Supervised Learning: the Probabilistic Approach

Linear and Quadratic Discriminant Analysis

November 22, 2022

本章探討監督式學習中的群組分析。有別於將變數間的關係直接套用迴歸模型（Deterministic），這次從群組界線的後驗機率相等的角度切入，建立群組界線以劃分資料空間。從理論的推敲、Python 程式碼撰寫，到直接利用 **sklearn** 套件，一步步了解問題並學習解決問題的方式。而依假設情況之不同，探討線性與非線性的群組空間的界線。

本章將學到關於程式設計：
sklearn 套件使用、特殊散佈圖、特殊線性方程式的繪圖、Python 矩陣式的計算技巧。

〈本章關於 Python 的指令與語法〉
指令：
numpy: loadtxt, random.randint
matplotlib:colors.LinearSegmentedColormap
matplotlib.pyplot:cm.register_cmap, contour
sklearn.discriminant_analysis:LinearDiscriminantAnalysis, QuadraticDiscriminantAnalysis, fit, score, predict, predict_proba
sklearn.model_selection:train_test_split

1 基本觀念

監督式學習常用在群組分析中，學習資料與其所屬群組間的關係。一但關係建立，便能預測（判別）新資料（未知群組）所屬的群組。「監督」之名來自利用已知資料的群組別來確立資料與群組間的關係。建立資料與其所屬群組間的關係可以從機率的觀點切入。譬如，後驗機率就是最直覺的出發點。

1.1 線性判別式分析（LDA）

假設 X 代表多變量樣本的變數，G 代表群組的類別變數，則後驗機率寫為 $P(G = k|X = \mathbf{x})$，這個條件機率解釋為，在出現資料 $X = \mathbf{x}$ 的條件下，該資料屬於群組 $G = k$ 的機率。在分別計算資料屬於不同群組的機率後，得到最大機率的群組便是該資料之所屬。於是這個群組判別寫成計算最大後驗機率問題，如式 (1)，也稱為判別式分析（Discriminant Analysis）。

$$G(\mathbf{x}) = \arg\ \max_k \ln Pr(G = k|X = \mathbf{x}) \tag{1}$$

後驗機率 $P(G|X)$ 符合直覺的想法，卻不知從何計算，必須依賴貝氏定理的協助

$$P(G = k|X) = \frac{P(X|G = k)P(G = k)}{\sum_l P(X|G = l)P(G = l)} \tag{2}$$

其中 $P(X|G = k)$ 表示第 k 組資料發生的機率密度函數，而 $P(G = k)$ 代表群組 k 發生的機率。[1]相較於不知從何計算的後驗機率 $P(G = k|X)$，這兩個機率函數比較容易從已知的資料中估計得到。當然估計過程的假設與計算品質的好壞也間接影響了這個想法的準確度。本章假設群組的機率密度函數 $P(X|G = k) = f_k(X)$ 服從多變量常態分配（Multivariate Normal Distribution），寫成

$$f_k(X) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(X-\mu_\mathbf{k})^T\Sigma_k^{-1}(X-\mu_\mathbf{k})} \tag{3}$$

其中資料變數 $X \in R^p$，$\mu_\mathbf{k}$ 與 Σ_k 分別代表第 k 群資料常態假設的均值與共變異矩陣（Covariance Matrix）。這是對於母體的假設，至於真實的資料是否具備這個分配的特性還需要進一步檢驗。為簡化問題的複雜性，進一步假設所有群組

[1]一般稱 $P(X|G = k)$ 為概似函數，稱 $P(G = k)$ 為先驗機率。

的共變異矩陣都相等，即 $\Sigma_k = \Sigma, \forall k$。當然這個假設的合理性也是需要從實際的資料中做進一步的檢驗。[2]

加入貝氏定理與資料的常態假設後，式 (1) 的最大後驗機率問題改寫為

$$\begin{aligned} G(\mathbf{x}) &= \arg\max_k \ln Pr(G=k|X=\mathbf{x}) \\ &= \arg\max_k \ln(Pr(X=\mathbf{x}|G=k)Pr(G=k)) \\ &= \arg\max_k \ \mathbf{x}^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \ln Pr(G=k) \end{aligned} \tag{4}$$

第一行的意思很直覺；一筆新資料 $X = \mathbf{x}$ 來自哪一個群組的機率最高？經過貝氏定理 (2) 的轉換並去除與組別 k 無關的分母，變成了第二行。再將式 (3) 假設的常態函數代入（共變異矩陣相同），同樣去除與組別 k 無關的項目，變成了第三行，其中的目標函數又稱為線性判別式函數（Linear Discriminant Function），即

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \ln Pr(G=k) \tag{5}$$

在式 (5) 中，除了資料 **x** 已知外，其餘都未知，即便如此，我們仍可以利用已知的資料來估計這些值，譬如 μ_k 用第 k 組資料的樣本平均值，Σ 可以用各組資料算出來的樣本共變異矩陣（Sample Covariance Matrices）的加權平均，$Pr(G=k)$ 則是已知資料中各組數量的比例，即

- $Pr(G=k) \approx \hat{\pi}_k = N_k/N$ ，其中 N_k 代表第 k 組的數量，N 代表所有資料的總數。
- $\hat{\mu}_k = \sum_{group=k} \mathbf{x}_i/N_k$
- $\hat{\Sigma} = \sum_{k=1}^{K}\sum_{group=k}(\mathbf{x}_i - \hat{\mu}_k)(\mathbf{x}_i - \hat{\mu}_k)^T/(N-K)$ ，K 代表群組數。這個估計又稱為 Pooled within-group covariance matrix。

重回本章標題「從機率角度出發的監督式學習」，「機率」指「後驗機率」，佐以貝氏定理轉換及對群組資料的常態分配假設為方法，從已知資料及其明確的群組標示（此謂監督式）估計常態假設的參數，最後以判別函數的大小作為判別新資料群組的依據。這個過程有別於「從決定性方法出發的監督式學習」，出發的角度不同，但結果卻有驚人的相似之處。

[2]在科學與工程的研發過程，面對複雜或困難的問題無法立即解決時，常先將問題簡單化或理想化，去除某些障礙，使研究得以繼續推進。當簡化過的問題得到解決後，再一一恢復原先的條件。即使短時間無法完美解決，至少有近似的方案可以替代。

1.2 群組分界線

監督式學習的群組分析，其幾何意義猶如在資料所在的 $\Re^p$ 空間切割出 K 個領域（K 是群組數），切割的依據當然是給定的 N 筆已知資料及其群組別。而判別新資料的群組別時，便只是看資料落在哪個區域而已。在概念的表達上，我們喜歡從二維資料去繪製平面或空間的群組分界線，再推至 p 度空間的群組間的分界線。[3]

繪製分界線的做法必須先找出兩個群組的共同條件，譬如，群組 k 與 l 分界線滿足

$$Pr(G = k|X = \mathbf{x}) = Pr(G = l|X = \mathbf{x}) \tag{6}$$

也就是，在資料所在的空間裡，屬於群組 k 與群組 l 的機率相同的地方，或者說，切割群組 k 與群組 l 的線必須滿足以上的條件。在後驗機率相等的群組分界原則下，結合由貝式定理 (2) 與資料的常態分配假設 (3)，分界線的函數可以從以下的轉換得到：

$$\begin{aligned} \ln \frac{Pr(G = k|X = \mathbf{x})}{Pr(G = l|X = \mathbf{x})} &= \ln \frac{f_k(\mathbf{x})}{f_l(\mathbf{x})} + \ln \frac{Pr(G = k)}{Pr(G = l)} \\ &= \ln \frac{Pr(G = k)}{Pr(G = l)} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + \\ &\quad \mathbf{x}^T \Sigma^{-1}(\mu_k - \mu_l) = 0 \end{aligned} \tag{7}$$

這裡巧妙的運用對數轉換（logit transformation）的技巧，並令 log-odds 為零去除指數，得到一組線性的方程式（請注意：線性關係來自「不同群組有相同共變異矩陣」的假設）。於是從資料的後驗機率相等，得到式 (7) 的線性方程式，也決定群組的分野，當然也可供判斷一筆新資料的究竟落在哪個群組。在資料為二度空間的維度裡，這條線性的分界線可以被畫出來。

1.3 二次判別式分析（QDA）

如前段所述，群組間的線性分界線來自群組的共變異矩陣相同的假設。如果拿掉這個假設，令各群組的共變異矩陣不同時，則如式 (5) 的線性判別式函數將改寫為：

[3]這裡提到的「分界線」，除二度空間上是「線」，三度空間是「面」，更高維度則是更以廣泛的「Separating Hyperplanes」明之，一般通稱為 hyperplane 的幾何平面。

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\ln|\Sigma_k| - \frac{1}{2}(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k) + \ln Pr(G=k) \tag{8}$$

式 (8) 稱為二次判別式函數。稱為二次（Quadratic）的原因來自切割群組 k 與群組 l 間的分界線不是直線，而是變數的二次方，這條分界線寫為：

$$\{\mathbf{x}|\delta_{\mathbf{k}}(\mathbf{x}) = \delta_{\mathbf{l}}(\mathbf{x})\} \tag{9}$$

若令 $\mathbf{x} = [X_1\ X_2]$，則式 (9) 經過妥善地展開、推導為

$$c = c_1X_1 + c_2X_2 + c_3X_1X_2 + c_4X_1^2 + c_5X_2^2 \tag{10}$$

式 (10) 便是一條 (X_1, X_2) 平面上的二次曲線。口說無憑，以下的練習帶領讀者實際畫出二度空間裡的線性與二次式分界線，從理論的分析到程式的撰寫與 Python 工具的探索，都是學習必要的元素。

2 練習

範例 1. 下載測試資料 `la_1.txt`（散佈圖如圖 1），這是一組內含兩個已知群組的雙變量 $\Re^2$ 資料。想看看式 (7) 的 LDA 分群效果如何，試著畫出所示的 LDA 的分界線。

首先估計出分界線函數 (7) 所需的 $\mu_1, \mu_2, \Sigma, Pr(G=1), Pr(G=2)$ ：

- 估計之前，先檢視該資料的結構與內容，譬如群組的記號以 0 與 1 表達。
- 當兩個群組的數量相等時，Σ 的估計為 $\hat{\Sigma} = (\hat{\Sigma}_1 + \hat{\Sigma}_2)/2$，即兩群組之 pooled within group covariance matrix，定義為 $\Sigma = \frac{(n_1-1)\Sigma_1+(n_2-1)\Sigma_2}{n_1+n_2-2}$

參考程式碼如下：

```
import numpy as np
import numpy.linalg as LA
import matplotlib.pyplot as plt

data_dir = '../Data/'
D = np.loadtxt(data_dir + 'la_1.txt', comments='%')
```

```
X = D[:, 0:2]
y = D[:,2]
C1, C2 = X[y==0,:], X[y==1,:]

plt.plot(C1[:,0], C1[:,1], 'r>', label = 'Group A')
plt.plot(C2[:,0], C2[:,1],'b<', label = 'Group B')

# Estimatr the group parameters
n = D[:,0].size
n1, n2 = C1[:,0].size, C2[:,0].size
pi1, pi2 = n1/n, n2/n
mu1, mu2 = np.mean(C1, axis = 0), np.mean(C2, axis = 0)
Sigma = (np.cov(C1.T) + np.cov(C2.T))/2
```

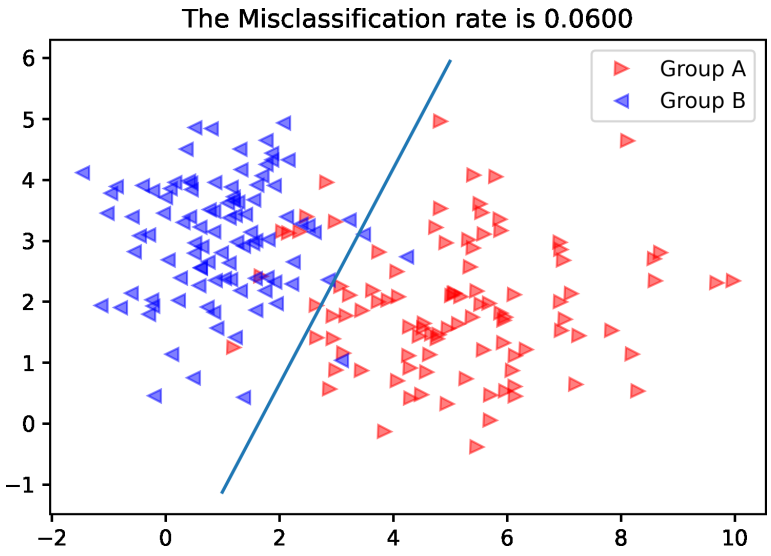

圖 1: LDA 群組分界線

在只有兩個變數的情況下，將式 (7) 中的 $\mathbf{x}^T$ 以 $[x_1\ x_2]$ 代入，改寫為 $K+[x_1\ x_2]L$，其中常數 K 與 2×1 向量 L 分別為

$$\begin{aligned} K &= \ln\frac{Pr(G=1)}{Pr(G=2)} - \frac{1}{2}(\mu_k+\mu_l)^T\Sigma^{-1}(\mu_k-\mu_l) \\ L &= \Sigma^{-1}(\mu_k-\mu_l) \end{aligned}$$

於是直線方程式寫成

$$K + L(1)x_1 + L(2)x_2 = 0$$

當 $\mu_1, \mu_2, \Sigma, Pr(G=1), Pr(G=2)$ 以估計值帶入計算 K 與 L 時，便能畫出圖 1 所示中間那條直線。從上述程式碼加入下列的直線函數估計與繪圖碼：

```
K = np.log(pi1/pi2) - 0.5 * (mu1 + mu2) \
        @ LA.inv(Sigma) @ (mu1 - mu2).T
L = LA.inv(Sigma) @ (mu1 - mu2).T

f = lambda x : -L[0]/L[1] * x - K/L[1]
x = np.linspace(1, 5, 10)
plt.plot(x, f(x))
```

圖 1 順帶計算了 LDA 的群組誤判率，請讀者試著模仿上一章以迴歸模型分類的誤判率計算（見習題說明），將誤判率寫在圖 1 的抬頭。

範例 2. 承上一個範例，再來看看如何使用 **sklearn** 套件提供的群組分類模組 `sklearn.discrimant_analysis` 裡面的 LDA 方法 `LinearDiscriminantAnalysis` 與 QDA 方法 `QuadraticDiscriminantAnalysis`。

以下程式碼展示 `LinearDiscriminantAnalysis` 執行 LDA 學習與錯判率的計算。

```
from sklearn.discriminant_analysis \
        import LinearDiscriminantAnalysis
from sklearn.discriminant_analysis \
        import QuadraticDiscriminantAnalysis

data_dir = '../Data/'
D = np.loadtxt(data_dir + 'la_1.txt', comments='%')
X = D[:, 0:2]
y = D[:,2]
Lda = LinearDiscriminantAnalysis(tol = 1e-6)
Lda.fit(X, y)
K = Lda.intercept_
L = Lda.coef_
MissClassRateLDA = 1 - Lda.score(X, y)
```

上述 LDA 學習器（也稱群組分析器 Classifier）的定義 `Lda = LinearDiscriminantAnalysis(tol = 1e-6)` 與資料配適 `Lda.fit(X, y)` 都是典型的機器學習模式，而學習過後的「副產品」都可以在 `Lda` 的 attributes 裡找到。譬如，分界線函數的係數：`Lda.intercept_, Lda.coef_`。學習器 `LinearDiscriminantAnalysis` 的種種參數的設定值，可以從 `Lda` 的 method `LDA.get_params` 取得，譬如在上述程式碼中，只設定一個參數值 `tol =`

`1e-6`，至於其意義與其他可設定的參數請參考線上手冊。[4]

學習器 `LinearDiscriminantAnalysis` 提供幾個常用的 methods，譬如計算分群的準確率 `Lda.score(X, y)`、預測資料群組的 `Lda.predict(X)` 及計算後驗機率值的 `Lda.predict_prob`。上述程式碼利用 `Lda.score(X, y)` 計算錯判率（或稱訓練誤差 Training error）。

至於 QDA 學習器 `QuadraticDiscriminantAnalysis` 的使用方式也是比照辦理，如下：

```
Qda = QuadraticDiscriminantAnalysis(tol = 1e-6, \
        store_covariance = True)
Qda.fit(X, y)
MissClassRateLDA = 1 - Qda.score(X, y)
```

不管 `LinearDiscriminantAnalysis` 或 `QuadraticDiscriminantAnalysis` 的使用都非常簡單。一般而言，只要計算錯判率與對新資料的預測就算完成任務了。比較複雜，或說，比較精彩的部分在於繪圖，譬如多個群組資料的散佈圖、LDA 與 QDA 的分界線函數繪製。下一個範例與圖 2 展示 Python 常見利用色彩的分佈來呈現群組的地域關係。

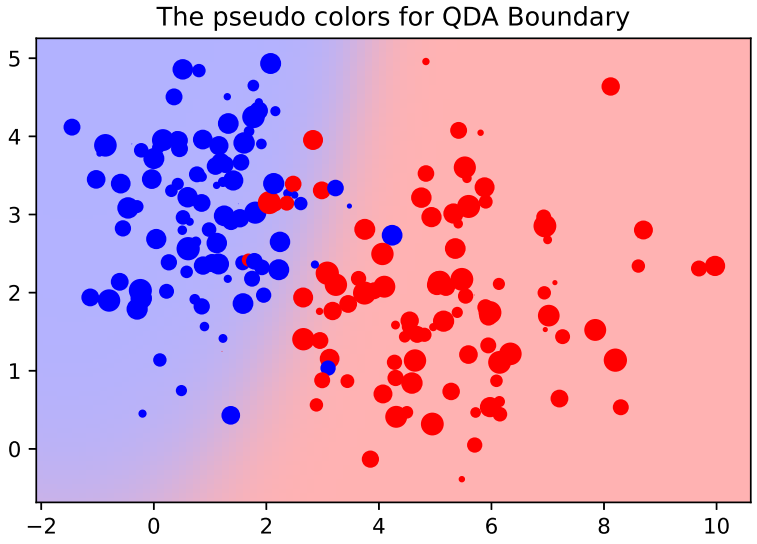

圖 2: QDA 對於資料 `la_1.txt` 的色彩界線

[4]參考手冊前，可以先 `print(dir(Lda))` 來看看 `LinearDiscriminantAnalysis` 學習器有哪些 attributes 與 methods。

範例 3. 群組資料的散佈圖（scatter），通常以圖案（marker）與顏色（color）來表達。本範例利用群組的後驗機率為一多變量函數（式 (2)），將函數值對應到不同顏色，並在函數值之間產生漸層效應，畫出如圖 2 具群組分辨的色彩分佈圖。在 **matplotlib.pyplot** 套件裡，稱為 pseudo color，指令為 `pcolormesh`。

畫出圖 2 的程式碼如下：

```
from matplotlib import colors
import matplotlib.pyplot as plt

data_dir = '../Data/'
D = np.loadtxt(data_dir + 'la_1.txt', comments='%')
X = D[:, 0:2]
y = D[:,2]

area = 2 * np.random.randint(50, size = D[:, 0].size)
grp_color = [[1,0,0] if i == 0 else [0,0,1] for i in y]
plt.scatter(D[:, 0], D[:, 1], c = grp_color, s = area, \
        alpha = 0.5, marker = 'o' )

Qda = QuadraticDiscriminantAnalysis(\
        tol = 1e-6, store_covariance = True)
Qda.fit(X, y)

nx, ny = 100, 100
x_min, x_max = plt.xlim()
y_min, y_max = plt.ylim()
x_ = np.linspace(x_min, x_max, nx)
y_ = np.linspace(y_min, y_max, ny)
xx, yy = np.meshgrid(x_, y_)

Z = Qda.predict_proba(np.c_[xx.ravel(), yy.ravel()])
Z = Z[:, 1].reshape(xx.shape)

# Define pseudo colors
cdit = {'red': [(0, 1, 1), (1, 0.7, 0.7)],
        'green': [(0, 0.7, 0.7), (1, 0.7, 0.7)],
        'blue': [(0, 0.7, 0.7), (1, 1, 1)]}
cmap = colors.LinearSegmentedColormap(
    'red_blue_classes', cdit)
plt.cm.register_cmap(cmap = cmap)

plt.pcolormesh(xx, yy, Z, cmap = 'red_blue_classes',\
    norm = colors.Normalize(0., 1.),\
    shading = 'auto', zorder = 0)
```

圖 2 顏色的漸層效果來自程式碼中的 `colors.LinearSegmentedColormap`，而色調來自前一行對於 RGB 三個元素的設定，並朝向紅色與藍色的搭配。[5]指令 `pcolormesh()` 將 `Z` 值對應顏色，而 `Z` 值代表群組的後驗機率值（式 (2)）。從圖 2 隱約可以看出在左邊的藍色調與右邊的紅色調中，有一條邊線，這條邊線對應的後驗機率值為 0.5（在兩群組的條件下），即在線上每一點的資料屬於任一群組的後驗機率值皆為 0.5。這便是分界線的意義。圖加上後驗機率函數的等高線圖為 0.5 的那條線，正好貼在顏色的分水嶺。程式碼加上這一行。

```
contoursQDA = plt.contour(xx, yy, Z, [0.5],colors = 'k')
```

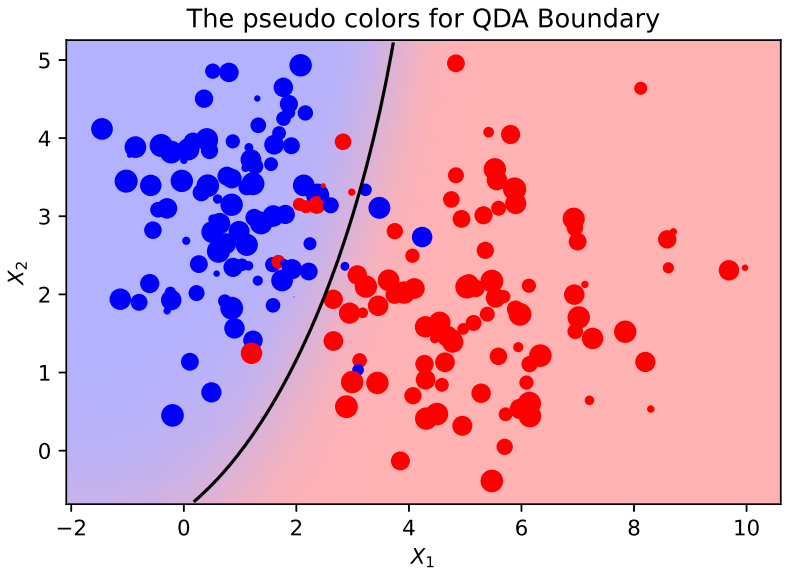

圖 3: QDA 對於資料 `la_1.txt` 的色彩界線與後驗機率（群組）分界線

讀者可以試著在圖 3 上加入 LDA 的群組分界線。做法可以模仿上述計算後驗機率的程式碼，或直接從前一個範例計算得到的 `Lda.intercept_` 與 `Lda.coef_` 畫直線函數。另外，再從 `Lda.score(X, y)` 與 `Qda.score(X, y)` 分別計算 LDA 與 QDA 對於同一筆資料的誤判率。

範例 4. 同前一範例利用漸層顏色來區隔資料空間，但是用一個比較簡單的方式。先將資料空間畫分為網格狀（grids），再使用 LDA 或 QDA 來預測每個網格點的群組屬性，依群組屬性畫上一個同顏色的點（紅或藍），當網格密度夠大

[5]這些顏色的定義將對應到 0 到 1 的數字。當對應的數字接近 0 時，分配的顏色接近紅色；當數字接近 1 時，則分配接近藍色。關於配置的細節與理論，可以參考：https://matplotlib.org/stable/api/_as_gen/matplotlib.colors.LinearSegmentedColormap.htm

時，同樣能用顏色切割資料空間。如圖 4 所示。

本範例一方面練習 LDA 與 QDA 學習器的預測功能，另一方面也練習資料的視覺表達技巧。參考程式碼如下：

```
Qda = QuadraticDiscriminantAnalysis(tol = 1e-6)
Qda.fit(X, y)

nx, ny = 200, 100
x_min, x_max = plt.xlim()
y_min, y_max = plt.ylim()
x_ = np.linspace(x_min, x_max, nx)
y_ = np.linspace(y_min, y_max, ny)
xx, yy = np.meshgrid(x_, y_)
x1, x2 = xx.ravel(), yy.ravel()
zz = Qda.predict(np.c_[x1, x2])

# colors = ['r', 'b']
colors =['#F2CBCB', '#CBEFF2']
for i in range(2) :
    plt.scatter(x1[zz==i], x2[zz==i], marker='.',
    color=colors[i])
```

上述程式碼共預測 $X_1 - X_2$ 空間裡的 200×100 筆資料點，預測結果放在變數 `zz`，讀者可以去查看 `zz` 的內容。最後將這些資料點的預測結果依群組值分開繪製散佈圖，而顏色選擇接近藍與紅的淡色系。

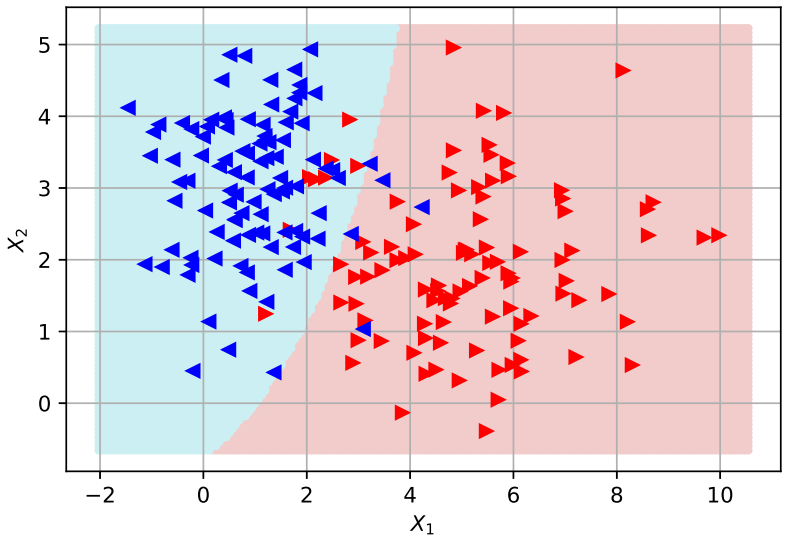

圖 4: QDA 對於資料空間的點預測與色彩分界線

範例 5. 前面的範例中使用等量的群組資料，但當群組大小不同時，結果會有什麼差別呢？不妨自己產生模擬資料，模擬兩個群組的位置與規模，再利用這些人工資料看看群組間距與規模大小對 LDA 與 QDA 的表現的影響？同樣的是否也試試三個群組或以上的分群狀況。

假設三個群組的中心點與共變異矩陣分別為：

$$\mu_1 = \begin{bmatrix} 0.5 \\ -0.2 \end{bmatrix}, \mu_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \mu_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 2 & 0.3 \\ 0.3 & 0.5 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

群組大小分別為 $n_1 = 300, n_2 = 200, n_3 = 100$。圖 5 呈現三群資料的分佈情況與 QDA 的分界線。讀者可以試著加入資料空間的顏色分群效果，並計算分群的準確率。[6]

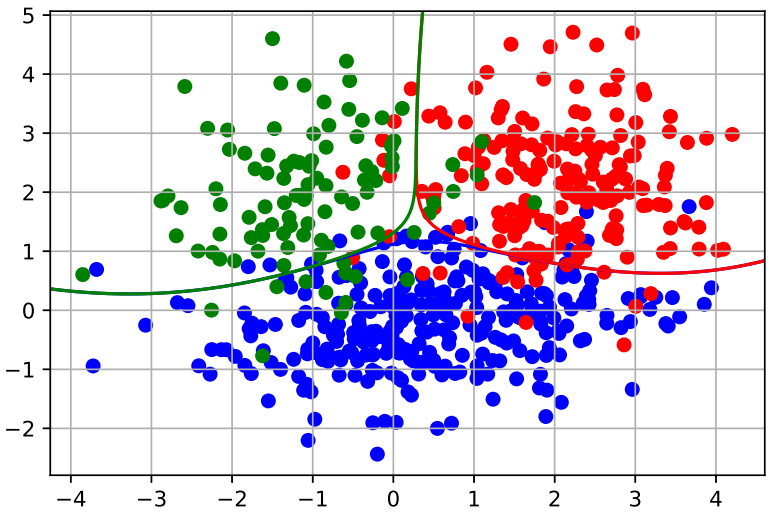

圖 5: 數量不同的三群組模擬資料與 QDA 分群。

範例 6. 利用模擬資料，可以很方便比較各種學習器的表現。本範例示範將一筆模擬資料依隨機方式分成兩部分，比例為 8:2，較大的部分做為訓練資料，其餘為測試資料。因為資料的切割採隨機抽取方式，為公平起見，共分割 K 次，進

[6]至於資料生成的方式，可以參考前一章的「觀察與延伸」小節中的程式碼。

行 K 次的 LDA 與 QDA 分群學習與測試，[7]並記錄每次的誤判率，最後採 K 次平均值作為誤判率的代表。

利用模擬資料作為分類學習器的評比前，必須仔細設計資料的「樣貌」，包括樣本數、組數、群組接進程度（中心點的位置）、共變異矩陣的差異等，才能進行一場公平的競爭。本範例的模擬資料只是方便說明一些程式技巧而已，並未考量諸多情境。以下程式碼僅列出關鍵的程式片段供參考。請讀者試著從前面的程式碼去拼湊出完整的程式。

```
from sklearn.model_selection import train_test_split

... Generate simulated data X, y

K =100
LDA.trainingError = np.zeros(K)

...
Lda = LinearDiscriminantAnalysis(tol = 1e-6)

for i in range(K) :
# split data into TRAINing and TESTing parts
        X_train, X_test, y_train, y_test = \
                train_test_split(X, y, test_size = 0.2)
        Lda.fit(X_train, y_train)
        LDA_trainingErr[i] = ...
        ...

print('LDA␣training␣Error:{:.4f}'.format(LDA_trainingErr.mean()))
```

程式最後呈現誤判率平均值的方式，是一般程式寫作過程的習慣，用來檢查程式是否執行正確。至於訓練後的誤差如何使用，就看後續的的用途了，譬如搜集五種情境資料，分別計算誤判率後，繪製折線圖，方便比較。

3 觀察與延伸

1. 在比較各種學習器（或演算法、統計量...）時，需要模擬大量符合各種情境的資料，因此有必要寫一支副程式（`def`）來供應各種需求。譬如，寫一支副程式生成多組服從多變量常態的資料，其中輸入參數（input arguments 包括各群組的平均數、共變異矩陣及樣本數；輸出參數則為 `X`

[7]讀者可以自行加入其他學習器一起評比。

矩陣，代表所有群組的輸入資料及 y 向量，代表群組別資料（譬如 0, 1, 2,...）。譬如，

```
def mvn_multiclass_data(mean, cov, n) :
    '''
    generate multiclass data with multivariate Normal dist.
    Ex. for a 3 class data
    n = [200, 300, 400] # sample size for each group
    mean = np.array([[0.5, -0.2], [2, 2],[-1, 2]])
    cov = np.array([[2.0, 0.3], [0.3, 0.5], \
    [1.0, 0.], [0., 1.], [1.0, 0.], [0., 1.]])
    '''
    ...
    ...
    ...
    return X, y
```

上述函數內的說明，以三群組資料服從雙變量常態為例，其中輸入參數 mean 是三組平均數疊成的 3 x 2 矩陣，參數 cov 是三個共變異矩陣疊成的 6 x 2 矩陣，而參數 n 是個向量含三個群組的樣本數（[200, 300, 400]）。則輸出資料 X 為 900 x 2 的矩陣，而群組別 y 為 ndarray(900,)。請讀者自行練習填補中間的程式碼（可參考前一篇講義的程式碼）。

2. 在兩個群組的情況下，如果訓練資料中兩群組的數量不同，會產生什麼狀況？譬如：$Pr(G = 1) = 2Pr(G = 2)$。建議讀者先想想看，再從程式的執行結果印證。

3. LDA 是從後驗機率的角度做出群組間的分界線，相對於從決定性（Deterministic）角度出發的迴歸模式（最小平方法），在只有兩個群組的情況下，有何相似之處嗎？試著針對同一組資料把這兩條線同時畫出來，比較看看。要仔細看喔！

4. 羅吉斯迴歸（Logistic regression）是處理類別型輸出資料的重要的工具。運用在群組分析上可以避免如 LDA 對概似函數 (likelihood function) 的常態性假設。這算是對常態分配的解套，[8]不過另一方面，它也對後驗機率 $P(G|X)$ 做了假設

[8]LDA 的分群方法來自對群組資料的兩個假設限制。其一是群組資料服從多變量常態分配，另一個假設是群組的共變異矩陣相等。QDA 分群法解除了共變異矩陣相等的限制，而羅吉斯迴歸則是解除常態分配的假設。

$$p = Pr(G = 1|X, \beta) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}} \tag{11}$$

這個假設對許多的資料來源而言，具有相當程度的合理性。令其 log odds ratio 等於 0 時，求得最佳的分界線：

$$c = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x = 0 \tag{12}$$

參數 β_0, β_1 由已知的資料估計而得。讀者可以試著將邏輯斯迴歸的模型應用在本單元的資料，進行群組判別。

4 習題

1. 從式 (4) 第一行的先驗分配利用貝氏定理與常態分配假設，推導至第三行的判別式函數。
2. 詳細地從式 (6) 的條件推導至式 (7) 的分界線函數。
3. 推導式 (8) 的二次判別函數。
4. 推導式 (9) 在二度空間中的二次式分界線函數。
5. 圖 1 將 LDA 的誤判率寫在抬頭。而誤判率的計算來自式 (5)，比較 $\delta_1(\mathbf{x})$ 與 $\delta_2(\mathbf{x})$ 的大小來決定資料 (x) 所屬的群組別。
6. 將你的程式運用在 `la_3.txt` 這組資料，畫出分界線，並允許輸入新資料，且立即輸出該資料經判別後的組別。
7. 請比較 Logistic regression 與 LDA 在假設上、方法上及最後做出的分界線有何不同。Hint: 式 (7) 與 (12)。
8. 試著產生具三個群組別的資料，每組 100 筆，群組間的距離自己拿捏。利用 LDA 的方法在不同的兩個群組間畫一條線，共可畫出三條分界線以區隔三個群組。這三條線是否交於同一點？圖 5 的 QDA 非線性分界線並沒有相交於一點。

References

[1] T. Hastie, R. Tibshirani, J. Friedman, ”The Elements of Statistical Learning:Data Mining, Inference, and Prediction,” Springer.