Supervised Learning：the Mixed Approach

Logistic Regression

July 30, 2019

迴歸分析會因爲資料的型態的不同而採取不同的模式，例如簡單線性迴歸與多項式迴歸都是觀察自變數與因變數資料間的關係所採取的適當模式。本章要探討當因變數爲類別資料時，邏吉斯迴歸（Logistic Regression ）是一個不錯的迴歸模式。這個模式在參數的估計上通常配合最大概似法（Maximum likelihood）進行非線性的參數估計。這□非線性的估計在研究領域很常見，本章提供最簡單的解決方案，作爲非線性估計的開端。就此目的而言，邏吉斯迴歸只是簡單的開場白。另 MATLAB 的機器學習（Machine Learning）套件也提出簡潔的解決方式，是學習邏吉斯迴歸的概念與應用最佳的工具。

本章將學到關於程式設計

- MATLAB 提供關於邏吉斯迴歸的解決方式。
- 兩個變數函數的繪圖方式。

〈本章關於 MATLAB 的指令與語法〉
指令：fminsearch, fitnlm, glmfit, predict

1 背景介紹

1.1 邏吉斯迴歸

迴歸模型用來描述自變數與因變數間的關係。當因變數爲群組性質的類別資料且其發生的機率如圖 1 的趨勢時，前章提過的 LDA（Linear Discriminant Analysis）的概念，結合簡單線性迴歸模式，便適合處理這類的問題。LDA 對群組的區別以後驗機率的 Logit Transformation 等於 0 作爲分野，即

$$\log \frac{Pr(G=k|X=\mathbf{x})}{Pr(G=l|X=\mathbf{x})} = 0$$

其中 X 與 G 分別代表自變數及因變數所代表的的群組別。利用貝氏定理與群組資料的常態分配假設（且群組間的共變異矩陣相同），上式可以簡化爲一線性函數

$$\log \frac{Pr(G=k|X=\mathbf{x})}{Pr(G=l|X=\mathbf{x})} = \lambda_0 + \lambda^T \mathbf{x}$$

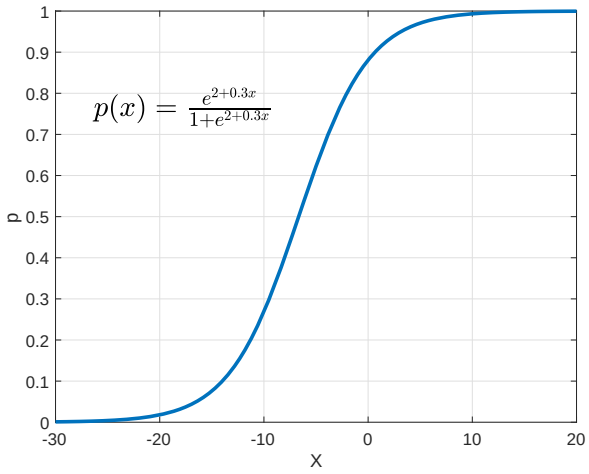

圖 1: 因變數群組發生的機率與自變數的關係：A Sigmoid function

邏吉斯迴歸維持群組分野如上式的簡單線性關係，並免除對群組資料的常態分配假設，直接採用如圖 1 的函數做爲群組的後驗機率。以兩個群組爲例，假設

$$Pr(G=1|X=\mathbf{x}) = \frac{e^{f(\mathbf{x})}}{1+e^{f(\mathbf{x})}} \tag{1}$$

其中 $f(x) = \beta_0 + \beta_1 x + \beta_2 x_2 + \cdots + \beta_k x_k = \beta_0 + \beta^T \mathbf{x}$，$\beta = [\beta_1\ \beta_2\ \cdots\ \beta_k]^T, \mathbf{x} = [x_1\ x_2\ \cdots\ x_k]^T$。

則後驗機率的 Logit Transformation 寫成

$$z = \log \frac{Pr(G=1|X=\mathbf{x})}{Pr(G=2|X=\mathbf{x})} = \beta_0 + \beta^{\mathbf{T}}\mathbf{x} \tag{2}$$

式 (2) 也是個簡單的迴歸模型。本單元的重點將是探討如何利用最大概似估計法估計未知參數 $\theta = [\beta_0\ \beta_1\ \cdots, \beta_k]$。[1]

關於式 (1) 對後驗機率的假設並非全無道理，它其實符合自然界某些運作法則。隨著自變數逐漸變大，因變數的反應機率從一個極端 (0) 走向另一個極端 (1)。中間轉換的幅度可能是和緩的，也有些是比較陡峭的。例如生物細胞受電壓的激發，其電壓大小與激發與否的關係也是如此，不過中間轉換曲線比較直聳，有點像 Step function，一般也叫做 Sigmoid function。式 (2) 亦稱爲 Log odds ratio（對數優勢比）。將優勢比取對數後對 **x** 作迴歸稱爲邏吉斯迴歸。這種透過變數的轉換將較複雜的模式 (1) 變爲簡單的模式 (2) 是迴歸分析常見的手段,所謂「山不轉,路轉。」不管對因變數或自變數都可以。

式 (1) 的後驗機率也可以解讀爲「成功的比例,」是一個遞增函數，兩端是近乎穩定平緩的水平線，中間部分變化較大。譬如，假設 **x** 與 $Pr(G=1|X=\mathbf{x})$ 代表年齡與罹患心臟病比例，雖兩者的關係是一平滑連續的漸增曲線，即年齡越大罹患心臟病的比例越高，不過當進行實際資料的蒐集時，原始資料僅呈現年齡（X）與罹患心臟病與否（G）的二元性數據（0 或 1），如圖 2所示，G 可視爲伯努力分配的變數，其成功比例爲 $Pr(G=1|X=\mathbf{x})$ 。

當拿連續型變數 X 與二元性資料 G 作邏吉斯迴歸時，一般採最大概似法的觀念來估算參數值。假設在已知自變數 X 及參數 θ 下，因變數 G 的條件機率密度函數爲

$$f(G|\mathbf{x},\theta) = Pr(\text{G: 罹患心臟病與否}|\mathbf{x},\theta) \tag{3}$$

藉由式 (1) 罹患心臟病的機率及因變數觀察值的二元性，因變數 G 的條件機率密度函數 (3) 可以進一步寫成

$$\begin{aligned} f(G=1|\mathbf{x},\theta) = Pr(\text{罹患心臟病}|x,\theta) &= \frac{e^{\beta_0+\beta^T\mathbf{x}}}{1+e^{\beta_0+\beta^T\mathbf{x}}} = p(\mathbf{x},\theta) \\ f(G=0|\mathbf{x},\theta) = 1 - Pr(\text{罹患心臟病}|x,\theta) &= \frac{1}{1+e^{\beta_0+\beta^T\mathbf{x}}} = 1-p(\mathbf{x},\theta) \end{aligned} \tag{4}$$

[1]以現在軟體的規格來說，根本不需要程式設計者寫程式計算最大概似估計的參數值，往往一個指令便解決邏輯斯迴歸的所有計算。在此仍從最大概似估計的問題出發，從理論的推導到程式的撰寫亦步亦趨的完成，最大的用意著眼在訓練讀者從理論到實務的過程，加強程式撰寫能力，而不是只會每個巨大的指令間跳躍。有一天你也需要從細節中創造偉大。

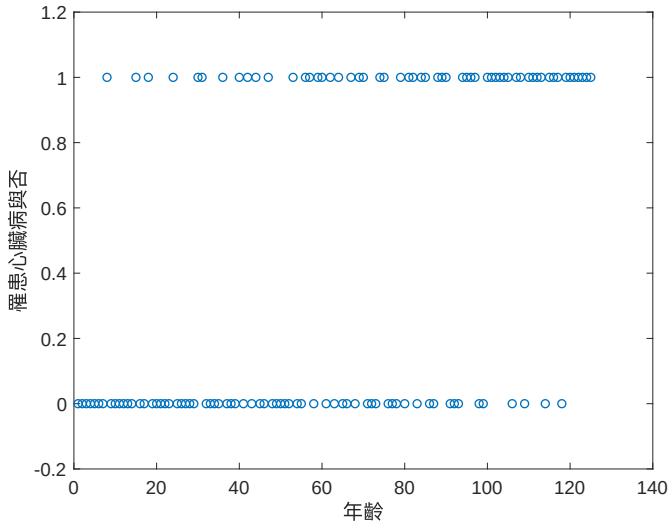

圖 2: 年齡與罹患心臟病的調查結果。0 代表「否」，1 代表「是」。

1.2 最大概似估計

在已知 N 個獨立觀察值 $(\{\mathbf{x}_k\}, \{g_k\}), k = 1, 2, \cdots, N$，因變數 G 的聯合機率密度函數（Joint Probability Density Function ）寫成

$$f(\mathbf{g}|\{\mathbf{x}_k\}, \theta) = \prod_{k=1}^{N} f(g_k|\mathbf{x}_k, \theta) \tag{5}$$

上式稱爲概似函數（Likelihood Function）。套用式 (4)，這個概似函數可以改寫爲

$$L(\theta) = f(\mathbf{g}|\{\mathbf{x}_k\}, \theta) = \prod_{k=1}^{N} f(g_k|\mathbf{x}_k, \theta) = \prod_{k=1}^{N} p(\mathbf{x}_k, \theta)^{g_k}(1 - p(\mathbf{x}_k, \theta))^{1-g_k} \tag{6}$$

所謂最大概似值的觀念便是選擇一組參數值 $\theta = \theta^o$ 使得概似函數 $L(\theta)$ 最大，這表示當 $\theta = \theta^o$ 的情況下，出現 N 個觀察值 $(\{\mathbf{x}_k\}, \{g_k\}), k = 1, 2, \cdots, N$ 的機會最大。這相當於要解決下列最佳化問題，

$$\max_{\theta} L(\theta)$$

由於概似函數的「長相」，對概似函數取對數後再求最佳值在計算上比較簡單，在不影響最佳值的情況下，問題變爲

$$\max_{\theta} \ln L(\theta) \tag{7}$$

其中對數概似函數經化簡後（習題 1），變成

$$\ln L(\theta) = \sum_{k=1}^{N} \left(g_k \theta^T \tilde{\mathbf{x}}_k - \log(1 + e^{\theta^T \tilde{\mathbf{x}}_k}) \right) \tag{8}$$

其中 $\tilde{\mathbf{x}}_k = [1 \ \mathbf{x}_k]^T$。從對數概似函數的一次導數 (即梯度向量 Gradient Vector)

$$\nabla \ln L(\theta) = \sum_{k=1}^{N} \left(g_k - \frac{e^{\theta^T \tilde{\mathbf{x}}_k}}{1 + e^{\theta^T \tilde{\mathbf{x}}_k}} \right) \tilde{\mathbf{x}}_k \tag{9}$$

當 $\nabla \ln L(\theta) = \mathbf{0}$ 時, 對 θ 而言是一組非線性的方程式, 沒有解析解 (Analytic Solution)。必須採用演算法以迭代遞迴（Iteration）的方式逐步迭代出最佳解。由於多數的演算法都以求最小值爲主，最大概似值的問題可以改寫成

$$\max_{\theta} \ln L(\theta) = \min_{\theta} - \ln L(\theta) \tag{10}$$

求解如式 (10) 的多變量函數最小值的演算法多如繁星，不在此介紹。各品牌軟體都有表現不錯的指令或套件可供使用，使用者只要對演算法的觀念有些粗淺的概念，多能正確使用並且確定找到裡想的答案。以下練習詳細的說明 MATLAB 提供的解決方案，若使用其他軟體，也大概可以比照辦理，找到相對應的指令。

2 MATLAB 的作法

範例 1 附件 heart_attack.txt 是一組 *125* 筆紀錄年齡與罹患心臟病與否的眞實資料，如圖 *2* 所示。請試著利用這組資料畫出對數概似函數 (8)，其中 $\theta = [\beta_0 \ \beta_1]$。請注意未知參數 β_0, β_1 範圍的訂定非常關鍵，否則即程式碼正確了，依然看不出所以然。

圖 3 的概似函數圖是經過適當旋轉後的樣子。當 MATLAB 自行決定的角度不利觀察時，可以利用繪圖區的旋轉功能，適度的旋轉，找到一個最佳的觀察視野。從圖上可以看出概似函數在 β_0 的方向一片平坦，說明 β_0 比較不具鑑別能力，反觀在 β_1 的方向則是清楚的凹線。以下程式碼利用式 (8) 與附件資料畫出圖 3。[2]

[2]圖 3 呈現函數的立體樣貌，由於該函數特別的長相，從圖形不易觀察最小值的位置。此時可以改採等高線圖（contour）試試，指令如 contour(B0, B1, Z) 或在後面加上等高線的線條數，譬如 mesh(B0, B1, Z, 200) 幫助「鎖定」最小值。

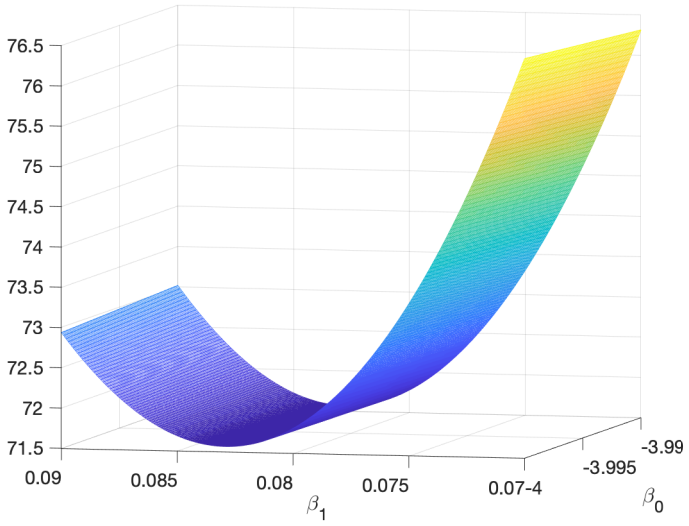

圖 3: 在「谷底」的部分比較平滑的對數概似函數

```
D=load('heart_attack.txt'); % 讀入資料 heart_attack
x=D(:, 1) % 年齡資料
g=D(:, 2); % 罹病與否
[B0, B1]=meshgrid(-4:0.0001:-3.99, 0.07:0.0001:0.09);
Z1=sum(g)*B0+g'*x*B1;
Z2=zeros(size(Z1));
for  i=1:N
        Z2=Z2 + log(1+exp(B0+B1*x(i)));
end
Z=-Z1+Z2;
mesh(B0, B1, Z)
```

範例 2 使用 MATLAB 指令 fminsearch 計算式 (10) 的最大概似估計問題。[3]

---

[3]關於指令 fminsearch 詳盡的使用方式，請參考 MATLAB 使用手册，或作者的專書:《Coding Math: 寫 MATLAB 程式解數學》

```
D=load('heart_attack.txt'); % 讀入資料 heart_attack
x=D(:, 1) % 年齡資料
g=D(:, 2); % 罹病與否
Lf=@(b) sum(log(1+exp(b(1)+b(2)*x))-g.*(b(1)+b(2)*x)); % 對數概似函數之負數
ini=[4 1];
[x_opt, fmin]=fminsearch(Lf, ini)
```

這裡採用匿名函數的方式，方便程式運作。其中變數爲 1×2 的向量 b，$b(1)$ 代表 β_0 ，$b(2)$ 代表 β_1。執行結果當得到 x_opt=[-3.9933 0.0828]，函數之最小值約爲 fmin=71.5678，也就是最大概似函數值爲 -71.5678。

範例 3 MATLAB 也提供指令 fitglm 與 glmfit 兩個指令，[4]直接進行各種迴歸模式的參數估計與相關統計量計算。本範例利用這兩個指令與上述的資料，進行羅吉斯迴歸分析。

MATLAB 指令 glmfit 提供廣義線性模型的迴歸分析（Generalized Linear Model Regression）。所謂「廣義」，指的是對於因變數的分配函數不再局限於常態分配，譬如本章探討的因變數爲群組類別，屬二元分配函數，一般線性迴歸模型並不適用。[5]先示範指令 glmfit 的使用方式:

```
D=load('heart_attack.txt'); % 讀入資料 heart_attack
x=D(:, 1) % 年齡資料
g=D(:, 2); % 罹病與否
b = glmfit(x, g, 'binomial', 'link', 'logit'); % 選擇 logit 轉換模式
yfit = glmval(b, x, 'logit');
plot(x, yfit, 'o')
```

這個程式碼執行結果與上一個範例得到一樣的迴歸參數。至於這個估計好或是不好，可以利用指令 glmval 計算擬合值 yfit，並與圖 1 配適看看。結果如圖 4。

MATLAB 自從推出 Machine Learning 套件之後，將相關的指令重新編排，維持使用的一致性。於是 glmfit 另設爲 fitglm，使用方式如下程式碼所示:

---

[4]glmfit 已經被 fitglm 取代。指令 glmfit 推出時間較早，後來配合 MATLAB 推出的 Machine Learning 套件，使用一致化的指令檔頭 fit— 做爲各種模型的學習（訓練）工具，並使用相同的變數格式與預測、誤差計算、、、等指令。

[5]因變數的型態（分配），可能是二項（Binomial）、常態（Normal）、卜瓦松（Poisson）等，透過適當的轉換函數或稱鏈結函數（Link Function）建立與自變數的關係。本章主要針對二項式的因變數，譬如成功或失敗，適合的鏈結函數爲 Logit function，即 $\ln \frac{\mu}{1-\mu}$，其中 μ 爲因變數的期望值。

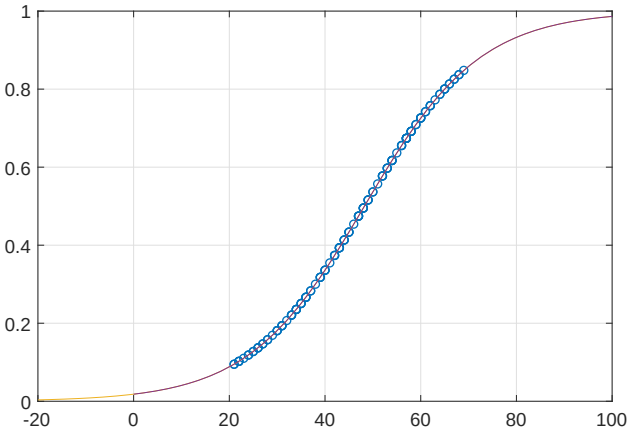

圖 4: 羅吉斯迴歸的模型配適情形

```
mdl = fitglm(x, y,'Distribution', 'binomial'); % 選擇因變數爲二項分配
yfit =predict(mdl, x);
```

輸出結果 mdl 也是標準化的形式，如圖 5 所示。而指令 predict 是機器學習套件的共用指令，任何學習模式的結果都可以用 predict 作爲訓練資料的擬合值計算與新資料的預測。上述指令 fitglm 只指定因變數分配爲 binomial，沒有指定轉換函數。不過從圖 5 的執行結果看到 logit(y) ∼ 1+x1，表明了轉換函數。

```
mdl =


Generalized linear regression model:
    logit(y) ~ 1 + x1
    Distribution = Binomial

Estimated Coefficients:
                   Estimate       SE        tStat       pValue
                   ________    ________    _______    __________

    (Intercept)     -3.9933     0.83558    -4.7791    1.7609e-06
    x1             0.082793    0.017323     4.7793     1.759e-06


125 observations, 123 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 29.2, p-value = 6.59e-08
```

圖 5: 指令 fitglm 執行結果

範例 4 上述範例的資料來自眞實案例，所以估計結果的「品質」好壞無法判斷。一般來說，會以模擬資料來評估模型與演算法估計的優劣。本範例示範因變數爲二項分配資料

的模擬，並代入 MATLAB 指令 fitglm 與 predict 做估計與產生擬合值。資料生成的模擬參數設定爲:

$$\beta_0 = -2, \beta_1 = 1,\ N = 100,\ X \sim Unif(0, 5)$$

資料模擬是很重要的訓練，幫助讀者更理解統計原理與應用的模型。本範例的自變數訂爲單一變數且來自均勻分配，這是沒有根據任何實務面的考量，純粹以亂數當作自變數資料。因變數的資料產生便是依據本章對於邏輯斯迴歸模型的描述產生的。讀者最好先自己想一想，再參考如下的程式碼。

```
RandStream.setGlobalStream(RandStream('mt19937ar','seed',sum(100*clock)));
b0=-2; b1=1; N=100;
x=unifrnd(0,5,N,1); % 先產生自變數資料
A=exp(b0+b1*x);
p=A./(1+A); % 計算因變數的期望值
g=binornd(1, p); % 生成二項分配的因變數資料
mdl = fitglm(x, g,'Distribution', 'binomial'); % 進行估計與計算相關統計量
yfit =predict(mdl, x); % 計算擬合值
f=@(x) exp(b0+b1*x)./(1+exp(b0+b1*x)); % 繪製機率圖
fplot(f), [-5 10])
hold on, plot(x, yfit, 'o'), hold off % 貼上擬合值
```

圖 6 展示上述程式碼的某次執行結果。之所以說是「某次」執行的結果，來自自變數從亂數生成來，而且亂數的「種子」（seed）並不固定。程式碼第一行定義了亂數取用的「種子」來源爲 sum(100*clock)，也就是根據電腦當時的時間做些處理後的數字。如果希望產生固定的模擬資料，必須將種子設爲固定數字。不同的模擬資料當然估計出不同的參數值，因此不能從一次的模擬資料判斷估計的品質。本章在習題中提出一種在學術研究上常見的模擬情境，讀者不妨試著寫程式做做看。

3 習題

1. 推導式 (8)、(9)。
2. 繪製圖 3 的等高線圖，必須能清楚地看出最小值的大概位置。
3. 將本章的羅吉斯迴歸方法用在他章所使用的資料，如 la_1.txt,la_2.txt，試著畫出那條分界線，並與之前做過的一般線性迴歸模型與 LDA 比較。

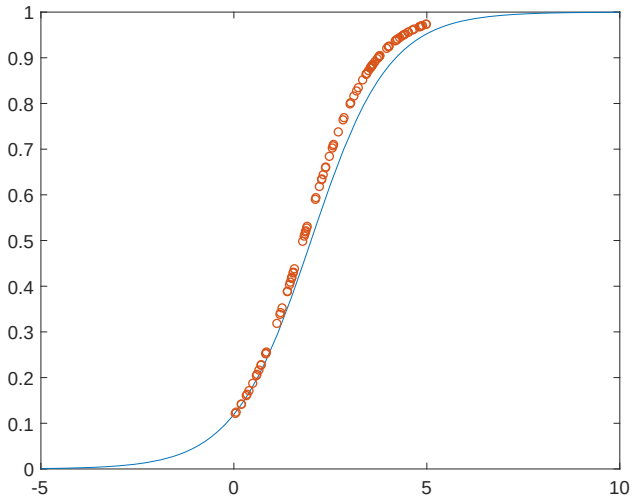

圖 6: 模擬資料的羅吉斯迴歸配適情形

4. 比較羅吉斯迴歸與線性判別分析（LDA）之異同。

5. 當自變數超過 1 個時，羅吉斯迴歸的估計結果必須作適當的評估，依所估計參數的大小檢定其顯著性 (Significance)，請試著以實際資料 [1]（請從網站下載）估計所有的參數，並評估哪一個變數該被去除。

6. 將範例 4 擴展到兩個自變數，即線性模式變爲

$$f(\mathbf{x}) = \beta_0 + \beta_1 x + \beta_2 x_2$$

其中參數 β_0, β_1 與 β_2 自訂。試著模仿範例 4 的做法，模擬兩個自變數的資料，並估計參數值 β_0, β_1 與 β_2。

7. 範例 4 展示模擬資料的邏吉斯迴歸模型參數估計。請針對樣本數 N=50, 100, 500, 1000，各模擬 1 萬筆資料並計算相對應的參數估計值共 1 萬個，計算每個參數估計值的平均數、標準差（SEE: Standard Error of Estimators ）及 95% 的覆蓋機率（CP: Coverage Probability）。

References

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

[2] 陳順宇，迴歸分析-三版，華泰書局。

[3] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization, Prentice Hall