

Stat 432 Homework 9

Assigned: Oct 21, 2024; Due: 11:59 PM CT, Oct 31, 2024

- Instruction
- Question 1: A Simulation Study for Random Forests [50 pts]
- Question 2: Parameter Tuning with OOB Prediction [20 pts]
- Question 3: Using `xgboost` [30 pts]

Instruction

Please remove this section when submitting your homework.

Students are encouraged to work together on homework and/or utilize advanced AI tools. However, **sharing, copying, or providing any part of a homework solution or code to others** is an infraction of the University's rules on Academic Integrity (<https://studentcode.illinois.edu/article1/part4/1-401/>). Any violation will be punished as severely as possible. Final submissions must be uploaded to Gradescope (<https://www.gradescope.com/courses/570816>). No email or hard copy will be accepted. For **late submission policy and grading rubrics** (<https://teazrq.github.io/stat432/syllabus.html>), please refer to the course website.

- You are required to submit the rendered file `HWx_yourNetID.pdf`. For example, `HW01_rqzhu.pdf`. Please note that this must be a `.pdf` file. `.html` format **cannot** be accepted. Make all of your R code chunks visible for grading.
- Include your Name and NetID in the report.
- If you use this file or the example homework `.Rmd` file as a template, be sure to **remove this instruction** section.
- Make sure that you **set seed** properly so that the results can be replicated if needed.
- For some questions, there will be restrictions on what packages/functions you can use. Please read the requirements carefully. As long as the question does not specify such restrictions, you can use anything.
- **When using AI tools**, you are encouraged to document your comment on your experience with AI tools especially when it's difficult for them to grasp the idea of the question.
- **On random seed and reproducibility**: Make sure the version of your R is $\geq 4.0.0$. This will ensure your random seed generation is the same as everyone else. Please note that updating the R version may require you to reinstall all of your packages.

Question 1: A Simulation Study for Random Forests [50 pts]

We learned that random forests have several key parameters and some of them are also involved in trading the bias and variance. To confirm some of our understandings, we will conduct a simulation study to investigate each of them:

1. The terminal node size `nodesize`
2. The number of variables randomly sampled as candidates at each split `mtry`
3. The number of trees in the forest `ntree`

For this question, we will use the `randomForest` package. This package is quite slow, so you may want to try smaller amount of simulations first to make sure your code is correct.

a. [5 pts] Generate the data using the following model:

$$Y = X_1 + X_2 + \epsilon,$$

where the two covariates X_1 and X_2 are independently from standard normal distribution and $\epsilon \sim N(0, 1)$. Generate a training set of size 200 and a test set of size 300 using this model. Fit a random forest model to the training set with the default parameters. Report the MSE on the test set.

b. [15 pts] Let's analyze the effect of the terminal node size `nodesize`. We will consider the following values for `nodesize`: 2, 5, 10, 15, 20 and 30. Set `mtry` as 1 and the bootstrap sample size as 150. For each value of `nodesize`, fit a random forest model to the training set and record the MSE on the test set. Then repeat this process 100 times and report (plot) the average MSE against the `nodesize`. Same idea of the simulation has been considered before when we worked on the KNN model. After getting the results, answer the following questions:

- Do you think our choice of the `nodesize` parameter is reasonable? What is the optimal node size you obtained? If you don't think the choice is reasonable, re-define your range of tuning and report your results and the optimal node size.
- What is the effect of `nodesize` on the bias-variance trade-off?

c. [15 pts] In this question, let's analyze the effect of `mtry`. We will consider a new data generator:

$$Y = 0.2 \times \sum_{j=1}^5 X_j + \epsilon,$$

where we generate a total of 10 covariates independently from standard normal distribution and $\epsilon \sim N(0, 1)$. Generate a training set of size 200 and a test set of size 300 using the model above. Fix the node size as 3, the bootstrap sample size as 150, and consider `mtry` to be all integers from 1 to 10. Perform the simulation study with 100 runs, report your results using a plot, and answer the following questions:

- * What is the optimal value of ``mtry`` you obtained?
- * What is the effect of ``mtry`` on the bias-variance trade-off?

d. [15 pts] In this question, let's analyze the effect of `ntree`. We will consider the same data generator as in part (c). Fix the node size as 10, the bootstrap sample size as 150, and `mtry` as 3. Consider the following values for `ntree`: 1, 2, 3, 5, 10, 50. Perform the simulation study with 100 runs. For this question, we do not need to calculate the prediction of all subjects. Instead, calculate just the prediction on a target point that all the covariate values are 0. After obtaining the simulation results, calculate the variance of the random forest estimator under different `ntree` values (for the definition of variance of an estimator, see our previous homework on the bias-variance simulation). Comment on your findings.

Question 2: Parameter Tuning with OOB Prediction [20 pts]

We will again use the MNIST dataset. We will use the first 2600 observations of it:

```

# inputs to download file
fileLocation <- "https://pjreddie.com/media/files/mnist_train.csv"
numRowsToDownload <- 2600
localFileName <- paste0("mnist_first", numRowsToDownload, ".RData")

# download the data and add column names
mnist2600 <- read.csv(fileLocation, nrow = numRowsToDownload)
numColsMnist <- dim(mnist2600)[2]
colnames(mnist2600) <- c("Digit", paste("Pixel", seq(1:(numColsMnist - 1)), sep = ""))

# save file
# in the future we can read in from the local copy instead of having to redownload
save(mnist2600, file = localFileName)

# you can load the data with the following code
#load(file = localFileName)
dim(mnist2600)

```

```
## [1] 2600 785
```

- [5 pts] Similar to what we have done before, split the data into a training set of size 1300 and a test set of the remaining data. Then keep only the digits 2, 4 and 8. After this screen the data and only keep the top 250 variables with the highest variance.
- [15 pts] Fit classification random forests to the training set and tune parameters `mtry` and `nodesize`. Choose 4 values for each of the parameters. Use `ntree = 1000` and keep all other parameters as default. To perform the tuning, you must use the OOB prediction. Report your results for each tuning and the optimal choice. After this, use the random forest corresponds to the optimal tuning to predict the testing data, and report the confusion matrix and the accuracy.

Question 3: Using xgboost [30 pts]

- [20 pts] We will use the same data as in Question 2. Use the `xgboost` package to fit the MNIST data multi-class classification problem. You should specify the following:
 - Use `multi:softmax` as the objective function so that it can handle multi-class classification
 - Use `num_class = 3` to specify the number of classes
 - Use `gbtree` as the base learner
 - Tune these parameters:
 - The learning rate `eta = 0.5`
 - The maximum depth of trees `max_depth = 2`
 - The number of trees `nrounds = 100`

Report the testing error rate and the confusion matrix.

- [10 pts] The model fits with 100 rounds (trees) sequentially. However, you can produce your prediction using just a limited number of trees. This can be controlled using the `iterationrange` argument in the `predict()` function. Plot your prediction error vs. number of trees. Comment on your results.