Statistik IV Assignment 9

Christian Peters

No. 16

First, read the data

```
train <- read.csv('data_train.csv')
validation <- read.csv('data_valid.csv')
```

a)

Get an estimate of the covariance matrix and its inverse

```
cov_estimate <- cov(train[, 1:2])
cov_inv <- solve(cov_estimate)
```

Now, get the means for each class

```
mean_1 <- colMeans(train[train$Class==1, 1:2])
mean_2 <- colMeans(train[train$Class==2, 1:2])
mean_3 <- colMeans(train[train$Class==3, 1:2])
```

Build the discrimination functions

```
# Calculates the value of the discrimination function for a given class
discriminator <- function(x, mean, cov_inv) {
  t(mean) %*% cov_inv %*% x - 1/2 %*% t(mean) %*% cov_inv %*% mean
}

# Returns the predicted class
d <- function(x) {
  scores <- c(discriminator(x, mean_1, cov_inv),
              discriminator(x, mean_2, cov_inv),
              discriminator(x, mean_3, cov_inv))
  return(which.max(scores))
}
```

Get the predictions

```
predictions_lda_train <- apply(train[, 1:2], 1, d)
predictions_lda_validation <- apply(validation[, 1:2], 1, d)
```

Estimate the error on the training as well as on the validation set

```
(error_lda_train <- 1 - mean(predictions_lda_train == train$Class))
```

```
## [1] 0.3929619
```

```
(error_lda_validation <- 1 - mean(predictions_lda_validation == validation$Class))
```

```
## [1] 0.3919598
```

b)

Do a quadratic discriminant analysis on the training set assuming equal a'priori probabilities

```
qda_model <- qda(Class ~ X1 + X2, data = train, prior = c(1/3, 1/3, 1/3))
```

Get the predictions

```
predictions_qda_train <- predict(qda_model, train)$class
predictions_qda_validation <- predict(qda_model, validation)$class
```

Estimate the error on the training as well as on the validation set

```
(error_qda_train <- 1 - mean(predictions_qda_train == train$Class))
```

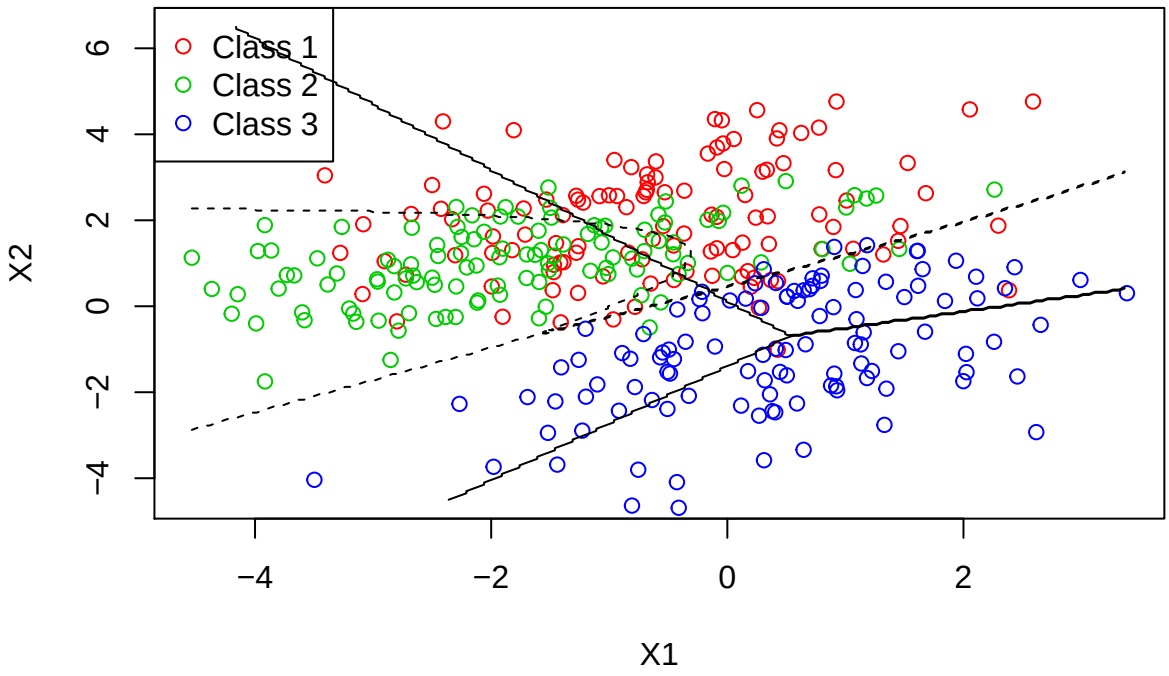
```
## [1] 0.2199413
```

```
(error_qda_validation <- 1 - mean(predictions_qda_validation == validation$Class))
```

```
## [1] 0.2361809
```

c)

Visualize the data as well as the decision boundaries for the lda and for the qda. The boundaries of the lda are drawn with solid lines while the boundaries of the qda are drawn with dashed lines.



d)

Implement the k-Nearest-Neighbor classifier based on the training set

```r
knn_classifier <- function(x, k) {
  distances <- apply(train[, 1:2], 1, function(w) norm(x-w, type='2'))
  neighbor_indices <- order(distances)[1:k]
  neighbor_votes <- table(train$Class[neighbor_indices])
  vote_result <- as.integer(names(which.max(neighbor_votes)))
  return(vote_result)
}
```

Get the predictions on the validation set using k=4

```r
predictions_knn_validation <- apply(validation[, 1:2], 1, function(x) knn_classifier(x, 4))
```

Estimate the validation error

```r
(error_knn_validation <- 1 - mean(predictions_knn_validation == validation$Class))
```

```
## [1] 0.2763819
```

e)

As we can see by looking at the error rates, the simple lda delivered the worst results, it's validation error being roughly 39.2%. The error on the validation set is roughly the same as the error on the training set for this method. The qda lead to better results, showing a validation error of about 23.6%. This is most likely due to the higher complexity of this model.

With a validation error of roughly 27.6%, the kNN classifier ranks second among the models. The interesting thing about this is that this error is only about 4 percentage points worse than that of the qda despite not making any assumptions on the distribution of the data.