Knowledge Discovery and Data Mining

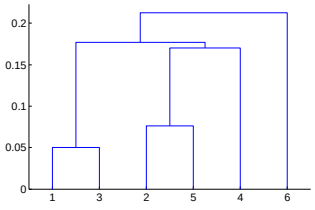
Unit # 12

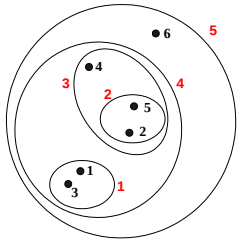
Acknowledgement

- Most of the slides in this presentation are taken from course slides provided by
  - Han and Kimber (Data Mining Concepts and Techniques) and
  - Tan, Steinbach and Kumar (Introduction to Data Mining)
  - Several other online sources

Hierarchical Clustering

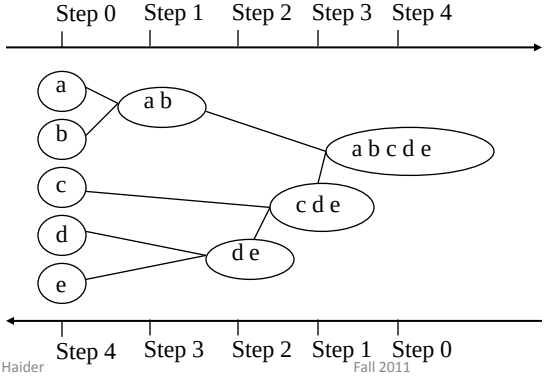
- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits





Hierarchical Clustering (Cont'd)

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

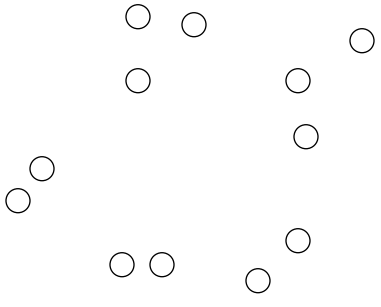
- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

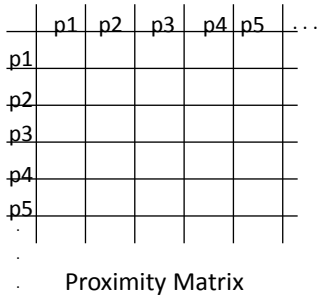
Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  1. Compute the proximity matrix
  2. Let each data point be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

- Start with clusters of individual points and a proximity matrix

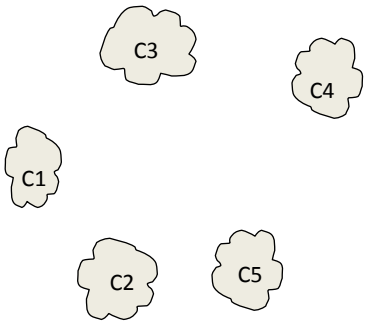




Proximity Matrix

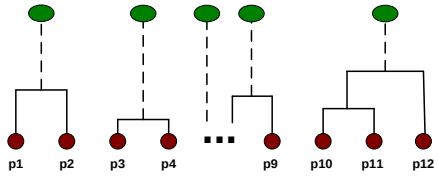
Intermediate Situation

- After some merging steps, we have some clusters



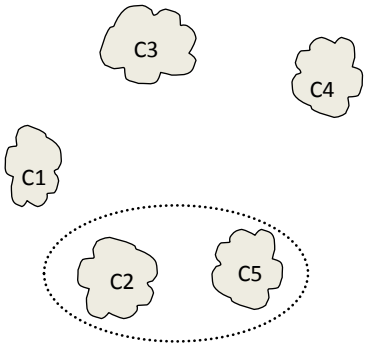
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



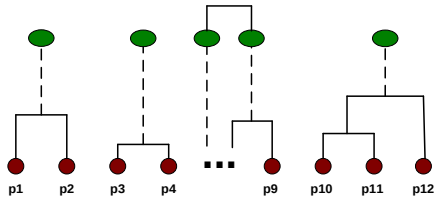
Intermediate Situation

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix.



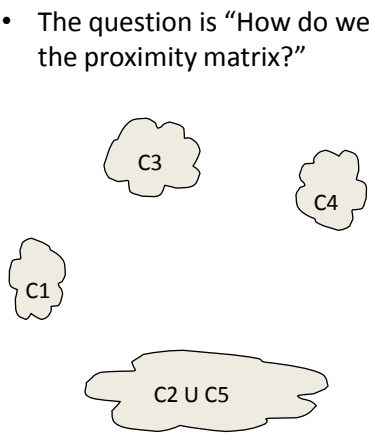
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



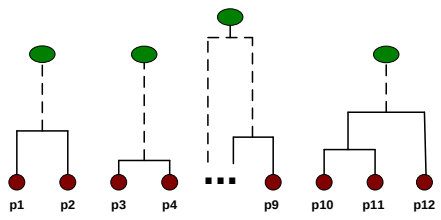
After Merging

- The question is "How do we update the proximity matrix?"

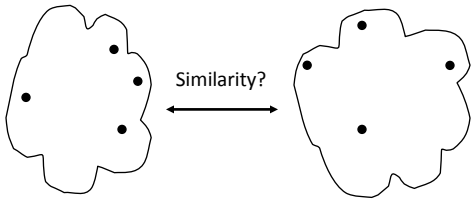


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



How to Define Inter-Cluster Similarity

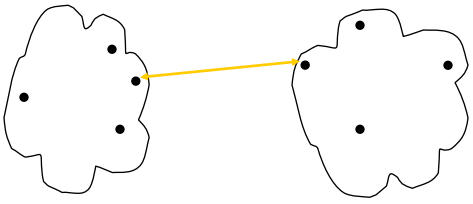


- MIN
- MAX
- Group Average
- Distance Between Centroids

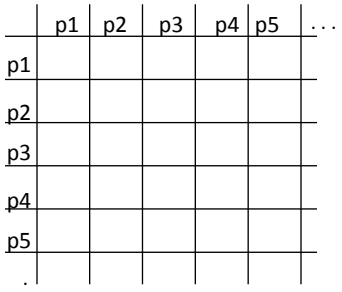
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity (Cont'd)

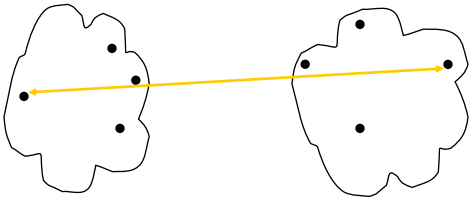


- MIN
- MAX
- Group Average
- Distance Between Centroids

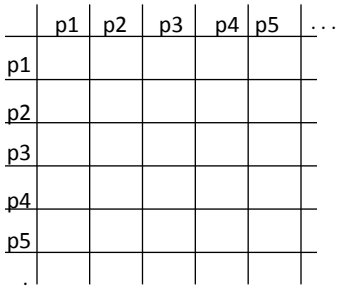


Proximity Matrix

How to Define Inter-Cluster Similarity (Cont'd)

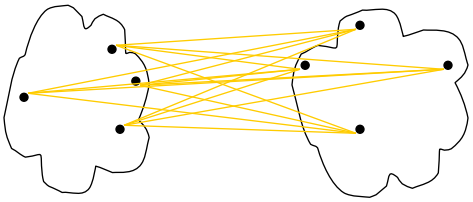


- MIN
- MAX
- Group Average
- Distance Between Centroids

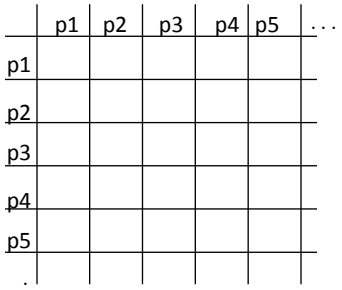


Proximity Matrix

How to Define Inter-Cluster Similarity (Cont'd)

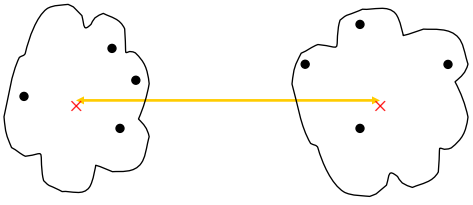


- MIN
- MAX
- Group Average
- Distance Between Centroids

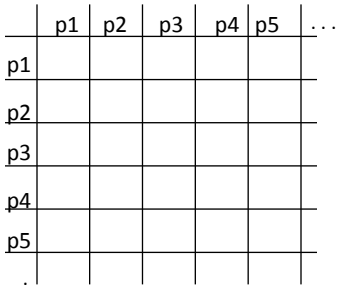


Proximity Matrix

How to Define Inter-Cluster Similarity (Cont'd)



- MIN
- MAX
- Group Average
- Distance Between Centroids



Proximity Matrix

Single, Complete and Average Linkage

- In *single-linkage* clustering, the distance between one cluster and another cluster is equal to the shortest distance from any member of one cluster to any member of the other cluster: $D(c_i,c_j) = \min d(a,b)\ a \in c_i, b \in c_j$. It is obvious that:

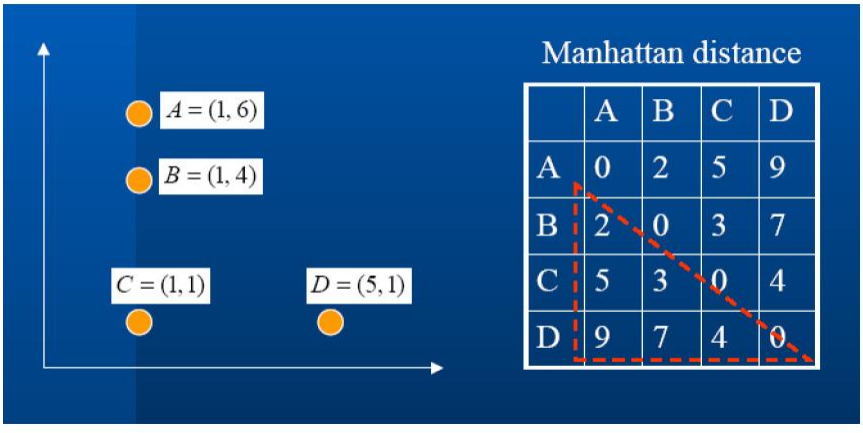
$$D(c_k,c_l) = \min\{D(c_i,c_l), D(c_j,c_l)\} \quad \text{for} \quad c_k = c_i \cup c_j$$

- In *complete-linkage* clustering, the distance between one cluster and another cluster is equal to the greatest distance from any member of one cluster to any member of the other cluster: $D(c_i,c_j) = \max d(a,b)\ a \in c_i, b \in c_j$.

- In *average-linkage* clustering, the distance between one cluster and another cluster is equal to the average distance from any member of one cluster to any member of the other cluster: $D(c_i,c_j) = \frac{1}{|c_i||c_j|} \sum_{a \in c_i, b \in c_j} d(a,b)$. It is obvious that

$$D(c_k,c_l) = \frac{|c_i|}{|c_k|} D(c_i,c_l) + \frac{|c_j|}{|c_k|} D(c_j,c_l) \quad \text{for} \quad c_k = c_i \cup c_j$$

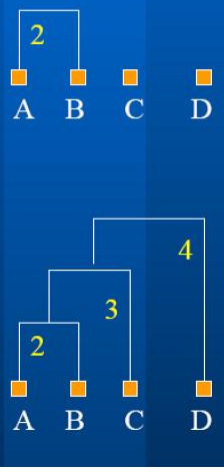
Example: Distance Computation



	A	B	C	D
A	0	2	5	9
B	2	0	3	7
C	5	3	0	4
D	9	7	4	0

Example: Single Linkage

Single linkage

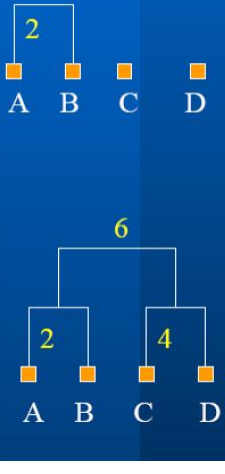


dist((A, B), C) = min{dist(A, C), dist(B, C)
= min{5, 3} = 3
dist((A, B), D) = min{dist(A, D), dist(B, D)}
= min{9, 7} = 7
dist(C, D) = 4

dist((A, B, C), D)
= min{dist((A, B), D), dist(C, D)}
= min{7, 4} = 4

Example: Average Linkage

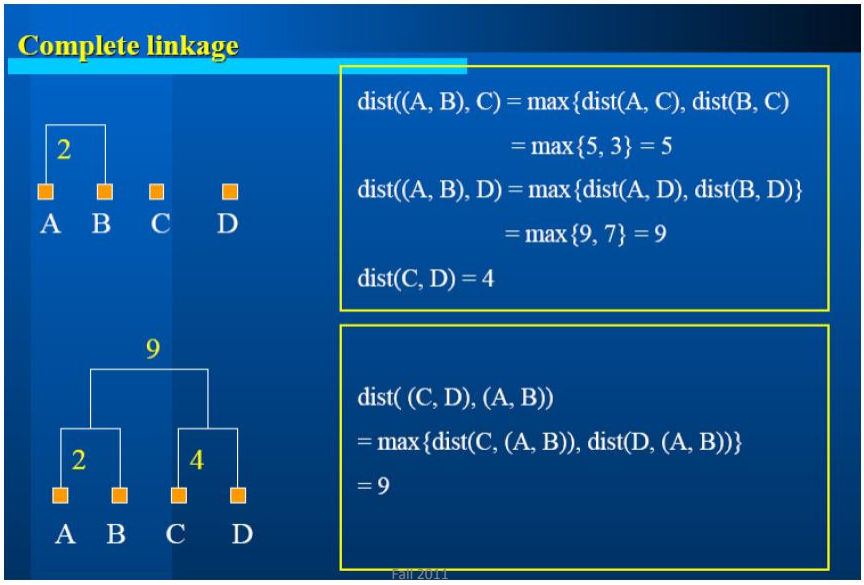
Average linkage



dist((A, B), C) = avg{dist(A, C), dist(B, C)
= (5+3)/2 = 4
dist((A, B), D) = avg{dist(A, D), dist(B, D)}
= (9+7)/2 = 8
dist(C, D) = 4

dist((C, D), (A, B))
= avg{dist(C, (A, B)), dist(D, (A, B))}
= (4+8)/2 = 6

Example: Complete Linkage



Hierarchical Clustering: Problems and Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

Cluster Validity

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the "goodness" of the resulting clusters?
- But "clusters are in the eye of the beholder"!
- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two sets of clusters
  - To compare two clusters

Measures of Cluster Validity

- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - Entropy
  - Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
    - Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy

Internal Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
  - Example: SSE
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

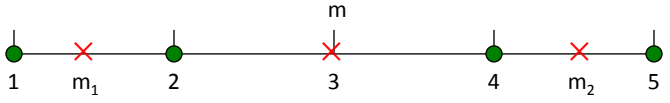
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} |C_i| (m - m_i)^2$$

    - Where $|C_i|$ is the size of cluster i

Cohesion and Separation: Example

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

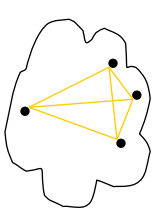
$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$BSS = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

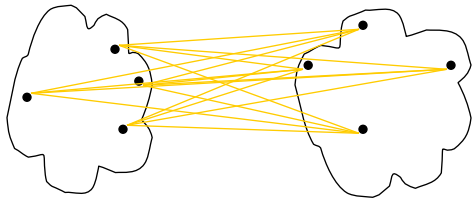
$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation (Cont'd)

- A proximity graph based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



separation

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can primarily be categorized into partitioning methods and hierarchical methods (there are many other less popular schemes too)
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis