머신러닝응용 제05강

Discriminant Analysis

첨단공학부 김동하교수

제05강 Discriminant Analysis

1	판별분석의 개념에 대해 학습한다.
2	선형판별분석(LDA)에 대해 학습한다.
3	이차판별분석(QDA)에 대해 학습한다.

핵심 단어

- 판별분석
- 선형판별분석
- 이차판별분석

05강. Discriminant Analysis

01. 판별분석

1) 판별분석이란?

◆두 개 이상의 모집단에서 추출된 데이터를 이용.

◆해당 샘플이 어느 모집단으로부터 추출된 것인지를 결정.

◆즉, 주어진 입력값에 대해서 이 데이터가 어떤 모집단에 속하는지를 판별하는 방법론

- 분류 문제를 해결하는 방법론 중 하나

1) 판별분석이란?

◆용어 정리

- 판별변수 (Discriminant Variable)
  - 어떤 집단에 속하는지 판별하기 위해 사용되는 독립변수
- 판별함수 (Discriminant Function)
  - 판별변수들의 선형/비선형 조합
  - 종속변수의 집단을 정확하게 분류할 수 있도록 학습

1) 판별분석이란?

◆용어 정리

- 판별점수 (Discriminant Score)
  - 개체가 어떤 집단에 속하는지 판별하기 위해 판별함수에 대입하여 얻은 값.

1) 판별분석이란?

◆독립변수값의 모집단 정보를 판별하는 판별함수의 형태에 따라서 다음의 두 가지 방법론으로 나뉨.

- 선형판별분석 (Linear Discriminant Analysis)
  - 선형 판별식을 사용
- 이차판별분석 (Quadratic Discriminant Analysis)
  - 이차 판별식을 사용

2) 판별분석의 모형

- X: 설명변수

- Y: 설명변수값이 속한 모집단의 라벨
  - 0, 1 두 개의 값만을 갖는다고 가정하자.
  - 판별분석은 3개 이상의 범주를 갖는 라벨도 분류가 가능하다.

2) 판별분석의 모형

◆판별분석모형은 다음의 분포를 가정

$$Y \sim Ber(\pi)$$

$$X|Y = y \sim N\left(\mu_y, \Sigma_y\right), \qquad \mathbf{y} \in \{\mathbf{0}, \mathbf{1}\}$$

◆설명변수 X의 주변 분포는 다음과 같이 계산할 수 있음.

$$X \sim (1-\pi)N(\mu_0, \Sigma_0) + \pi N(\mu_1, \Sigma_1)$$

- 혼합 정규 분포 (Mixture of Gaussian)

2) 판별분석의 모형

◆입력값 X가 주어졌을 때 종속변수 Y의 분포는 베이즈 정리에 의해 계산되어짐.

$$\mathrm{P}(Y=1|X=x)=\frac{P(Y=1)P(X=x|Y=y)}{P(X=x)}$$

$$=\frac{\pi_1\phi(x;\mu_1,\Sigma_1)}{\sum_{y=0}^{1}\pi_y\phi(x;\mu_y,\Sigma_y)}$$

- $\pi_0=1-\pi, \pi_1=\pi$
- $\phi(x;\mu_y,\Sigma_y)$: 정규분포 확률 밀도 함수

3) 판별분석을 이용한 예측

◆ 입력값 X가 주어졌을 때 종속변수 값 예측하기

$$\hat{y} = 1 \ \ if \ \mathrm{P}(Y = 1|X = x) > 0.5$$
$$\hat{y} = 0 \ \ o.w.$$

◆ 예측 기준을 다시 쓰면 아래와 같음.

$$\hat{y} = 1 \ \ if \ P(Y = 1|X = x) > P(Y = 0|X = x)$$
$$\hat{y} = 0 \ \ o.w.$$

3) 판별분석을 이용한 예측

◆앞선 수식을 대입하면 다음과 같은 결과를 얻을 수 있음.

$$\hat{y} = 1 \ \ if$$

$$-\frac{1}{2}x^T\left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)x + x^T\left(\Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0\right)$$

$$+\log\left(\frac{\pi_1 det(\Sigma_1)^{-1/2}}{\pi_0 det(\Sigma_0)^{-1/2}}\right) - \frac{1}{2}\left(\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_0^T\Sigma_0^{-1}\mu_0\right) > 0$$

$$\hat{y} = 0 \ \ \mathrm{o.w.}$$

4) 이차판별분석

- 판별식은 다음과 같이 입력값에 대한 이차식으로 정리됨.

$$\hat{y} = 1 \ \ if \ \ x^T A x + x^T b + c > 0$$
$$\hat{y} = 0 \ \ \mathrm{o.w.}$$

- 입력값의 이차식을 통해 종속변수를 예측
  - 이차판별분석 (QDA)

5) 선형판별분석

- 공분산 행렬 Σ_0, Σ_1이 같다는 가정을 하면?
  - $\Sigma_0 = \Sigma_1 = \Sigma$

- 판별식은 다음과 같이 정리됨.

$\hat{y} = 1 \ \ if$

$$x^T\Sigma^{-1}(\mu_1 - \mu_0) + \log\left(\frac{\pi_1}{\pi_0}\right) - \frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\right) > 0$$

$\hat{y} = 0 \ \ \text{o.w.}$

5) 선형판별분석

◆판별식은 다음과 같이 입력값에 대한 일차식으로 정리됨.

$$\hat{y} = 1 \ \ if \ \ x^T b + c > 0$$
$$\hat{y} = 0 \ \ \mathrm{o.w.}$$

◆입력값의 선형식을 통해 종속변수를 예측

- 선형판별분석 (LDA)

6) 판별분석 모수의 추정

◆판별분석모형의 모수

- 선형판별분석 (LDA)
  - $\pi,\ \mu_0,\ \mu_1,\ \Sigma$
- 이차판별분석 (QDA)
  - $\pi,\ \mu_0,\ \mu_1,\ \Sigma_0,\ \Sigma_1$

6) 판별분석 모수의 추정

◆LDA, QDA 모두 학습 데이터를 이용한 음의 로그우도함수를 최소화하는 모수를 추정한다.

- 학습 데이터: $(x_1, y_1), \cdots, (x_n, y_n)$
- 음의 로그우도 함수

$$-\sum_{i=1}^{n} \log P(Y = y_i, X = x_i)$$

7) LDA vs Logistic model

- 선형판별분석에서 설명변수가 주어졌을 때 종속변수의 분포는 다음과 같이 나타내어진다.

$$P(Y=1|X=x)=\frac{exp(b^T x+c)}{1+exp(b^T x+c)}$$

  - 로지스틱모형과 같은 형태!

- 어떤 차이점이 있을까?

7) LDA vs Logistic model

◆로지스틱모형

- 로지스틱모형은 종속변수의 조건부분포만을 모형화
- Discriminative model
- $P(Y = y|X = x)$

◆LDA

- LDA는 독립변수, 종속변수의 결합분포를 모형화
- Generative model
- $P(Y = y, X = x)$

7) LDA vs Logistic model

◆선형판별분석은 로지스틱모형보다 더 많은 수의 모수를 가짐.

◆데이터의 실제 분포(특히 설명변수)가 판별분석에서 가정한 분포와 일치한다면 로지스틱 모형보다 더 우수한 성능.

02. Python을 이용한 판별분석 실습

1) 데이터 설명

◆ Penguins 데이터셋

- 남극 펭귄 344마리에 대한 데이터
- species: 펭귄 종류 (총 3가지)
- island: 서식하는 남극섬 종류
- bill_length_mm: culmen length (mm)
- bill_depth_mm: bill_depth (mm)
- flipper_length_mm: 물갈퀴 길이 (mm)
- body_mass_g: 몸무게 (g)
- sex: 성별

1) 데이터 설명

◆분석 목표

- 수치형 설명변수 4개로 펭귄의 종을 판별하는 판별분석 모형을 학습하자.

2) 환경설정

◆필요한 패키지 불러오기

```
import os
import numpy as np
import pandas as pd
import collections
from sklearn.model_selection import train_test_split
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

3) 데이터 불러오기

◆ sns 패키지에 내장되어있는 penguins 데이터를 불러오기.

```
penguins = sns.load_dataset('penguins')
print(penguins.shape)
penguins.head()
```

```
(344, 7)
```

	species	island	bill_length_mm	bill_depth_mm
0	Adelie	Torgersen	39.1	18.7
1	Adelie	Torgersen	39.5	17.4
2	Adelie	Torgersen	40.3	18.0

4) 데이터 전처리

◆수치형 설명변수 4개에 있는 결측치는 각각의 평균값으로 대체한다.

```
penguins['bill_length_mm'].fillna(value=penguins['bill_length_mm'].mean(), inplace=True)
penguins['bill_depth_mm'].fillna(value=penguins['bill_depth_mm'].mean(), inplace=True)
penguins['flipper_length_mm'].fillna(value=penguins['flipper_length_mm'].mean(), inplace=True)
penguins['body_mass_g'].fillna(value=penguins['body_mass_g'].mean(), inplace=True)
```

4) 데이터 전처리

- 종속변수와 독립변수 구분
- 학습데이터와 평가데이터 나누기

```
X = penguins[['bill_length_mm','bill_depth_mm',\
              'flipper_length_mm','body_mass_g']]
y = penguins.iloc[:,0]
```

```
X_train, X_test, y_train, y_test = \
train_test_split(X, y, test_size = 0.3, random_state=123)
```

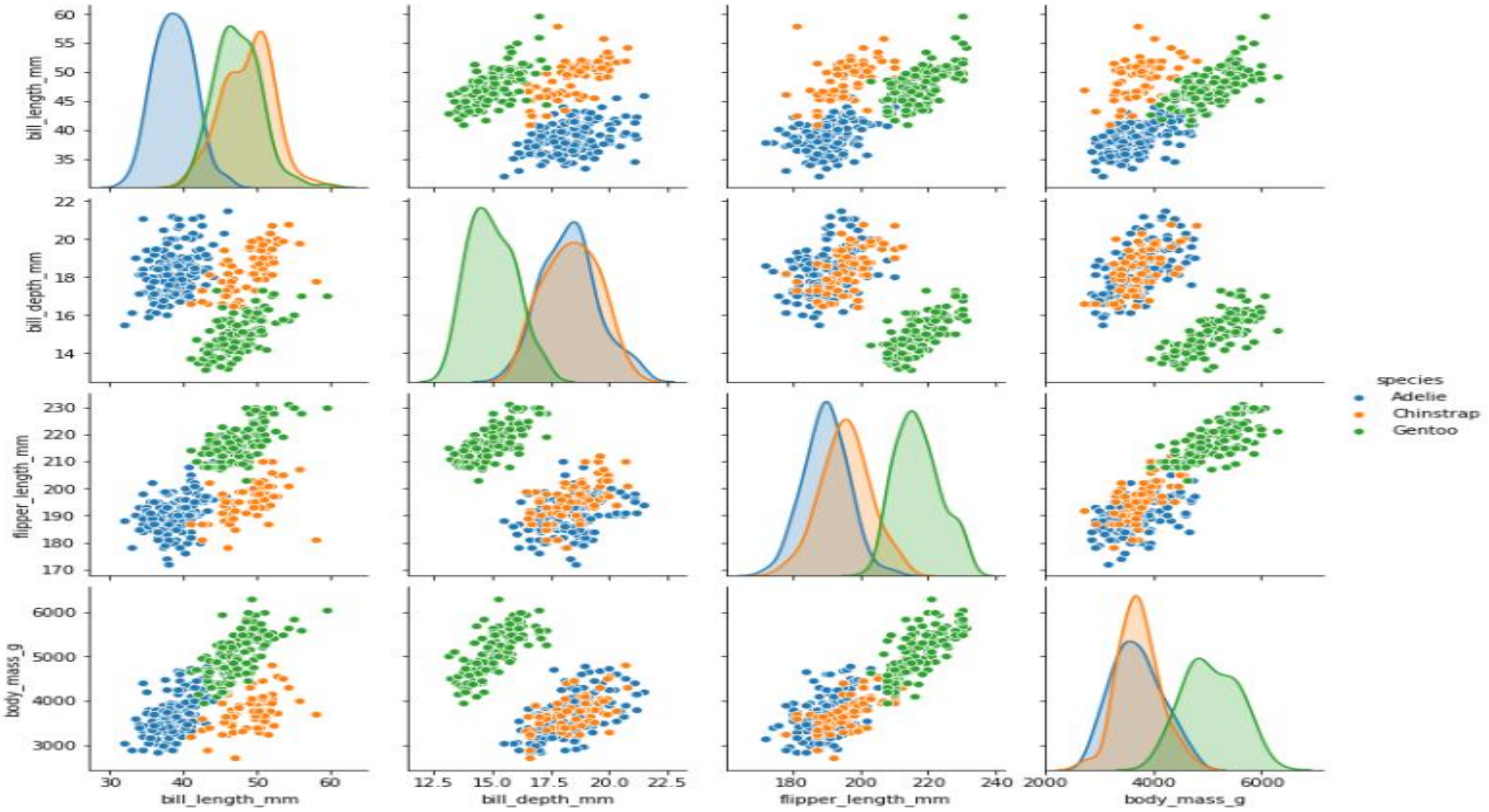
5) 탐색적 자료분석

- 설명변수 자료들의 특성을 살펴본다.
- 히스토그램과 산점도를 통해 시각화를 한다.

```
sns.pairplot(penguins, diag_kind='kde', hue='species')
plt.show()
```

5) 탐색적 자료분석

◆ 히스토그램과 산점도를 통해 시각화를 한다.



6) 선형판별분석

◆ LDA를 활용하여 데이터를 분석한다.

```
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
```

```
LinearDiscriminantAnalysis()
```

6) 선형판별분석

◆ 적합한 LDA 모형에 평가 데이터를 대입하여 분류 정확도와 오차행렬을 계산하자.

```
y_test_pred_lda = lda.predict(X_test)
confusion_matrix(y_test, y_test_pred_lda)
```

```
array([[47,  0,  0],
       [ 0, 21,  0],
       [ 0,  0, 36]])
```

```
lda.score(X_test, y_test)
```

```
1.0
```

7) 이차판별분석

◆ QDA를 활용하여 데이터를 분석한다.

```
qda = QuadraticDiscriminantAnalysis()
qda.fit(X_train, y_train)
```

```
QuadraticDiscriminantAnalysis()
```

7) 이차판별분석

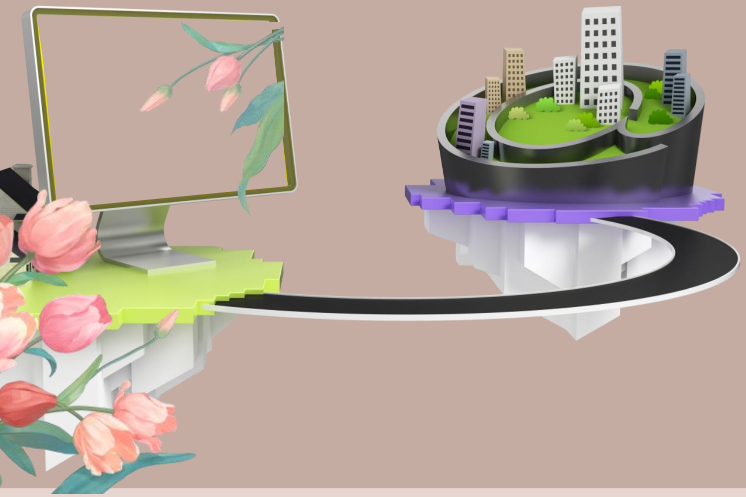
◆ 적합한 QDA 모형에 평가 데이터를 대입하여 분류 정확도와 오차행렬을 계산하자.

```
y_test_pred_qda = qda.predict(X_test)
confusion_matrix(y_test, y_test_pred_qda)
```

```
array([[47,  0,  0],
       [ 0, 21,  0],
       [ 0,  0, 36]])
```

```
qda.score(X_test, y_test)
```

```
1.0
```



다음시간안내

제06강

K-NN & Naïve Bayes