Gaussian Discriminant Analysis

Chih-Hui Wang(Jason)

May 06, 2015; Revised: March 03, 2016

Gaussian Discriminant Analysis

The idea of Gaussian discriminant analysis is that we use the Bayes theorem to compute the posterior probability $P(Y = c|X = x)$. We assume the underlying distribution of X for each class is a normal distribution. According to Bayes theorem, we can calculate the posterior probability as follow:

$$P(Y = c|X = x) = \frac{P(X = x|Y = c)P(Y = c)}{\sum_{i=1}^{C} P(X = x|Y = i)P(Y = i)}$$

- Prior probability: $p(Y = c) = \pi_c$
- Posterior probability: $P(Y = c|X = x)$
- Density function: $P(X = x|Y = c)$

We first consider that there are only one independent variable X. The assumption of normal distribution means that

$$P(X = x|Y = c) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x - \mu_c)^2}{2\sigma_c^2}\right)$$

For each class c, we need to estimate the mean, μ_c, variance, σ_c^2, and prior probability, $\pi_c = P(Y = c)$. Given x, we will choose the class c that maximize the posterior probability $P(Y = c|X = x)$ or equivalently $P(X = x|Y = c)P(Y = c)$ because the denominator of posterior probabilities are the same for all classes. We define $P(X = x|Y = c)$ as our **discrimiant function** and since log is a monotonically increasing function, it's equivalent to maximize $\delta_c(X) = \log P(X = x|Y = c)P(Y = c)$.
We can recover the posterior probability from discrimiant function, for 2-class case,

$$\begin{aligned} P(Y = C|X = x) &= \frac{P(X = x|Y = C)\pi_C}{P(X = x|Y = C)\pi_C + P(X = x|Y = D)\pi_D} \\ &= \frac{e^{\delta_C(x)}}{e^{\delta_C(x)} + e^{\delta_D(x)}} = \frac{1}{1 + e^{\delta_D(x) - \delta_C(x)}} \\ &= s(\delta_C(x) - \delta_D(x)) \end{aligned}$$

where $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ is the **logistic function** or **sigmoid function**.

1. Linear Discriminant Analysis (LDA)

Fundamental assumption: all the normal distribution have the same variance σ^2. We suppose that there are only two classes C and D. Our **desicion rule** will be

$$r^*(x) = \begin{cases} C, \text{if } \delta_C(x) - \delta_D(x) > 0 \\ D, \text{otherwise.} \end{cases}$$

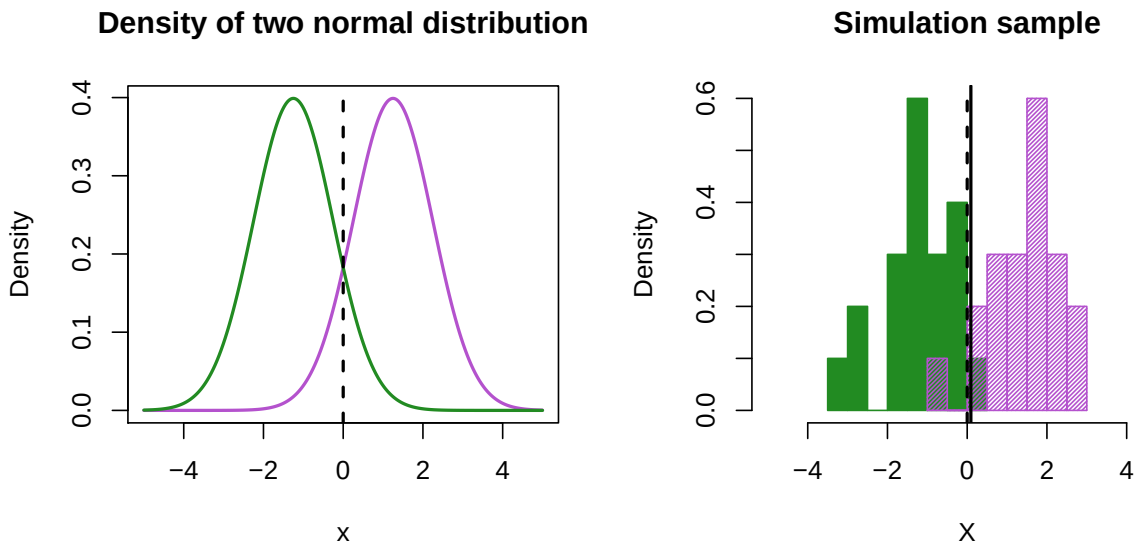


Figure 1: Special Case of LDA

The **prediction function** becomes

$$\begin{aligned}\delta_C(x) - \delta_D(x) &= -\frac{(x-\mu_C)^2}{2\sigma^2} - \log\sigma + \log\pi_C + \frac{(x-\mu_D)^2}{2\sigma^2} + \log\sigma - \log\pi_D \\ &= \frac{\mu_C - \mu_D}{\sigma^2}x - \frac{\mu_C^2 - \mu_D^2}{2\sigma^2} + \log\pi_C - \log\pi_D\end{aligned}$$

In 2-class case, the decision boundary can be written as $wx + \alpha = 0$ (a linear classifier). For multi-class case, we just choose class C that maximizes the $\delta_C = \frac{\mu_c}{\sigma^2}x - \frac{\mu_C^2}{2\sigma^2} + \log\pi_C$.

A special cass in 2-class: $\pi_C = \pi_D$. The decision boundary is

$$\delta_C(x) - \delta_D(x) = \frac{\mu_C - \mu_D}{\sigma^2}x - \frac{\mu_C^2 - \mu_D^2}{2\sigma^2} = 0$$

$$\Rightarrow (\mu_C - \mu_D)x - (\mu_C - \mu_D)\frac{\mu_C + \mu_D}{2} = 0$$

As shown in Figure 1, we simulated data from two normal distribution with mean equal to 1.25 and -1.25 and variance equal to 1. The dashed line is the real decision boundary and the solid line in the left figure is the decision boundary of LDA.

2. Quadratic Discriminant Analysis (QDA)

If we don't have the assumption of the same variance across classes, it becomes QDA. As you can see in the discriminant function,

$$\delta_C = -\frac{(x-\mu_C)^2}{2\sigma_C^2} - \log\sigma_C + \log\pi_C$$

which is quadratic in x. For multi-class case, the decision rule is the same as the one in LDA. We choose class C that maximizes the discriminant function.

3. Extension to Multivariate Case and Maximum Likelihood Estimation

Recall that $X \sim N(\mu, \Sigma)$ and the density function is

$$p(x) = \frac{1}{\sqrt{2\pi}^d \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

For multivariate case, the idea is the same as before. The LDA assume that the covariance matrix is the same for all classes while, in QDA, the covariance matrix may be different for each class. We can plug in the density of multivariate normal distribution and find the discriminant function $\delta_C(X) = \log P(X = x|Y = C)P(Y = C)$ for both cases.

Now, we turn to the question that how should we estimate the parameters for each parameter in normal distribution as well as the prior probability? We use maximum likelihood estimate (MLE)! We are not going to prove the MLE for mean and covariance matrix of each class. Instead, we will just summarize them below:

LDA

$$\delta_C(X) = \mu_C^T \Sigma^{-1} x - \frac{1}{2}\mu_C^T \Sigma^{-1} \mu_C + \log \pi_C$$

- $\hat{\mu}_C = \frac{1}{n_C} \sum_{i \in C} x_i$
- $\hat{\Sigma} = \frac{1}{n} \sum_C \sum_{i \in C} (x_i - \mu_C)(x_i - \mu_C)^T$
- $\hat{\pi_C} = \frac{n_C}{n}$

QDA

$$\delta_C(X) = -\frac{1}{2}(x-\mu_C)^T \Sigma_C^{-1}(x-\mu_C) - \frac{1}{2}\log|\Sigma_C| + \log \pi_C$$

- $\hat{\mu}_C = \frac{1}{n_C} \sum_{i \in C} x_i$
- $\hat{\Sigma}_C = \frac{1}{n_C} \sum_{i \in C} (x_i - \mu_C)(x_i - \mu_C)^T$
- $\hat{\pi_C} = \frac{n_C}{n}$

where n_c is the number of observation in class C and n is the total number of observation. For multi-class case, we will choose the class with the largest discriminant function.

4. Textbook Example

```
library(ISLR)
library(car)
library(MASS)

Default$default <- as.numeric(as.character(
recode(Default$default, "'Yes'='1'; 'No'='0'")))
```

```
# LDA
LDA <- lda(default ~ balance + student, data=Default)

# Fitted value
Prediction <- predict(LDA, Default)$class
t <- table(Predict=Prediction, True=Default$default)

# addmargins: comput all margin of the table
# ftable: make the table format nicer
ftable(addmargins(t))
```

```
        True     0     1   Sum
Predict
0              9644   252  9896
1                23    81   104
Sum            9667   333 10000
```

```
# QDA
QDA <- qda(default ~ balance + student, data=Default)

Prediction <- predict(QDA, Default)$class
t2 <- table(Predict=Prediction, True=Default$default)
ftable(addmargins(t2))
```

```
        True     0     1   Sum
Predict
0              9637   244  9881
1                30    89   119
Sum            9667   333 10000
```

```
threshold <- 0.2
Prediction_new <- (predict(LDA, Default)$posterior[, 2] > threshold)*1

t_new <- table(Predict=Prediction_new, True=Default$default)
ftable(addmargins(t_new))
```

```
        True     0     1   Sum
Predict
0              9432   138  9570
1               235   195   430
Sum            9667   333 10000
```

Performance Evaluation: ROC Curve

	0 (True)	1 (True)
0 (Predict)	True Negative (TN)	False Negative (FN)
1 (Predict)	False Positive (FP)	True Positive (TP)

There are several measure that can help us to determine the performance of our model or classifier.

1. Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
2. Specificity: $\frac{TN}{TN+FP}$
3. True Positive Rate (Sensitivity, Recall): $\frac{TP}{TP+FN}$
4. False Positive Rate (Type I error, 1 - Specificity): $\frac{FP}{TN+FP}$

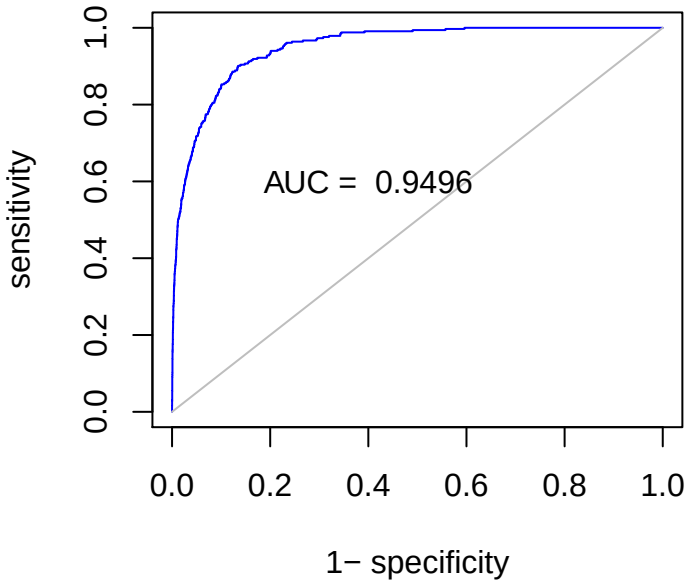
5. Positive Predicted Value (Precision, 1 - False Discovery Rate): $\frac{TP}{TP+FP}$

6. Negative Predicted Value: $\frac{TN}{TN+FN}$

```
library(caret)
# Confusion Matrix
confusionMatrix(t, positive="1")
```

```
Confusion Matrix and Statistics

       True
Predict    0    1
      0 9644  252
      1   23   81

               Accuracy : 0.9725
                 95% CI : (0.9691, 0.9756)
    No Information Rate : 0.9667
    P-Value [Acc > NIR] : 0.0004973

                  Kappa : 0.3606
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.2432
            Specificity : 0.9976
         Pos Pred Value : 0.7788
         Neg Pred Value : 0.9745
             Prevalence : 0.0333
         Detection Rate : 0.0081
   Detection Prevalence : 0.0104
      Balanced Accuracy : 0.6204

       'Positive' Class : 1
```

```
library(AUC)

# ROC and AUC of the classifier
plot(roc(predict(LDA, Default)$posterior[, 2], as.factor(Default$default)),
     col="blue")
auc_value <- auc(roc(predict(LDA, Default)$posterior[, 2],
                     as.factor(Default$default)))
text(0.4, 0.6, paste("AUC = ", round(auc_value, 4)))
```


sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.9496
0.0
0.2
0.4
0.6
0.8
1.0
1− specificity