KHU-BME
Pattern Classification
Lecture 18

Hierarchical Clustering (DHS 10.9)

J=no. of samples

C=J, Top, 1 cluster for all points (K=1)

C=1, Bottom, each point is a separate (K=J) cluster.

C=J-K+1, at the level K in the sequence.

Def: Hierarchical clustering (p 551)

Agglomerative: bottom up, merge clusters

Divisive: top down, divide clusters

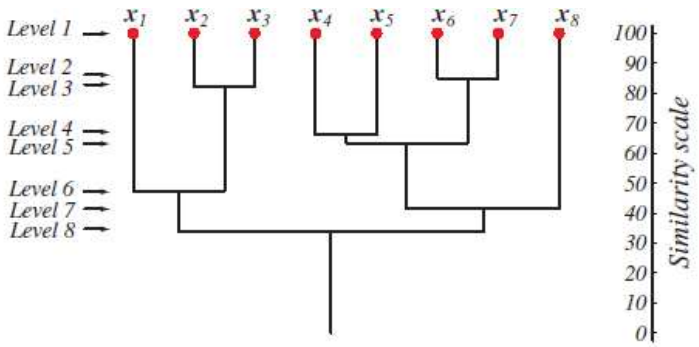

Figure 10.10: A dendrogram can represent the results of hierarchical clustering algorithms. The vertical axis shows a generalized measure of similarity among clusters. Here, at level 1 all eight points lie in singleton clusters; each point in a cluster is highly similar to itself, of course. Points x_6 and x_7 happen to be the most similar, and are merged at level 2, and so forth.

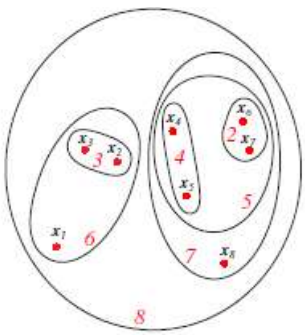

Figure 10.11: A set or Venn diagram representation of two-dimensional data (which was used in the dendrogram of Fig. 10.10) reveals the hierarchical structure but not the quantitative distances between clusters. The levels are numbered in red.

Agglomerative Hierarchical Clustering Procedure (DHS 10.9.2)

First choose K

1. Let $\hat{K}$ =J and $z_i=\{x_i\}$, for i=1,…,J
2. If $\hat{K} \le$K, stop
3. Find nearest pair of clusters, say z_i and z_l. Need distance function.
4. Merge z_i and z_l. Need rule for merging.
5. Go to Step 2.

If K=1, resulting hierarchy is a dendrogram.

Useful distance measures:

1. $d_{\min}(z_i, z_l) = \min_{\substack{x \in z_i \\ x' \in z_l}} \|x - x'\|$
2. $d_{\max}(z_i, z_l) = \max_{\substack{x \in z_i \\ x' \in z_l}} \|x - x'\|$
3. $d_{avg}(z_i, z_l) = \frac{1}{J_i J_l} \sum_{x \in z_i} \sum_{x \in z_l} \|x - x'\|$
4. $d_{mean}(z_i, z_l) = \|m_i - m_l'\|$

- All of these measures have a minimum-variance flavor
- They produce the same results if the clusters are compact and well separated.

Nearest Neighbor Algorithm (DHS 10.9.2)

- K set in advance
- Use d_{min} -> algorithm called the nearest-neighbor cluster algorithm, or minimum algorithm.
- or **Single Linkage** Algorithm
- Merge rule – connect 2 closest points of clusters to be merged, 1 from each cluster.

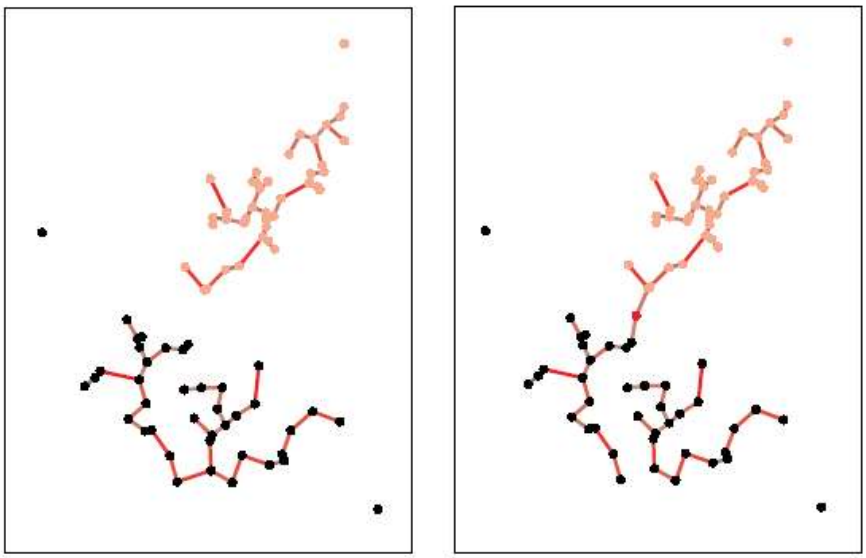

Figure 10.12: Two Gaussians were used to generate two-dimensional samples, shown in pink and black. The nearest-neighbor clustering algorithm gives two clusters that well approximate the generating Gaussians (left). If, however, another particular sample is generated (red point at the right) and the procedure re-started, the clusters do not well approximate the Gaussians. This illustrates how the algorithm is sensitive to the details of the samples.

Resulting graph is a tree (no closed loops)

Go to K=1 => spanning tree.

Using d_{min} => Minimal Spanning Tree (min. total length of all links)

Farthest Neighbor Algorithm

- Terminate at K clusters
- Use d_{max} -> farthest neighbor clustering algorithm or maximum algorithm
- **Complete Linkage** Algorithm

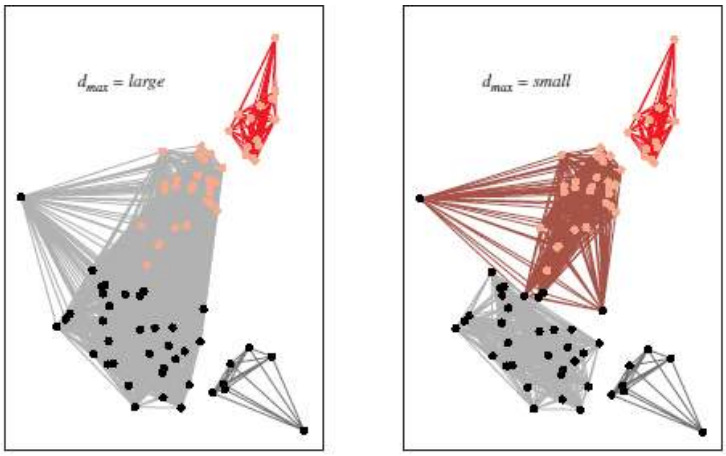

Figure 10.13: The farthest-neighbor clustering algorithm uses the separation between the most distant points as a criterion for cluster membership. If this distance is set very large, then all points lie in the same cluster. In the case shown at the left, a fairly large d_{max} leads to three clusters; a smaller d_{max} gives four clusters, as shown at the right.

If a larger dmax is allowed vs. If a smaller dmax is allowed.

Compromises

d_{mean}, d_{ave} are compromises between d_{min} and d_{max}.