EM Algorithm Implement

Step 1.　初始化參數μ_k^t、Σ_k^t、π_k^t，分別代表第k群在迭代t的 mean、covariance matrix 與 occurrence probability，$\theta^t = \{\pi_k^t, \mu_k^t, \Sigma_k^t, k = 1,2,3\}$。

在初代$t = 1$，令μ_k^t為資料群中任意點，$\Sigma_k^t = I_{2\times 2}$，$\pi_k^t = \frac{1}{3}$

Step 2.　計算在已知群體分布下，新的參數μ_k^{t+1}、Σ_k^{t+1}、π_k^{t+1}

Step 3.　迭代 Step2 直到參數不變或變化量之和小於T (預設 0.001)

$$T > V = \sum_{k=1}^{3} norm(\mu_k^t - \mu_k^{t-1}) + norm(\Sigma_k^t - \Sigma_k^{t-1}) + norm(\pi_k^t - \pi_k^{t-1})$$

或是迭代直到所有資料點機率變化量之和小於T

$$T > V = \sum_{k=1}^{3} \pi_k^t f_k\left(x_i \mid \mu_k^t, \Sigma_k^t\right) - \pi_k^{t-1} f_k\left(x_i \mid \mu_k^{t-1}, \Sigma_k^{t-1}\right)$$

Step 4.　計算各點在各群分布的機率$\mathrm{Prob}(x_i \in c_k \mid x_i, \theta^t)$並顯示結果圖

Method

E-Step:

使用參數θ^t求每點x_i在c_k群分布的期望值

$$T_k^t\left(x_i, \theta^t\right) = \mathrm{Prob}(x_i \in c_k \mid x_i, \theta^t) = \frac{\pi_k^t f_k\left(x_i \mid \mu_k^t, \Sigma_k^t\right)}{\sum_{k=1}^{3} \pi_k^t f_k\left(x_i \mid \mu_k^t, \Sigma_k^t\right)}, \theta^t = \{\pi_k^t, \mu_k^t, \Sigma_k^t, k = 1,2,3\}$$

$f_k\left(x_i \mid \mu_k^t, \Sigma_k^t\right)$為 2-Dimension normal distribution probability density function

M-Step:

求新的分布參數

$$\pi_k^{t+1} = \frac{\sum_{i=1}^{n} T_k^t\left(x_i, \theta^t\right)}{\sum_{k=1}^{3}\left(\sum_{i=1}^{n} T_k^t\left(x_i, \theta^t\right)\right)} = \frac{\sum_{i=1}^{n} T_k^t\left(x_i, \theta^t\right)}{n}$$

$$\mu_k^{t+1} = \sum_{i=1}^{n} x_i [\frac{T_k^t\left(x_i, \theta^t\right)}{\sum_{i=1}^{n} T_k^t\left(x_i, \theta^t\right)}], k = 1, 2,3$$

$$\Sigma_k^{t+1} = \sum_{i=1}^{n} \{[(x_i - \mu_k^{t+1})(x_i - \mu_k^{t+1})^T][\frac{T_k^t\left(x_i, \theta^t\right)}{\sum_{i=1}^{n} T_k^t\left(x_i, \theta^t\right)}]\}, k = 1, 2,3$$

Program

主程式 hw6.m，每次迭代會秀出目前分群結果的 figure，並在 CMD 顯示收斂情形，收斂後會秀出過程影片。
收斂結果方程式如下

$$V = \sum_{k=1}^{3} norm(\mu_k^t - \mu_k^{t-1}) + norm(\Sigma_k^t - \Sigma_k^{t-1}) + norm(\pi_k^t - \pi_k^{t-1})$$

收斂情形分別顯示

$$norm(\pi_k^t - \pi_k^{t-1}) \;\; norm(\mu_k^t - \mu_k^{t-1}) \;\; norm(\Sigma_k^t - \Sigma_k^{t-1}) \;\; V$$

當V小於 0.001 則停止迭代運算並秀圖

主程式	hw6.m
計算$T_k^t(x_i, \theta^t)$	Estep.m
計算π_k^{t+1}	MstepWeight.m
計算μ_k^{t+1}	MstepMean.m
計算Σ_k^{t+1}	MstepSigma.m

影片

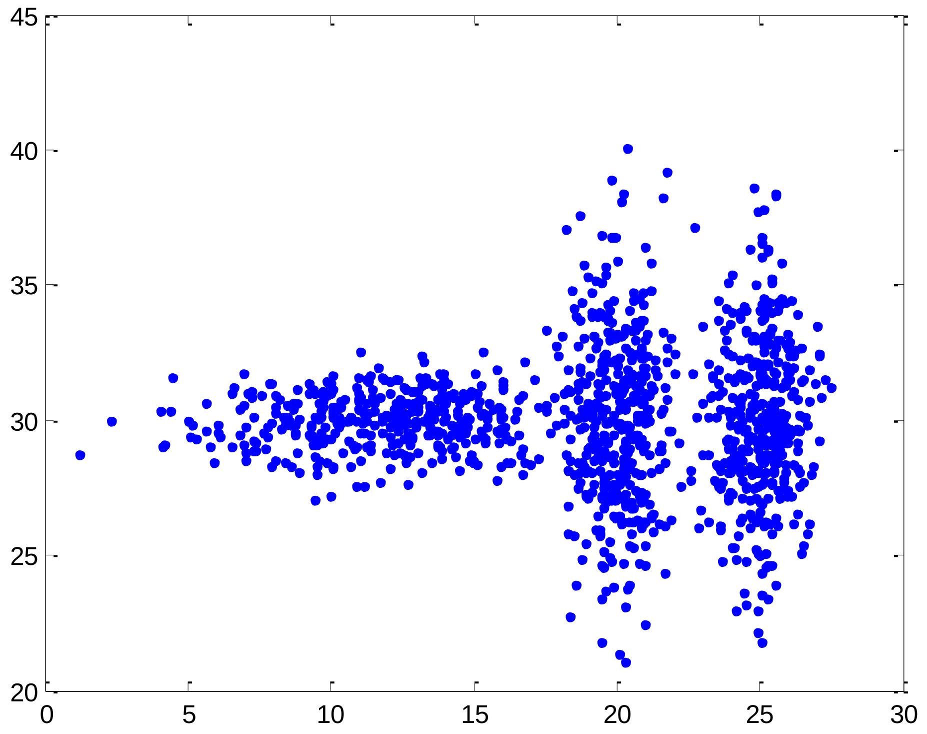
變數 a 儲存過程圖像，可使用 movie(a)播放
或是 movie2avi(a, 'result.avi', 'compression', 'None')儲存為 avi 檔案

Result

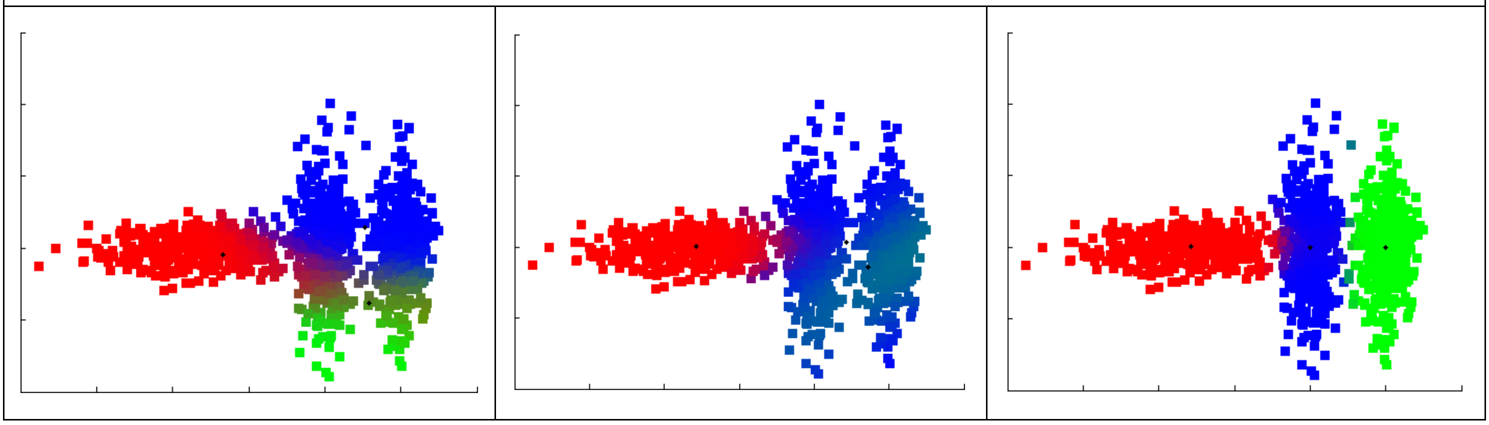
RGB Intensity 分別代表三群的 Probability $T_k^t\left(x_i,\theta^t\right)$，黑點為各群中心$\mu_k^t$

avi 影片與結果 mat 資料放在 result 目錄下

Data1

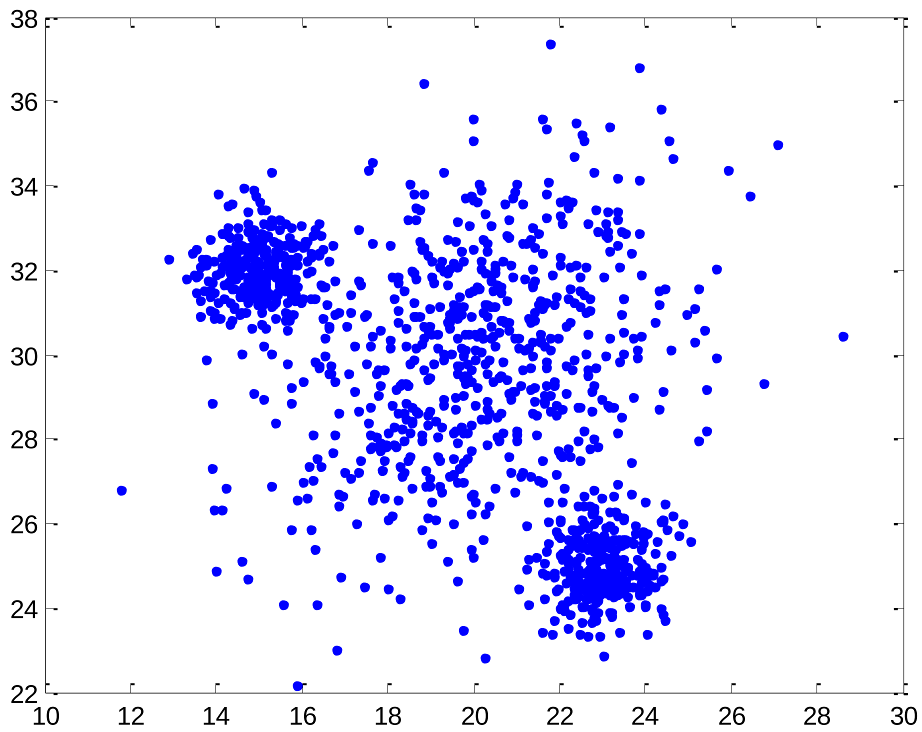


開始 ---> 收斂

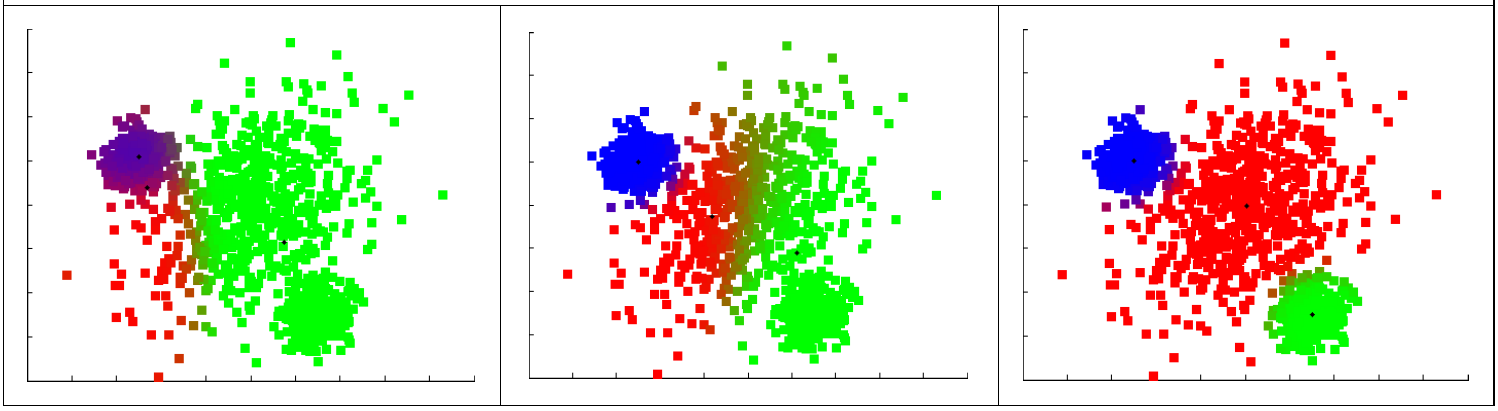


result1.avi & result1.mat

Data2



開始 ---> 收斂



result2.avi & result2.mat

Discussion

Initial μ_k^1會影響到收斂速度，若隨機取三點μ_k^1，很有可能都在非常接近的地方，會使得群體的μ_k^t移動緩慢，甚至在某些情況下會掉入意外的 Local optimum。若要加速收斂以及增加正確性，起始三點μ_k^1要盡量接近真實的群體中心並且散開。

為解決掉入 Local optimum 的問題，我們可以平行計算複數組μ_k^1並回傳最佳的分群結果。

平行運算M組μ_k^1，分群收斂後，加總各點與最可能 cluster mean 的 Euclidean distance D_M，取D_M最小的為最正確分群結果。

$$\mathrm{D}_M = \sum_{i=1}^{n} norm(\mu_{maxProb(x_i)} - x_i)$$