Linear Disriminant Analysis

Olena Smotrova

18/02/2018

Contents

Bayes' Theorem for Classification

Suppose that Y is a qualitative response variable and can take $K, K \geq 2$ distinct and unordered values. We denote p different predictors as $X = (X_1, X_2, ..., X_p)$. Rather than modeling response Y directly, linear disriminant analysis models the probability that Y belongs to a paticular category. Bayes' Theorem states that

$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \tag{1}$$

where π_k is a probability that a random choosen observations belongs to kth class. $f_k(X) = Pr(X = x|Y = k)$ is the density function of X for an observation that comes from kth class. $p_k(x) = Pr(Y = k|X = x)$ is the probability that an observation $X = x$ belongs to kth class, given the predictor value for observation.

Linear Disriminant Analysis for $p = 1$

The classification method is described by [1]. We would like to obtain an estimate for $f_k(x)$ in order to estimape $p_k(x)$. We will classify an observation to the class for wich $p_k(x)$ is greatest. Assume that $p = 1$, we have only one predictor. Assume that $f_k(x)$ is normal or Gaussian. The normal density in one dimension takes form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2) \tag{2}$$

where μ_k and σ_k^2 are the mean and variance parameters for the kth class. Assume further that $\sigma_1^2 = \sigma_2^2 = ... = \sigma_k^2 = \sigma^2$ is the same variance across all K classes. Put (2) in expression (1), we obtain

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2\sigma^2}(x - \mu_k)^2)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2\sigma^2}(x - \mu_l)^2)} \tag{3}$$

Classifier assigns an observation $X = x$ to the class for wich p_k is largest. Taking log of (3) and rearranging the terms we obtain

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + ln(\pi_k) \tag{4}$$

This equalent to assigning observation to the class for which $\delta(x)$ is largest. In practice we have to estimate parameters $\mu_1, ..., \mu_K$, $\pi_1, ..., \pi_k$ and σ.

$$\hat{\delta_k}(x) = x \cdot \frac{\hat{\mu_k}}{\hat{\sigma^2}} - \frac{\hat{\mu_k}^2}{2\hat{\sigma^2}} + ln(\pi_k) \tag{5}$$

Discriminant functions $\hat{\delta}_k$ in (5) are linear functions of x.

Now perform LDA on wine data set. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

```
wine = read.csv("./Data/wine.data.txt")
names(wine) = c('Label', 'Alcohol', 'Malic acid', 'Ash',
                'Alcalinity of ash','Magnesium', 'Total phenols',
                'Flavanoids', 'Nonflavanoid phenols', 'Proanthocyanins',
                'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline')

wine$Label = as.factor(wine$Label)
str(wine)
```

```
## 'data.frame':    177 obs. of  14 variables:
##  $ Label                       : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Alcohol                     : num  13.2 13.2 14.4 13.2 14.2 ...
##  $ Malic acid                  : num  1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 2.16 ...
##  $ Ash                         : num  2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 2.3 ...
##  $ Alcalinity of ash           : num  11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 18 ...
##  $ Magnesium                   : int  100 101 113 118 112 96 121 97 98 105 ...
##  $ Total phenols               : num  2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 2.95 ...
##  $ Flavanoids                  : num  2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 3.32 ...
##  $ Nonflavanoid phenols        : num  0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 0.22 ...
##  $ Proanthocyanins             : num  1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 2.38 ...
##  $ Color intensity             : num  4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 5.75 ...
##  $ Hue                         : num  1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 1.25 ...
##  $ OD280/OD315 of diluted wines: num  3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 3.17 ...
##  $ Proline                     : int  1050 1185 1480 735 1450 1290 1295 1045 1045 1510 ...
```

We separate `wine` data set into training and test data set.

```
spl = sample.split(wine$Label, SplitRatio = 0.735) # CaTools
wineTrain = subset(wine, spl == TRUE)
wineTest = subset(wine, spl == FALSE)

table(wineTrain$Label)
```

```
##
##  1  2  3
## 43 52 35
```

```
table(wineTest$Label)
```

```
##
##  1  2  3
## 15 19 13
```

Normal density distributions $\hat{f}_k(x) \sim N(\hat{\mu}_k, \hat{\sigma}_k)$ for 13 predictors are ploted below. For the feature Flavonoids three classes seem to be the most separated.

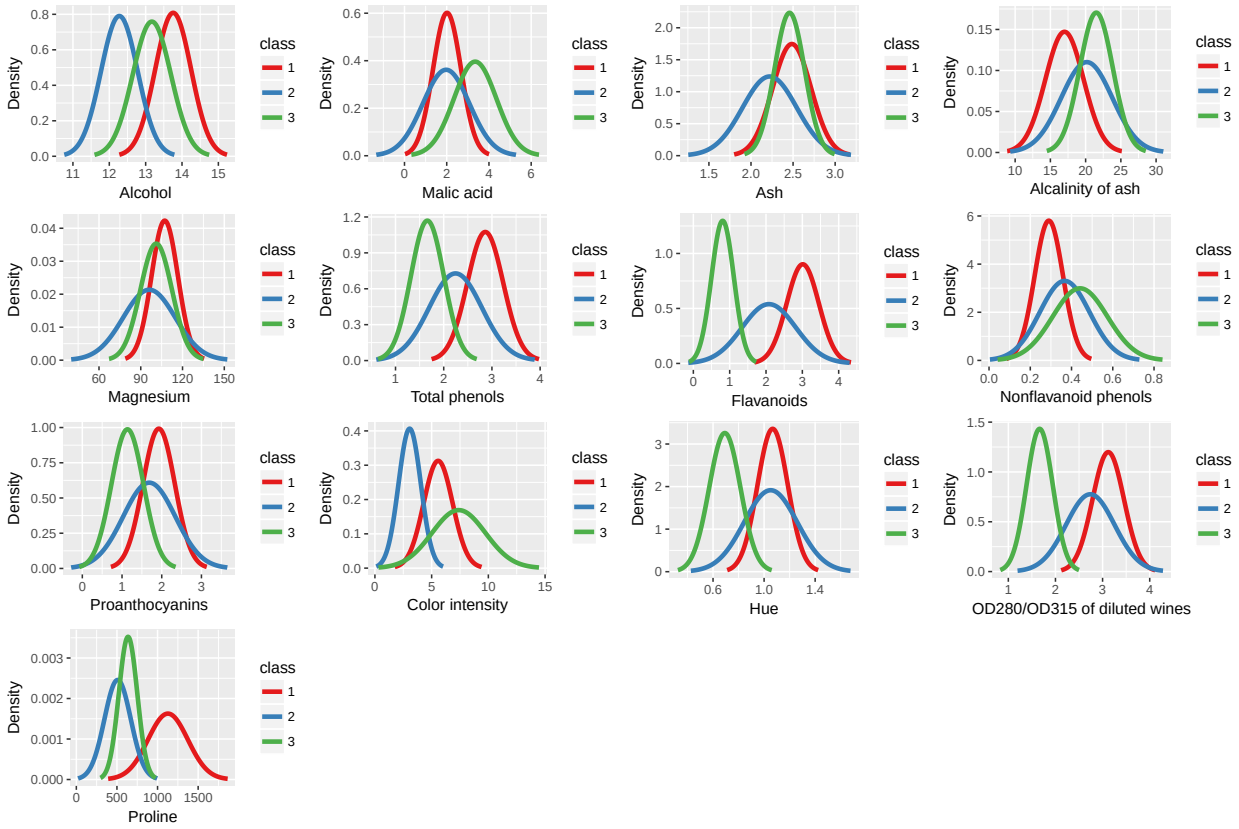

Figure 1: An example with three classes. One-dimensional Gaussian density functions are shown for all 13 features

We peek one feature `Alcohol` to build LDA using `MASS` library.

```
library(MASS)
lda.fit = lda(Label ~ Alcohol, data = wineTrain)
lda.fit
```

```
## Call:
## lda(Label ~ Alcohol, data = wineTrain)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3307692 0.4000000 0.2692308
##
## Group means:
##    Alcohol
## 1 13.75651
## 2 12.27615
## 3 13.17029
##
## Coefficients of linear discriminants:
##              LD1
## Alcohol 1.973001
```

The `predict()` function returns a list with three elements. The first element `class` contains LDA's predictions about wine labels.

```
lda.pred = predict(lda.fit, newdata = wineTest)
names(lda.pred)
```

```
## [1] "class"     "posterior" "x"
```

```
lda.class = lda.pred$class
```

A *confusion matrix* compares the LDA predictions to the true classes for test set observations.

```
table(lda.class, wineTest$Label)
```

```
##
## lda.class  1  2  3
##         1 13  1  5
##         2  0 16  5
##         3  2  2  3
```

The last commands computs the test set error rate. This level is high. To improve model we have to use more than one predictor.

```
mean(lda.class != wineTest$Label)
```

```
## [1] 0.3191489
```

Linear Discriminant Analysis for $p > 1$

Now we extend LDA classifier to the case of multile predictors. To do this we will assume that $X = (X_1, X_2, ..., X_p)$ is drawn from a multivariate Gaussian distribution with a class-specific mean vector μ and a common covariance matrix Σ. We write $X \sim N(\mu, \Sigma)$. The multivariate Gaussian density is defined

$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)) \tag{6}$$

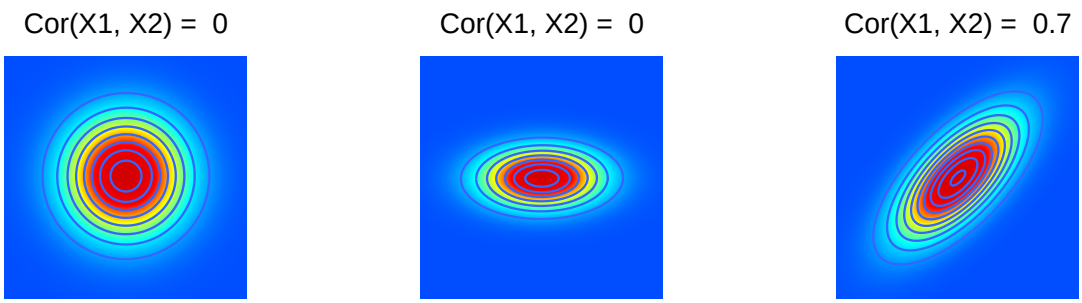

Figure 2: Bivariate Gaussian density functions. Red and blue colors correspond to max and min values of density functions. Left: Uncorrelated random variables with equal variances. Middle: Uncorrelated random variables with different variances. Right: Correlated random variables with different variances.

where $|\Sigma| = det(\Sigma)$ and $\Sigma = Cov(X)$ is the $p \times p$ matrix.

$$\Sigma = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) & \cdots & Cov(X_1, X_p) \\ Cov(X_2, X_1) & Var(X_2) & \cdots & Cov(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & Cov(X_p, X_2) & \cdots & Var(X_p) \end{pmatrix} \tag{7}$$

The multivariate Gaussian distribution assumes that each individual predictor follows a one-dimensional normal distribution with some correlation between each pair of predictors. Three examples of multivariate Gaussian distributions with $p = 2$ are shown in Fig2.

The LDA classifier assumes that the observations in the kth class are drawn from a multivariate Gaussian distribution $N(\mu_k, \Sigma)$ with a class-specific mean vector μ_k and a common to all K classes covariance matrix Σ.

Futher plugging the density function in Bayes' Theorem and preforming some transformations we obtain linear discriminat functions for many predictors

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + ln(\pi_k) \tag{8}$$

Classifer assigns an observation $X = x$ to the class for wich δ_k is largest. Decision boundaries are defined by $\delta_k(x) = \delta_l(x)$ for $k \neq l$:

$$x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + ln(\pi_k) = x^T \Sigma^{-1} \mu_l - \frac{1}{2} \mu_l^T \Sigma^{-1} \mu_l + ln(\pi_l) \tag{9}$$

Assuming that

$$a_0 = ln(\frac{\pi_k}{\pi_l}) - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1} (\mu_k - \mu_l) \tag{10}$$

$$(a_1, a_2, ..., a_p)^T = \Sigma^{-1}(\mu_k - \mu_l), \tag{11}$$

classification boundary can be written in the following form

$$a_0 + \sum_{i=1}^{p} a_i x_i = 0 \tag{12}$$

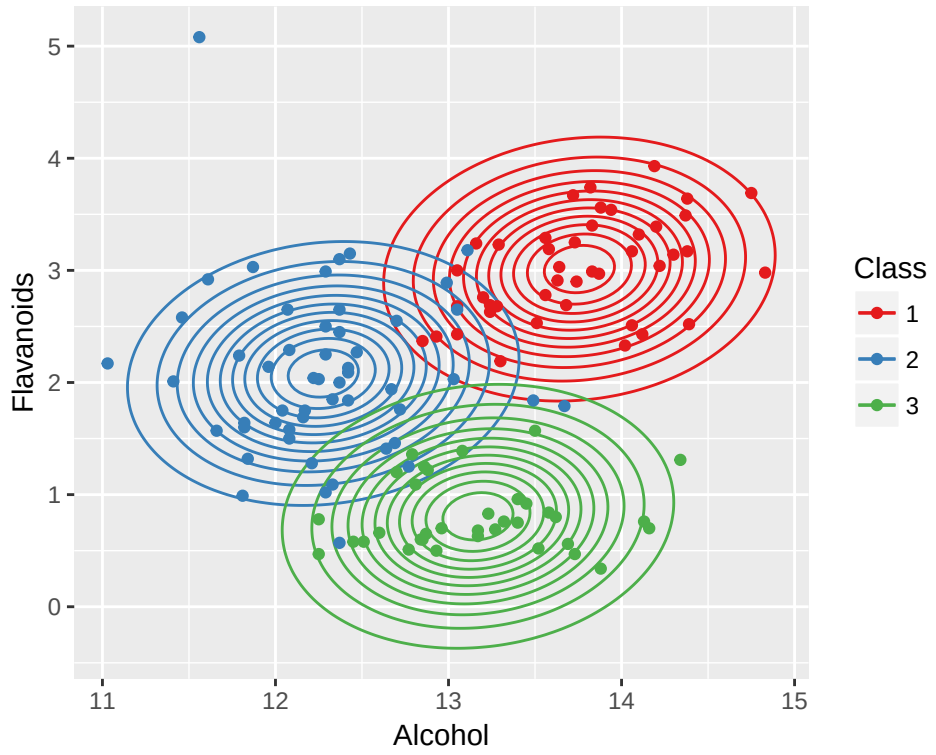

Figure 3: An example with three classes for two variables with a class-specific mean vector and a common covariance matrix. Ellipses are contour-plots of class-specific Gaussian density functions.

Now we perform LDA on `wine` data.

```
# Fit a linear discriminant model
lda.fit2 = lda(Label ~ Alcohol + Flavanoids, data = wineTrain)
lda.fit2
```

```
## Call:
## lda(Label ~ Alcohol + Flavanoids, data = wineTrain)
##
## Prior probabilities of groups:
##         1         2         3
## 0.3307692 0.4000000 0.2692308
##
## Group means:
##    Alcohol Flavanoids
## 1 13.75651  3.0116279
## 2 12.27615  2.0805769
## 3 13.17029  0.8065714
##
## Coefficients of linear discriminants:
##                    LD1        LD2
## Alcohol    -0.7504188  1.8337514
## Flavanoids -1.5994476 -0.8333091
##
## Proportion of trace:
##    LD1    LD2
```

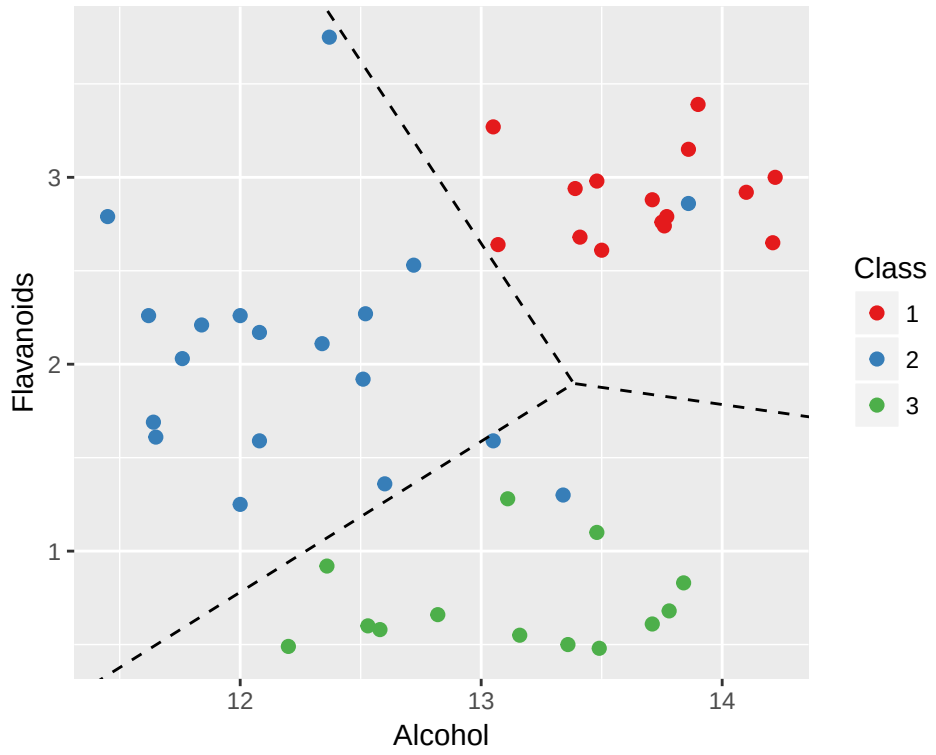

Figure 4: Decision boundaries on test data set for two wine features.

```
## 0.6516 0.3484
```

Evaluate model performance with test data set.

```
lda.pred = predict(lda.fit2, newdata = wineTest)
lda.class = lda.pred$class
```

A confusion matrix and test error are

```
table(lda.class, wineTest$Label)
```

```
##
## lda.class  1  2  3
##         1 15  1  0
##         2  0 16  0
##         3  0  2 13
```

```
mean(lda.class != wineTest$Label)
```

```
## [1] 0.06382979
```

The test error rate drop down compared to LDA with only one predictor variable.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, **2013**.