Machine Learning Block2 Lab1

Lepeng Zhang, Xuan Wang, Priyarani Patil

2023-11-24

The group report was made based on the discussion after all of us had finished all three assignments.

Assignment 1 was mainly contributed by Lepeng Zhang.

Assignment 2 was mainly contributed by Xuan Wang.

Assignment 2 was mainly contributed by Priyarani Patil.

Assignment 1. ENSEMBLE METHODS

Q1

See appendix.

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Train and predict for 1, 10, and 100 trees with condition "x1<x2" and nodesize=25
```

```
##   ntrees mean_misclassification_error variance_misclassification_error
## 1      1                       0.220                       0.17177177
## 2     10                       0.150                       0.12762763
## 3    100                       0.098                       0.08848448
```

Q2

See appendix.

```
## Train and predict for 1, 10, and 100 trees with condition "x1<0.5" and nodesize=25
```

```
##   ntrees mean_misclassification_error variance_misclassification_error
## 1      1                       0.001                             0.001
## 2     10                       0.001                             0.001
## 3    100                       0.001                             0.001
```

Q3

See appendix.

```
## Train and predict for 1, 10, and 100 trees with condition "(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5)" an
```

```
##   ntrees mean_misclassification_error variance_misclassification_error
## 1      1                       0.097                       0.08767868
## 2     10                       0.093                       0.08443544
## 3    100                       0.039                       0.03751652
```

Q4

question4.1

As the number of trees in the random forest grows, the mean and variance error rate typically decreases. This is because a random forest is an ensemble method that averages the predictions of many decision trees to reduce overfitting and improve prediction accuracy.

question4.2

This is because a random forest can model complex interactions between features by averaging the predictions of many decision trees, each of which can capture different aspects of the feature space. Therefore, as the complexity of the problem increases, the benefit of using a random forest over a single decision tree becomes more apparent. However, this also depends on the specific characteristics of the data and the problem at hand.

Assignment 2. MIXTURE MODELS

Q1

See appendix.

Appendix

Code for Assignment 1

Q1

```
# Load necessary library
library(randomForest)

generate_train_test_data <- function(size,condition) {
  # Check the arguments
  if ( !is.numeric(size) ) {
    stop("The argument size should be numeric type.")
  }
  conditions <- c("x1<x2", "x1<0.5", "(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5)")
  if ( !condition %in% conditions) {
    stop("The argument condition should be one of those: `x1<x2`, `x1<0.5`, `(x1<0.5 & x2<0.5) | (x1>0.5
  }

  # Set seed for reproducibility
  set.seed(1234)

  # Generate dataset
  x1 <- runif(size)
  x2 <- runif(size)
  #datasets <- cbind(x1, x2)
  y <- as.numeric(eval(parse(text=condition)))
  #true_labels <- as.factor(y)
  datasets <- data.frame(x1=x1, x2=x2, true_labels=as.factor(y))
  return(datasets)
}

# Function to train and predict with random forest
```

```r
train_and_predict <- function(train_dataset,test_dataset,ntree,nodesize) {
  rf_model <- randomForest(true_labels ~ ., data = train_dataset, ntree = ntree, nodesize = nodesize, k
  predicted_labels <- predict(rf_model, newdata = test_dataset)
  mean_mis_error <- mean(predicted_labels != test_dataset$true_labels)
  var_mis_error <- var(predicted_labels != test_dataset$true_labels)
  return(list(predicted_labels=predicted_labels,mean_mis_error=mean_mis_error,var_mis_error=var_mis_err
}

# Initialize variables
ntrees <- c(1, 10, 100)
nodesizes <- c(25, 12)

# Train and predict for 1, 10, and 100 trees with condition "x1<x2" and nodesize=25
results_list1 <- list()
train_dataset1 <- generate_train_test_data(size = 100, condition = 'x1<x2' )
test_dataset1 <- generate_train_test_data(size = 1000, condition = 'x1<x2' )
for ( i in 1:length(ntrees) ) {
  results_list1$ntrees[i] <- ntrees[i]
  rf_model_results1 <- train_and_predict(train_dataset = train_dataset1, test_dataset = test_dataset1, r
  results_list1$mean_misclassification_error[i] <- rf_model_results1$mean_mis_error
  results_list1$variance_misclassification_error[i] <- rf_model_results1$var_mis_error
  #plot(test_dataset1$x1,test_dataset1$x2,col=(as.numeric(rf_model_results1$predicted_labels)+1), main =
}
results_df1 <- as.data.frame(results_list1)
cat("Train and predict for 1, 10, and 100 trees with condition \"x1<x2\" and nodesize=25")
print(results_df1)
```

Q2

```r
# Train and predict for 1, 10, and 100 trees with condition "x1<0.5" and nodesize=25
results_list2 <- list()
train_dataset2 <- generate_train_test_data(size = 100, condition = 'x1<0.5' )
test_dataset2 <- generate_train_test_data(size = 1000, condition = 'x1<0.5' )
for ( i in 1:length(ntrees) ) {
  results_list2$ntrees[i] <- ntrees[i]
  rf_model_results2 <- train_and_predict(train_dataset = train_dataset2, test_dataset = test_dataset2, r
  results_list2$mean_misclassification_error[i] <- rf_model_results2$mean_mis_error
  results_list2$variance_misclassification_error[i] <- rf_model_results2$var_mis_error
  #plot(test_dataset2$x1,test_dataset2$x2,col=(as.numeric(rf_model_results2$predicted_labels)+1), main =
}
results_df2 <- as.data.frame(results_list2)
cat("Train and predict for 1, 10, and 100 trees with condition \"x1<0.5\" and nodesize=25")
print(results_df2)
```

Q3

```r
# Train and predict for 1, 10, and 100 trees with condition "(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5)" and
results_list3 <- list()
train_dataset3 <- generate_train_test_data(size = 100, condition = '(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5
test_dataset3 <- generate_train_test_data(size = 1000, condition = '(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5
for ( i in 1:length(ntrees) ) {
  results_list3$ntrees[i] <- ntrees[i]
  rf_model_results3 <- train_and_predict(train_dataset = train_dataset3, test_dataset = test_dataset3, r
```

```
  results_list3$mean_misclassification_error[i] <- rf_model_results3$mean_mis_error
  results_list3$variance_misclassification_error[i] <- rf_model_results3$var_mis_error
  #plot(test_dataset3$x1,test_dataset3$x2,col=(as.numeric(rf_model_results3$predicted_labels)+1), main =
}
results_df3 <- as.data.frame(results_list3)
cat("Train and predict for 1, 10, and 100 trees with condition \"(x1<0.5 & x2<0.5) | (x1>0.5 & x2>0.5)\"
print(results_df3)
```

Code for Assignment 2

Q1