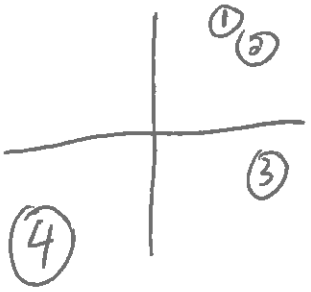
Lecture #8: September 27, 2018

- K-means inputs data $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$ and outputs labels $\{g_i\}_{i=1}^n$ where each $g_i \in \{1, \dots, K\}$. The user specifies K.
- Another approach is to get a family of clusterings, parametrized by K.
- That is, get a family of clusterings $\{C_K\}_{K=1}^n$, where C_K is a clustering of the data into K clusters.
- Of course, we could just run K-means and allow K to range from 1 to n. This would be slow and there is not necessarily a good relationship between the K-mean clusters as K increases.
- Hierarchical methods give a family of clusterings that relate well to each other.
- Let $C = \{C_1, \dots, C_K\}$ be a clustering of X: 1) $X = \bigcup_{i=1}^K C_i$
  2) $C_i \cap C_j = \emptyset$ if $i \neq j$.
- Another clustering $\mathcal{B} = \{B_1, \dots, B_L\}$ is nested inside C if for every $i = 1, \dots, L$ there is some j such that $B_i \subset C_j$.
- Hierarchical methods provide nested families of clusterings. We will focus on agglomerative methods, which start with each point in their own cluster, and iteratively merge clusters one at a time.

ex:

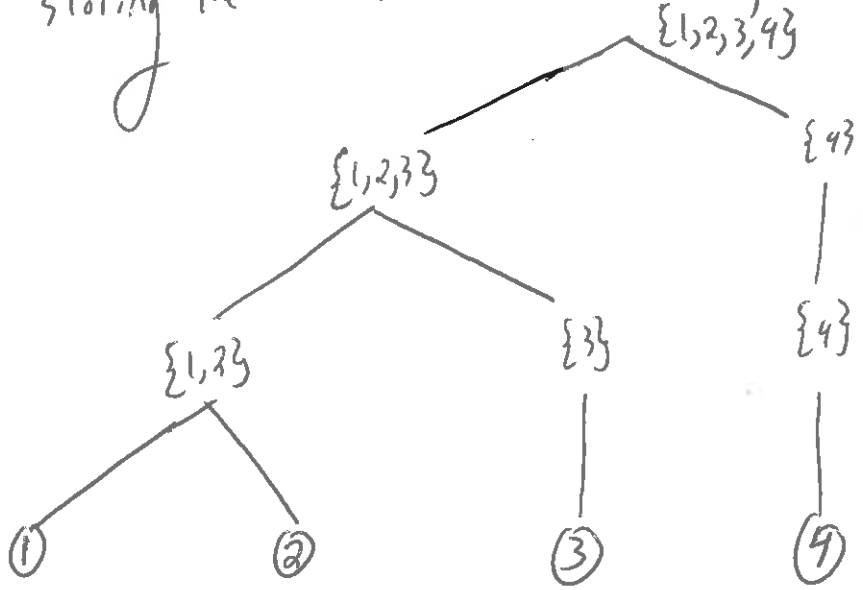


Step	# Clusters	Clusters
1	4	{1}, {2}, {3}, {4}
2	3	{1,2}, {3}, {4}
3	2	{1,2,3}, {4}
4	1	{1,2,3,4}

Merge the two clusters with the ~~nearest~~ smallest minimal distance between points.

- Could other notion of merger be used? Absolutely.

Remark: Instead of storing the results in a table as above, we could use a dendrogram:



- This way, all clusters can be viewed "at once" though maybe a little hard to parse.

- A major concern is how to algorithmically merge clusters?

- One approach is to: 1) Define a notion of distance between clusters, call it Δ
2) Iteratively merge clusters by merging the clusters C_i, C_j with minimal $\Delta(C_i, C_j)$.

Two very standard definitions of Δ are:

1) $\Delta_{SL}(C_i, C_j) = \min_{\substack{x \in C_i \\ y \in C_j}} \|x - y\|_2$ This links ~~point~~ clusters according to the nearest two points in the clusters. Single linkage distance requires only one short connection between clusters to have small value.

2) $\Delta_{CL}(C_i, C_j) = \max_{\substack{x \in C_i \\ y \in C_j}} \|x - y\|_2$. This links clusters according to the furthest two points in the clusters. Complete linkage distance requires all paths between the clusters to be short.

- These are natural, but extreme (min/max). Some other notion interpolate between these extremes; a reasonable one is average distance between the clusters

3) $\Delta_{GA}(C_i, C_j) = \sum_{\substack{x \in C_i \\ y \in C_j}} \|x - y\|_2 \Big/ |C_i| \cdot |C_j|$ Group average distance requires the average path between clusters to be short.

- Could also do distance between centroids, choose to merge clusters in a manner

minimizing the resultant variance increase.

Computationally, this could be slow. Need to know distances between all points ($O(n^2)$) then do n merges. A naive approach would take $O(n^3)$, using a data management structure like a heap drops this to $O(n^2 \log n)$, being really clever to $O(n^2)$.