結合GMM與K-means之語者分群方法
A Speaker-Clustering Method Using GMM and K-means

古鴻炎　　游政人
Hung-Yan Gu　and　Zheng-Ren You

國立台灣科技大學資訊工程系
E-mail: guhy@mail.ntust.edu.tw

摘要

本文提出一種結合GMM和K-means分群之語者分群方法。此方法首先為各個語者建立一個GMM模型，然後採取基於散異度(divergence)的距離函數來量測兩GMM模型之間的距離，接著使用我們修正的K-means分群演算法來進行語者的分群。當以本文的語者分群方法來對303個語者作40群的分群，然後只把40群中心語者的語料拿去訓練HMM辨識模型，就可使語音命令的辨識率達到飽和，亦即辨識率和303個語者都拿去訓練HMM的辨識率相當。

關鍵詞：語者分群、GMM、K-means、語音辨識。

1. 前言

語者分群的研究約興起於八十年代，當時作語者分群的目的，主要是作為語者調適(speaker adaptation)的前置處理，以便用來改進語者無關(speaker independent)語音辨識系統的辨識率；近年來則有學者研究以基於PCA (principle component analysis)的 eigenvoice方法，來作語者調適[16]。在此我們研究語者分群的主要目的，則是希望透過語者分群來找出聲學(acoustics)特徵上具有代表性的少量語者，然後就可以只錄製這些少量語者的語音命令之發音，去訓練語音命令辨識系統的聲學模型(如HMM, hidden Markov model)，以讓語者無關之語音命令辨識正確率，其衰減的量能夠儘量減小。

過去在語者分群上的研究，Padmanabhan等人所作的語者分群[1]，是依據待測語者(test speaker)所提供的少量語料，來從訓練語者中找出聲學(acoustic)相似度(likelihood)上最靠近的幾名語者，然後取出這幾名語者的語料，用以訓練待測語者的專屬的聲學模型，所以這種語者分群的意義並不是我們所要的。另外，Naito等人提出了一種語者分群之方法[2]，先對每一位語者估計其聲道(vocal tract)大小相關的聲學參數，再以歐基里德距離來量測兩組聲學參數之間的距離，然後根據Kosaka等人提出的樹狀分群法[3]，去進行語者分群。

後來，以聲學模型之間距離為基礎的語者分群方法逐漸被研究、提出，例如Yamada等人的作法是，先為各個語者建造隱藏式馬可夫模型(HMM)之聲學模型[4]，再採取Bhattachayya距離公式去量測聲學模型之間的距離，然後使用前面提到的樹狀分群法來作分群。Peng等人則提出以高斯混合模型(Gaussian Mixture Model, GMM)來為各個語者建立GMM聲學模型，然後以量測兩兩GMM之間的散異(divergence)程度，來作逐次合併(merge)式的語者分群[5]。除此之外，語者分群的另一種應用是，在多人交替說話的一段廣播語音中，偵測相鄰語者發音的轉換點[6, 7]。

回顧前人的研究成果之後，我們決定採取的語者分群作法是，先為各個訓練語者建造各自的GMM聲學模型；再使用基於散異度(divergence)觀念的差異值函數來量測GMM模型之間的距離，由於差異值函數的計算很費時間，因此必需把GMM模型之間的差異值先計算出來，再存入距離表；然後，我們修正K-means分群法[8]，以用來作預訂群數的語者分群處理，這裡需作修正的原因是，原始的K-means分群法不能夠直接拿來對GMM聲學模型作分群的計算。依據前述的步驟，我們研究的語者分群方法，其主要的處理流程就如圖1所示。

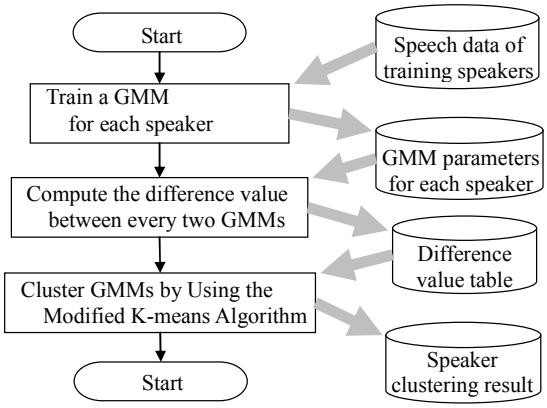

圖1　語者分群方法之主流程

2. GMM 模型建造

由於 GMM 能夠平滑地近似任意形狀的機率分佈，因此近年來常被用在語音與語者辨識，且得到不錯的效果[8]。當我們使用 GMM 來描述特徵向量 x 的機率密度時，實際上就是使用 M 個高斯密度

函數的加權和，來逼近 x 的機率密度，公式為

$$P(x|\lambda)=\sum_{j=1}^{M} w_j \cdot G(x;\mu_j;V_j) \quad (1)$$

其中 λ 代表 GMM 模型所有參數的集合，權重 w_j 需滿足總和為 1，μ_j 代表第 j 個高斯函數的中心點或平均向量，V_j 代表第 j 個高斯函數的共變異矩陣(covariance matrix)，$G(x;\mu;V)$表示多變數高斯(正規)機率分佈。

2.1 特徵向量計算

當要為一個語者建造 GMM 模型時，我們首先對該位語者所錄的語音信號作端點偵測[8, 9]，以把靜音(silence)部分去除，然後把各個片段的語音信號串接成一個大音檔，以方便作特徵參數的擷取。之後，將這個大音檔切割成一序列的 T 個音框(frame)，音框長度為 20ms，音框位移(shift)則為 10ms，接著對各個音框去計算出 39 維度的特徵向量，39 維度其實是由 13 個 MFCC (Mel frequency cepstral coefficient) [8, 9]係數 x(j)、13 個一階係數差值Δx(j)、及 13 個二階係數差值$\Delta\Delta x$(j)所構成，這裡 j 表示第 j 個係數。至於Δx_k(j)的定義是，$\Delta x_k(j) = x_k(j) - x_{k-1}(j)$，而$\Delta\Delta x_k(j)$的定義是$\Delta\Delta x_k(j) = \Delta x_k(j) - \Delta x_{k-1}(j)$，這裡下標 k 表示第 k 個音框。

2.2 GMM 模型訓練

在此我們設公式(1)裡的 M 值為 32，也就是使用 32 個高斯混合來建造一個語者的 GMM 模型，由於考慮到計算量會呈現平方成長，並且把重點放在距離量測上，因此我們未再嘗試更大的 M 值。GMM 模型的訓練，在觀念上是要去找出一組最佳的模型參數 λ_a，來滿足下列公式

$$\lambda_a = \arg\max_{\lambda} \prod_{k=0}^{T-1} P(x_k|\lambda) \quad (2)$$

其中 x_k 表示語料音檔第 k 個音框所算出的特徵向量，λ 表示一組可能的模型參數。但是實作上不可能窮舉所有組合的參數值來計算，因此本論文採取 Maximum Likelihood Estimate (MLE)的迭代估計方法[8]，來尋找最佳的參數值 λ_a。

訓練的過程可以分成初始化及疊代階段。模型初始化的作法是，把語料擷取出的特徵向量透過 k-means 演算法分成 32 群，然後去計算出 32 個中心向量 μ_j, j=0, 1, …, 31，再計算出全部特徵向量的共變異矩陣 V，並且給予每個高斯混合相同的權重值 1/32，如此就可以得到各個高斯混合的初始參數值。為了簡化計算，本論文假設特徵向量 x_k 的各維度之間是互相無關的，如此共變異矩陣 V 就變成了一個對角矩陣。

得到 GMM 模型的初始值後，就可用以計算出語料對於模型的 Log Likelihood，LL(λ)，計算公式為：

$$LL(\lambda)=\log\prod_{k=0}^{T-1} P(x_k|\lambda) \quad (3)$$

當我們把 LL(λ)分別去對 μ_j、V_j、w_j 作微分，並且令微分值為 0，就可以推導出下列的模型參數值之重估公式[8, 10]:

$$\hat{\mu}_j = \frac{\sum_{k=0}^{T-1} \gamma_j(x_k)\cdot x_k}{\sum_{k=0}^{T-1} \gamma_j(x_k)} \quad (4)$$

$$\hat{V}_j = \frac{\sum_{k=0}^{T-1} \gamma_j(x_k)\cdot(x_k-\mu_j)^t(x_k-\mu_j)}{\sum_{k=0}^{T-1} \gamma_j(x_k)} \quad (5)$$

$$\hat{w}_j = \frac{1}{T}\sum_{k=0}^{T-1} \gamma_j(x_k) \quad (6)$$

上面公式(4)(5)(6)裡的機率參數$\gamma_j(x_k)$，表示第 j 個高斯混合會被使用的事後機率(Post Probability)，也就是看到 x_k 時，推測 x_k 是由第 j 個高斯密度函數所產生之機率，其定義是

$$\gamma_j(x) = P(j|x) = \frac{P(j)P(x|j)}{P(x)} = \frac{w_j \cdot G(x;\mu_j;V_j)}{\sum_{i=0}^{M-1} w_i \cdot G(x;\mu_i;V_i)} \quad (7)$$

在此我們應用上述公式來作模型參數的重估計算，並且重估的疊代次數設為 15 次。

3. 距離量測與模型距離

有了各個語者的 GMM 模型之後，接著需要考慮如何量測兩個 GMM 模型之間的距離，以用來定義相對應語者之間的距離。而在定義兩 GMM 模型間的距離量測公式之前，應先考慮高斯分佈之間距離的量測方式。

3.1 高斯分佈之距離量測

回顧文獻得知，常被用於計算高斯分佈之間距離(相似度)的量測公式包括: Mahalanobis 距離、Bhattacharyya 距離、 Hellinger 距離等。Mahalanobis 距離是由一位印度統計學者 Mahalanobis 於 1936 年所提出[9]。假設給定兩個高斯分佈，$G_1=G(\mu_1;\Sigma_1)$ 及 $G_2=G(\mu_2;\Sigma_2)$，其中 μ_1、μ_2 表示平均值向量，Σ_1、

Σ_2 表示共變異矩陣，則兩者的 Mahalanobis 距離的計算方式，如公式(8)所示:

$$D_{MA}(G_1,G_2)=(\mu_1-\mu_2)(\frac{\Sigma_1+\Sigma_2}{2})^{-1}(\mu_1-\mu_2)^t \quad (8)$$

觀察公式(8)可發現，若給予的平均值向量 μ1、μ2，數值完全相同，則 Mahalanobis 距離將無法被計算。因此，Bhattacharyya 於 1943 年提出了 Bhattacharyya 距離來解決這個問題[11]。假設給定兩個高斯分佈，$G_1=G(\mu_1;\Sigma_1)$及 $G_2=G(\mu_2;\Sigma_2)$，則兩者的 Bhattacharyya 距離，計算方式就如公式(9)所示:

$$D_{BA}(G_1,G_2)=\frac{1}{8}(\mu_1-\mu_2)(\frac{\Sigma_1+\Sigma_2}{2})^{-1}(\mu_1-\mu_2)^t + \frac{1}{2}\ln\frac{\left|\frac{1}{2}(\Sigma_1+\Sigma_2)\right|}{|\Sigma_1|^{\frac{1}{2}}|\Sigma_2|^{\frac{1}{2}}} \quad (9)$$

Hellinger 距離是由 E. Hellinger 於 1909 所提出的 Hellinger integral 發展而來[12]，它可以由 Bhattacharyya 距離經非線性映射求出。假設給定兩個高斯分佈，$G_1=G(\mu_1;\Sigma_1)$及 $G_2=G(\mu_2;\Sigma_2)$，兩者的 Hellinger 距離的計算方式就如公式(10)所示[6]：

$$D_{HE}(G_1,G_2)=1-e^{-D_{BA}(G_1,G_2)} \quad (10)$$

3.2 基於 Pseudo Divergence 的模型距離

依據上一小節中所列舉的三種可用於計算高斯分佈之間的距離的量測公式，接著說明如何將它們應用於量測兩 GMM 模型之間的距離。一種基本的量測方法是，先將兩 GMM 模型 A 和 B 的高斯混合作一對一的配對，如令 A_j 對應到 B_j，A_j 和 B_j 分別表示 A 和 B 的第 j 個高斯混合，然後就可以如下公式來計算 A 和 B 兩 GMM 模型之間的距離:

$$Dist(A,B)=\sum_{j=0}^{M-1}D(A_j,B_j) \quad (11)$$

其中 $D(\bullet,\bullet)$代表距離量測，它可以是公式(8)(9)(10)之一所定義者。

不過，如果我們無法找到一個好的一對一配對高斯混合的方式，那麼公式(11)將不適用，並且各個高斯混合的重要性也不一樣，也就是應把權重值納入考慮。因此，我們決定採用 Peng 等人所提出的基於散異度之模型間距離的量測方法[5]，這個方法首先要去計算的是，兩 GMM 模型之間的分散度(dispersion)，其計算公式是:

$$Dispr(A,B)=\sum_{i=0}^{M-1}w_i\left(\sum_{j=0}^{M-1}w_j\cdot D(A_i,B_j)\right) \quad (12)$$

其中 w_i 和 w_j 分別表示 GMM 模型 A 和 B 的高斯混合的權重值，而 $D(\bullet,\bullet)$代表距離量測，它可以是公式(8)(9)(10)之一。接著，依據公式(12)的分散度定義，再去計算兩個單向散異度(unilateral pseudo divergence)的值，分別是 GMM 模型 A 到 B 的 *Divrg*(A,B)和 GMM 模型 B 到 A 的 *Divrg*(B,A)，而 *Divrg*(•, •)函數的定義是:

$$Divrg(X,Y)=\frac{Dispr(X,Y)}{Dispr(X,X)} \quad (13)$$

由於公式(13)所計算出的單向散異度不是對稱的，所以 Peng 等人再定義了一種具有對稱特性的差異值(difference)函數[5]，來量測 GMM 模型間的距離，其計算公式是:

$$Diffr(A,B)=\frac{Divrg(A,B)}{2}+\frac{Divrg(B,A)}{2} \quad (14)$$

4. 修正之 K-means 分群法

K-means 是一種相當廣泛被使用的分群演算法[8, 9]，每疊代一次後各群的中心向量就會更新一次，使得成員向量到中心向量的平均距離減少，而經過若干次疊代後，各群的中心向量就會趨於穩定而不再變動，也就是說 K-means 演算法保證會收斂，不過，K-means 演算法並不保證會找到最佳的分群結果。

4.1 傳統的 K-means 分群

K-means 演算法的處理步驟如下：

(i) 決定初始的中心向量。假設我們有 N 個訓練向量，要分成 K 群，則先從 N 個向量中隨機挑選出 K 個當作初始的中心向量。

(ii) 對每一個訓練向量 X_n，計算它離每個中心向量 C_k 的距離。當 X_n 和 C_k 都是 GMM 模型時，就是要去計算公式(14)。

(iii) 依據步驟(ii)算出的距離值，找出各個訓練向量 X_n 最靠近的中心向量 C_k 所屬的群，也就是計算

$$grp(X_n)=\arg\min_{0\le k\le K-1} distance(X_n, C_k) \quad (15)$$

然後將各個訓練向量分配到離它最近的群中。

(iv) 計算各群的新中心向量。對 K 群的向量集合分別去作取重心的處理，而得到新中心向量 $\overline{C}_k$。

(v) 決定是否結束分群處理，若否則回到步驟(ii)。設前一次疊代的失真距離為 R'，而本次疊代後的失真距離為 R，則當 R / R' > Thr (如 0.98)時，就表示 K-means 分群效果已不顯著，而可以結束疊代，此時 grp(X_n)就是最後的分群結果。

步驟(v)裡提到的失真距離 R，其計算公式為:

$$R=\sum_{n=0}^{N-1}\left(\min_{k=0}^{K-1} distance(X_n, C_k)\right) \quad (16)$$

4.2 修正的 K-means 分群

當我們嘗試應用傳統的 k-means 演算法來作語

者分群時，遭遇到至少兩個問題，第一個問題是關於新的群中心的決定，在傳統 k-means 演算法的步驟(iv)，每一次疊代裡都要以取重心的方式來計算新的群中心，並且這個群中心其實是不存在於訓練向量集合內。可是在語者分群上，如何去計算出同一群語者的 GMM 模型的重心，以作為新的群中心之 GMM 模型，是一個必需解決但不易解決的問題，因為必需維持 K-means 演算法的收斂性。此外，第二個問題是計算量很大的問題，傳統 k-means 分群法的步驟(ii)裡，需花費很多的計算量，來計算各語者的 GMM 模型和中心點 GMM 模型之間的距離，以 300 個語者和 30 群為例，就會需要依照公式(14)作 270*30=8,100 次計算，這需花費 48,600 秒(每次計算約需 6 秒)。

因此，我們研究了一個修正的 K-means 分群方法，在此訂定新的群中心的作法是，以逼近的方式來求取新的群中心，並且這個群中心是由某一個語者的 GMM 模型來代表。作法是，在每一次的疊代中，先去計算各群的成員對於該群當中其它成員的幾何平均距離，計算的公式為：

$$S_k^i = \left(\prod_{all\ grp(X_j)=k,\ j\neq i} Diffr(X_i, X_j) \right)^{\frac{1}{M_k - 1}} \quad (17)$$

其中 k 代表第 k 群，i 代表目前被計算的成員 X_i，j 代表在第 k 群中不是 X_i 的其他成員 X_j，M_k 代表第 k 群中的成員數。然後，我們取出幾何平均距離為最小的 GMM 模型 X_j 來作為此群的新中心。在減少計算量方面，我們的解決方法如圖 1 的第二個方塊所示，是在執行修正的 K-means 分群之前，先對所有的 X_i、X_j 組合，依公式(14)計算出 GMM 模型之間的距離，並且存入距離表，如此 K-means 演算法的步驟(ii)和公式(17)裡 *Diffr*(•, •)函數的計算，就變成是查表的方式，而大幅減少了 K-means 分群的計算量。

5. 語者分群實驗

接著，我們進行語者分群的實驗，來驗證圖 1 所示的語者分群方法。首先我們對語者分群之用的語料作預先處理，也就是對 TCC-300 語料庫中的 303 位語者(151 位男性, 152 位女性)，區分出各語者的錄音檔，再作端點偵測及語音片段的串接，然後切割成一序列的音框去計算出特徵向量。之後，把語料帶入圖 1 去作語者分群的處理。

5.1 評估方法

當完成分群後，我們如何知道分群的結果是好還是不好? 直觀的判斷是，屬於同一群內的資料點其相似度要高，而分屬於不同群的資料點必須要明確的分開，基於這樣的觀念，已有學者提出具體的評估方法，包括了 CH 評估法[13]和 Dunn 評估法[14]。以 CH 評估法為例，它的計算公式為

$$CH\ index = \frac{\frac{1}{K\text{-}1}\sum_{k=1}^{K} n_k \cdot \|z_k - z\|^2}{\frac{1}{N\text{-}K}\sum_{k=1}^{K}\sum_{i=1}^{n_k} \|x_i - z_k\|^2} \quad (18)$$

其中 K 為群落數，N 為總資料點數，n_k 為群落 k 中的資料點數，z_k 為群落 k 的中心點，z 為所有資料點的中心點。

此外，由於我們研究語者分群的目的是，要依據分群結果來取出各群群中心語者的錄音語料，用以訓練語音命令辨識所需的音節 HMM 模型。因此，我們覺得要評估一個語者分群的結果是好或是壞，不如就直接以語音命令辨識的辨識正確率來作為指標。在此，我們首先設定語音命令的內容為"前進", "後退", "左轉", "右轉", "開始", "停止", "準備", "機器人", "向後轉", "煞車"等 10 個命令，然後邀請訓練語料之外的 30 位語者(男女性各 15 人)來錄測試用的語料，也就是每人對這 10 個命令分別發音 5 遍，使用的是 Bluetooth 無線麥克風，取樣率已被限定為 8KHz。

5.2 語音命令辨識率

當分群的群數少時，拿去訓練音節 HMM 模型的語料也較少，如此辨識正確率直覺上也會較低；相反地，當分群的群數變多時，拿去訓練音節 HMM 模型的語料變多，辨識正確率直覺上會升高，而極端情況是每個語者各自為一群。作語者分群的語音命令辨識時，所使用的音節 HMM 模型，本文是以 HTK 軟體[15]來訓練，音節模型內的狀態數設為 7，而各狀態上的高斯混合數則設為 6。在此所以選擇狀態數 7 和混合數 6，是因為我們已先在 303 群(每人一群)的情況下，測試了狀態數由 5 變化到 9、及混合數由 5 變化到 9 的各種組合，以了解辨識率和這兩因素之間的關連性，結果發現辨識率在狀態數設為 7 或 8 時會比較高，但是在同一種狀態數時，混合數由 5 變化到 9 並不會讓辨識率有明顯的差異；辨識率最高與最低的組合分別是，狀態數 7 和混合數 6 時的辨識率 87.7%為最高，而狀態數 5 和混合數 8 時的辨識率 84.3 為最低。

在此我們將群數設定為 10、15、20、30、40、50、303 群等等，分別去作圖 1 的分群處理。由於 k-means 分群法有隨機挑選初始中心的步驟，所以每次分群的結果都會不一樣，因此我們對於各個分群群數都反複作五次的分群實驗，並且五次分群的結果都再拿去作語音命令辨識的測試，然後我們取五次測試中的辨識正確率最大值和平均值來作比較。至於公式(8)(9)(10)等三種距離量測的影響，從實驗結果來看，公式(10)的 Hellinger 距離大體上都可獲得較高的辨識率，因此這裡就以 Hellinger 距離所得到的辨識率數值作代表。如此，我們得到的辨識正確率數值，就如表 1 所列出的，觀察表 1 的辨

表 1　語音命令辨識正確率

群數	10	15	20	30	40	50	303
最高辨識率	**69.3**	**80.8**	**82.6**	**86.5**	**87.1**	**85.2**	**87.7**
平均辨識率	**66.8**	**76.7**	**77.4**	**81.6**	**83.4**	**83.5**	**void**

識率數值，可發現最高值和平均值兩者都會隨者群數的增加而升高，並且比較顯著的改變，是發生於 10 群變成 15 群、及 20 群變成 30 群時。此外，當群數增加到 40 群以上時，辨識率數值就開始出現飽和的現象，亦即和 303 群(每人一群也就是不分群)之辨識率 87.7 非常靠近。至於整體上來看，辨識率數值都偏低未超過 90%，其主要原因應是，我們要求了 30 位測試語料的錄音者，分別以快速、中庸、緩慢的速度來發出語音命令，而發音速度的確會對辨識率造成影響。

5.3 CH 值與辨識率之相關性

另外我們想要了解的一點是，辨識正確率和公式(18)所算出的 CH 值之間是否有相關性? 因此，這裡就把群數為 40 和 50 時，分別作的 5 次修正的 K-means 分群的結果，去計算出 CH 值，然後把 CH 值和辨識率值配對列出，就如表 2 所示的情況，觀察表 2 的數值，我們並未發現 CH 值和辨識率之間有明顯的相關性存在，例如 40 群的語者分群，CH 值最大可達 11.58，可是它對應的辨識率只有 81.0，比最高的 87.1 還低很多，不過在 50 群的語者分群裡，CH 值最大可達 9.52，而且它對應的辨識率 85.2，也是 5 次實驗中最高的。此外，我們也測量了 Dunn 評估法的 Index 值去作觀察，不過也是未發現相關性存在。

表 2　CH 值與辨識率

40 群分群實驗	1	2	3	4	5
CH 值	**11.58**	**8.98**	**8.71**	**9.57**	**8.47**
辨識率	**81.0**	**85.6**	**83.9**	**87.1**	**79.2**
50 群分群實驗	1	2	3	4	5
CH 值	**8.33**	**7.23**	**8.18**	**9.52**	**7.93**
辨識率	**81.4**	**84.2**	**83.9**	**85.2**	**82.7**

6. 結論

本文研究語者分群的目的是，希望使用少量語者的錄音語料，便能夠建立強健的 HMM 模型，來作為語音命令辨識之用。在我們所提出的語者分群方法裡，首先使用 MFCC 特徵參數來訓練各語者自己的 GMM 模型。接著，考慮了兩高斯分佈之間距離的量測方法，我們測試了 Mahalanobis 距離、Bhattacharyya 距離、和 Hellinger 距離，由實驗結果來看，Hellinger 距離是較好的選擇；由於前述三種距離量測僅限用於兩個高斯分佈之間，因此我們採用基於散異度的方法來量測 GMM 模型之間的距離，不過散異度的計算相當耗費 CPU 時間，所以我們使用了距離表和查表的作法。

在分群的演算法方面，原始的 K-means 分群法不能夠直接被使用，因為不知道如何去計算同一群語者的 GMM 模型的重心，因此我們提出一個修正的 K-means 分群方法，它訂定新的群中心的作法是以逼近的方式來決定，作法是在每一次的疊代中，先去計算各群的成員對於該群中其它成員的幾何平均距離，然後取出幾何平均距離為最小的 GMM 模型來作為此群的新中心。

當以本文的語者分群方法來對 303 個語者作 40 群的分群，然後只把 40 群中心語者的語料拿去訓練 HMM 辨識模型，就可使語音命令的辨識率達到飽和，亦即辨識率和 303 個語者都拿去訓練 HMM 的辨識率相當。此外，觀察 CH 分群評估值和辨識率之間的相關性，我們並未發現兩者之間有相關性存在。

致謝

感謝國科會計畫之支援，國科會計畫編號 NSC 95- 2218- E- 011- 009。

參考文獻

[1] Padmanabhan, M., *et al.*, “Speaker Clustering and Transformation for Speaker Adaptation in Speech Recognition Systems,” *IEEE trans. Speech and Audio Processing*, Vol. 6, No. 1, pp. 71-77, 1998.

[2] Natio, M., L. Deng, and Y. Sagisaka, “Speaker Clustering for Speech Recognition Using the Parameters Characterizing Vocal Tract Dimensions”, *ICASSP* (Seattle, USA), pp. 981-984, 1998.

[3] Kosaka, T. and S. Sagayama, “Tree-Structured Speaker Clustering For Fast Speaker Adaptation”, *ICASSP*, pp. 245-248, 1994.

[4] Yamada, M., *et al.*, “Fast Algorithm for Speech Recognition Using Speaker Cluster HMM”, *EuroSpeech* (Rhodes, Greece), pp. 2043-2046, 1997.

[5] Peng, X., W. Xu, and B. Wang, “Speaker Clustering via Novel Pseudo-Divergence of Gaussian Mixture Models”, *Int. Conf. on Natural Language Processing and Knowledge Engineering* (Wuhan, China), pp. 111-114, 2005.

[6] Liu, D. and F. Kubala, “Online Speaker Clustering”, *ICASSP* (Hong Kong), Vol 1, pp. 572-575, 2003.

[7] Tsai, W. H. and S. M. Wang, “Speaker Clustering Based on Minimum Rank Index”, *ICASSP* (Honolulu, U.S.A), Vol 4, pp. 15-20, 2007.

[8] Rabiner, L. and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

[9] O`Shaughnessy, D., *Speech Communication: Human and Machine*, 2'nd ed., IEEE Press, 2000.

[10] 林俊青，多國語言辨識系統之特徵設計研究，碩士論文，國立中山大學電機研究所，2002。

[11] Wikipedia, Bhattacharyya distance, http://en.wikipedia.org/wiki/Bhattacharyya_distance.

[12] Wikipedia, Hellinger distance, http://en.wikipedia.org/wiki/Hellinger_distance.

[13] Davies, D. L. and D. W. Bouldin, "A Cluster Separation Measure", IEEE trans. Pattern Analysis and Machine Intelligence, Vol. 1., No. 2, pp. 224-227, 1979.

[14] Dunn, J. C., "Well Separated Clusters and Optional Fuzzy Partitions", Journal of Cybernetics, Vol. 4, pp. 95-104, 1974.

[15] University of Cambridge, Hidden Markov Model Toolkit (HTK)", http://htk.eng.cam.ac.uk/.

[16] Kuhn, R., et al., "Rapid speaker adaptation in eigenvoice space", IEEE trans. Speech and Audio Processing, Vol. 8, pp. 695-707, 2000.