群組分類
Nearest Neighbor Method

last modified May 29, 2008

對空間中的群組資料作分組切割, 在應用上很常見, 因此衍生的方法也很多。上一個單元採用線性迴歸的模式, 並運用最小平方法確立模型的參數。這個方式雖簡單, 但是切割的效果往往隨著資料來源的佈散愈趨緊密而顯得「心有餘而力不殆。」這個單元將面對相同的問題提出另一個簡單的方式, 看看從不同的思維模式是否能得到比較令人滿意的結果。

從二度空間的資料來看, 上個單元線性迴歸模式試圖在交錯的資料群中畫出一條分界線, 將空間一分爲二, 進而確立群組在空間中的的範圍。這條線切的好不好, 影響了後續做群組判別的準確性。這個單元從「可能性、」「機率論」觀點作爲切入點, 多了些假設, 卻要看看效果如何。

本章將學到關於程式設計

群組資料的繪製技巧、各種等高線圖(contour plot) 的繪製、排序資料的索引技巧及亂數資料的產生技巧。

⟨本章關於 MATLAB 的指令與語法⟩

指令:sort, find, contour

1 背景介紹:Nearest-Neighbor Method

有時候我們對資料的 (或可能的) 來源並非一無所知, 但是迴歸模式採用的最小平方法, 並沒有充分利用資料本身的訊息。從資料的變異性, 我們希望資料的來源或資料的本身可以提供足夠的訊息, 做爲新資料群組判別的依據。這樣的需求把這個問題帶進「機率」的範疇來解決。假設 Y 及 $f(X)$ 代表輸出變數與輸出預測值, [1] 其中 $X \in R^p$ 表示 p 個輸入變數。我們期望輸出值與預測值的誤差愈小愈好, 如果輸入變數 X 與輸出變數 Y 的聯合機率密度函數 $Pr(X,Y)$ 已知的話, 這個問題可以寫成:

$$\min_{f(X)} E_{XY}\left[(Y - f(X))^2\right] \tag{1}$$

也就是找一個輸入與輸出變數間的關係式 $f(\cdot)$, 使得輸出值 Y與其預測值 $f(X)$ 差的平方期望值越小越好。有別於不論機率特性的「最小平方法 (least squares)」, 這個方法的名稱也有個"squares"叫做「Minimum Mean Squares」。在已知樣本值 $X = \mathbf{x}$ 的條件下, 經過一番推導之後(作業1), 它的最佳解如下:

$$y = \hat{f}(\mathbf{x}) = E_{Y|X}\left(Y|X = \mathbf{x}\right) \tag{2}$$

其中 X, Y 代表輸入輸出變數,$\mathbf{x}, y$ 表示輸入值及輸出的預測值。式 (2) 說明當輸入值爲 $\mathbf{x}$ 時, 最佳的輸出預測值爲輸出變數的「條件式均值(conditional mean)」。

接下來的問題是如何計算 $y = E_{Y|X}\left(Y|X = \mathbf{x}\right)$? 期望值的表示法是理論上的東西, 要如何落實到實際的應用呢? 假設不知道機率密度函數 $Pr(Y|X)$, 如何得到這個期望值呢?

一般我們會使用平均數來估計這個期望值。譬如下面是個不錯的估計式

$$\hat{y} = Ave(y_i|X = \mathbf{x}) \tag{3}$$

[1]函數 $f(\cdot)$代表輸入與輸出間的關係, 譬如上個單元的迴歸模式 $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$。

其中 $Ave(\cdot)$ 代表求平均值。

這個估計式還是面臨實際的困難:

- 當$X = \mathbf{x}$ 是從已知的輸入資料中取得, 通常資料同爲 $\mathbf{x}$ 的筆數不多, 多半都只有一筆資料, 其輸出的樣本平均數估計並無意義。[2]
- 當$X = \mathbf{x}$是需要預測輸出值的新資料時, 更沒有計算輸出樣本平均數的可能。

下面這個輸出預測值定義舒緩了這些困擾

$$\hat{y} = Ave\left(y_i | \mathbf{x}_i \in N_k(\mathbf{x})\right) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y_i \tag{4}$$

從已知的資料中找到 k 個最靠近 $\mathbf{x}$ 的資料(這是 $N_k(\mathbf{x})$ 的意義), 將這些鄰近資料所對應的的 y 值平均起來作爲「條件式均值」的估計, 這個方法叫做 Nearest-Neighbor method。

目前爲止所提到的輸出變數並不侷限任何型態, 但若輸出變數爲群組屬性的類別資料時, 如式 (1) 的「Minimum Mean Squares」問題可以寫成,

$$\min_{g(X)} E_{XG}\left[L(G, g(X))\right] \tag{5}$$

由於是類別資料的關係, 其輸出群組 G 與預測群組 $g(X)$ 的誤差以「Loss function」$L(G, g(X))$ 取代原先的平方差。當 $L(G, g(X))$ 定義爲

$$L(G, g(X)) = \begin{cases} 0 \text{ if } G = g(X) \\ 1 \text{ if } G \neq g(X) \end{cases}$$

式 (5) 的最佳解爲

$$\hat{g}(X) = g_k \text{ if } Pr(g_k | X = \mathbf{x}) = \max_{g \in G} Pr(g | X = \mathbf{x}) \tag{6}$$

[2]當然也可以嘗試自 $P(Y|X = \mathbf{x})$ 中抽取適量的樣本並計算其均值。

其中 g_k 代表第 k 個群組 (group),G 是所有群組的集合。這個結果說明: 當輸入值爲 $\mathbf{x}$ 時, 其所屬群組的Minimum Mean Squares 預測爲

「在所有的群組中, 群組機率密度函數在 $\mathbf{x}$ 處的值最大者」

這也稱爲 Bayes classifier。[3] 式子(2)(6) 的推導將在課堂上說明, 或是參考[1]。不管哪一種輸出的型態, 這裡都使用到「後驗機率」(posterior probability) 的觀念, 也就是當給定輸入變數 $X = \mathbf{x}$, Y(或G) 值的可能性 (機率)。Bayes 之名也來自於此。

群組判別: 從式(6) 中似乎看不出一個明顯的「分界線」方程式, 無法像前一個單元那樣根據方程式畫出一條分界線, 更何況機率密度函數$Pr(G|X = x)$也是未知。不過如同前一個單元將迴歸模式應用在類別資料上, 當假設兩個群組的輸出爲 0 (群組1) 與 1 (群組2) 時, 式 (4) 可以當作式 (6) 的估計式, 並配合下列的群組判別式,

$$\mathbf{x} \in \begin{cases} g_1 & \text{if } \hat{y} \leq 0.5 \\ g_2 & \text{if } \hat{y} > 0.5 \end{cases} \tag{7}$$

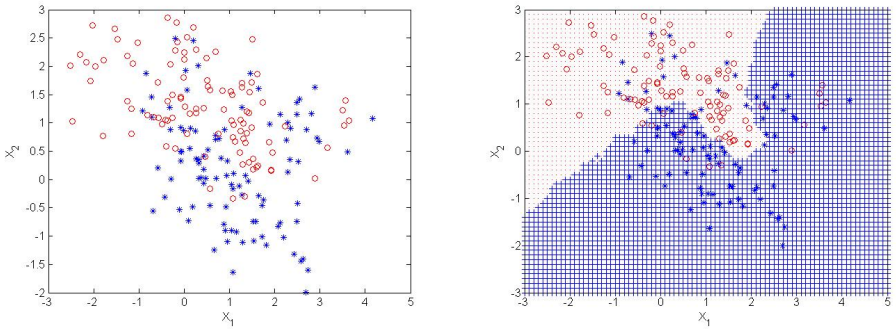
式 (4) 的 $\hat{y} \leq 0.5$ 相當於 式 (6) 的 $Pr(g_1|X = x) > Pr(g_2|X = x)$。這個方法的表現到底如何? 比起前面的線性迴歸模式好或壞? 有什麼缺點? 有什麼限制? 只要把程式寫出來, 拿幾組資料實際來測試一番便知分曉。

2 練習

範例1: 舉模擬資料 mix.mat 爲例 (從網頁下載的, 資料來源 [1]), 按前一個單元的繪圖方式, 畫出如圖1(a)所示的群組資料。圖 (b) 便是利用 Nearest-Neighbor method 切割空間的結果。試著按下列的說明複製圖1(b)。

由於 Nearest-Neighbor method 並沒有定義出一個分界線的方程式, 無法畫出這條線, 不過可以如圖1(b) 的作法, 在一定的空間範圍內, 將空間等份成格子狀 (grids),

[3]這個結果看起來似乎是「廢話,」因爲當後驗機率 $Pr(g|X)$ 未知時, 如何知道當 $X = \mathbf{x}$ 時的群組機率密度函數值呢? 不過這卻是很重要的理論, 它說明在期望預測值與實際值的誤差最小的前提下, 最好的預測值是來自在 $X = \mathbf{x}$ 處有最大後驗機率的群組。



(a) (b)

圖 1: 群組資料與空間分割

每個格子的座標代表一個資料值 $\mathbf{x}$, 將每一筆資料當作新資料一樣的拿出來判斷其類別(大部分的格子點都不在已知的資料裡), 依式(4) 與 (7), 爲每個位置劃上不同的符號做爲群組的區別。格子狀的粗細與組別符號決定了畫出來的感覺, 不妨試試看不同的格子密度與群組符號或顏色。

式 (4) 的估計式中需要找出「靠近 $\mathbf{x}$ 的 k 個已知資料」, 這個「靠近」的測量方式可以採用 Euclidean Distance。假設 $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}$ 爲 N 個已知資料, $\mathbf{x}$ 爲空間中某個待判別群組的資料, 程式中需要計算 $\mathbf{x}$ 與所有已知資料的距離, 再從中選取最靠近的 k 筆資料, 最後再將這 k 筆資料的群組值 (0或1) 平均起來, 即爲式 (7) 中的 $\hat{y}$ 值, 程式片段如下

```
% 參數準備
interval=0.1;%格子點距
k=15; % k nearest points
xmin=-3; xmax= 5;ymin=-2;ymax= 3; % X 與 Y 的範圍
gx=xmin:interval:xmax;%格子 X 座標
gy=ymin:interval:ymax;%格子 Y 座標
B=zeros(length(gx),length(gy));%存放相鄰 k 個點的 y 值平均
for i=1:length(gx)%計算距離並取前 k 個 y 值的平均
      for j=1:length(gy)
            d=(x1-gx(i)).^2 + (x2-gy(j)).^2;%計算每個格子點與所有樣本的距離
            [dd,I]=sort(d);% d 值由小到大排序
            B(i,j)=mean(y(I(1:k)));%取出最相鄰 k 個點的 y 值平均
            if B(i,j) >0.5%決定群組並描繪其群組符號
               plot(gx(i),gy(j),'-r')
            else
               plot(gx(i),gy(j),'-g')
            end
       end
end
```

程式中的 x1,x2 與 y 變數分別代表樣本資料與其群組別。

範例2: 運用繪圖的技巧爲圖1(b) 的兩個顏色的邊界劃上線條, 如圖 2。

雖然 Nearest-Neighbor method 並未定義出分界線方程式, 不過運用繪圖的技巧, 仍可以巧妙的描繪出這條線。其作法是先將前一個範例所提到的每一個格子點所計算出來的 Y 值平均, 依式 (7) 轉換成組別資料 (0或1), 如以下程式片段

```
%呈上述程式
B(B > 0.5)=1; % group 2
B(B <= 0.5)=0; % group 1
```

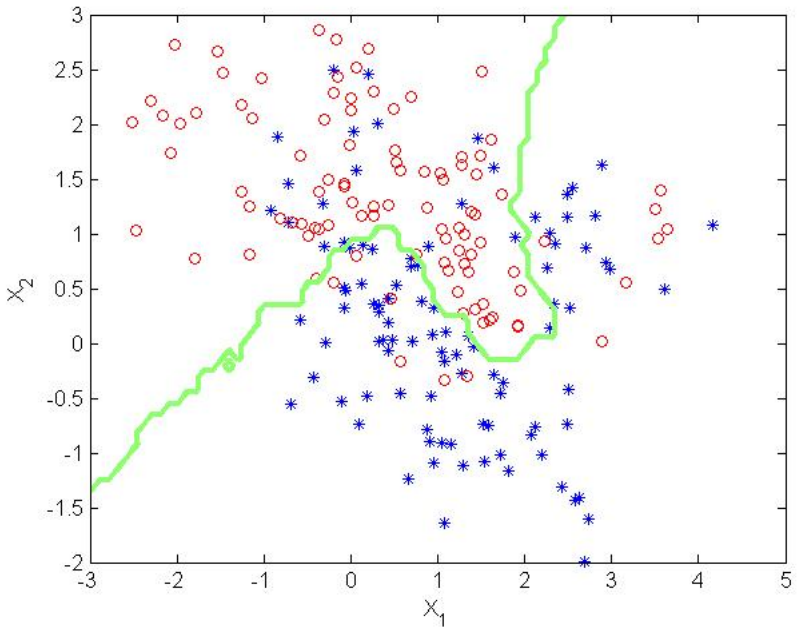


圖 2: 分界線的繪圖技巧

根據這些組別資料 (B 值) 與其座標值 (X-Y 值) 繪製 contour 的等高線圖, 或只畫 0.5這條線。程式片段如下

```
%呈上述程式
[X,Y]=meshgrid(xmin:interval:xmax,ymin:interval:ymax);
contour(X,Y,B',0.5) %注意矩陣 B 的轉置以配合 XY 矩陣
```

請先瞭解在 Matlab 裡面 contour 的用法, 它同時也是繪製不規則函數圖型的利器。下列指令可供參考

```
contour(X,Y,Z)          %多條等高線圖
contour(X,Y,Z,0.5)      %繪製 1 條 Z 值爲0.5的等高線
contour(X,Y,Z,[0 1])    %繪製 2 條 Z 值爲0,1的等高線
```

範例3:Nearest Neighbor method 對空間的切割並不一定是連續的, 試試看不同的 k 值 (譬如 $k = 15, 14, \cdots, 1$), 觀察空間被切割的情況。

當 $k = 1$ 時, 可說是依據已知資料對空間的切割到了極致, 如圖3所示。與前一個單元的迴歸直線對照, 恰似兩個極端。其他的分割方式都介於這兩者之間。

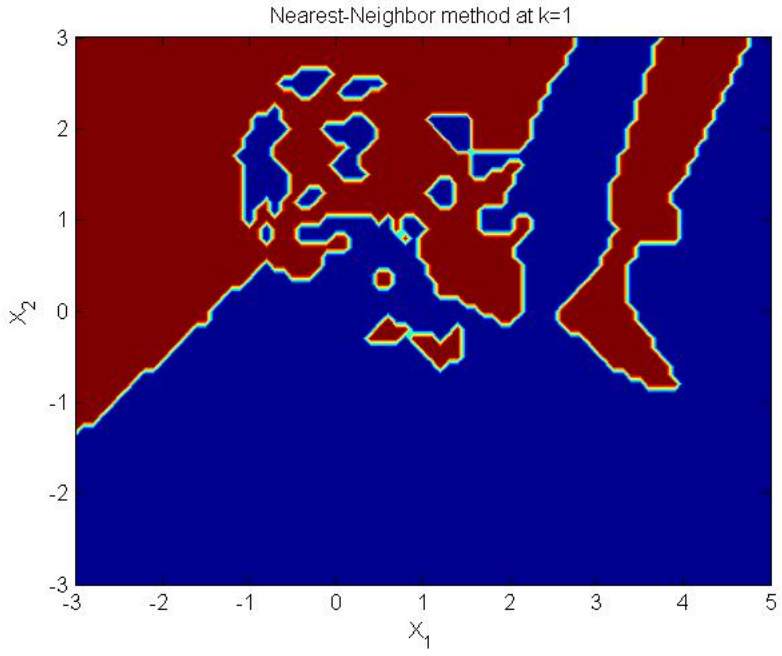


圖 3: k=1的空間分割圖

圖3的繪圖方式相當具有視覺效果, 可以兩種方式達到, 其一

```
contour(X,Y,Z,'Fill','on')
```

或是以繪圖編輯器 (Property Editor) 來注入顏色。

3 觀察

1. Nearest-Neighbor method 做出來空間切割並非直線, 往往都是不規則形狀。換句話說, 分界線應該是很複雜的非連續函數。

2. k 值的選擇對於空間的切割有重大的影響, 對新資料作分類預測時當然也會不同。試著嘗試不同的 k 值, 看看切割出來的空間有何不同?

3. 採用 Nearest-Neighbor method 並不是在找一條空間的分界線, 相反的它是比較區域型 (範圍較小) 的, 在一個小區域的空間中找界線, 這個區域的大小由 k 及資料的分佈情形來決定。

4. 如果將估計值 (4) 改爲 $\hat{y} = Median\,(y_i|\mathbf{x}_i \in N_k(\mathbf{x}))$, 結果會變好還是變壞? 做做看。

5. 式 (4) 的估計值將鄰近點視爲等值影響, 對某些區域來說並不「公平」, 如果依距離之遠近改變其權重, 是否能提供更平滑的分界線? 請試試看依常態分配的機率值來權衡其重要性。

4 作業

1. 推導出式 (2), 其中

$$\begin{aligned} E_{XY}\left(Y - f(X)\right)^2 &= \int\int (Y - f(X))^2 Pr(X,Y) dX dY \\ &= \int\int (Y - f(X))^2 Pr(Y|X) Pr(X) dX dY \\ &= E_X E_{Y|X}((Y - f(X))^2|X) \end{aligned}$$

2. 推導出式 (6), 其中

$$E_{XG}\left[L(G, g(X))\right] = E_X \sum_{k=1}^{K} L(g_k, g(X)) Pr(g_k|X)$$

3. 以 mix.mat 資料爲例, 利用本單元的方法, 依 k=15,10,5,1 各畫出分界線 (共四張圖)。

4. 任選一個 k 值, 按「觀察 (4)」的建議採 median 取代 average, 畫一張以顏色及符號分組的圖, 另一張只畫分界線。

5. 不同的 k 值其分割組別的能力不同。要判斷分割的精準度, 可以將所有的擬合值與原始值做比較, 看看整體的誤差 (譬如所有誤差值的平方合) 就可以看出精準度。請根據 k 值的不同, 從 k=15,10,5,1 分別計算其誤差平方合, 並畫一張圖來表示其趨勢。

6. (optional) 從上題的精準度問題來看, 實際上精準度愈高並不代表其分辨新資料的能力愈好, 針對不同的資料, 會有一個最佳的 k 值, 其對新資料的分辨能力最好。請自行產生一組資料 (俗稱 training data) 來決定分界線, 再產生另一組資料作爲測驗這個分界線的測試資料 (test data), 一樣採誤差平方合來作爲優劣的依據。畫一張圖含兩條線, 一條是根據 training data 所產生的誤差, 另一條是根據 testing data 產生的。橫軸都是 k 值 (譬如 k= $15, 14, \cdots, 1$)。請注意, 這裡的資料可以依下一題的方式產生。

7. k- Nearest-Neighbor classifier(式 (4)(7)) 其實是 Bayes classifier(式 (6)) 的近似版, 當條件式機率分配 $P(X|G)$ 已知時,Bayes optimal boundary 可以被準確的計算出來, 如圖 4所示。

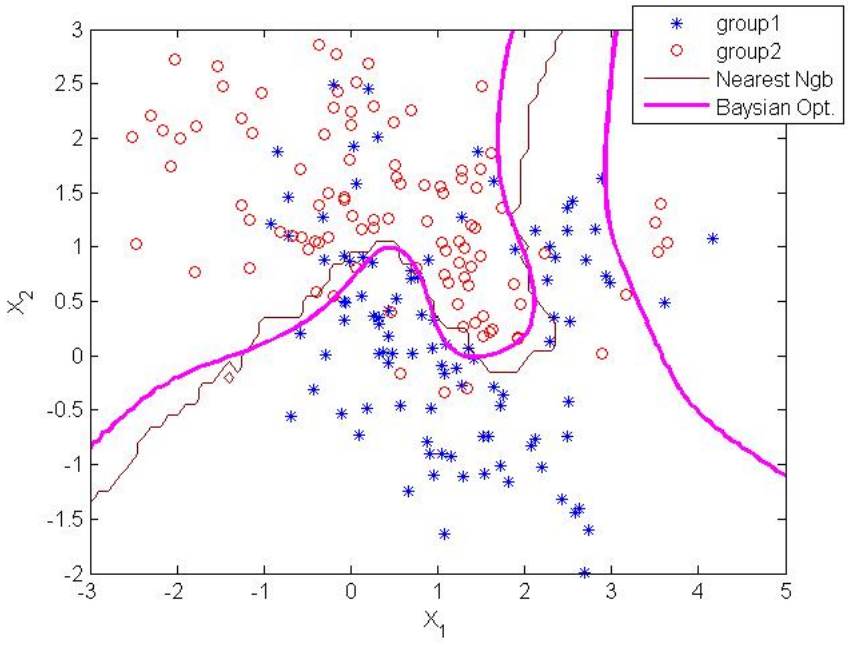


圖 4: Bayes optimal boundary

計算 Bayes optimal boundary 的前提是隨機資料的產生機率密度函數 (generating density) 必須已知。圖1(b)的資料 (mix) 來自兩組混合常態 (Normal Mixtures) 的母體, 每組各由 10 個 bivariate 常態母體組成, 如

$$\text{第一組} \quad : \quad P(X|G=1) = \sum_{k=1}^{10} \alpha_k \phi(X, \mu_k^1, \Sigma_k)$$

$$\text{第二組} \quad : \quad P(X|G=2) = \sum_{k=1}^{10} \alpha_k \phi(X, \mu_k^2, \Sigma_k)$$

其中, 混合比例 α_k 皆爲 $\frac{1}{10}$, 共變異矩陣 Σ_k 皆爲 $I/5$, 第一組常態分配的 mean 由一個 bivariate 常態 $N((1,0), I)$ 產生 10 個, 另一組由 $N((0,1), I)$ 產生 10 個。參考資料 mix 亦包含這些 mean.mat 的樣本 (變數名稱 x_mean)。Bayes optimal boundary 爲

$$Pr(G = g_1|X) = Pr(G = g_2|X) \tag{8}$$

利用貝氏定理並將上述的假設代入 (8), 可以得到這條線的數學式子, 用 MATLAB 畫等高線圖, 即爲圖 4 所示。從式 (8) 的角度來看,Nearest-Neighbor method 對兩個群組的分割可說是接近完美的演出。試著將這條理論上最好的分界線畫出來。

參考文獻

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