Clustering

- ✓ Classification 與 clustering 差別在於: classification 有用到 class label，然 clustering 則無。
- ✓ clustering 中的 data 無 label，因此為 unsupervised 的過程。也因為沒有 label，clustering 的結果無一定的好壞標準，通常會依據問題或需求而定標準。
- ✓ dataset X，clustering 後分成 m 個 clusters。其中 clusters 的集合稱為 clustering。

$$C_i \neq \phi, \ \forall i$$
$$C_1 \cup C_2 \cup ... \cup C_m = X$$

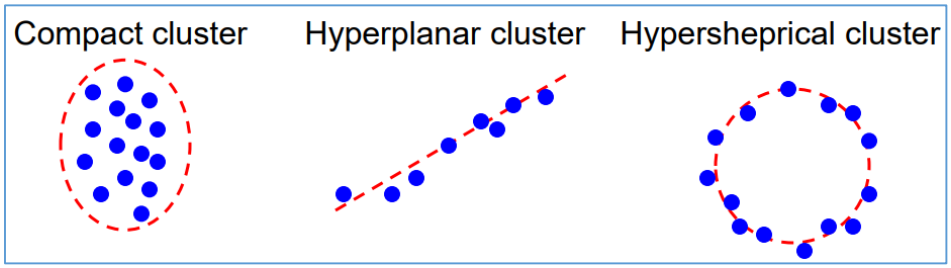
$$\mathcal{R} = \{C_1, C_2, ..., C_m\}$$

- ✓ (cluster representative)為了要降低資料運算量，通常將一個 cluster 用較少的 parameters 表示。常見的兩種表示方式分別為: point representative (compact cluster)與 shell representative (shell cluster)。Point representative 可以用 cluster 中所有 sample 點的平均(mean point)表示，也可以用實際在 cluster 中的 sample 點表示(找到一個點使得相異度加總最小或是相似度加總最大)。

mean point $\boldsymbol{m}_p$: $\boldsymbol{m}_p = \frac{1}{n_C} \sum_{\boldsymbol{y} \in C} \boldsymbol{y}$

mean center $\boldsymbol{m}_c$ (one of the samples in the cluster):

$$\boldsymbol{m}_c = \arg\min_{z \in C} \sum_{\boldsymbol{y} \in C} d(\boldsymbol{y}, z) \quad \text{or} \quad \boldsymbol{m}_c = \arg\max_{z \in C} \sum_{\boldsymbol{y} \in C} s(\boldsymbol{y}, z)$$



- ✓ 在分群的過程中，會遇到要將相似的 patterns 分在同一個 cluster；將相異的 patterns 分在不同 clusters。Patterns 之間多相似、多相異，可以透過 measure 得知，這種 measure 我們稱之為 proximity measure。當使用不同的 measure，產生出 cluster 的形狀也有所不同。Data 經過 proximity measure 後，就會成為 relational data。

✓ 相異度測量(dissimilarity measure)。須滿足以下兩個條件: 1.自身相異度最小 2. 兩點之間的相異度不變。常見的方法有: Weighted lp distance、Mahalanobis distance、Hamming distance、Edit distance。

$$d(\boldsymbol{x},\boldsymbol{y}) \geq d(\boldsymbol{x},\boldsymbol{x}) = d_0, \ \forall \boldsymbol{x},\boldsymbol{y} \in X$$
$$d(\boldsymbol{x},\boldsymbol{y}) = d(\boldsymbol{y},\boldsymbol{x}), \ \forall \boldsymbol{x},\boldsymbol{y} \in X$$

It is a metric dissimilarity measure if it also satisfies

$$d(\boldsymbol{x},\boldsymbol{y}) = d_0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$$
$$d(\boldsymbol{x},\boldsymbol{z}) \leq d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z}), \ \forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in X$$

(triangle inequality)

- Weighted lp distance:

$$d_p(\boldsymbol{x},\boldsymbol{y}) = \left[\sum_{i=1}^{l} w_i \mid x_i - y_i \mid^p \right]^{1/p}$$

l_2 with wi=1: Euclidean distance; l_1: Manhattan norm。

- Mahalanobis distance:其中 B 為對稱正定矩陣。

$$d(\boldsymbol{x},\boldsymbol{y}) = \left[(\boldsymbol{x}-\boldsymbol{y})^T B(\boldsymbol{x}-\boldsymbol{y}) \right]^{1/2}$$

- Hamming distance: 常用於離散的點。Hamming distance 就是兩 vectors 相異處的個數。例如: 101 與 111 的 hamming distance 為 1。
- Edit distance: 用於資料為字串形式。Edit distance 就是可以使得兩字串相同的總改變量。例如: kitten 與 sitting 的 edit distance 為 3。

✓ 相似度測量(similarity measure)。須滿足以下兩個條件: 1. 自身相似度最大 2. 兩點之間的相似度不變。常見的方法有: cosine similarity、Tanimoto measure。

$$s(\boldsymbol{x},\boldsymbol{y}) \leq s(\boldsymbol{x},\boldsymbol{x}) = s_0, \ \forall \boldsymbol{x},\boldsymbol{y} \in X$$
$$s(\boldsymbol{x},\boldsymbol{y}) = s(\boldsymbol{y},\boldsymbol{x}), \ \forall \boldsymbol{x},\boldsymbol{y} \in X$$

It is a metric similarity measure if it also satisfies

$$s(\boldsymbol{x},\boldsymbol{y}) = s_0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{y}$$
$$s(\boldsymbol{x},\boldsymbol{y}) + s(\boldsymbol{y},\boldsymbol{z}) \leq [s(\boldsymbol{x},\boldsymbol{y}) + s(\boldsymbol{y},\boldsymbol{z})]s(\boldsymbol{x},\boldsymbol{z}), \ \forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in X$$

- Cosine similarity: 所有 vectors 皆已 normalized 為前提。其中 x^Ty 代表 x 與 y 內積的結果。

$$s_{cosine}(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

- Tanimoto measure: 所有 vectors 皆已 normalized 為前提。

$$s_T(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^T\boldsymbol{y}}$$

✓ 對於一個 clustering 演算法，我們需要一個 function 作為要 assign sample x 給哪個 cluster 使用，也就是找出 point 與 cluster 之間的距離。它們之間的距離可以定義成: sample x 與 cluster 最遠(或最近)的距離、sample 與 cluster 中所有點距離的平均等等。

$$\mathcal{P}_{max}(\boldsymbol{x},C) = \max_{\boldsymbol{y}\in C}\mathcal{P}(\boldsymbol{x},\boldsymbol{y})$$

$$\mathcal{P}_{min}(\boldsymbol{x},C) = \min_{\boldsymbol{y}\in C}\mathcal{P}(\boldsymbol{x},\boldsymbol{y})$$

$$\mathcal{P}_{avg}(\boldsymbol{x},C) = \frac{1}{n_C}\sum_{\boldsymbol{y}\in C}\mathcal{P}(\boldsymbol{x},\boldsymbol{y})$$

✓ 前面提到: 通常我們會用 cluster representative 來表示一個 cluster。因此，point 與 cluster 之間的距離就可以定義成 point 與 cluster representative 之間的距離。

Point representatives $\boldsymbol{m}$: $\mathcal{P}(\boldsymbol{x},C) = \mathcal{P}(\boldsymbol{x},\boldsymbol{m})$

Shell representatives (examples):

Hyperplanes: $d(\boldsymbol{x},C) = |\boldsymbol{a}^T\boldsymbol{x} + a_0| / \|\boldsymbol{a}\|$

Hyperspheres: $d(\boldsymbol{x},C) = \big|\|\boldsymbol{x}-\boldsymbol{c}\| - r\big|$

✓ 兩 clusters 之間的距離，可以定義成如上述提到的 point 與 cluster 距離、或是兩 cluster representatives 之間的距離。

$$\mathcal{P}_{max}(C_1, C_2) = \max_{\boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{min}(C_1, C_2) = \min_{\boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{avg}(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{\boldsymbol{x} \in C_1} \sum_{\boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{mean}(C_1, C_2) = \mathcal{P}(\boldsymbol{m}_{C_1}, \boldsymbol{m}_{C_2})$$

✓ Sequential clustering:

➢ BSAS: 每加入一個 sample 進 cluster，就會更新 cluster representative。根據設定的Θ值，判斷要加入哪一個 cluster。

Basic Sequential Algorithmic Scheme (BSAS):

$m \leftarrow 1$
$C_m \leftarrow \{\boldsymbol{x}_1\}$
For $i = 2$ to N
  $k \leftarrow$ argmin $d(\boldsymbol{x}_i, C_k)$
  If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$
    $m \leftarrow m + 1$
    $C_m \leftarrow \{\boldsymbol{x}_i\}$
  Else
    $C_k \leftarrow C_k \cup \{\boldsymbol{x}_i\}$
    Update cluster representatives if necessary
  End
End

➢ MBSAS: 一開始根據Θ值建立初始的 clusters。有了這些初始的 clusters，將 sample 加入哪一個 cluster 的方法與 BSAS 完全相同。

Modified BSAS (MBSAS)

$m \leftarrow 1$
$C_m \leftarrow \{\boldsymbol{x}_1\}$
For $i = 2$ to N
  $k \leftarrow$ argmin $d(\boldsymbol{x}_i, C_k)$
  If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$
    $m \leftarrow m + 1$
    $C_m \leftarrow \{\boldsymbol{x}_i\}$
  End
End
For $i = 1$ to N
  If $\boldsymbol{x}_i$ is not in any cluster
    $k \leftarrow$ argmin $d(\boldsymbol{x}_i, C_k)$
    $C_k \leftarrow C_k \cup \{\boldsymbol{x}_i\}$
    Update cluster representative if necessary
  End
End

- ✓ 在 BSAS 與 MBSAS 中不同的Θ值會有不同的 clustering 結果。如何選出一個較好的結果? 找到一個 clustering，它對於Θ值影響最小。例如Θ= 1~5, q=6，對於資料順序為 328716，並且使用 MBSAS，Θ = 2~4 都是分成兩群，因此分成兩群是相對穩定的結果。

- ✓ (refinement by merging) 兩 clusters 可能非常相近，將它們合併成一個 cluster 會比較好，我們可以定義:相似程度超過門檻值時，就將它們合併。(refinement by reassignment) 計算 clusters 的 representatives，並根據 representataives 重新 assign 到合適的 cluster 裡 (與新的 representive 最近)。

- ✓ Hierarcical clustering: 常用在 clustering relational data，會產生 nest effect (對於 agglomerative clustering 而言，當兩個 samples 被分配到同一個 cluster，則它們最終仍會在同一個 cluster 裡；對於 divisive clustering 而言，當兩個 samples 被分配到不同的 clusters，則它們最終仍會被分隔) 。兩種常見的型態為 agglomerative clustering、divisive clustering。Agglomerative clustering 起始為一個 cluster，每次兩兩 clusters 合併直到所有 sample 都合併；division clustering 起始為一個 cluster，其中包含著所有的 samples，每次將 cluster 一分為二。

- ✓ 對於 agglomerative clustering，要如何選擇每次要合併的兩個 clusters? 如使用相異度測量(dissimilarity measure)，則選擇兩 clusters 為相異度最小的。常見的方法有 single link 與 complete link。Single link 定義的相異程度為兩 clusters 間的最小相異度；complete link 則為兩 clusters 間最大的相異度。當 clusters 被合併成另一個較大的 cluster 後，相異程度的判斷仍不變。以 single link 為例: 假設 Ci 與 Cj 合併成 Cq，則任一個 cluster Cs 與 Cq 的相異程度為 (Cs 與 Ci 最小相異度) 與 (Cs 與 Cj 最小相異度) 的最小值。

$$d_{min}(C_1,C_2)=\min_{x\in C_1,y\in C_2} d(x,y) \quad \text{(single link)}$$

$$d_{max}(C_1,C_2)=\max_{x\in C_1,y\in C_2} d(x,y) \quad \text{(complete link)}$$

$$d_{min}(C_q,C_s)=\min\{d_{min}(C_s,C_i),d_{min}(C_s,C_j)\}$$

$$d_{max}(C_q,C_s)=\max\{d_{max}(C_s,C_i),d_{max}(C_s,C_j)\}$$

- ✓ Single link 容易形成狹長型的 clusters。缺點為 cluster 的頭尾兩點距離遠。Single link (complete link) works betters for elongated clusters (compact clusters). 除 single link 與 complete link 以外，也有其他的方法例如: WPGMA、UPGMA。

Weighted pair group method average (WPGMA)

$$d(C_q, C_s) = \frac{1}{2}[d(C_s, C_i) + d(C_s, C_j)]$$

Unweighted pair group method average (UPGMA; average-link)

$$d(C_q, C_s) = \frac{n_i}{n_i + n_j} d(C_s, C_i) + \frac{n_j}{n_i + n_j} d(C_s, C_j)$$

- ✓ 用暴力法(brute-force)產生 agglomerative clustering 需要 $O(N^3)$，而 single link 方法等同於建立 MST，因此只需要 $O(N^2)$；complete/average link 可以用 heap 建立，需花費 $O(N^2 \log N)$。
- ✓ 對於 divisive clustering，要如何選擇 split 成兩個 clusters? 我們可以找相異度最高(或相似度最低)的兩個 clusters，然而在實作上此方法非常耗費時間。
- ✓ C-Means algorithm: clusters 的數量與 cluster representative (prototype)需給定。常見的方法有: HCM、FCM、PCM。
  - ➢ HCM (Hard/crisp C-Means): 每一個 sample 被分配到唯一一個 cluster。其中 cost function 可以表示為: (d_{ij}^2為 sample i 與 cluster j 的距離)
    - $C = \sum_{i=1}^{N} \sum_{j=1}^{C} d_{ij}^2 \delta_{ij}$, subject to $\forall i$, $\sum_{j=1}^{C} \delta_{ij} = 1$, $\delta_{ij} = 0$ or 1

    它是一個 alternating optimization 過程。每次都會將 sample x 分配到最接近最合適的 cluster (固定d_{ij}^2降低 C)，接著再更新 prototype (固定δ_{ij}降低 C)，重複上述兩步驟直到 stopping critera (例如 prototype 於兩次 iteration 間不再改變)。
  - ➢ FCM (Fuzzy C-Means): 每一個 sample 可以被分配到多個 clusters，其隸屬於各群的程度為[0,1]之間，且所有程度加總為 1。它的 cost function 可以表示為:
    - $J = \sum_{i=1}^{N} \sum_{j=1}^{C} d_{ij}^2 u_{ij}^q$ where $q>1$, subject to $\forall i$, $\sum_{j=1}^{C} u_{ij} = 1$

    越大的 q 值，能接受 fuzzy 的程度越高。

- Recalculate the memberships (u_{ij})

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{q-1}}}$$

- Recalculate the prototypes (v_j)

This is for point prototypes only: →

$$v_j = \frac{\sum_{i=1}^{N} (u_{ij})^q x_i}{\sum_{i=1}^{N} (u_{ij})^q}$$

- PCM (Possibilistic C-Means): 將 FCM 中"隸屬程度加總為 1"的條件移除，也就是 sample 可以隸屬於多個 clusters，且程度加總允許大於 1，另一方面，也可以不隸屬任一個 cluster。對於 dataset 中有 outliers，FCM 會使得 prototype 外移，然而 PCM 可以讓 uij 變小，使 outliers 對 cluster 影響程度降至最小。它的 coust fucntion 可以表示為:

$$J=\sum_{i=1}^{N}\sum_{j=1}^{c}u_{ij}^{q}\,d_{ij}^{2}+\sum_{j=1}^{c}\eta_j\sum_{i=1}^{N}(1-u_{ij})^{q}$$

- Recalculate the memberships (u_{ij})

$$u_{ij}=\frac{1}{1+\left(\frac{d_{ij}^{2}}{\eta_j}\right)^{\frac{1}{q-1}}}$$

- Recalculate the prototypes (v_j)

Same as FCM →

$$\boldsymbol{v}_j=\frac{\sum_{i=1}^{N}(u_{ij})^{q}\,\boldsymbol{x}_i}{\sum_{i=1}^{N}(u_{ij})^{q}}$$

- 總結 HCM、FCM、PCM。
  - HCM 是(收斂)速度最快的。
  - HCM 是 FCM 中的一個特例(q=1)。
  - HCM 相較於 FCM 更可能收斂到 local optima。
  - PCM 對於 outliers 的影響最小，對於 initializtion 最敏感。(通常會先用 FCM 做 initialization，再做 PCM)
- C-Medoid clustering: 與 C-Means 不同的是: prototype 為 cluter 中真實的點，也就是找到 cluster 中的一個點與 cluster 中的其他點相異程度加總最小。

$$\boldsymbol{m}_c=\arg\min_{z\in C}\sum_{\substack{y\in C\\ y\neq z}}d(\boldsymbol{y},\boldsymbol{z})$$

➢ Gustafson-Kessel algorithm (FCM 的一個變形): 與 FCM 唯一不同之處為: prototype 以 normal distribution 取代。

$$J = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^{q}\, d_{ij}^{2}$$

$$d_{ij}^{2} = d_{GK}^{2}(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = \frac{1}{|\Sigma_j|^{1/l}} (\boldsymbol{x}_i - \boldsymbol{c}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_i - \boldsymbol{c}_j)$$

The update equations:

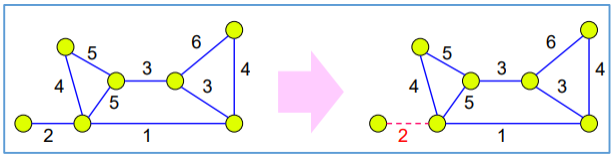
$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{q-1}}} \qquad \boldsymbol{c}_j = \frac{\sum_{i=1}^{N} (u_{ij})^q \boldsymbol{x}_i}{\sum_{i=1}^{N} (u_{ij})^q}$$

$$\Sigma_j = \frac{\sum_{i=1}^{N} (u_{ij})^q (\boldsymbol{x}_i - \boldsymbol{c}_j)(\boldsymbol{x}_i - \boldsymbol{c}_j)^T}{\sum_{i=1}^{N} (u_{ij})^q}$$

✓ Density-based Spatial Clustering of Application with Noise (DBSCAN): 一種 nonparameteric (no prototype) clustering 演算法。對於高密度點的會自成一群，低密度點的會被視為 noise。類似 single link agglomerative clustering，但此方法不會連接低密度點。首先，定義指定半徑 r 與密度閾值 t。每次拜訪尚未拜訪過的點，以此點為中心向外擴展半徑為 r 的圓形區域，如該區域包含到的點(自身點除外)之個數超過密度閾值 t，則將它們自成一群，同時包含到的其他點也向外擴展，找到滿足閾值的所有點，並將它們納入同一群。因此，只需要給定 r 與 t 就能分群。其餘不滿足的點(亦即為屬於任何一個 cluster)則視為 noise。DBSCAN 的困難點在於: t 與 r 的選定以及找 neighbors 的效率。

✓ Graph-cut based clustrering: 把 samples 描述成 weighted graph，其中 edges 上的 weight 為兩 samples 相似程度。透過移除 edges，可以將 graph 分成兩個(多個) subgraph。edges 上的 weights 為相似度，因此移除 edges 的 wieght 總和越小越好(如 wieghts 代表相異度，則移除 wieght 總和越大越好)。但這種方法容易產生不平均的 subgrpahs 或是產生 isolated

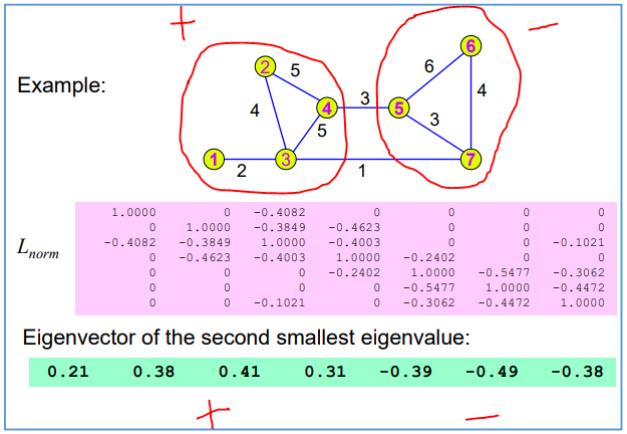
samples。我們可以透過 Normalized cut 找到最合適的 cut，使得 ncut(A,B) 最小。



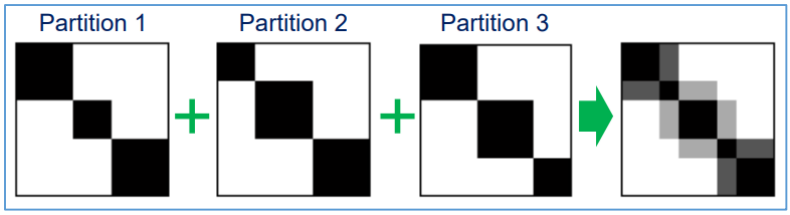
Normalized cut: $ncut(A,B) = \frac{w(A,B)}{w(A,V)} + \frac{w(A,B)}{w(B,V)}$

- V is the original set of vertices, and A and B are the two subsets of vertices.
- $w(A,B)$ is the total weights of edges that connect vertices between A and B.

✓ Sepctral clustering: 因為"最小化 ncut(A,B)" 的問題不容易。我們可以透過尋找 normalized Laplacian matrix 特徵向量的方式找到合適的切割。normalized Laplician matrix (NLM) $L_{norm} = I - D^{-1/2} A D^{-1/2}$ 中 A 為 weighed 相鄰矩陣(有連接才有值，且對角線的值為 0)，D 為對角矩陣(Dii 為連接到 vertex i 的 weight 總和)。因為 NLM 的最小特徵值為 0，故找第二小的特徵值。相對應的特徵向量，並依照其 component 的正負值來分群。例如第二小的特徵值對應到的特徵向量為(0.2, 0.3, 0.4, -0.3, -0.3, -0.4, -0.3)，前三個點、後四個點各自分在不同群。時間複雜度為 $O(N^3)$。



✓ 透過簡易的 clustering 演算法可以將 dataset 裡的 samples 分群。Cluster ensamble 就是將分群的結果再做分群(將某些 cluters 合併成更大的 clusters)。此時 similarity measure 就會是兩 samples 在同 cluster 但在不同 parition 的 likihood。根據相似度測量，我們可以使用前述的 spectral clustering 或是 hieracacal agglomeration 做 cluster ensamble。Co-association (affinity) matrix 是相似度測量結果對於不同的 partition 常見的表示方法。



✓ Clustering 中無使用到 label，因此對於 clustering 的結果沒有直觀的判斷標準(例如正確率)。一些常見的 cluster validity 方法: Dunn Index、DB Index、XB Index。

➢ Dunn Index: 用於 hard (crisp) clustering。Dm 的值越大，代表 clustering 結果越好。其中，分子為兩 clusters 間相異度的最小值 min(d(Ci,Cj))；分母為 cluster 中兩 samples 中最遠的距離 max(diam(Ck))。例如:有兩個 clusterings 結果: {1,2},{3,5},{7,8}與{1,2,3},{5,7,8}。前者 Dm=(3-2)/(5-3)=1/2；後者 Dm=(5-3)/(8-5)=2/3。因為 2/3>1/2，所以後者 clustering 結果較好。

$$D_m = \min_{\substack{i,j \\ i \neq j}} \left[\frac{d(C_i, C_j)}{\max_k [diam(C_k)]} \right]$$ The larger, the better.

Dissimilarity between clusters: $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$

Cluster diameter: $diam(C_k) = \max_{x, y \in C_k} d(x, y)$

➢ Davies-Bouldin (DB) Index: 其中 Si,Sj 以及分母的算法，如前述 weighted lp distance 雷同。Vi 為 cluster i 的 cluster representative (prototype)。q 為 fuzzy factor，q=1 為 hard clustering；q>1 為 fuzzy clustering。與 FLD 算 class separability 很像，分母如 between class variance；分子如 within class variance。因此 DBm 值越小，代表 clustering 結果越好。

DB index for hard clusterings:

$$DB_m = \frac{1}{m} \sum_{i=1}^{m} \max_{j \neq i} \left[\frac{s_i + s_j}{\| v_i - v_j \|_q} \right]$$

The smaller, the better.

where

$$s_j = \left[\frac{1}{n_j} \sum_{x \in C_j} \| x - v_j \|^r \right]^{1/r}$$

DB index for fuzzy clusterings:

Use

$$s_j = \left[\frac{\sum_{i=1}^{N} u_{ij}^q \| x_i - v_j \|^r}{\sum_{i=1}^{N} u_{ij}^q} \right]^{1/r}$$

- Xi-Beni (XB) Index: 用於 fuzzy clustering。XB 值越小，代表 clustering 結果越好。

$$XB = \frac{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^2 \| \boldsymbol{x}_i - \boldsymbol{v}_j \|^2}{\min_{j \neq i} \| \boldsymbol{v}_i - \boldsymbol{v}_j \|^2}$$ The smaller, the better.

✓ 對於 fuzzy parition 的 crisp 程度。可以透過 partition coefficient 或是 partition entropy coefficient 判斷。它們的值越大，代表 partition 越 crisp (越不 fuzzy)。

Partition coefficient:

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}^2$$ The larger, the better.

Partition entropy coefficient:

$$PE = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(u_{ij}\log u_{ij}\right)$$ The larger, the better.

✓ 當我們有正確的 parition 資訊時，就可以對 clustering 結果好壞做評估。常見的方法有: Adjusted RAND Index (ARI)、Normalized Mutual Information (NMI)、clustering accuracy (Hungarian Algorithm)。

- RAND Index: 為 ARI 的簡易版。R 值越高，代表 clustering accuracy 越高。$R = \frac{a+b}{\binom{n}{2}}$。其中，n 為 samples 的總數、a 為 sample pairs 在同一個 cluster 不同 parititon 的總數、b 為 sample pairs 在不同 clusters 不同 parititon 的總數。例如:有兩個 clusterings 結果: {1},{2,3}與{1,2},{3}。R=(0+1)/3=1/3;

- Ajusted RAND Index (ARI): 為 RI 的調整版。根據下表的公式可以算出 ARI。ARI 值越高，代表 clustering accuracy 越高。其中 nij 代表 sample 在第一個 partition 中被分配到 cluster i 且同時在第二個 partition 中被分配到 cluster j 的個數。

	Y_1	Y_2	$\cdots$	Y_s	$Sums$
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
$Sums$	b_1	b_2	$\cdots$	b_s	

the adjusted index is:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}$$

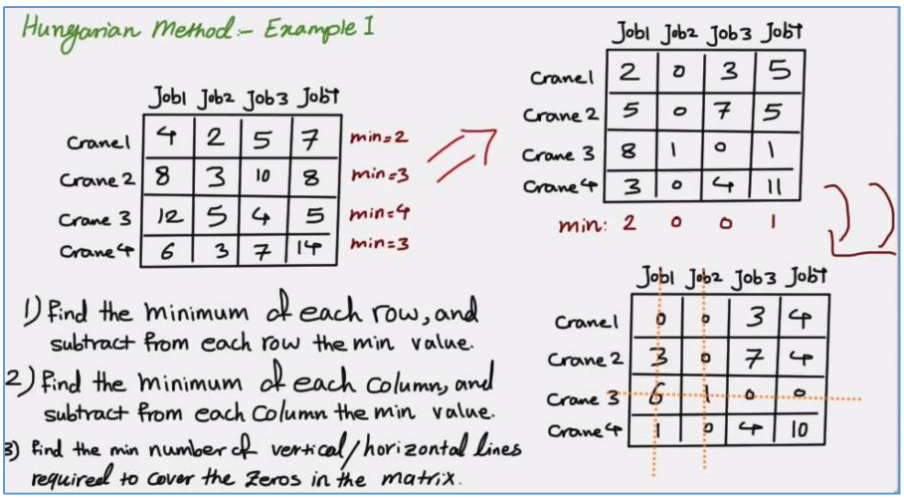
- Normalized Mutual Information (NMI): NMI 值越高，代表 clustering 結果越好。

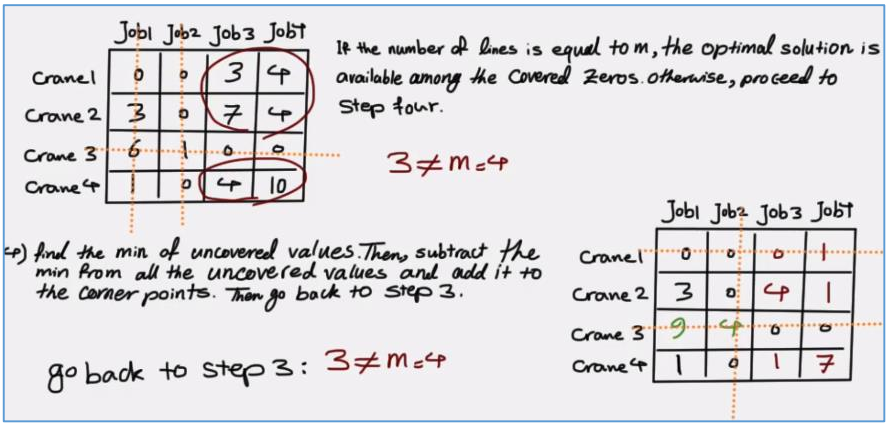
[Here is an example]

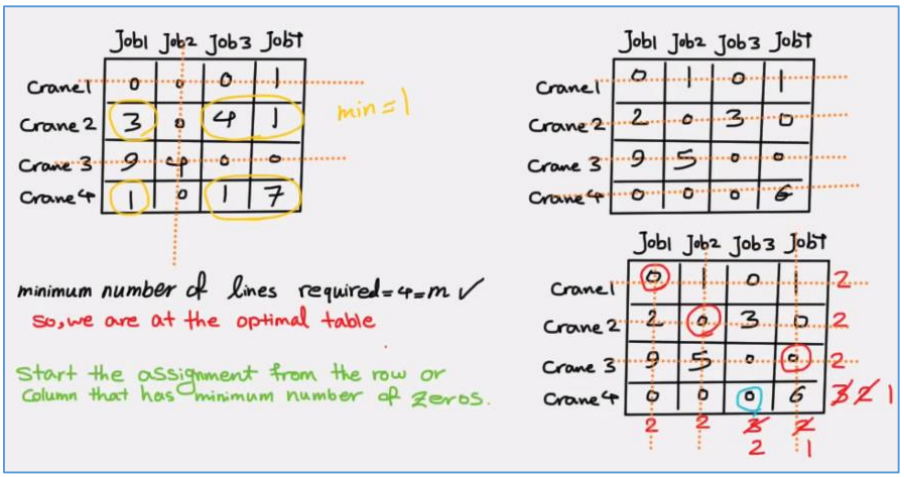
$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

- Hungarian Algorithm: 找到最低的分配成本。[slide reference]







Hidden Markov Model (HMM)

- ✓ 之前我們所學的 classifier (如 perceptron、Naïve Bayes 等等)都是假設 samples 之間互相獨立。然而對於許多實際應用上是不好的假設，因此有了 context-dependent classification 的議題產生，像是分析 speech、music、gesture 之問題等等。如 samples 之間都有關聯性，對於每個 sample 我們可以透過 bind 在此 sample 前所有的 observations (所謂的 combo-feature-vectors)，來得到更多的資訊加以分析? 然而，這會出現一些問題，例如:原本的 samples 維度低，但加上 observation 後維度就提高許多(curse of dimensionality)、不同的 class 可能有不同的 dimesion、太多的 redundancy 等等。Markov process 是一個有效率且很直觀的方法，可以用來記錄 samples 之間的關係。

- ✓ Markov process 的假設為: 在 time t 的 observeration，只跟前 n 個 observations 有關係，稱之為 n-order Markov process。

- ✓ Deterministic Markov process 為每個 state 由前一個 state 決定 (例如: 紅綠燈 traffic light)；Non-deterministic Markov process 與 determinstic 的概念相同，但對於 states 之間存在轉移的機率 (如 state i 有多少的機率會轉移到 state j)。aij 為 state i 到 state j 的機率、A 為所有 state 的轉移矩陣、對於所有的 state 到另一個 state 的機率加總為 1。

$$a_{ij} = P(j \mid i) \qquad A = [a_{ij}] \qquad \sum_j a_{ij} = 1$$

- ✓ 對於 state 無法直接可見的，我們稱之為 Hidden Markov process。一個常見的例子為: behind–the–cutain con–tossing。在這個例子中我們可以看到一連串硬幣正反面的結果，然而，無法得知是在哪個 state 下所產生的結果(每個 state 都有機會產生正反面結果)。要建立一個 Hidden Markov Model (HMM) 需要給定 state 的數量、states 之間的轉移機率 aij、state 初始的機率 P(i)以及在給定 state 情況下所產生的 observation 機率 P(x|j)。於許多實際應用中，observation 與 state 並不一定相同，以 text analysis 為例: observation 為獨立的 word，而 state 為 noun、verb、adjective 等等。

- ✓ HMM 又分為 discrete (離散) 與 continous (連續)。如果 observation 為 continous，在建立 dicrete HMM 時就會產生一些失真。對於建立 continuous HMM，它

的 observables 就會建立在 continuous space 上，P(x|j)就會是一個 pdf。雖然 continous HMM 會比較接近實際的應用，但計算上需耗費較多的時間。

✓ HMM 使用的範疇包含 evaluation、recognition、decoding、training problem。

➢ Evaluation problem: 給定 HMM 與 obervation sequence，評估此 obervations 由 HMM 產生的機率。 $P(X \mid S) = \sum_I p(X \mid I,S)P(I \mid S)$ 其中，S 為 HMM、X 為 observation sequence、I 為 state sequence。假設一個 HMM 為 fully connected 且有 K 的 state，observation sequence 長度為 t，如以暴力解(brute-force)，則需要遍巡 K^t 個長度為 t 的 path (時間複雜度為 $O(tK^t)$)。此方法耗費太多時間，我們可以用動態規劃來降低計算量。定義αt(j)為給定 HMM 下，從起始時刻到 time t ，observation seqeuece 為 $x_1x_2\ldots x_t$，且在 time t 時停在 state j 的機率。首先，需要初始化α1(j)，其關係式如下列圖示。[forward procdeure]

$$\alpha_t(j) = p(\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_t, i_t = j \mid S) \qquad \alpha_1(j) = p(\boldsymbol{x}_1 \mid j)P(j)$$

當到達 time t+1 時，其狀態與 time t 時的有關係。下列等式右方，左邊中括號項為在 time t 時所有 state 轉移到 state j 的機率總和；右邊項為 state j 產生下一個 observable (也就是 x_{t+1}) 的機率。

$$\alpha_{t+1}(j) = \left[\sum_i \alpha_t(i)P(j \mid i)\right] p(\boldsymbol{x}_{t+1} \mid j)$$

當到達最終時刻(time T)時，obervation sequence 由 HMM 產生的機率就是 time T 時所有 state 的機率加總。

$$P(X \mid S) = \sum_j \alpha_T(j)$$

- Recognition problem: 為 evaluation problem 延伸。給定多個 HMMs (每個 HMM 代表不同的 class)與 observation sequence，評估此 obervation 由哪個 HMM 產生的機率較高。以下圖為例，計算 P(HHH | HMM1)。

  α1(Left) = 0.5*0.8 = 0.4； α1(Right) = 0.5*0.2 = 0.1。

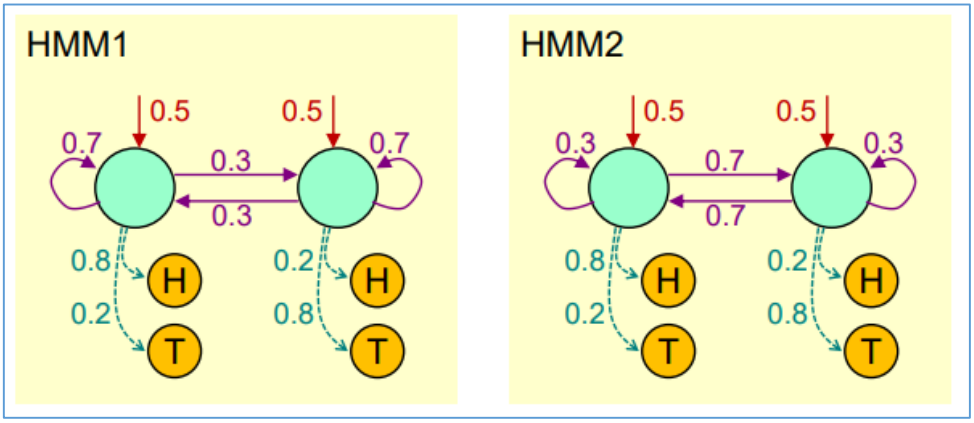
  α2(Left) = (0.4*0.7+0.1*0.3)*0.8 = 0.248

  α2(Right) = (0.4*0.3+0.1*0.7)*0.2 = 0.038

  α3(Left) = (0.248*0.7+0.038*0.3)*0.8 = 0.148

  α3(Right) = (0.248*0.3+0.038*0.7)*0.2 = 0.0202

  P(X|S) = α3(Left) + α3(Right) = 0.148 + 0.0202 = 0.1682



Observations X	$P(X \mid \text{HMM1})$	$P(X \mid \text{HMM2})$
H H H	**0.0168**	0.0096
H H H H H H	**0.0348**	0.0084
H H H T T T	**0.0218**	0.0101
H T H T H T	0.0084	**0.0348**
H T H H H T	0.0129	**0.0176**

- Decoding problem: 給定 HMM 與 obseration sequence，找到最有可能的 state sequence。 $I^* = \arg\max_I P(I \mid X,S)$ 如用暴力法(brute-force)，時間複雜度與 evalution problem 提及的相同。同理，我們可以使用動態規劃方法解決，一個常見的演算法 Viterbi algorithm。其方法與 evalution problem 很像，不同的是:每移動到下一步時，只選擇機率最大的那一步。因此，到達最後一步時也是選擇機率較大的結果。

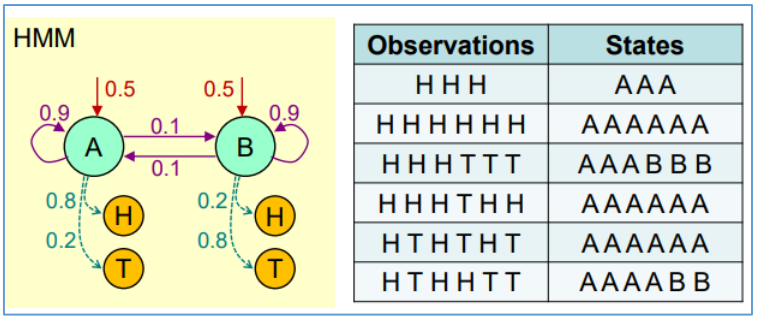
$$\delta_1(j) = p(\boldsymbol{x}_1 \mid j)P(j) \qquad \delta_{t+1}(j) = \left[\max_i \delta_t(i)P(j \mid i)\right] p(\boldsymbol{x}_{t+1} \mid j)$$

我們可以發現算出來機率與問題本身機率不同，但因為兩機率之間只差一個倍數 P(X | S)，且此倍數不變。因此找 P(I,X | S)的最大值就等同於找 P(I |X,S)的最大值。

$$P(I^*, X \mid S) = \max_j \delta_T(j)$$

$$P(I \mid X,S) = \frac{P(I,X,S)}{P(X,S)} = \frac{P(I,X,S)}{P(X \mid S)P(S)} = \frac{P(I,X \mid S)}{P(X \mid S)}$$

以下圖為例，找出 HTHHTT 最可能的 state sequence。



Observations	States
H H H	A A A
H H H H H H	A A A A A A
H H H T T T	A A A B B B
H H H T H H	A A A A A A
H T H T H T	A A A A A A
H T H H T T	A A A A B B

H → α1(A) = 0.5*0.8 = 0.4； α1(B) = 0.5*0.2 = 0.1 (0.4>0.1，紀錄 A)

T → α2(A) = (α1(A)*0.9)*0.2 = 0.072 (α1(A)*0.9>α1(B)*0.1，紀錄 A)

α2(B) = (α1(B)*0.9)*0.8 = 0.072 (α1(B)*0.9>α1(A)*0.1，紀錄 B)

H → α3(A) = (α2(A)*0.9)*0.8 = 0.05184 (α2(A)*0.9>α2(B)*0.1，紀錄 A)

α3(B) = (α2(B)*0.9)*0.2 = 0.01296 (α2(B)*0.9>α2(A)*0.1，紀錄 B)

H → α4(A) = (α3(A)*0.9)*0.8 =0.037348 (α3(A)*0.9>α3(B)*0.1，紀錄 A)

α4(B) = (α3(B)*0.9)*0.2 = 0.0023328 (α3(B)*0.9>α3(A)*0.1，紀錄 B)

T → α5(A) = (α4(A)*0.9)*0.2 = 0.00672264 (α4(A)*0.9>α4(B)*0.1，紀錄 A)

α5(B) = (α4(A)*0.1)*0.8 = 0.00298484 (α4(A)*0.1>α4(B)*0.9，紀錄 A)

T → α6(A) = (α5(A)*0.9)*0.2=0.0012100752(α5(A)*0.9>α5(B)*0.1，紀錄 A)

α6(B) = (α5(B)*0.9)*0.8=0.00214900848(α5(B)*0.9>α5(A)*0.1，紀錄 B)

到最後一步時，逆推每一步最大機率所記錄的 state，因此會得 AAAAAB。

- Training problem: 給定多個 observation sequences，用來估計 HMM，其中包含的 components 有: states 之間的轉移機率 aij、state 初始的機率 P(i) 以及給定 state 情況下所產生的 observation 機率 P(x|j)。一個常見的方法 Baum-Welch re-estimation method (EM algorithm)。首先，定義 visit 的機率(給定 HMM 與 observation sequence，time t 時在 state i 的機率)，state 轉移機率(給定 HMM 與 observation sequence，time t 時在 state i 且 time t+1 時在 state j 的機率)。

probability of "visit": $\gamma_t(i) = p(i_t = i \mid X, S)$

probability of "transition": $\xi_t(i,j) = p(i_t = i, i_{t+1} = j \mid X, S)$

βt(i)與前述定義的αt(i)不太一樣。βt(j)定義成給定 HMM 且 time t 時在 state i 的情況下，從 time t 到終止時刻 observation seqeucne 為 $x_{t+1}x_{t+2}\ldots x_T$ 的機率。

$$\beta_t(i) = p(\boldsymbol{x}_{t+1}\boldsymbol{x}_{t+2}\ldots\boldsymbol{x}_T \mid i_t = i, S)$$

$$\alpha_t(j) = p(\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_t, i_t = j \mid S)$$

βt(i)的關係是定義如下。

[backward procedure]

$$\beta_T(i) = 1, \quad \forall i$$

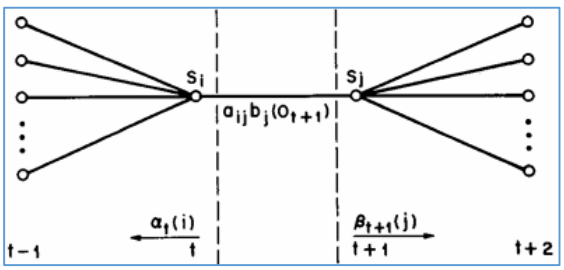
$$\beta_t(i) = \sum_j \beta_{t+1}(j)\, p(\boldsymbol{x}_{t+1} \mid j) P(j \mid i), \quad T > t \geq 1$$

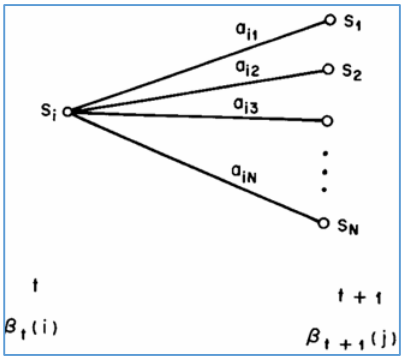
$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) P(j \mid i) p(\boldsymbol{x}_{t+1} \mid j) \beta_{t+1}(j)}{\sum_{i'}\sum_{j'} \alpha_t(i') P(j' \mid i') p(\boldsymbol{x}_{t+1} \mid j') \beta_{t+1}(j')}$$

$$r_t(i) = p(i_t = i \mid X, S) = \frac{p(i_t = i, X, S)}{p(X, S)} = \frac{p(i_t = i, X \mid S) p(S)}{p(X \mid S) p(S)} = \frac{p(i_t = i, X \mid S)}{p(X \mid S)} = \frac{\alpha_t(i)\beta_t(i)}{\sum \alpha_t(i)\beta_t(i)}$$

我們可以隨意初始化 HMM 的參數，並用上述的公式算出αt(i)、βt(i)、ξt(i)、γt(i)，再將求出的結果帶入下列的公式更新 HMM 參數。重複相同的步驟直到收斂。





$$P(i) = \gamma_1(i)$$

$$P(j \mid i) = \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \gamma_t(i)$$

$$P(\boldsymbol{x} \mid i) = \sum_{\substack{1 \leq t \leq T \\ x_t = x}} \gamma_t(i) \Big/ \sum_{1 \leq t \leq T} \gamma_t(i)$$