

程式語言期末專題書面報告

數學三 B 102201024 王鈺鎔、102201501 李子婕

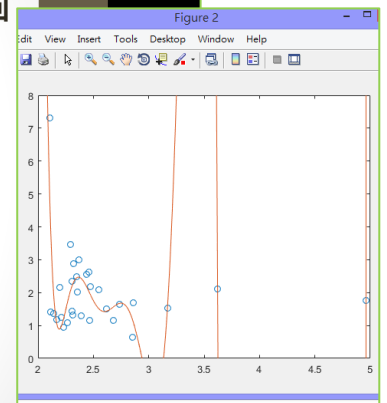
一、目標

找一個數學 Model 來校正 Prototype 的數據，使它的數據盡量符合 Moor 的數據。

二、數學方法與模型建構——演算法介紹

✧ 一開始我們用 polyfit 方法實作。

- 想法：假定五個向量間線性獨立，把五個向量分開實作，分別透過polyfit的方式，找到最符合的方程式。
- 缺點：
 - 五個向量彼此關聯，分開實作增加誤差
 - 多項式因幾個差異大的點，多轉幾個彎
 - 無法找出那個多項式函式是什麼



✧ 但是因為誤差太大，所以我們著手研究 Multivariate General Linear Model。

$$[y_1 \ y_2 \ y_3 \ y_4 \ y_5] = [1 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5] \times \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} & \beta_{15} \\ \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} & \beta_{25} \\ \beta_{31} & \beta_{32} & \beta_{33} & \beta_{34} & \beta_{35} \\ \beta_{41} & \beta_{42} & \beta_{43} & \beta_{44} & \beta_{45} \\ \beta_{51} & \beta_{52} & \beta_{53} & \beta_{54} & \beta_{55} \\ \beta_{61} & \beta_{62} & \beta_{63} & \beta_{64} & \beta_{65} \end{bmatrix}$$
$$y_1 = 1\beta_1 + x_1\beta_2 + x_2\beta_3 + x_3\beta_{14} + x_4\beta_5 + x_5\beta_6$$

常數項

✧ 這是我們後來實作的數學式子。

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ \vdots & \vdots \\ y_{n1} & y_{n2} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{bmatrix} \begin{bmatrix} \beta_{01} & \beta_{02} \\ \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \\ \beta_{31} & \beta_{32} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \\ \vdots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} \end{bmatrix}$$

Moor **Prototype** **Beta(model)**

- ✧ 找好方法後開始程式撰寫。一開始先透過助教所提供的函式，把數據轉成 31*5 的矩陣。
- ✧ 31 : 31 筆數據。5 : 5 維向量。

數據轉換(說明)

- 按照助教提供的程式碼，使用 **czt_ldf** 函式
- 結果:
 - 分別得到 6 個 31x5 的矩陣
 - dataMoorarray_finger
 - dataProtoarray_finger
 -
 - 將prototype的皆擴增為31x6的矩陣
 - (左項補1)

- ✧ 以下是轉換過程的部分程式碼。

數據轉換(程式碼)

```
load('sampleRate.mat');
moor = resample( loaddata.LDF_Moor, 10, rate );
proto = resample( loaddata.LDF_Prototype, 10,
rate );
moorEnergyS= czt_ldf( moor, 10, 1024 );
protoEnergyS = czt_ldf( proto, 10 ,1024);
for j=1:1:5
    dataMoorarray(i,j)=moorEnergyS(j);
    dataProtoarray(i,j)=protoEnergyS(j);
end
end
disp(dataMoorarray);
disp(dataProtoarray);
```

- ✧ 以下是轉換後 Moor 的矩陣樣子。

數據轉換(Finger data example)

```
dataMoorarray_finger = 1.0e+05 * [
    0.7887  0.4096  0.1954  0.0613  0.3204;
    6.5008  3.0269  0.7524  0.3171  0.9338;
    2.6806  1.8142  0.6201  0.2436  0.8309;
    9.2410  4.6035  2.8144  0.7603  0.9441;
    4.2486  3.1000  1.2534  0.3196  0.8194;
    .....
    5.5945  2.8807  0.9957  0.5222  1.0065;
    5.5344  4.0193  1.0763  0.5905  1.1023;
```

$$\begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \\ \vdots & \vdots \\ y_{n1} & y_{n2} \end{bmatrix}$$

✧ 以下是轉換後 Prototype 的矩陣樣子。

數據轉換(Finger data example)

```
dataProtoarray_finger = 1.0e+13 * [  
1 3.0101 1.8053 0.5670 0.2425 0.1169;  
1 2.5392 1.4485 0.4199 0.1851 0.0900;  
1 2.5237 1.4838 0.4595 0.1873 0.0912;  
1 3.3390 1.9361 0.5951 0.2499 0.1122;  
1 2.7114 1.5779 0.4899 0.2089 0.1032;  
.....  
1 2.1468 1.3590 0.4090 0.1614 0.0800;  
1 2.5317 1.4950 0.4535 0.1950 0.0883;
```

$$\begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{bmatrix}$$

✧ 接著是利用 Matlab 的內建函式來建構 beta 矩陣，也就是此專題所要得到的目標。

✧ (該函式用紅線框標示)

建構模型

```
[beta_finger,sigma_finger,E_finger,V_finger] =  
mvregress(X_dataProtoarray_fingercell,Y_dataMoorarray_finger);  
%% 將beta從 1 x 30矩陣轉為 6 x 5的矩陣 %%  
[a,b] = size(beta_finger)  
new_beta_finger = zeros(6,5);  
i = 1;  
for j = 1:1:30;  
    if mod ( j ,5 )==0  
        new_beta_finger(i,5) = beta_finger(j);  
        if j ~= 30  
            i = i+1;  
        end  
    else  
        new_beta_finger(i,mod(j,5)) = beta_finger(j);  
    end  
end  
end
```

✧ 以下是 beta 矩陣的樣子。

建構模型(Finger data example)

```
beta_finger = (1.0e-06) * [  
-0.0063 0.0024 -0.0037 0.0041 0.0042;  
0.0207 0.0078 0.0012 0.0057 0.0146;  
0.1356 0.1239 0.0537 0.0067 0.0150;  
-0.3915 -0.3447 -0.1849 -0.0126 0.0039;  
0.1954 0.0042 0.1287 -0.0864 -0.3310;  
-0.6732 -0.2275 -0.0862 -0.0224 0.0946];
```

$$\begin{bmatrix} \beta_{01} & \beta_{02} \\ \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \\ \beta_{31} & \beta_{32} \end{bmatrix}$$

- ✧ 接著就是轉換的部分，把 Prototype 的 array 乘上 beta array，就可以得到校正矩陣，也就是被修正過的 Prototype array。又為了避免有負數，所以又多了一個步驟，把校正矩陣加上絕對值。
- ✧ 為了減少誤差，我們分別為三種數據(Finger、data1、data2)個建立了一個 Model。

```
• TdataProtoarray_finger =  
  NdataProtoarray_finger * beta_finger ;  
• TdataProtoarray_finger =  
  abs( TdataProtoarray_finger ) ;
```

- ✧ 實作 Multivariate Generral Linear Model 之後我們發現它對於此專題的優缺點。

- 優點:
 - 有處理五種資料的相依關係
 - Matlab提供完整的函式
- 缺點:
 - 五種資料的相依關係可能不只是線性的
 - 誤差不夠小(依然無法準確表示五種資料間的相依關係)

三、 GUI 介面說明、執行方式

★ axes1, 2 : 印 Prototype 和 Moor 的訊號圖。

★ axes5, 6, 7 : 印 Prototype 和 Moor 以及 Moor, Transform 能量圖。

本來要畫三個一起的能量圖，但因為 Prototype 的數量級(次方)和其他兩個差太多，會導致其他兩個值小到看不見，所以最後選擇畫 Moor, Transform 兩個數據一起的能量圖。

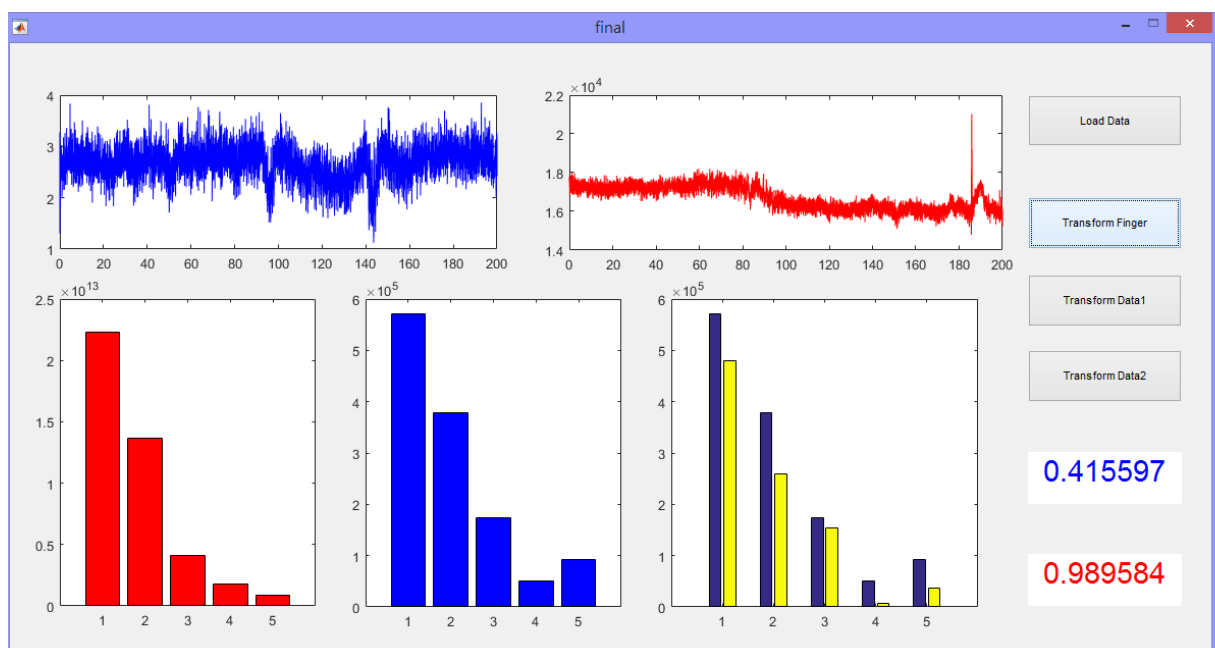
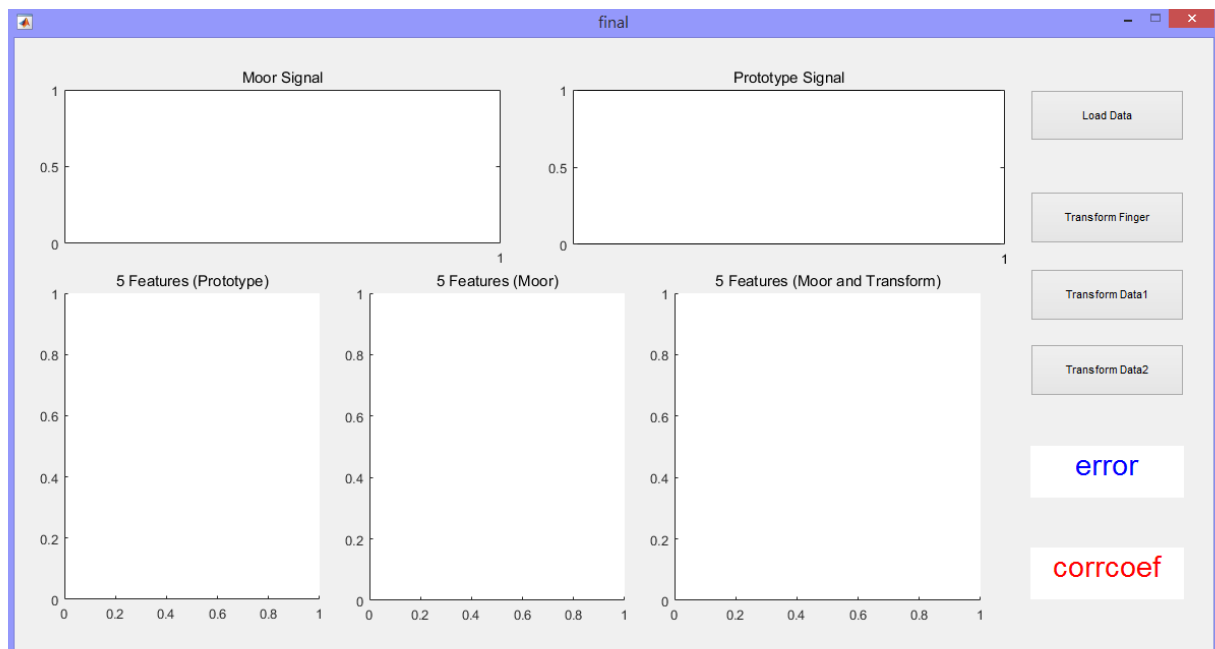
★ 四個按鈕：

一個為 Load Data：讀檔、轉換、印訊號圖、印 Moor 和 Prototype 能量圖。

其他三個分別為三種 Model：Transform Finger、Transform Data1、Transform Data2。

按下對應轉換按鈕會畫完 Moor, Transform 兩個數據一起的能量圖，印出誤差和相關係數。

★ 介面



四、實驗結果、心得

助教公佈實驗數據後，我們利用每周二一起到系館寫程式，一開始我們花了很多時間用 Polyfit 方法來實作，但我們發現即使調高方程式的次方也不會減少誤差，如同前面提到的缺點，五個向量彼此關聯，分開實作會增加誤差，另外，多項式為了配合幾個差異大的點而多轉幾個彎，並且無法找出那個多項式函式是什麼，所以我們決定換個方法實作。

後來我們利用 Matlab 的 help 功能看了很多方法，最後我們決定使用 Multivariate General Linear Model 這個迴歸方法，雖然最後的結果還是不太理想，因為依然無法準確表示五種資料間的相依關係，但已經比 Polyfit 好許多。

做專題過程中遇到的困難主要是花了一番功夫才好不容易看懂 Multivariate General Linear Model，另外是在實作 GUI 介面時因為不熟悉語法一直卡關，但還好最後都順利的解決了。原本很擔心 Matlab 版本問題無法順利 demo，因為我們用的 Matlab 版本開學校電腦、助教電腦都不一樣，但還好最後圖都有順利畫出來。

因為當時來不及加上「相關係數」，我們在 demo 後補上，然後加在程式與書面報告中。

五、工作分配

作期末專題的過程，我們都是約時間一起去系館寫程式，所以幾乎都是一起。

其中可能有工作分配的是，李子婕做投影片，王鈺鏐上台報告，

基本上所有部份我們都是約週二一起完成，以及利用星期三程語下課後的一小時空檔作報告。