Math189 HW4

Group Members:

Yaqi Chen (PID: A15742547; Section: A03)

Yuetong Lyu (PID: A13779993; Section: A04)

Siyi He (PID: A13400569; Section: A03)

Zhenyuan Xu (PID: A92067995; Section: A02)

Jiawei Chao (PID: A13818001; Section: A02)

Problem 1

```
library(datasets)
data(iris)
#Divide data into train and test
train=iris[c(1:40,51:90, 101:140),]
test=iris[c(41:50,91:100, 141:150),]

#Sample size
n_setosa=40
n_versicolor=40
n_virginica=40

#Prior=relative sample size in train data
for(i in 1:3){
  if(i == 1){
    p_setosa=0.8
    p_versicolor=0.1
    p_virginica=0.1
  }
  if(i == 2){
    p_setosa=0.1
    p_versicolor=0.8
    p_virginica=0.1
  }
  if(i == 3){
    p_setosa=0.1
```

```
    p_versicolor=0.1
    p_virginica=0.8
  }


  Mean_setosa=colMeans(train[1:40,1:4])
  Mean_versicolor=colMeans(train[41:80,1:4])
  Mean_virginica=colMeans(train[81:120,1:4])

  #Sample variance-covariance matrix for each species
  S_setosa=cov(train[1:40,1:4])
  S_versicolor=cov(train[41:80,1:4])
  S_virginica=cov(train[81:120,1:4])

  #Complete fomula
  S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-
1)*S_virginica)/(n_setosa+n_versicolor+n_virginica-3)

  S_inv=solve(S_pooled)

  #Simple way
  #S_pooled=(S_setosa+S_versicolor+S_virginica)/3

  ############## Calculate alpha_i ##############

  alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa
  alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor
  alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica


  ############## Calculate beta_i ##############

  beta_setosa=S_inv %*% Mean_setosa
  beta_versicolor=S_inv %*% Mean_versicolor
  beta_virginica=S_inv %*% Mean_virginica


  ############## Classification ##############
  prediction=c()
  d_setosa_vec=c()
  d_versicolor_vec=c()
  d_virginica_vec=c()
  label=c("setosa", "versicolor", "virginica")

  for(i in 1:nrow(test)){
    #Read an observation in test data
```

```
    x=t(test[i,1:4])

    #Calculate linear discriminant functions for each species
    d_setosa=alpha_setosa+ t(beta_setosa) %*% x
    d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
    d_virginica=alpha_virginica+ t(beta_virginica) %*% x

    #Classify the observation to the species with highest function value
    d_vec=c(d_setosa, d_versicolor, d_virginica)
    prediction=append(prediction, label[which.max( d_vec )])

    d_setosa_vec=append(d_setosa_vec, d_setosa)
    d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
    d_virginica_vec=append(d_virginica_vec, d_virginica)
  }

  #Combine the predicted results to the test dataset.
  test$prediction=prediction
  #see if there is any difference between the prediction and reality
  print(sum(test$Species!=test$prediction))
}
```

Output:

```
[1] 0 #difference between prediction and reality with prior (0.8,0.1,0.1)
[1] 0 #difference between prediction and reality with prior (0.1,0.8,0.1)
[1] 0 #difference between prediction and reality with prior (0.1,0.1,0.8)
```

Therefore, we think the LDA method is not sensitive to the choice of prior.

Problem 2

```
library(datasets)
data(iris)

for(i in 1:3){
  if(i == 1){
    #Divide data into train and test
    train=iris[c(1:30,51:80, 101:130),]
    test=iris[c(31:50,81:100, 131:150),]

    #Sample size
```

```
    n_setosa=30
    n_versicolor=30
    n_virginica=30

    ############## Choose Prior ##############
    #Prior=relative sample size in train data
    p_setosa=n_setosa/90
    p_versicolor=n_versicolor/90
    p_virginica=n_virginica/90

    ############## Calculate sample mean vectors ##############
    Mean_setosa=colMeans(train[1:30,1:4])
    Mean_versicolor=colMeans(train[31:60,1:4])
    Mean_virginica=colMeans(train[61:90,1:4])

    ############## Calculate pooled variance-covariance matrix ##############
    #Sample variance-covariance matrix for each species
    S_setosa=cov(train[1:30,1:4])
    S_versicolor=cov(train[31:60,1:4])
    S_virginica=cov(train[61:90,1:4])
  }
  if(i == 2){
    #Divide data into train and test
    train=iris[c(1:20,51:70, 101:120),]
    test=iris[c(21:50,71:100, 121:150),]

    #Sample size
    n_setosa=20
    n_versicolor=20
    n_virginica=20

    ############## Choose Prior ##############
    #Prior=relative sample size in train data
    p_setosa=n_setosa/60
    p_versicolor=n_versicolor/60
    p_virginica=n_virginica/60
    ############## Calculate sample mean vectors ##############
    Mean_setosa=colMeans(train[1:20,1:4])
    Mean_versicolor=colMeans(train[21:40,1:4])
    Mean_virginica=colMeans(train[41:60,1:4])

    ############## Calculate pooled variance-covariance matrix ##############
    #Sample variance-covariance matrix for each species
    S_setosa=cov(train[1:20,1:4])
    S_versicolor=cov(train[21:40,1:4])
    S_virginica=cov(train[41:60,1:4])
```

```
  }
  if(i == 3){
    #Divide data into train and test
    train=iris[c(1:10,51:60, 101:110),]
    test=iris[c(11:50,61:100, 111:150),]

    #Sample size
    n_setosa=10
    n_versicolor=10
    n_virginica=10

    ############## Choose Prior ##############
    #Prior=relative sample size in train data
    p_setosa=n_setosa/30
    p_versicolor=n_versicolor/30
    p_virginica=n_virginica/30
    ############## Calculate sample mean vectors ##############
    Mean_setosa=colMeans(train[1:10,1:4])
    Mean_versicolor=colMeans(train[11:20,1:4])
    Mean_virginica=colMeans(train[21:30,1:4])

    ############## Calculate pooled variance-covariance matrix ##############
    #Sample variance-covariance matrix for each species
    S_setosa=cov(train[1:10,1:4])
    S_versicolor=cov(train[11:20,1:4])
    S_virginica=cov(train[21:30,1:4])
  }

  #Complete fomula
  S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-
1)*S_virginica)/(n_setosa+n_versicolor+n_virginica-3)

  S_inv=solve(S_pooled)

  #Simple way
  #S_pooled=(S_setosa+S_versicolor+S_virginica)/3

  ############## Calculate alpha_i ##############

  alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa
  alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*% Mean_versicolor
  alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica


  ############## Calculate beta_i ##############
```

```
  beta_setosa=S_inv %*% Mean_setosa
  beta_versicolor=S_inv %*% Mean_versicolor
  beta_virginica=S_inv %*% Mean_virginica


  ############## Classification ##############
  prediction=c()
  d_setosa_vec=c()
  d_versicolor_vec=c()
  d_virginica_vec=c()
  label=c("setosa", "versicolor", "virginica")

  for(i in 1:nrow(test)){
    #Read an observation in test data
    x=t(test[i,1:4])

    #Calculate linear discriminant functions for each species
    d_setosa=alpha_setosa+ t(beta_setosa) %*% x
    d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
    d_virginica=alpha_virginica+ t(beta_virginica) %*% x

    #Classify the observation to the species with highest function value
    d_vec=c(d_setosa, d_versicolor, d_virginica)
    prediction=append(prediction, label[which.max( d_vec )])

    d_setosa_vec=append(d_setosa_vec, d_setosa)
    d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
    d_virginica_vec=append(d_virginica_vec, d_virginica)
  }

  #Combine the predicted results to the test dataset.
  test$prediction=prediction
  print(sum(test$Species!=test$prediction))
}
```

Output:

```
[1] 2 #number of mistake corresponding to training sample size 90
[1] 3 #number of mistake corresponding to training sample size 60
[1] 6 #number of mistake corresponding to training sample size 30
```

Therefore, the probablity of getting the correct predictions by using a 90 training sample size is 0.967.

The probablity of getting the correct predictions by using a 60 training sample size is 0.967.

The probablity of getting the correct predictions by using a 30 training sample size is 0.95.

Problem 3

```
library(datasets)
data(iris)
count<-c(1:100)
for(i in 1:100){
  randomn<-iris[sample(nrow(iris), 50,replace=F), ]
  n_setosa=length(which(randomn$Species=="setosa"))
  n_versicolor=length(which(randomn$Species=="versicolor"))
  n_virginica=length(which(randomn$Species=="virginica"))

  ############## Choose Prior ##############
  #Prior=relative sample size in train data
  p_setosa=n_setosa/50
  p_versicolor=n_versicolor/50
  p_virginica=n_virginica/50

  #Divide data into train and test
  train=randomn
  iris_test<-iris
  delete<-row.names(train)
  for(j in 1:50){
    test<-subset(iris_test,row.names(iris_test)!=delete[j])
    iris_test<-test}

  ############## Calculate sample mean vectors ##############
  Mean_setosa=colMeans(train[train$Species=="setosa",1:4])
  Mean_versicolor=colMeans(train[train$Species=="versicolor",1:4])
  Mean_virginica=colMeans(train[train$Species=="virginica",1:4])

  ############## Calculate sample mean vectors ##############
  #Sample variance-covariance matrix for each species
  S_setosa=cov(train[train$Species=="setosa",1:4])
  S_versicolor=cov(train[train$Species=="versicolor",1:4])
  S_virginica=cov(train[train$Species=="virginica",1:4])

  #Complete fomula
  S_pooled= ((n_setosa-1)*S_setosa+(n_versicolor-1)*S_versicolor+(n_virginica-
1)*S_virginica)/(n_setosa+n_versicolor+n_virginica-3)

  S_inv=solve(S_pooled)

  ############## Calculate alpha_i ##############
```

```
  alpha_setosa= -0.5* t(Mean_setosa) %*% S_inv %*% Mean_setosa + log(p_setosa)
  alpha_versicolor= -0.5* t(Mean_versicolor) %*% S_inv %*%
Mean_versicolor+log(p_versicolor)
  alpha_virginica= -0.5* t(Mean_virginica) %*% S_inv %*% Mean_virginica+
log(p_virginica)

  ############## Calculate beta_i ##############

  beta_setosa=S_inv %*% Mean_setosa
  beta_versicolor=S_inv %*% Mean_versicolor
  beta_virginica=S_inv %*% Mean_virginica


  ############## Classification ##############
  prediction=c()
  d_setosa_vec=c()
  d_versicolor_vec=c()
  d_virginica_vec=c()
  label=c("setosa", "versicolor", "virginica")

  for(k in 1:nrow(test)){
    #Read an observation in test data
    x=t(test[k,1:4])

    #Calculate linear discriminant functions for each species
    d_setosa=alpha_setosa+ t(beta_setosa) %*% x
    d_versicolor=alpha_versicolor+ t(beta_versicolor) %*% x
    d_virginica=alpha_virginica+ t(beta_virginica) %*% x

    #Classify the observation to the species with highest function value
    d_vec=c(d_setosa, d_versicolor, d_virginica)
    prediction=append(prediction, label[which.max( d_vec )])

    d_setosa_vec=append(d_setosa_vec, d_setosa)
    d_versicolor_vec=append(d_versicolor_vec, d_versicolor)
    d_virginica_vec=append(d_virginica_vec, d_virginica)
  }

  #Combine the predicted results to the test dataset.
  test$prediction=prediction
  result<-data.frame("observation"=test$Species,"classification"=test$prediction)
  falseclassify=0
  for(j in 1:100){
    if(result[j,1]!=result[j,2])
      falseclassify=falseclassify+1
```

```
  }
  count[i]=falseclassify
}
count
hist(count,breaks = seq(min(count)-0.5,max(count)+0.5,1))
```

```
[1] 3 4 4 3 2 2 3 5 2 4 2 2 3 2 6 3 7 2 5 2 1 6 2 3 2 3 4 2 3 2 3 5 3 3 6 3 1 3 2
3
 [41] 4 2 3 2 3 3 4 1 1 1 4 4 2 2 3 2 4 3 2 4 2 2 5 2 1 5 4 1 3 1 2 3 3 3 4 2 5 1
3 4
 [81] 2 1 5 3 1 3 1 2 2 5 3 5 1 1 2 2 2 4 3 4
```