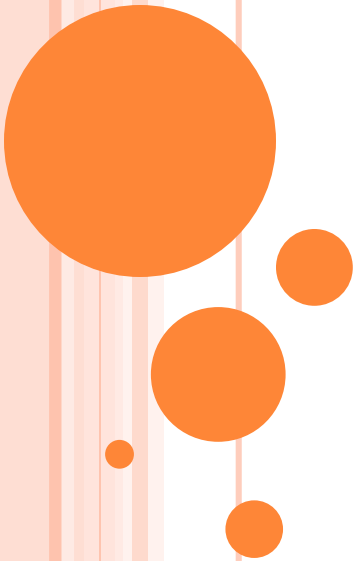


CLUSTERING

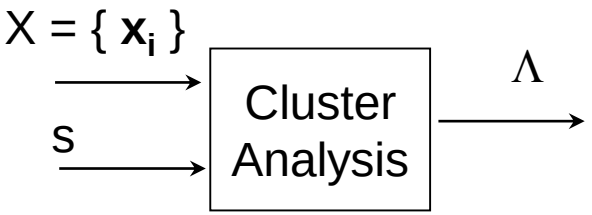
Bor-shen Lin

bslin@cs.ntust.edu.tw

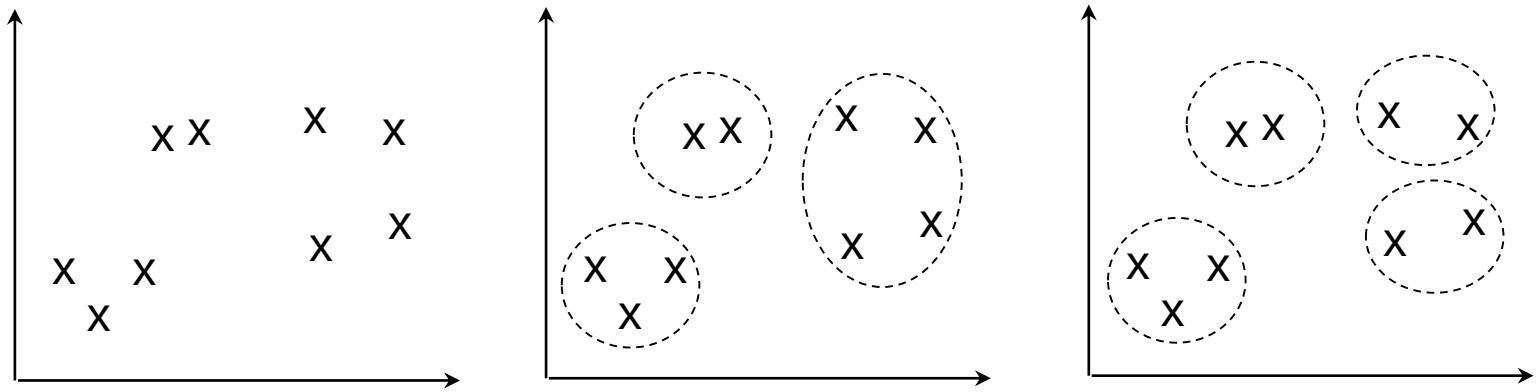
http:www.cs.ntust.edu.tw/~bslin

Cluster Analysis

- A form of unsupervised learning
- Automatic classification of samples into a number of groups using a measure of association
  - Data in each group are similar
- Input
  - A set of samples: X
  - A measure of similarity: s
- Output
  - A number of groups
  - A partition $\Lambda = \{ G_1, G_2, \ldots, G_N \}$
    $G_1 \cup G_2 \cup \ldots \cup G_N = X$ and $G_i \cap G_j = \phi$



Example



Clustering for anything

- Clustering for customers
- Clustering for documents
- Clustering for words
- Clustering for images
- Clustering for melodies
- Clustering for objects
- Clustering for design patterns
- Clustering for styles

Clustering of Design Patterns

Basic Concept of Cluster Analysis

- Sample
  - A point in a multi-dimensional space
- Objective of Cluster Analysis
  - Construct decision boundaries (classification surfaces) based on unlabeled training data set

    (Unsupervised learning)
  - Appropriate for exploration of interrelationships among samples to make a preliminary assessment of the sample structure

Difficulties of Clustering

- Shape and size of data
- Number of clusters
  - Determined according to desired resolution
  - There might be different results for the same data (different # of clusters)
- Human can visualize the data with dimension at most 3. Visualization of high dimensional data need to rely on clustering algorithms

Features

- Quantitative features
  - Continuous values: R, R^+
  - Discrete values: { 0, 1 }, Z
  - Interval values: $\{ x \leqq 20, 20 < x \leqq 40, x > 40 \}$
- Qualitative features
  - Nominal or unordered: color is “blue” or “red”
  - Ordinal: military rank: “general” and “colonel”

Similarity/Distance Measure

- Similarity
  - Symmetric: $s(\mathbf{x}, \mathbf{x}') = s(\mathbf{x}', \mathbf{x})$ for all $\mathbf{x}$, $\mathbf{x}'$ in X
  - Normalized: $0 \leqq s(\mathbf{x}, \mathbf{x}') \leqq 1$
- Distance (dissimilarity) measure
  - $d(\mathbf{x}, \mathbf{x}') \geqq 0$ ($d(\mathbf{x}, \mathbf{x}) = 0$)
  - $d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x})$
  - Called **metric** distance measure if **triangular inequality** holds:

    $d(\mathbf{x}, \mathbf{x}'') \leqq d(\mathbf{x}, \mathbf{x}') + d(\mathbf{x}', \mathbf{x}'')$

Metrics for Continuous Values

- Euclidean distance
  - $d_2(\mathbf{x_i}, \mathbf{x_j}) = [\Sigma_k(x_{ik} - x_{jk})^2]^{1/2}$
- Block distance (L1 metric)
  - $d_1(\mathbf{x_i}, \mathbf{x_j}) = \Sigma_k \mid x_{ik} - x_{jk} \mid$
- Minkowski metric
  - $d_p(\mathbf{x_i}, \mathbf{x_j}) = [\Sigma_k \mid x_{ik} - x_{jk} \mid^p]^{1/p}$
- Cosine similarity
  - $s_{cos}(\mathbf{x_i}, \mathbf{x_j}) = [\Sigma_k(x_{ik} \cdot x_{jk})] / [\Sigma_k x_{ik}^2 \cdot \Sigma_k x_{jk}^2]^{1/2}$
  - $s_{cos}(\mathbf{x_i}, \mathbf{x_j}) = 1$ if $x_i = c \cdot x_j$, $c > 0$

Metrics for Discrete Values

- $\mathbf{x_i}$, $\mathbf{x_j}$

  (1, 1) : a, (1, 0) : b

  (0, 1): c, (0, 0): d
- **Simple Matching Coefficient (SMC)**
  - $S_{smc}(\mathbf{x_i}, \mathbf{x_j}) = (a + d) / (a + b + c + d)$
- Jaccard Coefficient
  - $S_{jc}(\mathbf{x_i}, \mathbf{x_j}) = a / (a + b + c)$
- Rao's Coefficient
  - $S_{rc}(\mathbf{x_i}, \mathbf{x_j}) = a / (a + b + c + d)$

Example

- $\mathbf{x}_1 = (0,0,1,1,0,1,0,1)$

  $\mathbf{x}_2 = (0,1,1,0,0,1,0,0)$

  → a = 2 for (1, 1)

  b = 2 for (1, 0)

  c = 1 for (0, 1)

  d = 3 for (0, 0)

  → $S_{smc}(\mathbf{x}_1, \mathbf{x}_2) = 5 / 8$

  $S_{jc}(\mathbf{x}_1, \mathbf{x}_2) = 2 / 5$

  $S_{rc}(\mathbf{x}_1, \mathbf{x}_2) = 2 / 8$

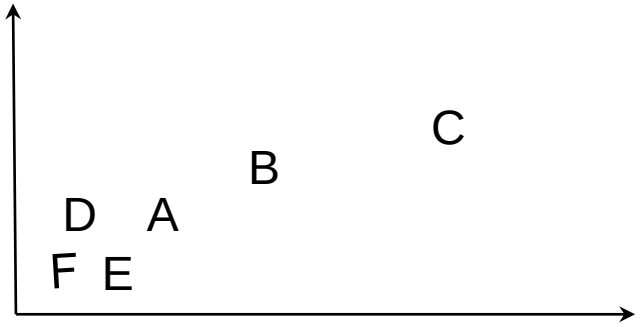
Advanced Distance Measures

- Consider the effect of the context (relative distance)
- Mutual neighbor distance (MND)
  $MND(\mathbf{x}_i, \mathbf{x}_j) = NN(\mathbf{x}_i, \mathbf{x}_j) + NN(\mathbf{x}_j, \mathbf{x}_i)$
  $NN(\mathbf{x}_i, \mathbf{x}_j)$: the neighbor number of $\mathbf{x}_i$ with respect to $\mathbf{x}_j$
  $\mathbf{x}_i$ is the n-th closest point for $\mathbf{x}_j$

NN(A, B) = 1, NN(B, A) = 1

NN(B, C) = 1, NN(C, B) = 2

MND(A,B) = 2, MND(B,C) = 3



NN(A, B) = 1, NN(B, A) = 4

NN(B, C) = 1, NN(C, B) = 2

MND(A,B) = 5, MND(B,C)=3

Positions of A & B are the same, but with different distances!

Clustering Algorithms

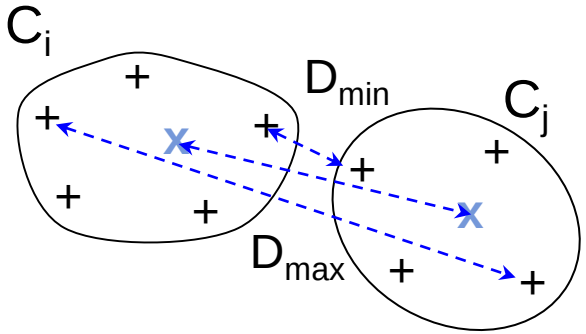
- Hierarchical Clustering Algorithm
  - Producing the hierarchy (dendrogram)
- Partitional Clustering algorithm
  - Producing the partition of the data

Hierarchical Clustering Algorithms

- Divisible algorithm
  - A Top-down process
  - Example: CART for clustering
    - Divided by discrete properties
  - Starts from the entire set of samples
  - Divides it into a partition of subsets
- Agglomerative algorithm (aggregation)
  - A bottom-up process
    - Rain drops (small → big)
  - Regards each object as a cluster initially
  - The clusters are merged into larger clusters
  - A dendrogram is constructed

Distance Between Clusters

- $D_{min}(C_i, C_j) = \min |\mathbf{x_i} - \mathbf{x_j}|$, for $\mathbf{x_i}$ in C_i, $\mathbf{x_j}$ in C_j
- $D_{mean}(C_i, C_j) = |\mathbf{m_i} - \mathbf{m_j}|$, for centroids $\mathbf{m_i}$ and $\mathbf{m_j}$
- $D_{avg}(C_i, C_j) = 1/(n_i n_j) \Sigma_i \Sigma_j |\mathbf{x_i} - \mathbf{x_j}|$, for $\mathbf{x_i}$ in C_i, $\mathbf{x_j}$ in C_j
- $D_{max}(C_i, C_j) = \max |\mathbf{x_i} - \mathbf{x_j}|$, for $\mathbf{x_i}$ in C_i, $\mathbf{x_j}$ in C_j

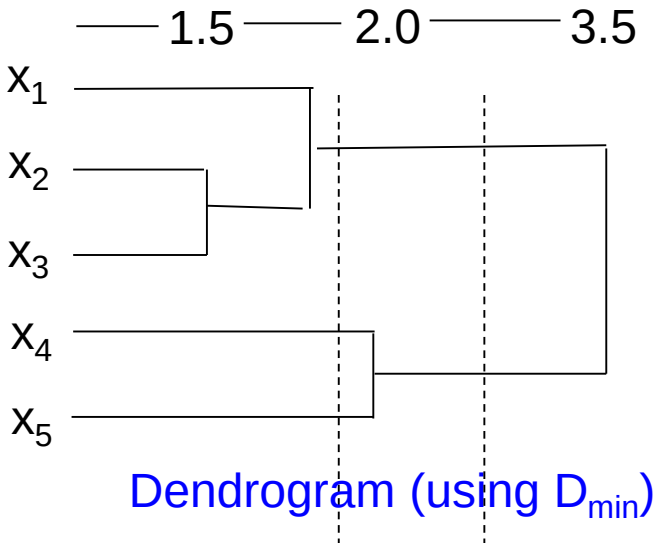


Agglomerative Clustering

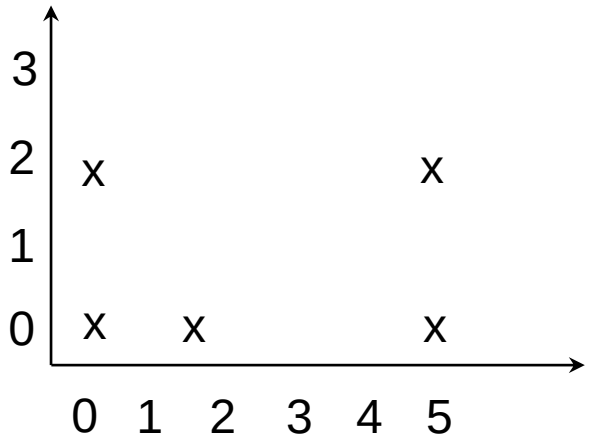
1. Place each sample in its own cluster. Construct a list of inter-cluster distances for all pairs of samples, and sort this list in ascending order.
2. Step through the sorted list of distances, forming for each distinct threshold value d_k a graph of the samples where pairs of samples closer than d_k are connected into a new cluster by a graph edge. If all the samples are members of a connected graph, stop. Otherwise, repeat this step.
3. The output of the algorithm is a nested hierarchy of graphs, which can be cut at the desired dissimilarity level forming a partition (clusters) identified by simple connected components in the corresponding subgraph.

EXAMPLE

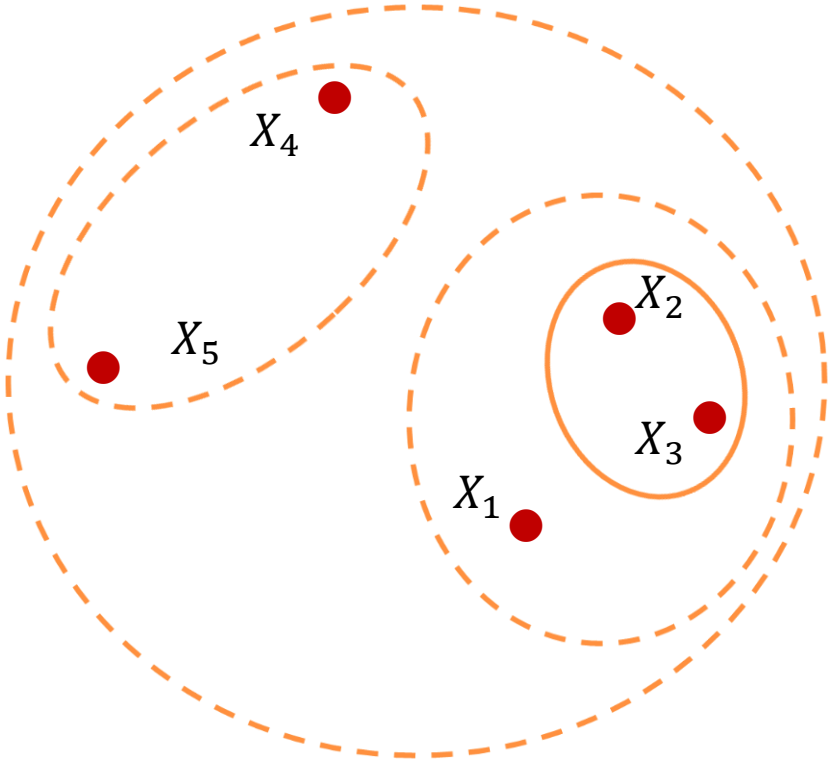
- Samples
  - x_1=(0,2), x_2=(0,0), x_3=(1.5,0), x_4=(5,0), x_5=(5,2)
- $d(x_1, x_2) = 2$, $d(x_1, x_3) = 2.5$, $d(x_1, x_4) = 5.39$, $d(x_1, x_5) = 5$, $d(x_2, x_3) = 1.5$, $d(x_2, x_4) = 5$, $d(x_2, x_5) = 5.29$, $d(x_3, x_4) = 3.5$, $d(x_3, x_5) = 4.03$, $d(x_4, x_5) = 2$

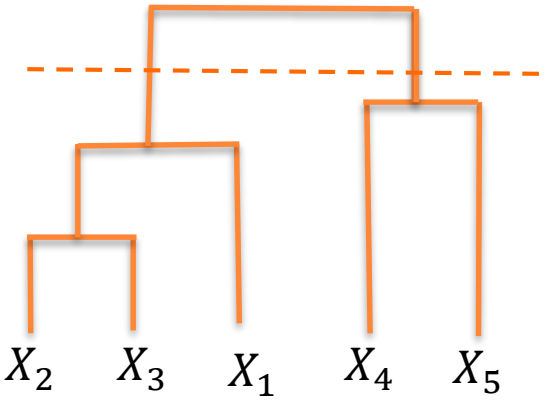


Dendrogram (using D_{min})



Merging of Clusters





Chameleon Algorithm

- Basic idea: Improve the clustering quality by using a more elaborate criterion when merging two clusters
  - C_i and C_j will be merged if the interconnectivity and closeness of the merged cluster is very similar to that of C_i and C_j before merging
- Min-cut on a graph (of cluster C_i)
  - Partitioning a graph into two parts of close, equal size such that the total weight of the edges being cut is minimized.
  - Total weight of the edges: stand for total interconnectivity (c.f. inner links in political parties)
  - min-cut: minimize loss of connectivity for the graph after being cut
  - Interconnectivity $I(C_i)$: total weight of edges being cut (total loss)
  - Closeness $C(C_i)$: average weight of edges being cut (average loss)

CHAMELEON ALGORITHM

1. Divide all data into *highly dense* sub-clusters.
   - Construct initial graph G=(V,E) by KNN with weighted edge $e(v_i,v_j)$: closeness between two samples
   - Recursively partition G into many *small, unconnected subgraphs* by doing min-cut
   - Stopped when certain criteria are satisfied
2. Merge sub-clusters into larger clusters.
   - For C_i and C_j, compute $RI(C_i, C_j)$ and $RC(C_i, C_j)$
     - Relative interconnectivity
       $RI(C_i, C_j) \equiv I(C_i \cup C_j) / [0.5 \cdot (I(C_i)+I(C_j))]$
     - Relative closeness
       $RC(C_i, C_j) \equiv C(C_i \cup C_j) / [0.5 \cdot (C(C_i)+C(C_j))]$
   - $s(C_i, C_j) \equiv RC(C_i, C_j) \cdot RI(C_i, C_j)^{\alpha}$, $0 \leqq \alpha \leqq 1$
     Minimizing the decrease after being merged

Partitional Clustering

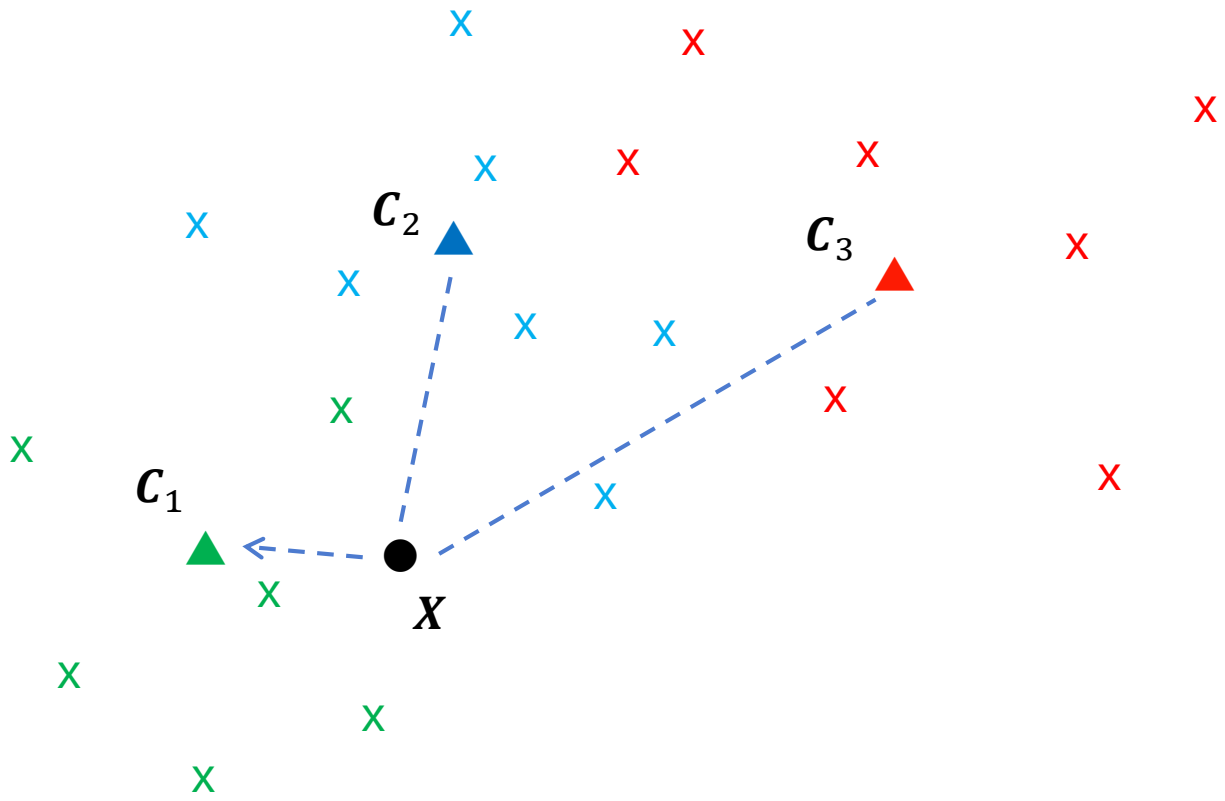
- Why?
  - Construction of dendrogram is computationally complex for large data set
- Global criterion
  - Euclidean square-error measure E^2 for k-th cluster

    $E^2 = \Sigma_k d_k$, $d_k = \Sigma_i (x_{ik} - M_k)^2$, $\mathbf{m_k} = (1/n_k)\Sigma_i \mathbf{x_{ik}}$
  - Euclidean distance cannot be used if the dimensions have different physical meaning → using stochastic model
- Local criterion
  - Minimal mutual neighbor distance (MND)
  - Utilizing the local structure or context in the data

K-Means Clustering Algorithm

- A partitional clustering algorithm employing a square-error criterion
- Simplest & most commonly used
  - Easy to implement
  - Time and space complexity is relatively small
  - Gives surprisingly good result if the clusters are compact, hyperspherical in shape, and well separated in the feature space
- Work well for data that are far apart
- Equivalent to Kohonen net in ANN
- Similar methods (e.g. C-means, K-mediods)

Redistribution of Samples



K-Means Algorithm

1. Select an *initial partition* with k clusters containing randomly chosen samples, and compute the centroids of the clusters.
2. Generate a new partition by (re-distributing) assigning each sample to its closest cluster center

   $k^* = argmin_k\, d(\boldsymbol{x}, \boldsymbol{m_k})$ for every $\boldsymbol{x}$
3. Compute new centroids of the clusters

   $\boldsymbol{m_k} = (1/n)\Sigma \boldsymbol{x_i}$ for cluster k
4. Repeat steps 2 and 3 until an optimum value of the criterion function is found.

Characteristics of K-Means

- Time complexity: O(nkl)

  n: number of samples

  k: number of clusters

  l: number of iterations
- Space complexity: O(k+n)
- Order-independent
  - It generates the same partition of the data at the end of the partitioning process irrespective of the order in which the samples are presented to the algorithm

Order-dependent Incremental Clustering

1. Initially, there is no cluster.
2. For every point, compute the distance between the point and any existing cluster.
3. If the minimum distance does exist, raise a new cluster with the point and go to step 2. Otherwise, go to step 3.
4. If the minimum distance is smaller than a threshold, add that point to the corresponding cluster. Otherwise raise a new cluster with the point.
5. Go to step 2.

- Order dependent
  - The order that data are presented influences the result

Major Problems of K-Means

- Sensitive to the selection of the initial partition
- May converge to a local minimum
- Lack of available guidelines for
  - Choosing the initial partition
  - Adjusting the number of clusters
  - Selecting the stopping criterion
- Very sensitive to noise and outlier data points
  - Every point has the same contribution to the centroid
  - K-mediods method: remove the outliers when computing the centroids

    → less sensitive to noise and outliers
- Based on global distance
  - Might not work well for the data with locally connected patterns

K-Means for Categorical Data

- Centroids for clusters can NOT be computed
  - $d(\boldsymbol{x}, \boldsymbol{m}_k)$ cannot be calculated without $\mathbf{m}_k$, but $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ or $s(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can still be found
- Use *K-nearest neighbor (KNN)* for reclustering data
  - **x** can be clustered according to the clusters of its KNNs through majority voting
  - Do not compute the distance between **x** and centroid
  - My choice → according to the choices of K-best friends
- Use *the distribution of cluster* for reclustering data
  - $P(\mathbf{x} \mid C_k)$ could be trained for each cluster
  - Example: customer **x** = [gender=male, age=20, occupation=engineer, income=900k, …]

Redistribution of Categorical Data (KNN)

- $\mathbf{x}_1$ = (A,B,A,B,C,B)
  $\mathbf{x}_2$ = (A,A,A,B,A,B)
  $\mathbf{x}_3$ = (B,B,A,B,A,B)
  $\mathbf{x}_4$ = (B,C,A,B,B,A)
  $\mathbf{x}_5$ = (B,A,B,A,C,A)
  $\mathbf{x}_6$ = (A,C,B,A,B,B)
- $C_1 = \{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \}$, $C_2 = \{ \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6 \}$
- For $\mathbf{y}$ = { A, C, A, B, C, A }

  $s_{smc}(\mathbf{y}, \mathbf{x}_1)=0.66$, $s_{smc}(\mathbf{y}, \mathbf{x}_2)=0.50$, $s_{smc}(\mathbf{y}, \mathbf{x}_3)=0.33$

  $s_{smc}(\mathbf{y}, \mathbf{x}_4)=0.66$, $s_{smc}(\mathbf{y}, \mathbf{x}_5)=0.33$, $s_{smc}(\mathbf{y}, \mathbf{x}_6)=0.33$
- KNN for K=3 are $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_4$ → voting C_1 for y

K-Means vs. GMM Training

- K-Means
  - Distance: Euclidean distance (for homogenous data)
  - Objective function: square error
  - Clusters have *mutually exclusive* data
  - Each point has equal contribution (weight) to mean

$$\boldsymbol{m}_k = \frac{1}{n_k}\sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i$$

- GMM
  - Similarity: Gaussian probability (distance is normalized by convariance $(\boldsymbol{x} - \boldsymbol{\mu}_k)^t \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_k)$)
  - Objective function: expectation of log probabilities
  - All data points contribute to each cluster

$$\boldsymbol{\mu}_k = \frac{\sum_i \boldsymbol{l}_i(k)\boldsymbol{x}_i}{\sum_{i,k} \boldsymbol{l}_i(k)}, \boldsymbol{\Sigma}_k = \frac{\sum_i \boldsymbol{l}_i(k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^t}{\sum_{i,k} \boldsymbol{l}_i(k)}$$

K-Means for Vector Quantization

- Select a set of prototype vectors to represent a large vector space
- Vectors in vector space are encoded into a discrete symbols
- *Codebook*: a set of *codewords* (vectors)
- Example: data compression

Evaluation for Cluster Analysis

- Assessment of the data domain
  - estimating the data in advance(how many clusters, distance between clusters, …)
- Cluster validity (verification on the results)
  - External assessment of validity
    - Compare the discovered structure to an a priori structure
  - Internal examination of validity
    - Determine if the discovered structure is intrinsically appropriate or meaningful
  - Relative test
    - Adjust the algorithms and parameters to produce different clusters. Compare the results to judge which is more applicable

Summary

- There is NO clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets.
- User's understanding of the problem and the corresponding data types will be the best criteria to select the appropriate method.
- The data should be subjected to tests for clustering tendency before applying a clustering algorithm, followed by a validation of the clusters generated by the algorithm
- There is no best clustering algorithm. Try several ones.
- Clustering analysis might be used as processing techniques in intermediate level.
  - e.g. word class n-gram (sharing probabilities)

Reference

- Data Mining: Concepts, Models, Methods and Algorithms

  Mehmed Kantardzic, Wiley Inter-science