CS 412 Intro. to Data Mining

Chapter 10. Cluster Analysis: Basic Concepts and Methods

ฮีท์ x ไม่ฮู้ ๆ

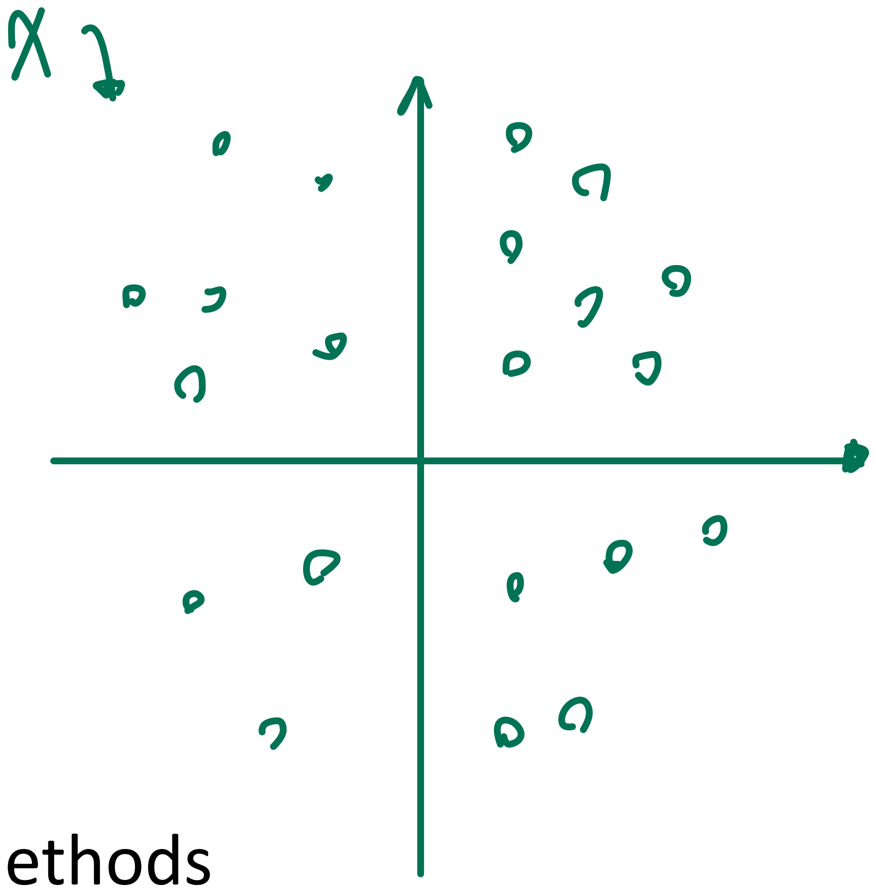
Jiawei Han, Computer Science, Univ. Illinois at Urbana-Champaign, 2017

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: An Introduction
- Partitioning Methods
- Hierarchical Methods
- Density- and Grid-Based Methods
- Evaluation of Clustering
- Summary

Partitioning-Based Clustering Methods

- Basic Concepts of Partitioning Algorithms
- The K-Means Clustering Method
- Initialization of K-Means Clustering
- The K-Medoids Clustering Method
- The K-Medians and K-Modes Clustering Methods
- The Kernel K-Means Clustering Method



Partitioning Algorithms: Basic Concepts

- Partitioning method: Discovering the groupings in the data by optimizing a specific objective function and iteratively improving the quality of partitions
- *K*-partitioning method: Partitioning a dataset ***D*** of ***n*** objects into a set of ***K*** clusters so that an objective function is optimized (e.g., the sum of squared distances is minimized, where c_k is the centroid or medoid of cluster C_k)
  - A typical objective function: **Sum of Squared Errors** (**SSE**)

$$SSE(C) = \sum_{k=1}^{K} \sum_{x_{i \in C_k}} || x_i - c_k ||^2$$

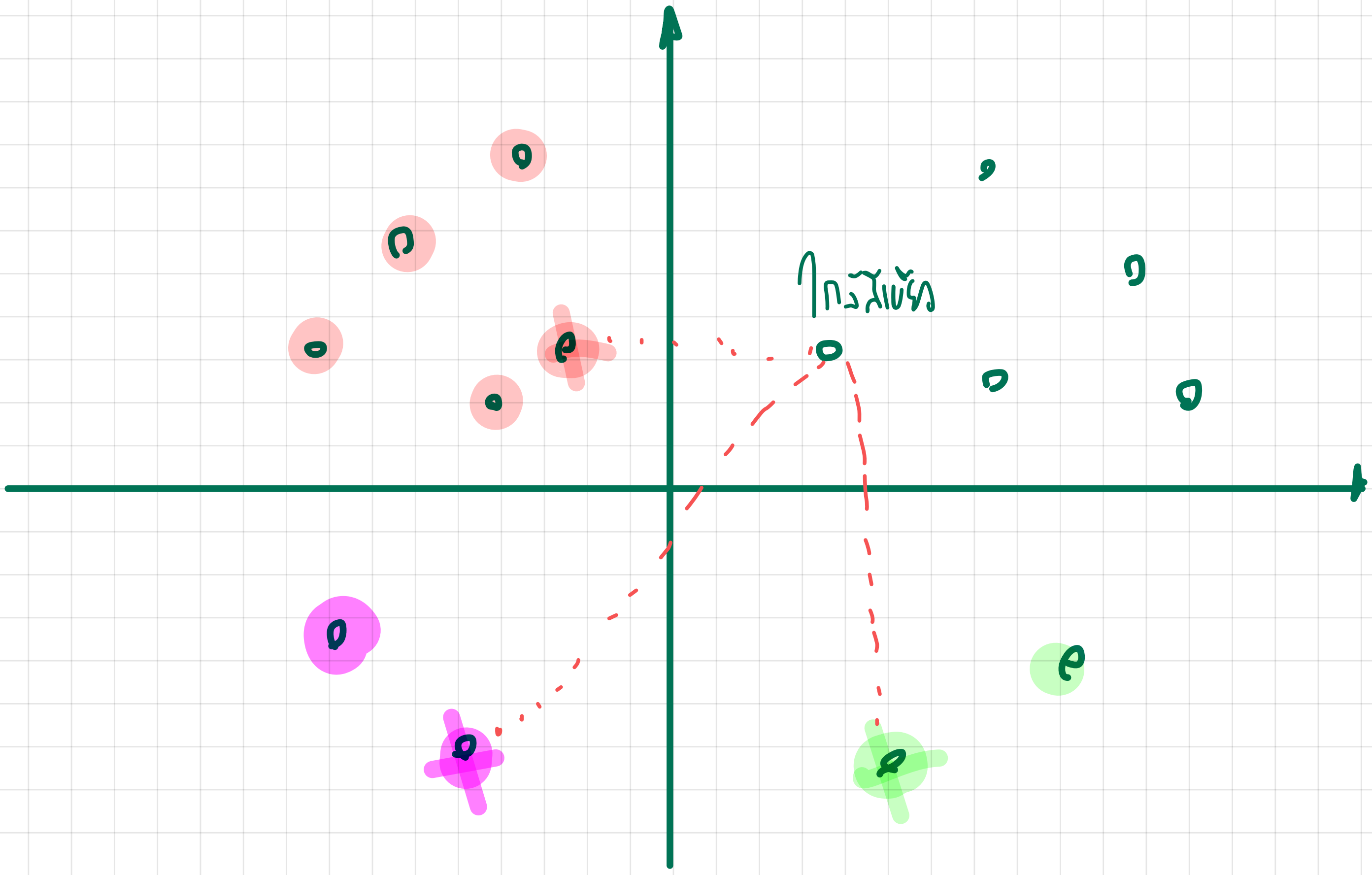
- Problem definition: Given *K*, find a partition of *K clusters* that optimizes the chosen partitioning criterion
  - Global optimal: Needs to exhaustively enumerate all partitions
  - Heuristic methods (i.e., greedy algorithms): *K-Means*, *K-Medians*, *K-Medoids*, etc.

คือ จน.ที่กำหนด(พารามิเตอร์)

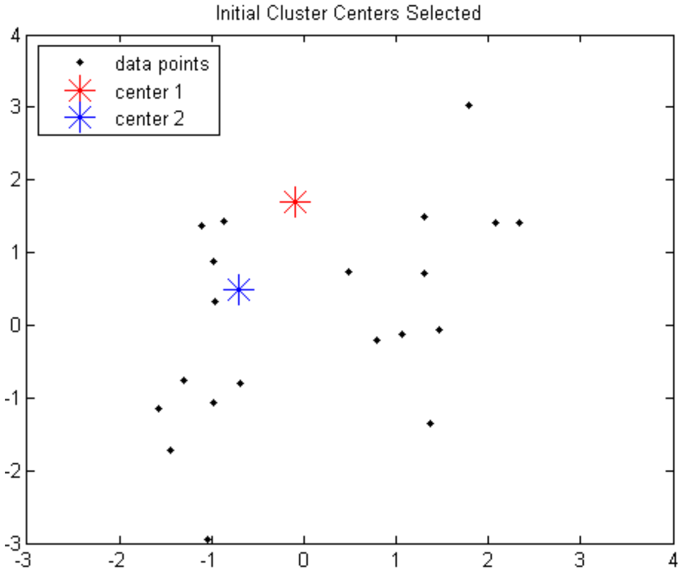
The *K-Means* Clustering Method

- *K-Means* (MacQueen'67, Lloyd'57/'82)
  - Each cluster is represented by the center of the cluster
- Given K, the number of clusters, the *K-Means* clustering algorithm is outlined as follows เริ่มมา กำหนดค่า K ก่อน ว่ามีกี่กลุ่ม
  - Select *K* points as initial centroids
  - **Repeat** → ทำซ้ำ จนกว่าจะครบรอบที่กำหนด
    - Form *K* clusters by assigning each point to its closest centroid
    - Re-compute the centroids (i.e., *mean point*) of each cluster
  - **Until** convergence criterion is satisfied
- Different kinds of measures can be used
  - Manhattan distance (L_1 norm), Euclidean distance (L_2 norm), Cosine similarity

K = 3

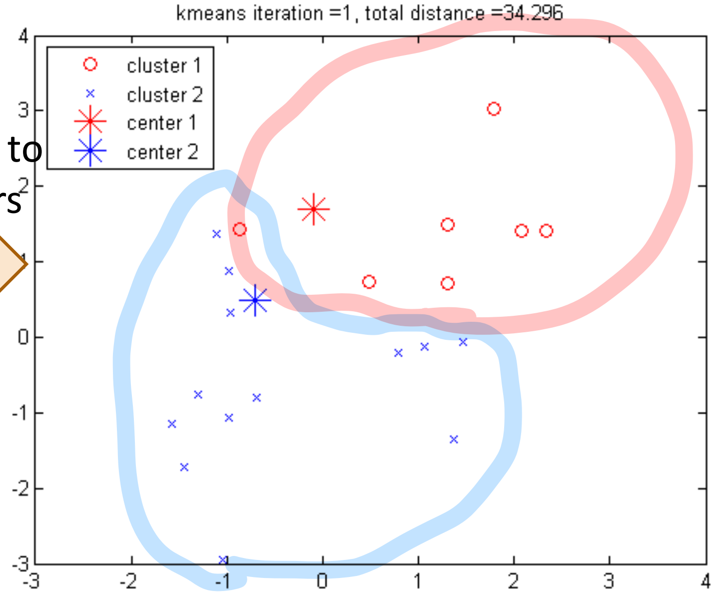


Example: *K-Means* Clustering

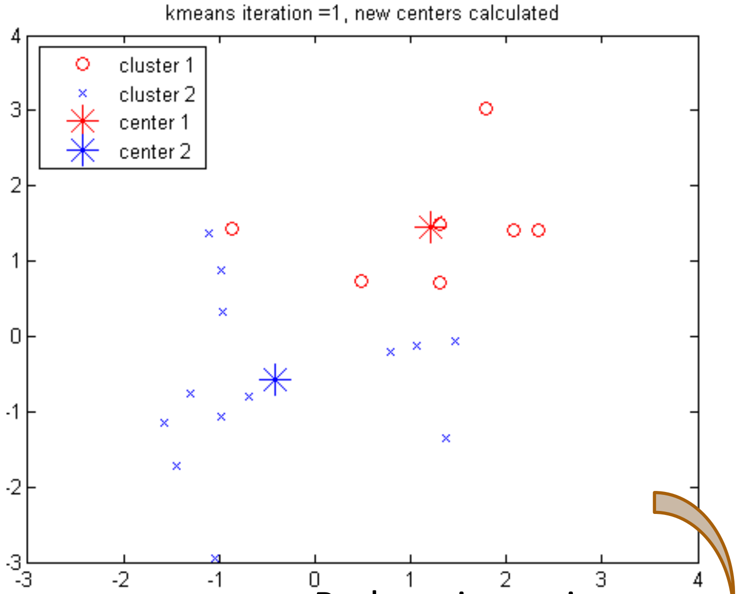


The original data points & randomly select *K* = 2 centroids

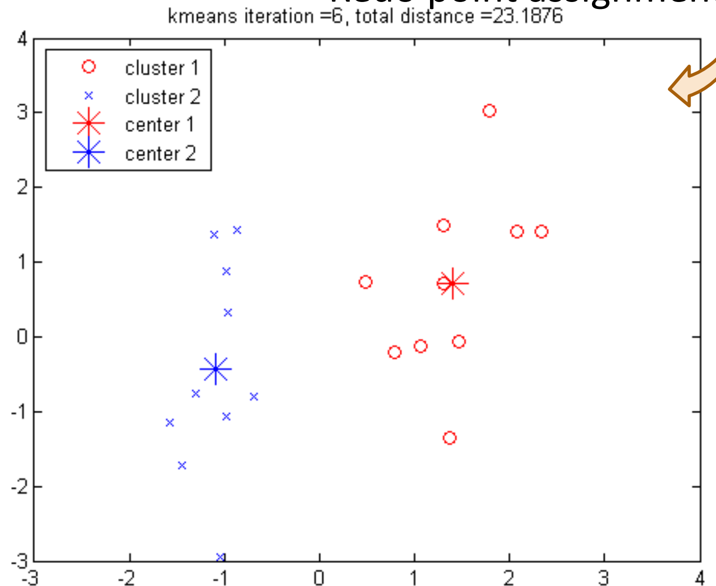
Assign points to clusters



Recompute cluster centers



Redo point assignment



Execution of the K-Means Clustering Algorithm

Select *K* points as initial centroids

Repeat

- Form *K* clusters by assigning each point to its closest centroid
- Re-compute the centroids (i.e., *mean point*) of each cluster

Until convergence criterion is satisfied

Discussion on the *K-Means* Method

- **Efficiency**: $O(tKn)$ where n: # of objects, K: # of clusters, and t: # of iterations
  - Normally, K, t << n; thus, an efficient method
- K-means clustering often ***terminates at*** **a** ***local optimal***
  - Initialization can be important to find high-quality clusters
- **Need to specify *K*,** the *number* of clusters, in advance
  - There are ways to automatically determine the "*best*" *K*
  - In practice, one often runs a range of values and selected the "*best*" *K* value
- **Sensitive to noisy data and *outliers***
  - Variations: Using K-medians, K-medoids, etc.
- K-means is applicable only to objects in a continuous n-dimensional space
  - Using the K-modes for ***categorical data***
- Not suitable to discover clusters with ***non-convex shapes***
  - Using density-based clustering, kernel *K*-means, etc.

Variations of *K-Means*

- There are many variants of the *K-Means* method, varying in different aspects
  - Choosing better initial centroid estimates
    - *K-means++, Intelligent K-Means, Genetic K-Means*

      To be discussed in this lecture
  - Choosing different representative prototypes for the clusters
    - *K-Medoids, K-Medians, K-Modes*

      To be discussed in this lecture
  - Applying feature transformation techniques
    - *Weighted K-Means, Kernel K-Means*

      To be discussed in this lecture

Chapter 10. Cluster Analysis: Basic Concepts and Methods

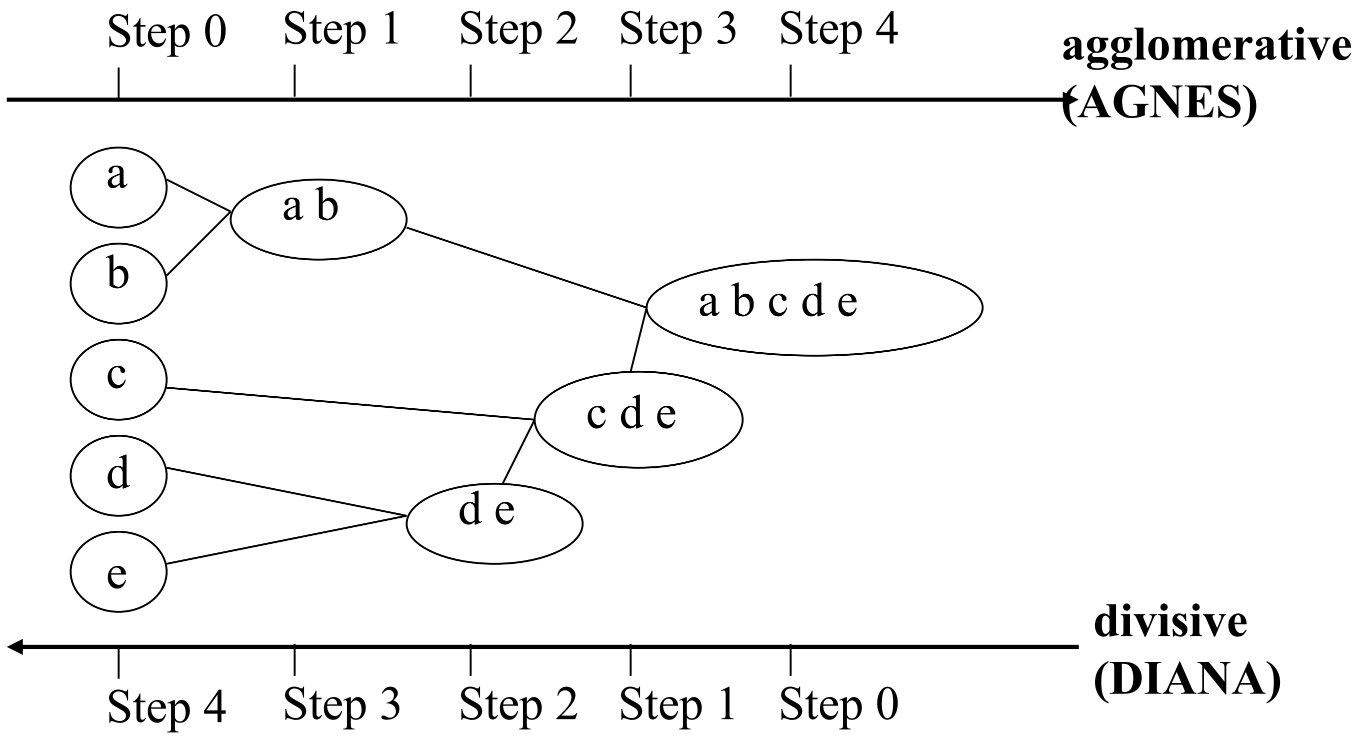
- Cluster Analysis: An Introduction
- Partitioning Methods
- Hierarchical Methods
- Density- and Grid-Based Methods
- Evaluation of Clustering
- Summary

Hierarchical Clustering Methods

- Basic Concepts of Hierarchical Algorithms
- Agglomerative Clustering Algorithms
- Divisive Clustering Algorithms
- Extensions to Hierarchical Clustering
- BIRCH: A Micro-Clustering-Based Approach
- CURE: Exploring Well-Scattered Representative Points
- CHAMELEON: Graph Partitioning on the KNN Graph of the Data
- Probabilistic Hierarchical Clustering

Hierarchical Clustering: Basic Concepts

- Hierarchical clustering
  - Generate a clustering hierarchy (drawn as a **dendrogram**)
  - Not required to specify ***K*,** the number of clusters
  - More deterministic
  - No iterative refinement
- Two categories of algorithms:

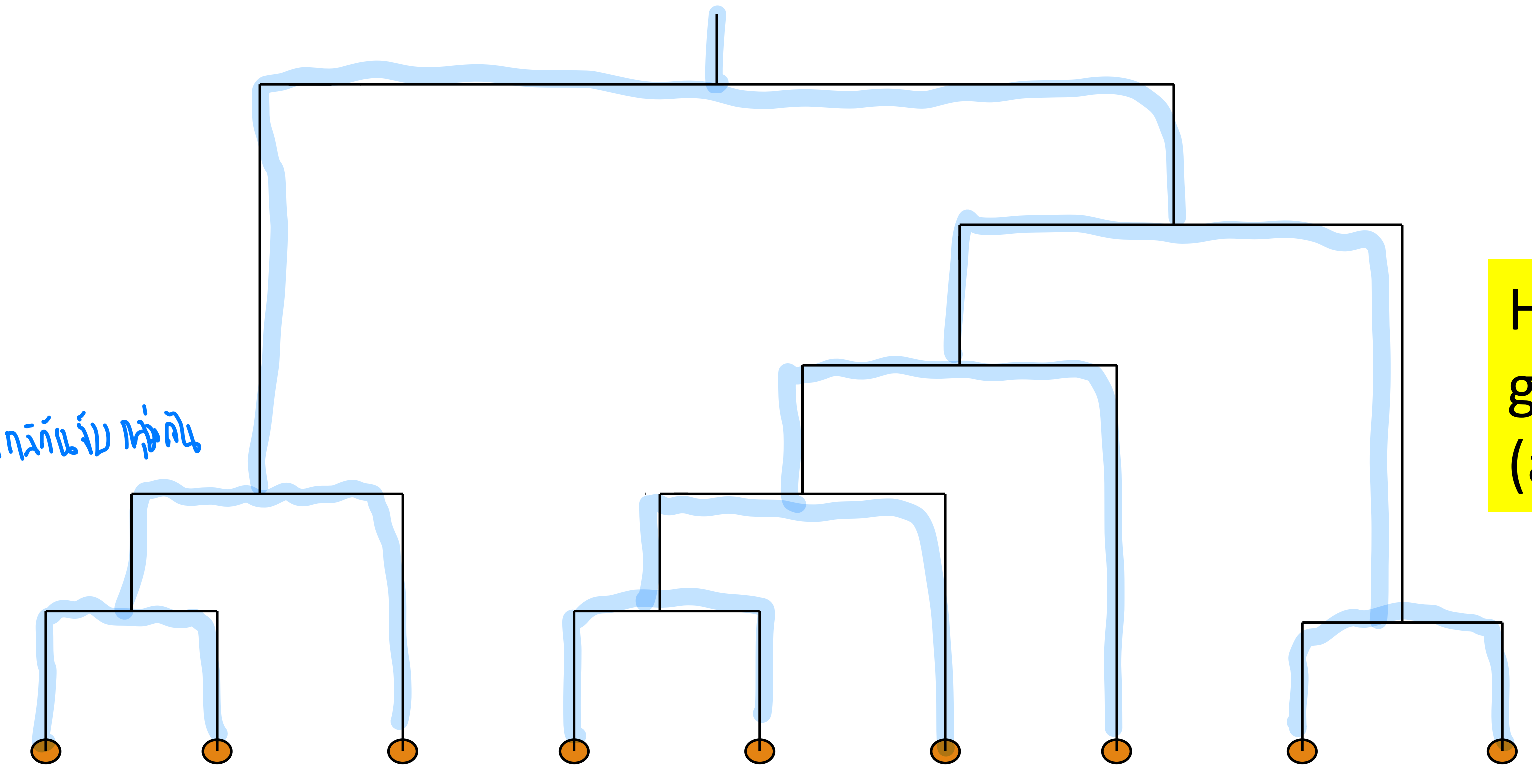


  - **Agglomerative**: Start with singleton clusters, continuously merge two clusters at a time to build a **bottom-up** hierarchy of clusters
  - **Divisive:** Start with a huge macro-cluster, split it continuously into two groups, generating a **top-down** hierarchy of clusters

Dendrogram: Shows How Clusters are Merged

- Dendrogram: Decompose a set of data objects into a tree of clusters by multi-level nested partitioning
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster

Dataใกล้กันจับ กลุ่มกัน



Hierarchical clustering generates a dendrogram (a hierarchy of clusters)

Chapter 10. Cluster Analysis: Basic Concepts and Methods

- Cluster Analysis: An Introduction
- Partitioning Methods
- Hierarchical Methods
- Density- and Grid-Based Methods
- Evaluation of Clustering
- Summary

Clustering Validation

- Clustering Validation: Basic Concepts
- Clustering Evaluation: Measuring Clustering Quality
- External Measures for Clustering Validation
  - I: Matching-Based Measures
  - II: Entropy-Based Measures
  - III: Pairwise Measures
- Internal Measures for Clustering Validation
- Relative Measures
- Cluster Stability
- Clustering Tendency

Clustering Validation and Assessment

- Major issues on clustering validation and assessment
  - **Clustering evaluation** ผลการทำ clustering ที่แบ่งได้จะมี 3 หัวข้อ
    - Evaluating the goodness of the clustering
  - **Clustering stability**
    - To understand the sensitivity of the clustering result to various algorithm parameters, e.g., # of clusters
  - **Clustering tendency**
    - Assess the suitability of clustering, i.e., whether the data has any inherent grouping structure

วิธีวัด

Measuring Clustering Quality

- **Clustering Evaluation**: Evaluating the goodness of clustering results
  - No commonly recognized best suitable measure in practice
- **Three categorization of measures**: External, internal, and relative
  - **External**: Supervised, employ criteria not inherent to the dataset
    - Compare a clustering against prior or expert-specified knowledge (i.e., the ground truth) using certain clustering quality measure
  - **Internal**: Unsupervised, criteria derived from data itself
    - Evaluate the goodness of a clustering by considering how well the clusters are separated and how compact the clusters are, e.g., silhouette coefficient
  - **Relative**: Directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm

Measuring Clustering Quality: External Methods

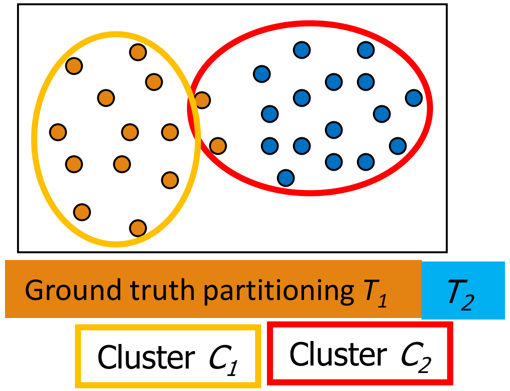
- Given the **ground truth** *T*, *Q*(*C*, *T*) is the **quality measure** for a clustering *C*
- *Q*(*C*, *T*) is good if it satisfies the following **four** essential criteria
  - **Cluster homogeneity**
    - The purer, the better
  - **Cluster completeness**
    - Assign objects belonging to the same category in the ground truth to the same cluster
  - **Rag bag better than alien**
    - Putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., "miscellaneous" or "other" category)
  - **Small cluster preservation**
    - Splitting a small category into pieces is more harmful than splitting a large category into pieces

C : (AAAA) (BABA) ✗

(AAAA) (BB) (AA) ✓

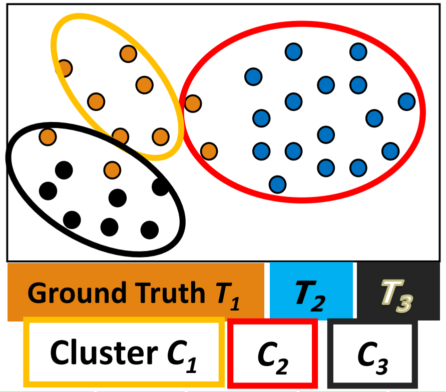
Commonly Used External Measures

- **Matching-based measures** (To be covered)
  - Purity, maximum matching, F-measure
- **Entropy-Based Measures**
  - Conditional entropy (To be covered)
  - Normalized mutual information (NMI) (To be covered)
  - Variation of information
- **Pairwise measures** (To be covered)
  - Four possibilities: True positive (TP), FN, FP, TN
  - Jaccard coefficient, Rand statistic, Fowlkes-Mallow measure
- **Correlation measures**
  - Discretized Huber static, normalized discretized Huber static



Matching-Based Measures (I): Purity vs. Maximum Matching

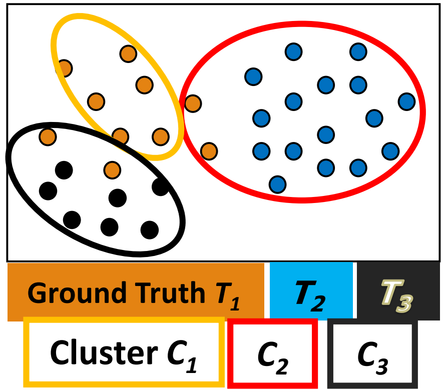
- **Purity**: Quantifies the extent that cluster C_i contains points only from one (ground truth) partition: $purity_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\}$
  - Total purity of clustering C:
    $$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1}^{k} \{n_{ij}\}$$
  - Perfect clustering if purity = 1 and $r = k$ (the number of clusters obtained is the same as that in the ground truth)
  - Ex. 1 (green or orange): $purity_1 = 30/50$; $purity_2 = 20/25$; $purity_3 = 25/25$; $purity = (30 + 20 + 25)/100 = 0.75$
  - Two clusters may share the same majority partition
- **Maximum matching**: Only one cluster can match one partition
  - Match: Pairwise matching, weight $w(e_{ij}) = n_{ij}$ $\quad \mathrm{w}(M) = \sum_{e \in M} w(e)$
  - Maximum weight matching: $match = \arg\max_{M} \{\frac{w(M)}{n}\}$
  - Ex2. (green) $match = purity = 0.75$; (orange) $match = 0.65 > 0.6$



C\T	T_1	T_2	T_3	Sum
C_1	0	20	30	50
C_2	0	20	5	25
C_3	25	0	0	25
m_j	25	40	35	100

C\T	T_1	T_2	T_3	Sum
C_1	0	30	20	50
C_2	0	20	5	25
C_3	25	0	0	25
m_j	25	50	25	100

Matching-Based Measures (II): F-Measure



- **Precision**: The fraction of points in C_i from the majority partition T_{j_i} (i.e., the same as purity), where j_i is the partition that contains the maximum # of points from C_i

$$prec_i = \frac{1}{n_i} \max_{j=1}^{k} \{n_{ij}\} = \frac{n_{ij_i}}{n_i}$$

  - Ex. For the green table
    - $prec_1$ = 30/50; $prec_2$ = 20/25; $prec_3$ = 25/25

C\T	T_1	T_2	T_3	Sum
C_1	0	20	30	50
C_2	0	20	5	25
C_3	25	0	0	25
m_j	25	40	35	100

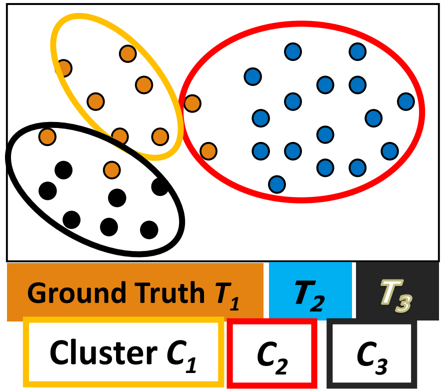
- **Recall**: The fraction of point in partition T_{j_i} shared in common with cluster C_i, where $m_{j_i} = |T_{j_i}|$

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}}$$

  - Ex. For the green table
    - $recall_1$ = 30/35; $recall_2$ = 20/40; $recall_3$ = 25/25
- **F-measure** for C_i: The harmonic means of $prec_i$ and $recall_i$: $F_i = \frac{2n_{ij_i}}{n_i + m_{j_i}}$
- F-measure for clustering C: average of all clusters: $F = \frac{1}{r}\sum_{i=1}^{r} F_i$
  - Ex. For the green table
    - F_1 = 60/85; F_2 = 40/65; F_3 = 1; F = 0.774

Entropy-Based Measures (I): Conditional Entropy

- Entropy of clustering *C*: $H(\mathcal{C}) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i}$ $\quad p_{C_i} = \frac{n_i}{n}$ (i.e., the probability of cluster C_i)
- Entropy of partitioning *T*: $H(\mathcal{T}) = -\sum_{j=1}^{k} p_{T_j} \log p_{T_j}$
- Entropy of *T* with respect to cluster C_i: $H(\mathcal{T}|C_i) = -\sum_{j=1}^{k} (\frac{n_{ij}}{n_i}) \log(\frac{n_{ij}}{n_i})$
- Conditional entropy of *T* with respect to clustering *C*: $H(\mathcal{T}|\mathcal{C}) = -\sum_{i=1}^{r} (\frac{n_i}{n}) H(\mathcal{T}|C_i) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log(\frac{p_{ij}}{p_{C_i}})$
  - The more a cluster's members are split into different partitions, the higher the conditional entropy
  - For a perfect clustering, the conditional entropy value is 0, where the worst possible conditional entropy value is *log k*

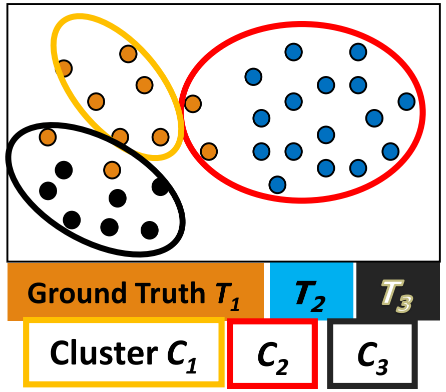


$$H(\mathcal{T}|\mathcal{C}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}(\log p_{ij} - \log p_{C_i}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log p_{ij} + \sum_{i=1}^{r} (\log p_{C_i} \sum_{j=1}^{k} p_{ij})$$

$$= -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log p_{ij} + \sum_{i=1}^{r} (p_{C_i} \log p_{C_i}) = H(\mathcal{C},\mathcal{T}) - H(\mathcal{C})$$

Entropy-Based Measures (II): Normalized Mutual Information (NMI)

- **Mutual information**:
  - Quantifies the amount of shared info between the clustering *C* and partitioning *T*
    $$I(C,T) = \sum_{i=1}^{r} \sum_{j=1}^{k} p_{ij} \log\left(\frac{p_{ij}}{p_{C_i} \cdot p_{T_j}}\right)$$
  - Measures the dependency between the observed joint probability p_{ij} of *C* and *T*, and the expected joint probability $p_{Ci} \cdot p_{Tj}$ under the independence assumption
  - When *C* and *T* are independent, $p_{ij} = p_{Ci} \cdot p_{Tj}$, $I(C, T) = 0$. However, there is no upper bound on the mutual information



- **Normalized mutual information** (NMI)

$$NMI(C,T) = \sqrt{\frac{I(C,T)}{H(C)} \cdot \frac{I(C,T)}{H(T)}} = \frac{I(C,T)}{\sqrt{H(C) \cdot H(T)}}$$

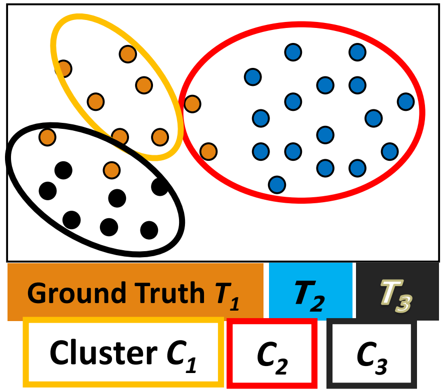
  - Value range of NMI: [0,1]. Value close to 1 indicates a good clustering

Pairwise Measures: Four Possibilities for Truth Assignment

- **Four possibilities** based on the agreement between cluster label and partition label
  - *TP*: true positive—Two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ belong to the same partition T, and they also in the same cluster C

    $$TP = |\{(\mathbf{x}_i, \mathbf{x}_j) : y_i = y_j \text{ and } \hat{y}_i = \hat{y}_j\}|$$

    where y_i: the true partition label, and $\hat{y}_i$: the cluster label for point $\boldsymbol{x}_i$



  - *FN*: false negative: $FN = |\{(\mathbf{x}_i, \mathbf{x}_j) : y_i = y_j \text{ and } \hat{y}_i \neq \hat{y}_j\}|$
  - *FP: false positive* $FP = |\{(\mathbf{x}_i, \mathbf{x}_j) : y_i \neq y_j \text{ and } \hat{y}_i = \hat{y}_j\}|$
  - *TN*: true negative $TN = |\{(\mathbf{x}_i, \mathbf{x}_j) : y_i \neq y_j \text{ and } \hat{y}_i \neq \hat{y}_j\}|$
- Calculate the four measures:

$$N = \binom{n}{2}$$

Total # of pairs of points

$$TP = \sum_{i=1}^{r}\sum_{j=1}^{k}\binom{n_{ij}}{2} = \frac{1}{2}\left(\left(\sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^{2}\right) - n\right) \qquad FN = \sum_{j=1}^{k}\binom{m_j}{2} - TP$$

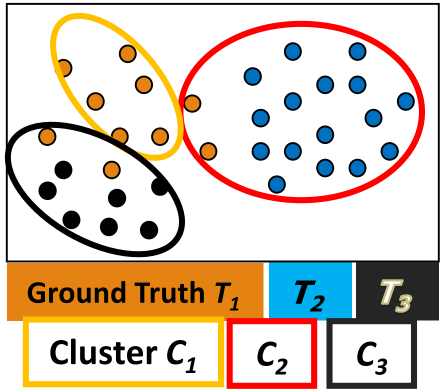
$$FP = \sum_{i=1}^{r}\binom{n_i}{2} - TP \qquad TN = N - (TP + FN + FP) = \frac{1}{2}\left(n^2 - \sum_{i=1}^{r} n_i^{2} - \sum_{j=1}^{k} m_j^{2} + \sum_{i=1}^{r}\sum_{j=1}^{k} n_{ij}^{2}\right)$$

Pairwise Measures: Jaccard Coefficient and Rand Statistic

- **Jaccard coefficient:** Fraction of true positive point pairs, but after ignoring the true negatives (thus asymmetric)
  - *Jaccard* = $TP/(TP + FN + FP)$ [i.e., denominator ignores TN]
  - Perfect clustering: *Jaccard* = 1
- **Rand Statistic**:
  - *Rand* = $(TP + TN)/N$
  - Symmetric; perfect clustering: *Rand* = 1
- **Fowlkes-Mallow Measure**:
  - Geometric mean of precision and recall

$$FM = \sqrt{prec \times recall} = \frac{TP}{\sqrt{(TP+FN)(TP+FP)}}$$

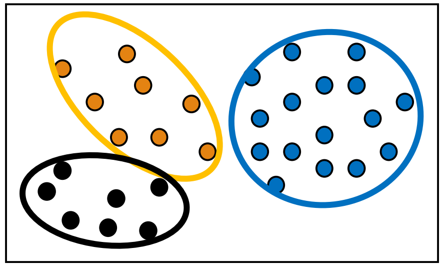
- Using the above formulas, one can calculate all the measures for the green table (leave as an exercise)



$C\backslash T$	T_1	T_2	T_3	Sum
C_1	0	20	30	50
C_2	0	20	5	25
C_3	25	0	0	25
m_j	25	40	35	100

Internal Measures (I): BetaCV Measure

- A trade-off in maximizing intra-cluster compactness and inter-cluster separation
- Given a clustering $C = \{C_1, . . ., C_k\}$ with k clusters, cluster C_i containing $n_i = |C_i|$ points
  - Let $W(S, R)$ be sum of weights on all edges with one vertex in S and the other in R
  - The sum of all the intra-cluster weights over all clusters: $W_{in} = \frac{1}{2}\sum_{i=1}^{k} W(C_i, C_i)$
  - The sum of all the inter-cluster weights: $W_{out} = \frac{1}{2}\sum_{i=1}^{k} W(C_i, \overline{C_i}) = \sum_{i=1}^{k-1}\sum_{j>i} W(C_i, C_j)$
  - The number of distinct intra-cluster edges: $N_{in} = \sum_{i=1}^{k}\binom{n_i}{2}$
  - The number of distinct inter-cluster edges: $N_{out} = \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} n_i n_j$



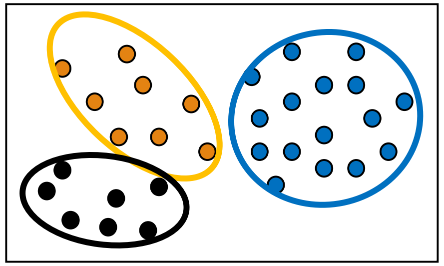
- **Beta-CV measure**: $BetaCV = \frac{W_{in} / N_{in}}{W_{out} / N_{out}}$
  - The ratio of the mean intra-cluster distance to the mean inter-cluster distance
  - The smaller, the better the clustering

Internal Measures (II): Normalized Cut and Modularity

- **Normalized cut**: $NC = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{vol(C_i)} = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{W(C_i, V)} = \sum_{i=1}^{k} \frac{W(C_i, \overline{C_i})}{W(C_i, C_i) + W(C_i, \overline{C_i})} = \sum_{i=1}^{k} \frac{1}{\frac{W(C_i, C_i)}{W(C_i, \overline{C_i})} + 1}$

  where $vol(C_i) = W(C_i, V)$ is the volume of cluster C_i

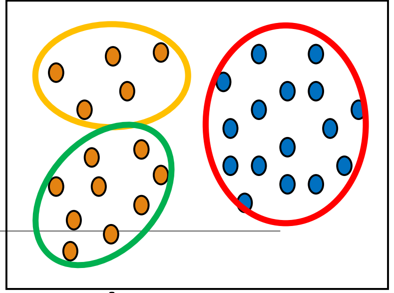
  - The higher normalized cut value, the better the clustering



- **Modularity** (for graph clustering) $Q = \sum_{i=1}^{k} \left(\frac{W(C_i, C_i)}{W(V, V)} - \left(\frac{W(C_i, V)}{W(V, V)} \right)^2 \right)$
  - Modularity Q is defined as

    where $W(V,V) = \sum_{i=1}^{k} W(C_i, V) = \sum_{i=1}^{k} W(C_i, C_i) + \sum_{i=1}^{k} W(C_i, \overline{C_i}) = 2(W_{in} + W_{out})$
  - Modularity measures the difference between the observed and expected fraction of weights on edges within the clusters.
  - The smaller the value, the better the clustering—the intra-cluster distances are lower than expected

Relative Measure



- Relative measure: Directly compare different clusterings, usually those obtained via different parameter settings for the same algorithm
- **Silhouette coefficient** as an **internal measure**: Check cluster cohesion and separation
  - For each point ***x**$_i$*, its silhouette coefficient s_i is: $s_i = \dfrac{\mu_{out}^{\min}(\mathbf{x}_i) - \mu_{in}(\mathbf{x}_i)}{\max\{\mu_{out}^{\min}(\mathbf{x}_i), \mu_{in}(\mathbf{x}_i)\}}$

    where $\mu_{in}(\mathbf{x}_i)$ is the mean distance from ***x**$_i$* to points in its own cluster

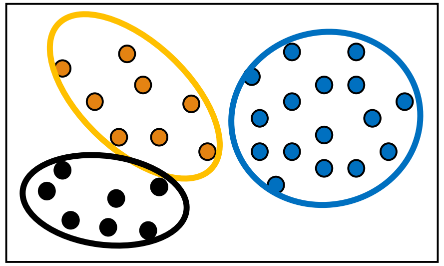
    $\mu_{out}^{\min}(\mathbf{x}_i)$ is the mean distance from ***x**$_i$* to points in its closest cluster
  - Silhouette coefficient (*SC*) is the mean values of s_i across all the points: $SC = \frac{1}{n}\sum_{i=1}^{n} s_i$
  - *SC* close to +1 implies good clustering
    - Points are close to their own clusters but far from other clusters
- **Silhouette coefficient** as a **relative measure**: Estimate the # of clusters in the data

$$SC_i = \frac{1}{n_i}\sum_{x_j \in C_i} s_j$$

Pick the *k* value that yields the best clustering, i.e., yielding high values for *SC* and SC_i $(1 \le i \le k)$

Cluster Stability

- Clusterings obtained from several datasets sampled from the same underlying distribution as ***D*** should be similar or "stable"



- Typical approach:
  - Find good parameter values for a given clustering algorithm
- Example: Find a good value of k, the correct number of clusters
- A **bootstrapping approach** to find the best value of k (judged on stability)
  - Generate t samples of size n by sampling from ***D*** with replacement
  - For each sample $\boldsymbol{D}_i$, run the same clustering algorithm with k values from 2 to k_{max}
  - Compare the distance between all pairs of clusterings $C_k(\boldsymbol{D}_i)$ and $C_k(\boldsymbol{D}_j)$ via some distance function
    - Compute the expected pairwise distance for each value of k
  - The value k^* that exhibits the least deviation between the clusterings obtained from the resampled datasets is the best choice for k since it exhibits the most stability