Linear discriminant analysis
DS351

Probability-based classifier

- Classification problem: $y =$ label, $x =$ features

Probability-based classifier

- Classification problem: $y =$ label, $x =$ features
- Given x. We predict $\hat{y}$ if $P(\hat{y}|x)$ gives the largest probability among all possible values of y. For example,

j	0	1	2	3	4	5	6	7	8	9
$P(y = j\|x)$	0	0	0.5	0.2	0	0.2	0	0	0.1	0

In this case, $\hat{y} = 2$.

Probability-based classifier

- Classification problem: $y = \text{label}$, $x = \text{features}$
- Given x. We predict $\hat{y}$ if $P(\hat{y}|x)$ gives the largest probability among all possible values of y. For example,

j	0	1	2	3	4	5	6	7	8	9
$P(y = j\|x)$	0	0	0.5	0.2	0	0.2	0	0	0.1	0

  In this case, $\hat{y} = 2$.
- However, if x does not appear in the training set (always happens when x is continuous), we cannot compute $P(y|x)$ directly!

Linear discriminant analysis

In Linear discriminant analysis (LDA), we model the the **distribution** in each group and use the **Bayes' rule**.

Bayes' Rule

$$P(y = j|x) = \frac{P(x|y = j)P(y = j)}{P(x)}$$

To compute $P(y = j|x)$, we need to know the following:

- $P(x|y = j) = f_j(x)$, a probability density function.

Linear discriminant analysis

In Linear discriminant analysis (LDA), we model the the **distribution** in each group and use the **Bayes' rule**.

Bayes' Rule

$$P(y = j|x) = \frac{P(x|y = j)P(y = j)}{P(x)}$$

To compute $P(y = j|x)$, we need to know the following:

- $P(x|y = j) = f_j(x)$, a probability density function.
- $P(y = j) = \frac{\text{\# of group } j\text{'s observations}}{\text{\# of all observations}} = \frac{n_j}{n}$.

Linear discriminant analysis

In Linear discriminant analysis (LDA), we model the the **distribution** in each group and use the **Bayes' rule**.

Bayes' Rule

$$P(y = j|x) = \frac{P(x|y = j)P(y = j)}{P(x)}$$

To compute $P(y = j|x)$, we need to know the following:

- $P(x|y = j) = f_j(x)$, a probability density function.
- $P(y = j) = \frac{\#\text{ of group } j\text{'s observations}}{\#\text{ of all observations}} = \frac{n_j}{n}$.
- $P(x)$ you don't need to compute this; more later.

Linear discriminant analysis

Suppose that there are three classes i.e. $y \in \{1, 2, 3\}$.

Given data with features x, we predict $y = j$ if $P(y = j|x)$ is the largest among

$$P(y = 1|x), \quad P(y = 2|x), \quad P(y = 3|x)$$

Linear discriminant analysis

Suppose that there are three classes i.e. $y \in \{1, 2, 3\}$.

Given data with features x, we predict $y = j$ if $P(y = j|x)$ is the largest among

$$P(y = 1|x), \quad P(y = 2|x), \quad P(y = 3|x)$$

Using Bayes' rule, we find the largest among

$$\frac{P(x|y{=}1)P(y{=}1)}{P(x)}, \quad \frac{P(x|y{=}2)P(y{=}2)}{P(x)}, \quad \frac{P(x|y{=}3)P(y{=}3)}{P(x)}$$

Good news! we don't need to compute $P(x)$ if we only care about predictions.

Linear discriminant analysis

Classification

Given data with features x, we predict $\hat{y} = j$ if $P(x|y = j)P(y = j)$ is the largest among

$$P(x|y = 1)P(y = 1), \quad P(x|y = 2)P(y = 2), \quad P(x|y = 3)P(y = 3)$$

Example

Wine classification

- A bottle of wine with no label, only know chemical features.
- What kind of wine is it from 1, 2 and 3?

Wine dataset

Training set of 130 wine bottles.

- Class 1: 43 bottles
- Class 2: 51 bottles
- Class 3: 36 bottles
- 13 Features: *Alcohol*, *Malic Acid*,... but we will use *Alcohol* only.

...and a test set of 48 bottles.

LDA on wine dataset

$y \in \{1, 2, 3\}$, $x = Alcohol$,
Given x, we make a prediction $y = j$ if $P(x|y = j)P(y = j)$ is the largest among

$$P(x|y = 1)P(y = 1), \quad P(x|y = 2)P(y = 2), \quad P(x|y = 3)P(y = 3)$$

LDA on wine dataset

$y \in \{1, 2, 3\}$, $x = Alcohol$,
Given x, we make a prediction $y = j$ if $P(x|y = j)P(y = j)$ is the largest among

$$P(x|y = 1)P(y = 1), \quad P(x|y = 2)P(y = 2), \quad P(x|y = 3)P(y = 3)$$

$$P(y = 1) = \frac{\text{\# of type 1 bottles}}{\text{\# of all bottles}} = \frac{n_1}{n}$$
$$P(y = 2) = \frac{\text{\# of type 2 bottles}}{\text{\# of all bottles}} = \frac{n_2}{n}$$
$$P(y = 3) = \frac{\text{\# of type 3 bottles}}{\text{\# of all bottles}} = \frac{n_3}{n}$$

LDA on wine dataset

$y \in \{1, 2, 3\}$, $x = Alcohol$,
Given x, we make a prediction $y = j$ if $P(x|y = j)P(y = j)$ is the largest among

$$P(x|y = 1)P(y = 1), \quad P(x|y = 2)P(y = 2), \quad P(x|y = 3)P(y = 3)$$

We infer these values from the training set.

$$P(y = 1) = \frac{43}{130} = 0.33$$
$$P(y = 2) = \frac{51}{130} = 0.39$$
$$P(y = 3) = \frac{36}{130} = 0.28$$

LDA on wine dataset

$y \in \{1, 2, 3\}$, $x = Alcohol$,
Given x, we make a prediction $y = j$ if $P(x|y = j)P(y = j)$ is the largest among

$$P(x|y=1)P(y=1), \quad P(x|y=2)P(y=2), \quad P(x|y=3)P(y=3)$$

How can we compute $P(x|y = 1)$ etc.?

We assume that x in each class is **Gaussian** distributed.

Gaussian distribution

$j = 1, 2, 3$. Suppose that the data in Class j is

$$x_{ij} : x_{1j}, x_{2j}, \ldots, x_{n_j j}.$$

The *density function* of Class j is

$$f_{\hat{\mu}_j, \hat{\sigma}_j^2}(x) = \frac{1}{\sqrt{2\pi\hat{\sigma}_j^2}} e^{-(x-\hat{\mu}_j)^2/2\hat{\sigma}_j^2},$$

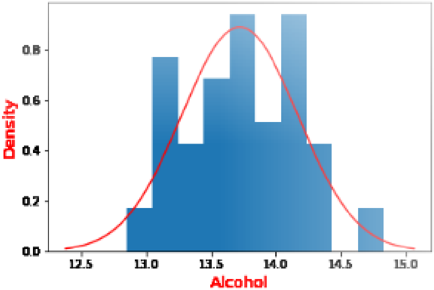
where

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_{ij}$$

$$\hat{\sigma}_j^2 = \frac{1}{n_j - 1} \sum_{i=1}^{n_j} (x_{ij} - \hat{\mu}_j)^2$$

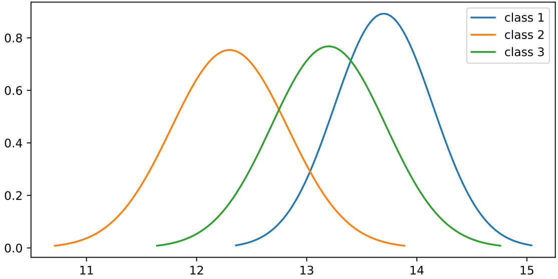
Distribution of class 1

Histogram of class 1:



$$\begin{array}{lll} \text{Class 1:} & \text{Mean } \hat{\mu}_1 = 13.72, & \text{Variance } \hat{\sigma}_1^2 = 0.20 \\ \text{Class 2:} & \text{Mean } \hat{\mu}_2 = 12.3, & \text{Variance } \hat{\sigma}_2^2 = 0.28 \\ \text{Class 3:} & \text{Mean } \hat{\mu}_3 = 13.2, & \text{Variance } \hat{\sigma}_3^2 = 0.27 \end{array}$$

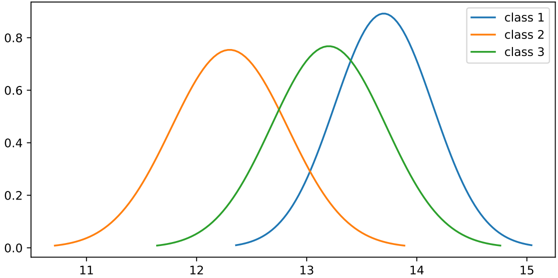
Predictions



$$P(x|y=1) = f_{13.7,0.20}(x), \qquad P(y=1) = 0.33$$
$$P(x|y=2) = f_{12.3,0.28}(x), \qquad P(y=2) = 0.39$$
$$P(x|y=3) = f_{13.2,0.27}(x), \qquad P(y=3) = 0.28$$

Pick $j \in \{1, 2, 3\}$ with the largest $P(x|y=j)P(y=j)$.

Example



Example: $x = 12$

$$P(12|y=1) = 0.0006, \qquad P(y=1) = 0.33$$
$$P(12|y=2) = 0.6420, \qquad P(y=2) = 0.39$$
$$P(12|y=3) = 0.0533, \qquad P(y=3) = 0.28$$

Model evaluation

Imbalanced data

Example:

model 1:

y_i	0	0	0	0	0	0	0	0	0	1
$\hat{y}_i$	0	0	0	0	0	0	0	0	0	0

vs

model 2:

y_i	0	0	0	0	0	0	0	0	0	1
$\hat{y}_i$	0	0	0	0	0	0	0	0	1	1

both have 90% accuracy, but which model would you prefer?

Prediction errors

A model can make two types of error:

		Label	
		1	**0**
Prediction	**1**	correct	False positive
	0	False negative	correct

- Type 1: **False Positive** (0 classified as 1)
  Ex: False alarm
- Type 2: **False Negative** (1 classified as 0)
  Ex: Dangerous items passing a security check

Confusion matrix

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	0	0	0	1	0	1	1	0	1	1

		Label	
		1	**0**
Prediction	**1**	TP	FP
	0	FN	TN

- **True Positive**: an instance correctly classified as 1
- **True Negative**: an instance correctly classified as 0

True Positive Rate

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	0	0	0	1	0	1	1	0	1	1

		Label	
		1	0
Prediction	1	**TP**	FP
	0	FN	TN

True Positive Rate (Recall or Sensitivity):

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

i.e. proportion of positives that are correctly classified as positives

True Negative Rate

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	0	0	0	1	0	1	1	0	1	1

Label

		1	0
Prediction	1	TP	FP
	0	FN	**TN**

True Negative Rate (Specificity):

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

i.e. proportion of negatives that are correctly classified as negatives

Precision

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	0	0	0	1	0	1	1	0	1	1

Label

		1	0
Prediction	1	**TP**	FP
	0	FN	TN

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

i.e. proportion of positive predictions that are actually positive

Accuracy

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	0	0	0	1	0	1	1	0	1	1

Label

		1	0
Prediction	1	**TP**	FP
	0	FN	**TN**

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

i.e. proportion of all instances that are predicted correctly

Example 1

y_i	0	0	0	0	0	0	0	0	0	1
$\hat{y}_i$	0	0	0	0	0	0	0	0	0	0

$$
\begin{aligned}
\text{Recall (TPR)} &= \frac{\text{TP}}{\text{TP} + \text{FN}} = \\
\text{TNR} &= \frac{\text{TN}}{\text{TN} + \text{FP}} = \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} =
\end{aligned}
$$

Example 2

y_i	0	0	0	0	0	0	0	0	0	1
$\hat{y}_i$	0	0	0	0	0	0	0	0	1	1

$$
\begin{aligned}
\text{Recall (TPR)} &= \frac{\text{TP}}{\text{TP} + \text{FN}} = \\
\text{TNR} &= \frac{\text{TN}}{\text{TN} + \text{FP}} = \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} =
\end{aligned}
$$

What to use?

- Use **Recall** if we want the model to "see" all the positive instances.
  Examples: security check, tests for deadly diseases

What to use?

- Use **Recall** if we want the model to "see" all the positive instances.
  Examples: security check, tests for deadly diseases
- Use **Precision** if we only care about correct positive predictions.
  Examples: Youtube video recommendation, hiring workers

But in some situation, we might want to find a balance between these two scores.

- want a way to combine both Precision and Recall

Precision & Recall

How about average of the two?

y_i	0	0	0	0	0	1	1	1	1	1
$\hat{y}_i$	1	1	1	1	1	1	1	1	1	1

$$\frac{\text{Recall} + \text{Precision}}{2} = \frac{1 + 0.5}{2} = 0.75$$

...probably too high for such a simple model.

F-score

F-score or **F1-score** is used to find a model that has a nice balance between Precision and Recall

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Fact: The value F_1 is always between Precision and Recall