Breiman in 2001 from scratch in R (source). We will compare the randomForest package against our naive implementation of the algorithm using the California housing dataset described in the Dataset section.

Random Forest Algorithm

The Random Forest algorithm is an enhancement to the bagging algorithm, also invented by Leo Breiman. One key difference between the two is when each decision tree is grown, a random subset of features are used to grow the tree at each terminal node. In bagging, the full set of features are used, whereas in a Random Forest, using a subset of features prevents correlation between the trees.

Dataset

The California housing dataset, found in section 10.14.1 of ESL, comes from the paper on "Sparse spatial autoregressions" found here. There are $20,640$ observations, 8 predictor variables and one response variable, median house value. A detailed description of each variable can be found below.

Variable	**Description**
median_income	median income of California block
housing_median_age	median age of the house
total_rooms	aggregate total number of rooms
total_bedrooms	aggregate total number of bedrooms
population	population of geographic block
households	number of houses
latitude	latitude of geographic block
longitude	longitude of geographic block
median_house_value (response*)	median house value

We downloaded the California housing dataset from the Statlib website found here. The features in this dataset are numeric, so we scale them to have mean zero and unit variance prior to fitting

our random forest model. Since we have few variables and no categorical data, the data processing step is straightforward.

First, we first and profile using tidymodels out of the box implementation of randomForest using the ranger engine.

```
housing_splits <- initial_split(housing)
housing_training <- training(housing_splits)
housing_testing  <- testing(housing_splits)

rf_recipe <-
  recipe(median_house_value ~ ., data = housing_training)

rf_mod <-
  rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine("ranger") %>%
  set_mode("regression")

rf_workflow <-
  workflow() %>%
  add_model(rf_mod) %>%
  add_recipe(rf_recipe)

rf_folds <- vfold_cv(housing_training)

rf_res <-
  rf_workflow %>%
  tune_grid(grid = 10,
            control = control_grid(save_pred = TRUE),
            metrics = metric_set(rmse, rsq, ccc),
            resamples = rf_folds)

best_rf_res <- select_best(rf_res, "rmse")
```