Homework 2: Classification Methods

Wen Li Teng

February 2 2020

```r
# Load packages
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 3.6.2
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(ggplot2)
```

Exploring Simulated Differences between LDA and QDA

```r
# 4(a) Vary n in non-linear Bayes decision boundary approach

finderror <- function(rownum) {

  # Simulate a dataset
```

```r
  set.seed(rownum)

  n_obs <- 1000
  min <- -1
  max <- 1

  q4_data <- tibble(X1 = runif(n_obs, min, max), X2 = runif(n_obs, min, max))

  codeY <- function(simY) {
    if(simY >= 0)
      output <- TRUE
    if (simY < 0)
      output <- FALSE
    return(output)
  }

  q4_data <- q4_data %>%
    mutate (E = rnorm(1000, 0, 1)) %>%
    mutate (simY = X1 + X1^2 + X2 + X2^2 + E) %>%
    rowwise() %>%
    mutate(Y = codeY(simY))

  # Randomly split dataset
  split <- initial_split(q4_data, prop = 0.7)
  train <- training(split)
  test <- testing(split)

  # Use training dataset to estimate LDA and QDA models
  lda_q4 <- MASS::lda(Y ~ X1 + X1^2 + X2 + X2^2 + E, data = q4_data)
  qda_q4 <- MASS::qda(Y ~ X1 + X1^2 + X2 + X2^2 + E, data = q4_data)

  # Calculate model's training and test error rate
  test_predicted_lda <- predict(lda_q4, newdata = test)
  test_predicted_qda <- predict(qda_q4, newdata = test)
  lda_cm <- table(test$Y, test_predicted_lda$class)
  qda_cm <- table(test$Y, test_predicted_qda$class)

  results <- as.data.frame(test) %>%
    mutate(lda.pred = (test_predicted_lda$class)) %>%
    mutate(qda.pred = (test_predicted_qda$class)) %>%
    summarize(lda.error = mean(Y != lda.pred),
              qda.error = mean(Y != qda.pred))

  return(results)
}
```

```r
vary_sim <- c(1e02, 1e03, 1e04, 1e05)
```

```r
for(i in vary_sim) {
  sim <- 1:i
  lda_error <- vector("numeric", i)
  qda_error <- vector("numeric", i)
  results <- as.data.frame(cbind(sim, lda_error, qda_error))
```

```r
  for (i in 1:i) {
    temp_result <- finderror(i)
    results$lda_error[i] <- temp_result$lda.error
    results$qda_error[i] <- temp_result$qda.error
    remove(temp_result)
  }
  assign(paste0("results", i), results)
  remove(results)
}

write.csv(results100, "results100.csv", row.names = F)
write.csv(results1000, "results1000.csv", row.names = F)
write.csv(results10000, "results10000.csv", row.names = F)
write.csv(results100000, "results100000.csv", row.names = F)
```

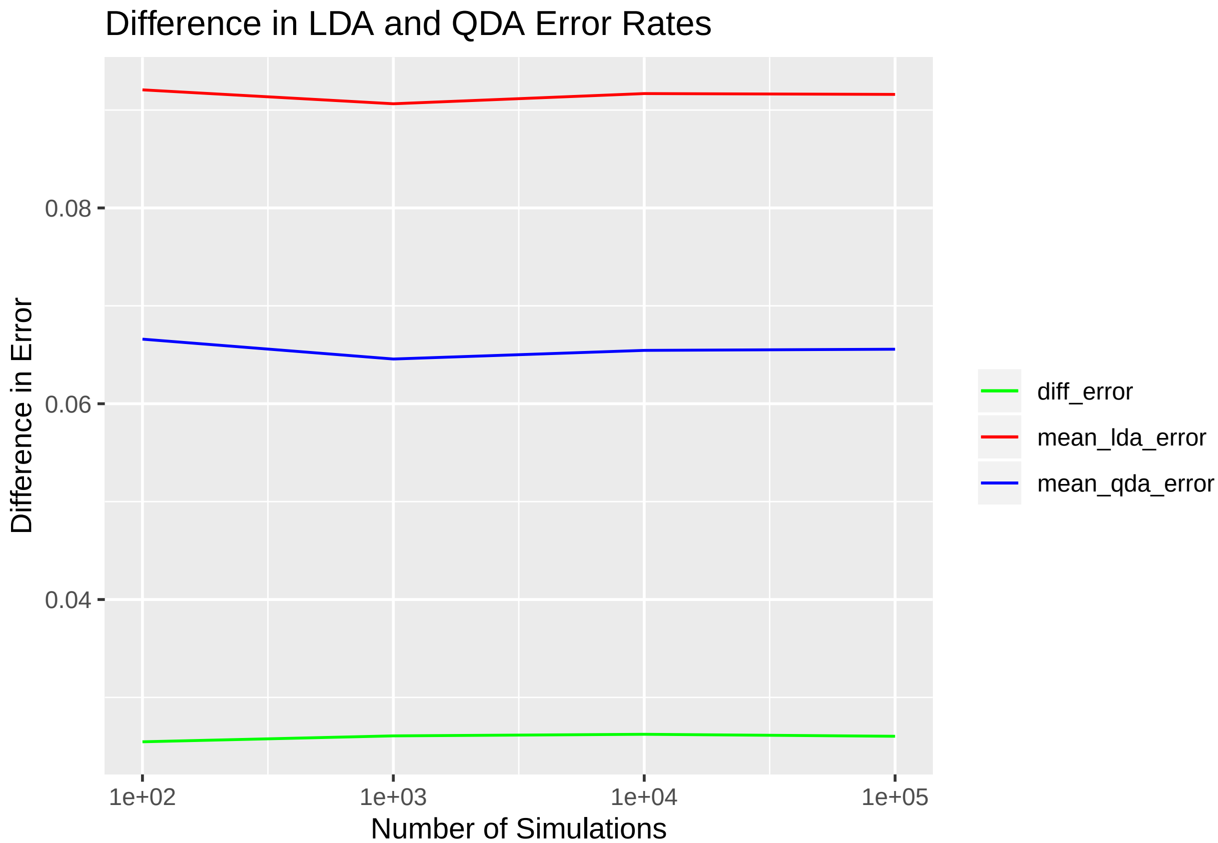
```r
# 4(b) Plot test error rate for LDA and QDA models

results100 <- read.csv("data/results100.csv")
results1000 <- read.csv("data/results1000.csv")
results10000 <- read.csv("data/results10000.csv")
results100000 <- read.csv("data/results100000.csv")

mean_error <- function(data) {
  data <- data %>%
    summarize(mean_lda_error = mean(lda_error),
              mean_qda_error = mean(qda_error)) %>%
    mutate(diff_error = mean_lda_error - mean_qda_error)
  return(data)
}

mean_100 <- mean_error(results100)
mean_1000 <- mean_error(results1000)
mean_10000 <- mean_error(results10000)
mean_100000 <- mean_error(results100000)
means_combined <- rbind(mean_100, mean_1000, mean_10000, mean_100000)
means_combined <- cbind(vary_sim, means_combined)

ggplot(means_combined, aes(x = vary_sim)) +
  geom_line(aes(y = diff_error, color = "diff_error")) +
  geom_line(aes(y = mean_lda_error, color = "mean_lda_error")) +
  geom_line(aes(y = mean_qda_error, color = "mean_qda_error")) +
  scale_x_log10(labels = scales::scientific) +
  scale_colour_manual("", values = c("diff_error"="green",
                                     "mean_lda_error" = "red",
                                     "mean_qda_error"="blue")) +
  labs(title = "Difference in LDA and QDA Error Rates",
       x = "Number of Simulations",
       y = "Difference in Error")
```



If the Bayes decision boundary is non-linear, we should expect the test error rate of QDA to improve relative to that of LDA. Observe that the plot shows how the difference between LDA and QDA is generally increasing from 1e02 to 1e04. With an increase in sample size, variance becomes less of a concern. As such, the high-variance limitation of QDA becomes less problematic as n increases.