StatsM254 Statistical Methods in Computational Biology Lecture 4 - 04/10/2014

Lecture 4

Lecturer: Jingyi Jessica Li *Scribe: Megan Roytman*

1 K-means

Algorithm:

1. Given a cluster of assignments C, we find cluster centers as $m_k = \frac{1}{n_k} \sum_{C(i)=k} x_i$, where $k = 1, ..., K$
2. Given the cluster centers $\{m_k\}_{k=1}^K$ we find the cluster assignment
$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \|x_i - m_k\|^2$$
3. Iterate steps 1 and 2 until C converges.

2 K-medoids

Similar to k-means, but requires $\{m_k\}_{k=1}^K$ to be data points. Advantage: robust to outliers.

Algorithm:

1. Given a cluster of assignments C, find the data point in the cluster to minimize the total distance to other data points in that cluster.
$$i_k^* = \underset{i:\ C(i)=k}{\operatorname{argmin}} \sum_{C(j)=k} d(x_i, x_j)$$
Then $m_k = x_{i_k^*}, k = 1, ..., K$ (new cluster centers)

Note: $d(x_i, x_j)$ can be any distance metric, e.g. $|x_i - x_j|$.
2. Given current cluster centers $\{m_k\}_{k=1}^K$, assign each data point to the closest center:
$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} d(x_i, m_k)$$
where $i = 1, ..., n$
3. Iterate steps 1 and 2 until C converges.

This algorithm is a heuristic search to

$$C(i) = \min_{C, \{i_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{C(i)=k} d(x_i, x_{i_k})$$

Comment:

1. k-means is based on Euclidian distance. If you change the metric to L_1 norm, you get k-medians. Both do not require the cluster centers to be data points.
2. In k-medoids, you can use any distance metric, but the cluster centers are restricted to be data points.

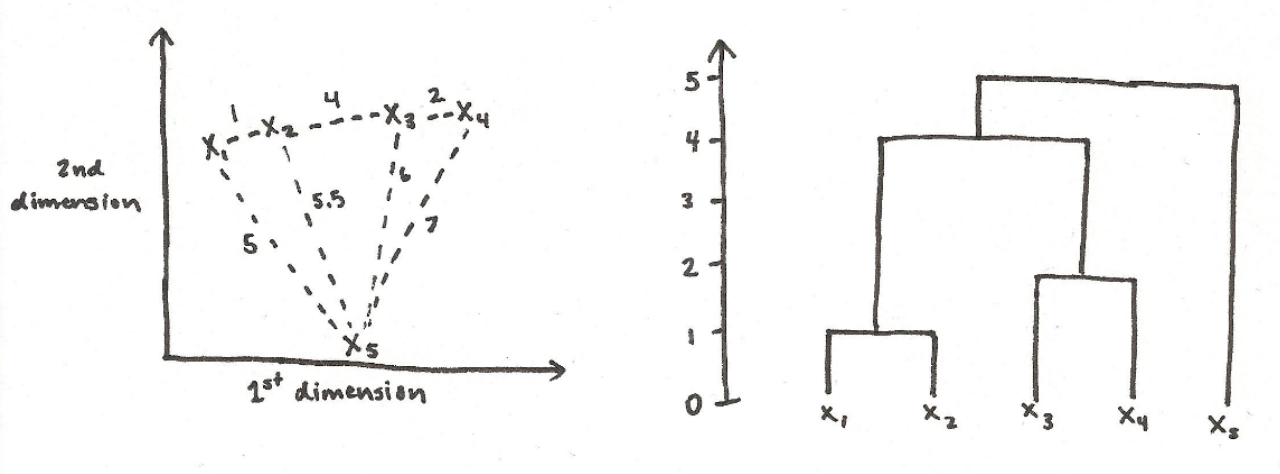
3 Heirarchical Clustering

Don't need to specify K, the number of clusters. A hierarchical tree will be built. This is agglomerative clustering.

Two things required:

1. distance metric (required by every clustering algorithm)
2. distance between a point and a cluster, and between 2 clusters
   (a) Single linkage (SL)
   $$d_{SL}(G,H) = \min_{\substack{i\in G \\ i'\in H}} d_{ii'}$$
   (b) Complete linkage (CL)
   $$d_{CL}(G,H) = \max_{\substack{i\in G \\ i'\in H}} d_{ii'}$$
   (c) Group average (GA)
   $$d_{GA}(G,H) = \frac{1}{N_G N_H} \sum_{i\in G} \sum_{i'\in H} d_{ii'}$$

Example: Euclidian distance + single linkage



Procedure:

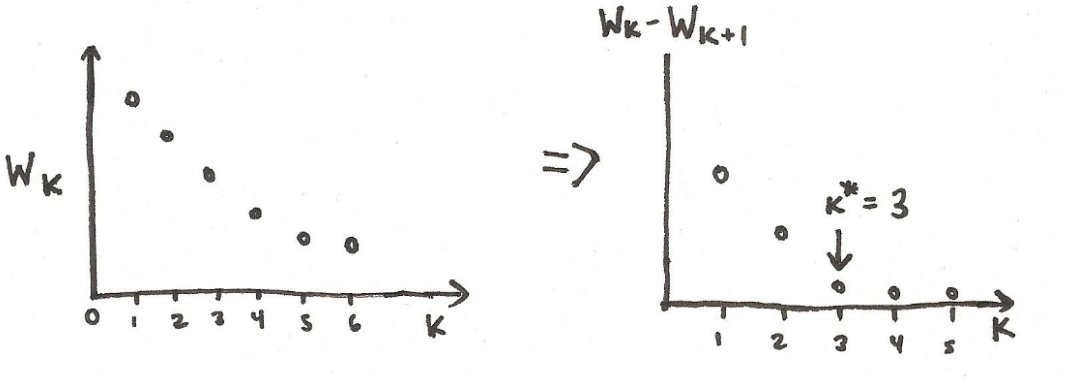
1. Start with the 2 points closest to each other. Merge them into 1 cluster. $x_1, x_2 \Rightarrow x_1^*$
2. Find the closest pair among x_1^*, x_3, x_4, x_5. $x_3, x_4 \Rightarrow x_3^*$
3. Find the closest pair among x_1^*, x_3^*, x_5. $x_1^*, x_3^* \Rightarrow x_1^{**}$
4. Merge x_1^{**} and x_5.

4 How to determine K, the number of clusters?

Within-cluster dissimilarity W_K as a function of K, e.g. Euclidian distance.
K clusters: $C_1, ..., C_K$, each is a set of indices of data points in each cluster.
$n_k = |C_k|$ = number of data points in cluster k.

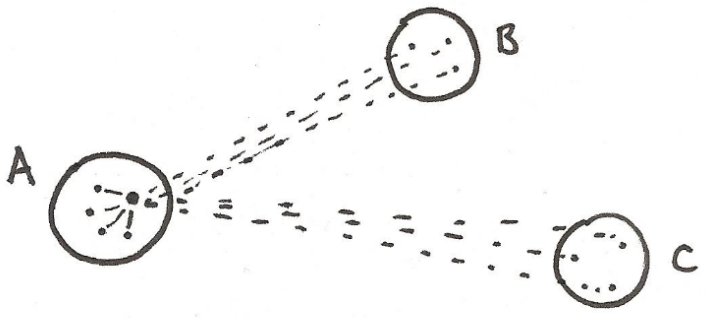
$$D_k = \sum_{i,i' \in C_k} d_{ii'} = \sum_{i,i' \in C_k} (x_i - x_{i'})^2$$

$$W_k = \sum_{k=1}^{K} \frac{1}{2n_k} D_k$$



Graphical method: Silhouettes (Rousseau, 1987) - R package

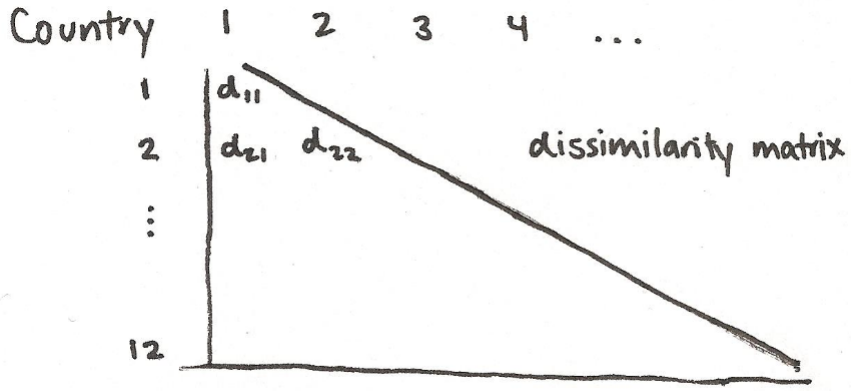
Suppose that $x_i \in A$.
$a(i)$ = average dissimilarity of x_i to other points in A.
$d(i, C)$ = average dissimilarity of x_i to all points in cluster C.
$b(i) = \min_{C \neq A} d(i, C)$.

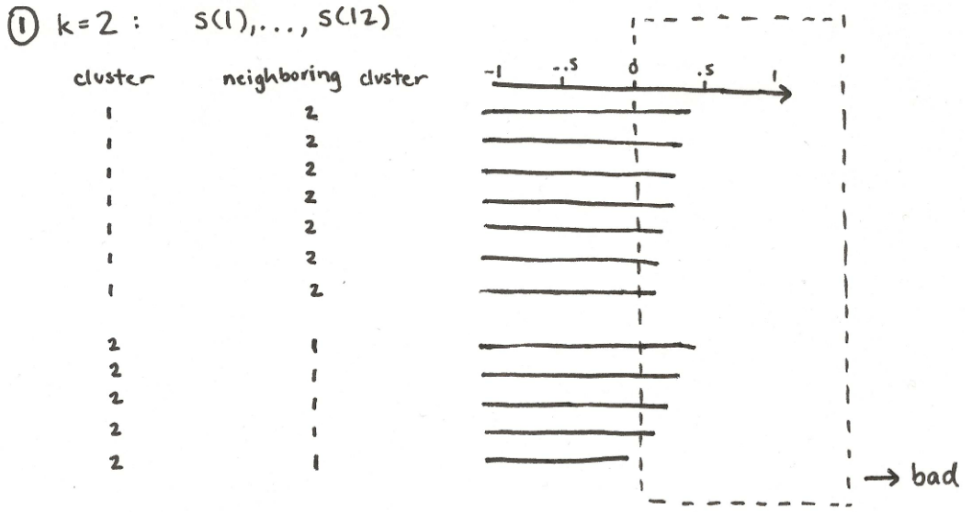


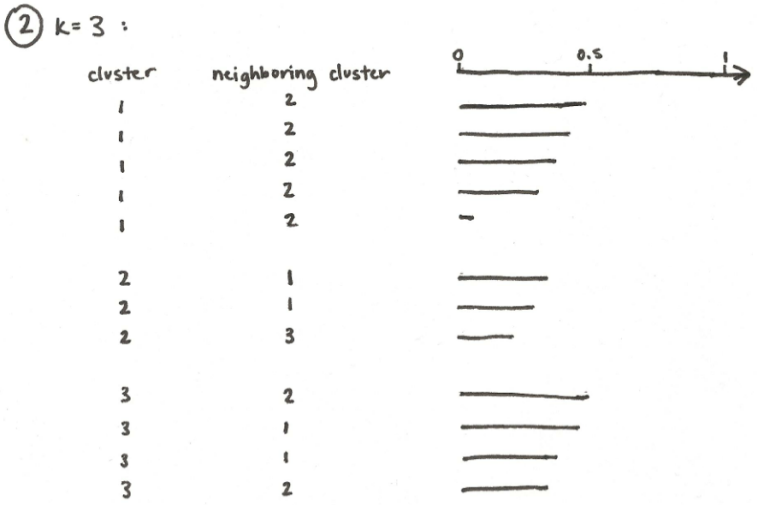
Silhouette:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases}$$

$-1 \leq S(i) \leq 1$







In the case above, $K = 3$ is a more reasonable choice than $K = 2$.