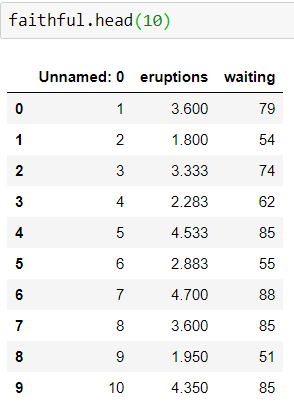
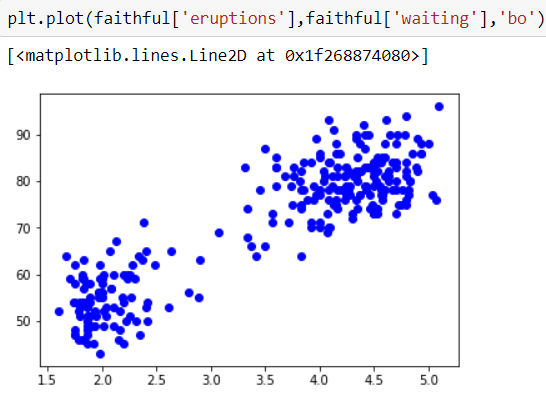
統計計算與模擬( EM )

目標 : 以EM演算法將Old Faithful Geyser Dataset 分群並估計母體參數

資料介紹 :

Old Faithful Geyser Dataset具272筆間歇泉資料，共有兩個變數

1. Eruptions: 噴發時間(秒) 2. Waiting: 噴發間隔時間(分鐘)，從散佈圖可看出資料大致上可分為兩群，因此可以決定k=2，並假設資料服從二元常態分配，在此假設之下，執行EM演算法分群並估計參數。

EM ( Expectation Maximization ) 簡介 :

EM是透過迭代找到一組分群及參數可使概似函數( likelihood function )最大化的演算法，已知我們手中有一筆觀測資料，生成一筆資料，又稱作“class label ”，可視作每個樣本的標籤，告訴我們這筆樣本屬於第類分配，再給定群數及起始參數，過程如下 :

首先以觀測資料及起始參數估計label ，這步驟稱作“E-Step”(分群)

再根據更新參數，這步驟稱作 “M-Step”

重覆E-Step及M-Step，直至概似函數收斂

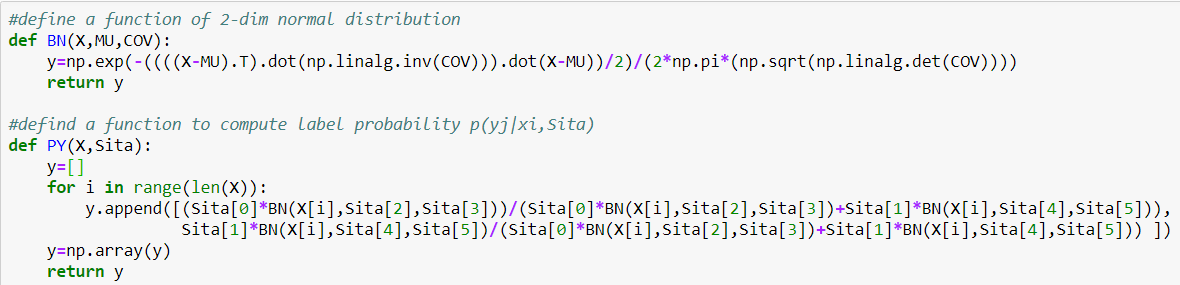
即是 : 以參數分群 🡪 以分群估計參數 🡪 以參數分群 🡪 … 反覆迭代

執行EM 演算法 :

在已知群數k=2、分配假設為二元常態的情況下，設定參數

;其中為兩群出現的比率，首先第一步就是計算在給定條件下， 機率

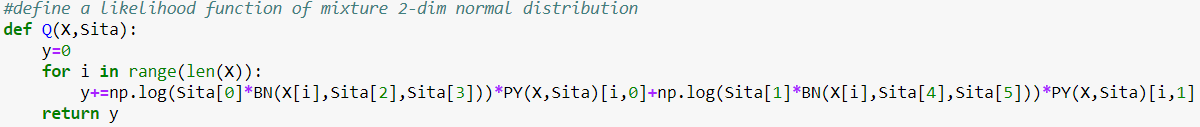
(其中皆為二元常態分配的)



函數BN是二元常態分配的，而函數PY()的結果會是((),…, ())的陣列

由於EM演算法的目的是求出可使概似函數最大化的參數，因此在迭代之前須計算概似函數，若概似函數，則停止迭代。

因為概似函數在樣本數多的情況下會太接近0，不好比較，因此在這裡將概似函數取對數 :



建立完概似函數後，建立每次迭代各參數的估計量

根據弱大數法則，，

因此如果我們要估計第j分配在母體中的比率，可用 :

( hard assigned )

在這裡提一下EM演算法在迭代過程中的分群是採用soft assigned

soft assigned並不會指定屬於特定的哪個分配，而是根據屬於這個分配的機率是多少來給予加權。而hard assigned就是直接指定屬於哪個分配，

K-means就是採用這樣的方法分群

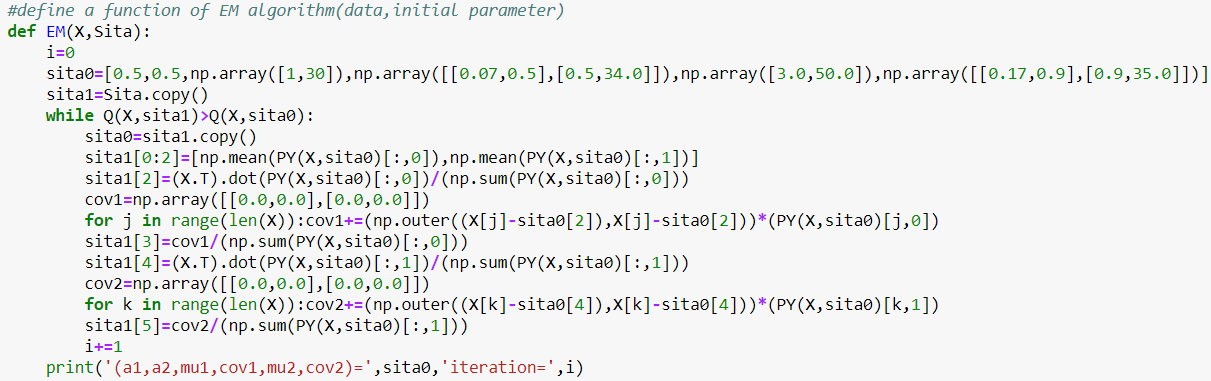
兩者比較如下圖:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | EM | | K-Means | |
|  |  |  |  |  |
|  | 0.7 | 0.3 | 1 | 0 |
|  | 0.9 | 0.1 | 1 | 0 |
|  | 0.2 | 0.8 | 0 | 1 |
|  | 0.6 | 0.4 | 1 | 0 |
|  | 0.3 | 0.7 | 0 | 1 |

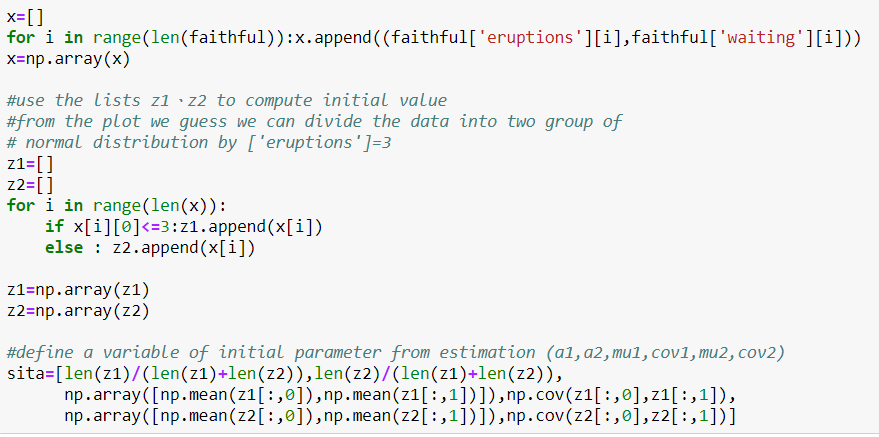
EM會用soft assigned的原因在於好的分群要有精確的分配參數資訊，而好的參數估計又需要好的分群，可以得知在迭代過程中，我們其實並不知道我們的分群是否正確，若以錯誤的分群來估計參數，顯然會得到錯的估計值，如果又用這錯誤的估計值再去更新分群，最後演算法完成的結果肯定差強人意，hard assigned就很可能造就這樣的結果，指定屬於其中一個分配，就一定得承擔屬於另一個分配的風險，相反地，soft assigned有將同時屬於兩個分配的可能考慮進加權當中，得到錯誤分群的可能就會大幅降低

根據soft assigned，參數的估計量為 :

有了估計量即可建立EM演算法 :

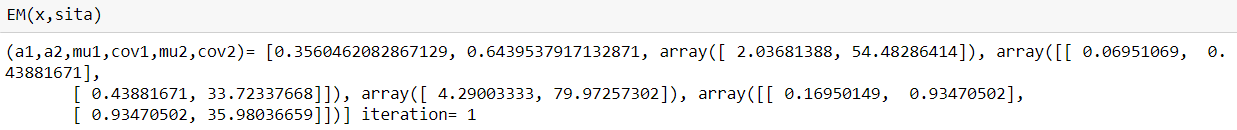


接下來給定參數起始值，第一組起始值是以”eruptions”=3切為兩個分群，並用python套件numpy中的函數直接進行估計，可以想像這一組起始值已經很接近MLE

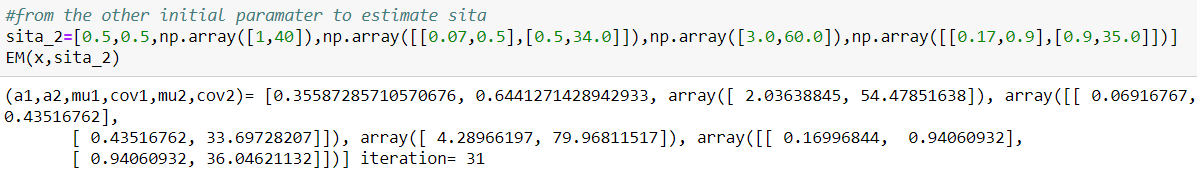


出來的結果

將第一組起始值執行EM演算法的結果如下，演算法只迭代一次即停止，結果也與起始值沒有太大的差異



接下來將第一組稍作改良，使它稍微偏離中心參數，並執行EM演算法



這筆較差的起始值迭代了31次才停止，但最後的結果與另一筆起始值的結果幾乎相同，直到小數點三位後才開始出現分歧，代表這個EM演算法的模型並沒有建立錯誤。這筆間歇性噴泉資料的分群工作到這邊告一段落，最後附上分群散佈圖，可看到我們用EM演算法清楚地將資料分成兩個群集。

