Classification-LDA

Mohammad Saqib Ansari

2023-12-18

Linear Discriminant Analysis (LDA)

Introduction

Linear Discriminant Analysis (LDA) is a supervised machine learning technique used for classification tasks. It's a method that finds the optimal linear combination of features to distinguish between two or more classes in a dataset. LDA is commonly used for dimensionality reduction and feature extraction in pattern classification.

Objective

The primary goal of LDA is to project the input features onto a lower-dimensional space while maximizing the separability between classes. This is achieved by finding a set of features that best discriminates among classes.

Assumptions

LDA is based on several key assumptions:

1. **Normality**: It assumes that the features within each class follow a normal distribution.
2. **Equal Covariance**: It assumes that all classes have the same covariance matrix.
3. **Independence of Features**: It assumes that the features are statistically independent within each class.

Linear Discriminant Analysis Steps

Step 1: Compute Class Means

Calculate the mean of each feature for each class.

Step 2: Compute Within-Class Scatter Matrix

Compute the scatter matrices for each class, representing the spread of data points within each class.

Step 3: Compute Between-Class Scatter Matrix

Compute the scatter matrix that measures the spread of the class means around the overall mean.

Step 4: Compute Eigenvectors and Eigenvalues

Calculate the eigenvectors and eigenvalues of the matrix obtained by the inverse of the within-class scatter matrix multiplied by the between-class scatter matrix.

Step 5: Select Discriminants

Select the top eigenvectors corresponding to the largest eigenvalues to form the transformation matrix.

Step 6: Project Data onto Lower-Dimensional Space

Project the dataset onto the subspace formed by the selected eigenvectors to obtain the transformed features.

Advantages of LDA

- LDA is effective in reducing the dimensionality of data while preserving most of the class discriminatory information.
- It assumes linear relationships between features, which can be beneficial in certain scenarios.
- LDA works well with small to medium-sized datasets.

Limitations of LDA

- LDA assumes that the data is normally distributed and classes have equal covariance matrices, which might not always hold true.
- It is sensitive to outliers in the data.
- LDA can only provide linear decision boundaries.

Conclusion

Linear Discriminant Analysis is a valuable technique for classification and dimensionality reduction. By maximizing class separability, LDA identifies the most informative features to distinguish between different classes in a dataset.

Certainly! Below is the provided R code translated into an R Markdown format along with explanations for each step:

Linear Discriminant Analysis in R

Step 1: Read the CSV Data

```r
# Read the CSV file
dati <- read.csv("insect.csv")

# Display the first few rows of the dataset
head(dati)
```

```
##   species joint1 joint2 aedeagus
## 1       a    191    131       53
## 2       a    185    134       50
## 3       a    200    137       52
## 4       a    173    127       50
## 5       a    171    128       49
## 6       a    160    118       47
```

```r
# Summary statistics of the dataset
summary(dati)
```

```
##    species              joint1          joint2         aedeagus    
##  Length:20          Min.   :160.0   Min.   :107.0   Min.   :43.00  
##  Class :character   1st Qu.:181.5   1st Qu.:121.0   1st Qu.:48.50  
##  Mode  :character   Median :189.5   Median :127.0   Median :50.00  
##                     Mean   :193.7   Mean   :125.6   Mean   :49.70  
##                     3rd Qu.:208.8   3rd Qu.:131.0   3rd Qu.:51.25  
##                     Max.   :242.0   Max.   :144.0   Max.   :54.00  
```

```r
# Dimensions of the dataset
dim(dati)
```

```
## [1] 20  4
```

Step 2: Constructing the LDA Model

```r
# Load necessary libraries
library(MASS)

# Construct the LDA model using all features except 'species'
model <- lda(species ~ ., data = dati)

# Construct a model using specific features ('joint1' and 'aedeagus')
model1 <- lda(species ~ joint1 + aedeagus, data = dati)
model1
```

```
## Call:
## lda(species ~ joint1 + aedeagus, data = dati)
## 
## Prior probabilities of groups:
##   a   b 
## 0.5 0.5 
## 
## Group means:
```

```
##   joint1 aedeagus
## a  179.1     50.5
## b  208.2     48.9
##
## Coefficients of linear discriminants:
##                 LD1
## joint1    0.1152350
## aedeagus -0.5813965
```

```r
# Construct a model with predefined prior probabilities
model2 <- lda(species ~ ., data = dati, prior = c(0.6, 0.4))
model2
```

```
## Call:
## lda(species ~ ., data = dati, prior = c(0.6, 0.4))
##
## Prior probabilities of groups:
##   a   b
## 0.6 0.4
##
## Group means:
##   joint1 joint2 aedeagus
## a  179.1  128.4     50.5
## b  208.2  122.8     48.9
##
## Coefficients of linear discriminants:
##                  LD1
## joint1    0.13225339
## joint2   -0.07941509
## aedeagus -0.52655608
```

Step 3: Estimating the Classification Error Rate

```r
# Predict classes using Cross-Validation and calculate the confusion matrix
pred <- lda(species ~ ., data = dati, CV = TRUE)$class
table(pred, dati$species)
```

```
##
## pred  a  b
##    a 10  2
##    b  0  8
```

Step 4: Making Predictions

```r
# Create a new dataset for prediction
mat <- matrix(c(157, 127, 56, 171, 125, 49), nrow = 2, ncol = 3, byrow = TRUE)
new_d <- as.data.frame(mat)
names(new_d) <- names(dati)[-1]  # Use column names from original dataset
```

```r
# Make predictions on the new dataset
predict(model, newdata = new_d)
```

```
## $class
## [1] a a
## Levels: a b
##
## $posterior
##           a            b
## 1 1.0000000 3.481760e-19
## 2 0.9999982 1.766086e-06
##
## $x
##         LD1
## 1 -8.275571
## 2 -2.579301
```

“‘

Explanation:

- **Step 1** involves reading the CSV data ("insect.csv") into R, displaying the first few rows, summarizing statistics, and checking the dimensions of the dataset.
- **Step 2** constructs the Linear Discriminant Analysis (LDA) models using different approaches:
  - `model`: Uses all features except 'species' to predict 'species'.
  - `model1`: Constructs a model using specific features ('joint1' and 'aedeagus').
  - `model2`: Constructs a model with predefined prior probabilities.
- **Step 3** involves estimating the classification error rate by using Cross-Validation (CV). It predicts classes and computes the confusion matrix to evaluate the model's performance.
- **Step 4** creates a new dataset (`new_d`) and uses the previously built model to make predictions on this new dataset.