4 Classfiers for Iris Data

Contents

1 Background

The goal of this practice is to compare the k-NN classifier, linear discriminant analysis (LDA), Logistic Regression Model in a binary classification problem.

Here we considerFisher's iris data set. Extract the data corresponding to the flower types versicolor and virginica, numbering a total of 100 flowers. Set aside the first 15 observations for each flower type as test data and use the remaining data consisting of 75 observations (with flower types as class labels) as training data

```
# loading the data
library("dplyr")
data("iris")
versicolor <- iris %>% filter(Species == "versicolor")
virginica <- iris %>% filter(Species == "virginica")

# testing data
test_data <- rbind(versicolor[1:15,],virginica[1:15,])
glimpse(test_data)
```

```
## Rows: 30
## Columns: 5
## $ Sepal.Length <dbl> 7.0, 6.4, 6.9, 5.5, 6.5, 5.7, 6.3, 4.9, 6.6, 5.2, 5.0,...
## $ Sepal.Width  <dbl> 3.2, 3.2, 3.1, 2.3, 2.8, 2.8, 3.3, 2.4, 2.9, 2.7, 2.0,...
## $ Petal.Length <dbl> 4.7, 4.5, 4.9, 4.0, 4.6, 4.5, 4.7, 3.3, 4.6, 3.9, 3.5,...
## $ Petal.Width  <dbl> 1.4, 1.5, 1.5, 1.3, 1.5, 1.3, 1.6, 1.0, 1.3, 1.4, 1.0,...
## $ Species      <fct> versicolor, versicolor, versicolor, versicolor, versic...
```

```
# training data
train_data <- rbind(versicolor[-(1:15),],virginica[-(1:15),])
glimpse(train_data)
```

```
## Rows: 70
## Columns: 5
## $ Sepal.Length <dbl> 6.7, 5.6, 5.8, 6.2, 5.6, 5.9, 6.1, 6.3, 6.1, 6.4, 6.6,...
## $ Sepal.Width  <dbl> 3.1, 3.0, 2.7, 2.2, 2.5, 3.2, 2.8, 2.5, 2.8, 2.9, 3.0,...
## $ Petal.Length <dbl> 4.4, 4.5, 4.1, 4.5, 3.9, 4.8, 4.0, 4.9, 4.7, 4.3, 4.4,...
## $ Petal.Width  <dbl> 1.4, 1.5, 1.0, 1.5, 1.1, 1.8, 1.3, 1.5, 1.2, 1.3, 1.4,...
## $ Species      <fct> versicolor, versicolor, versicolor, versicolor, versic...
```

```
# dropping unused factor levels `setosa` for the factor `Species`
train_data$Species <- droplevels(train_data$Species)
test_data$Species <- droplevels(test_data$Species)
```

2 Linear Discriminant Analysis (LDA)

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda.fit = lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = train_data)
knitr::kable(lda.fit$means,
             caption = "The class-specific means of the predictor variables for the training data.")
```

Table 1: The class-specific means of the predictor variables for the training data.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
versicolor	5.920000	2.765714	4.265714	1.322857
virginica	6.642857	3.002857	5.540000	2.014286

The confusion matrix for the test data is summarized in the following table. The precision rate is 100%.

```
lda.pred = predict(lda.fit, test_data)
lda.conf = table(true = test_data$Species, predicted =lda.pred$class)
knitr::kable(lda.conf,
             caption = "The confusion matrix for the test data using LDA")
```

Table 2: The confusion matrix for the test data using LDA

	versicolor	virginica
versicolor	15	0
virginica	0	15

```
# precision rate
sum(diag(lda.conf))/30
```

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

3 Logistic Regression Model

```
glm = glm(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
          family = binomial, data = iris)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm)
```

```
##
## Call:
## glm(formula = Species ~ Sepal.Length + Sepal.Width + Petal.Length +
##     Petal.Width, family = binomial, data = iris)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.173e-05  -2.100e-08   2.100e-08   2.100e-08   3.185e-05
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)      16.946 457457.097       0        1
## Sepal.Length    -11.759 130504.037       0        1
## Sepal.Width      -7.842  59415.373       0        1
## Petal.Length     20.088 107724.589       0        1
## Petal.Width      21.608 154350.604       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.9095e+02  on 149  degrees of freedom
## Residual deviance: 3.2940e-09  on 145  degrees of freedom
## AIC: 10
##
## Number of Fisher Scoring iterations: 25
```

The confusion matrix for the test data using the logistic regression model is given in the following table. The misclassification error rate is 50%.

```
# predicted probabilities
glm.pred.prob = predict(glm, test_data, type = "response")
# predicted labels
glm.pred = ifelse(glm.pred.prob > 0.5, " virginica", "versicolor")

# confusion matrix
glm.conf = table(true = test_data$Species, predicted = glm.pred)
knitr::kable(glm.conf,
             caption = "The confusion matrix for the test data")
```

Table 3: The confusion matrix for the test data

	virginica
versicolor	15
virginica	15

4 k-NN

When $k = 5$, the confusion matrix for the test data is summarized in the following table. The precision rate is 96.7%.

```
library("class")
## the kNN classifier with k = 5

knn5 = knn(
  train = train_data[,-5],
  test = test_data[,-5],
  cl = train_data$Species,
  k = 5
)
# confusion matrix
knn5.conf = table(true = test_data$Species, predicted = knn5)
knitr::kable(knn5.conf,
             caption = "The confusion matrix for test data using kNN (k=5)")
```

Table 4: The confusion matrix for test data using kNN (k=5)

	versicolor	virginica
versicolor	15	0
virginica	1	14

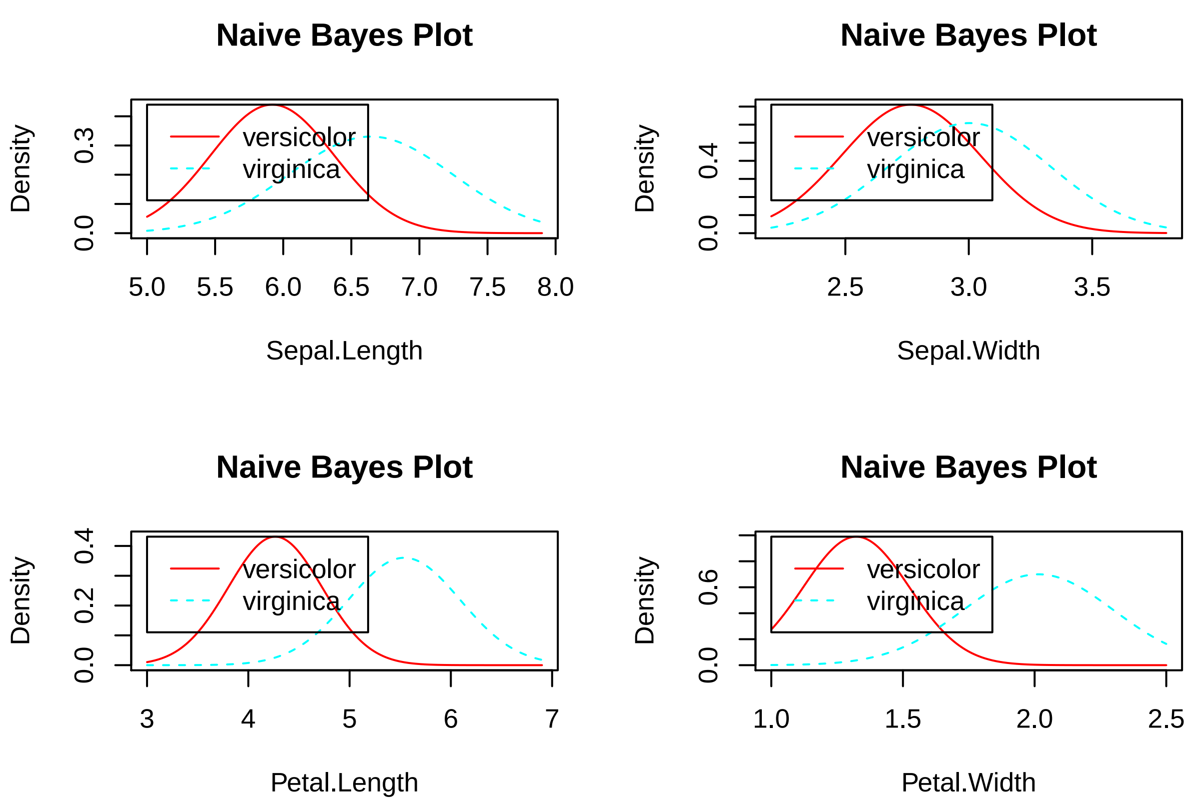
```r
# precision
sum(diag(knn5.conf))/30
```

```
## [1] 0.9666667
```

5 Naive Bayes Classifier

The confusion matrix of Naive Bayes Classifier on the test data is summarized in the following table. The precision rate is 93.3%.

```r
# fit the model
library(klaR)
fit.bayes <- NaiveBayes(Species ~., data = train_data)
# fit.bayes[1:length(fit.bayes)]
par(mfrow = c(2,2))
plot(fit.bayes)
```



```r
# predictions
bayes.prediction <- predict(fit.bayes, test_data[,-5])$class

# confusion matrix
bayes.conf = table(true = test_data$Species, predicted = bayes.prediction)
knitr::kable(bayes.conf,
             caption = "The confusion matrix for test data using Naive Bayes")
```

Table 5: The confusion matrix for test data using Naive Bayes

	versicolor	virginica
versicolor	14	1
virginica	1	14

```r
# precision
sum(diag(bayes.conf))/30
```

```
## [1] 0.9333333
```

6 References

Gareth James, Daniela Witten, Trevor Hastie Robert Tibshirani (2013), An Introduction to Statistical Learning with Applications in R.