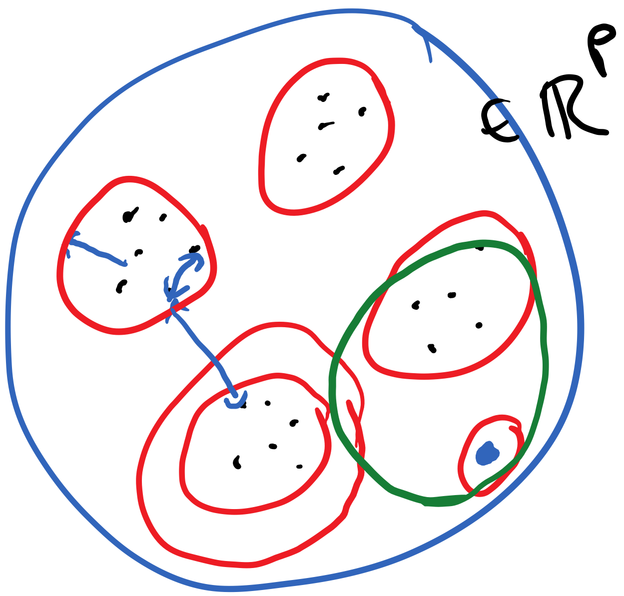
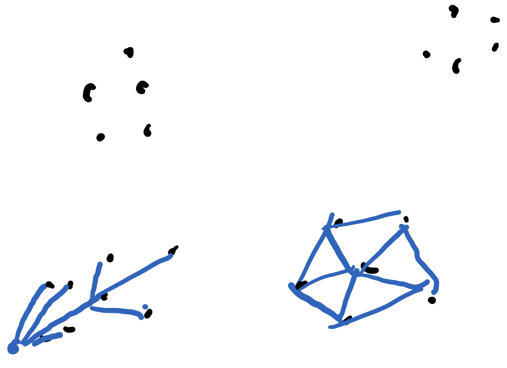
MITx:
Statistics, Computation & Applications

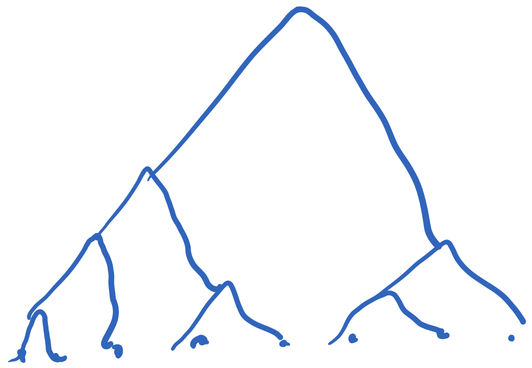
Genomics and High-Dimensional Data Module

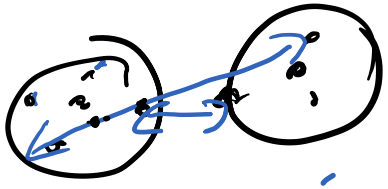
Lecture 3: Clustering with Hig-Dimensional Data

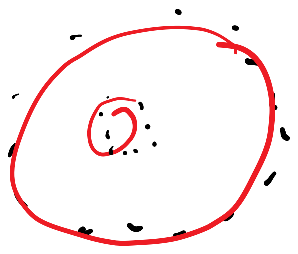
k-means





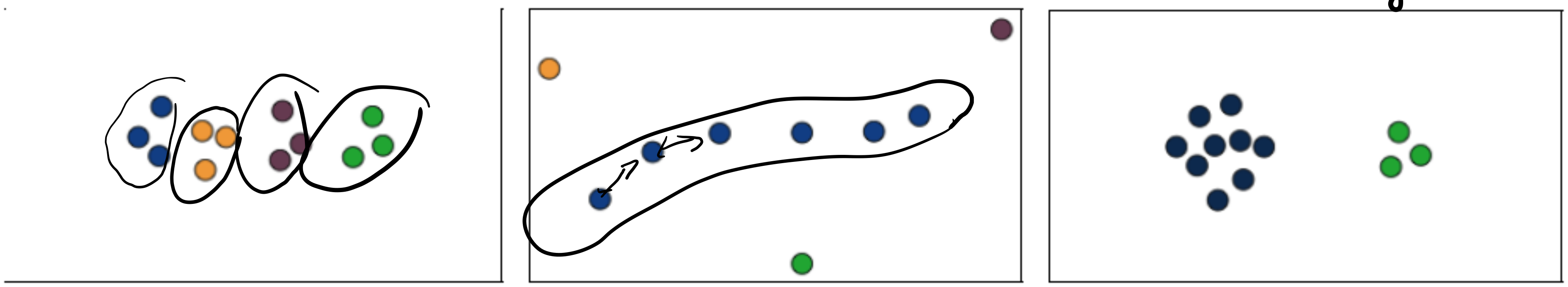






DBSCAN

max min average



k-means

min max average

distance between clusters

MITx:
Statistics, Computation & Applications

Genomics and High-Dimensional Data Module

Lecture 3: Clustering with Hig-Dimensional Data

Clustering

- Find groups, so that elements within cluster are very similar and elements between clusters are very different
- **Examples:**
  - Find customer groups to adjust advertisement
  - Find subtypes of diseases to fine-tune treatment
- N samples, k clusters: k^N possible assignments
  - E.g., $N = 100$, $k = 3$: $3^{100} = 5 * 10^{47}$!!
    $\Rightarrow$ impossible to search through all assignments

We will discuss:

- k-means clustering
- Gaussian mixture models
- Hierarchical clustering
- DBSCAN

K-means clustering

- K (fixed!) Clusters are obtained by minimizing some loss function
- Natural loss function given by within-groups sum of squares (WGSS):

$$W(C) = \sum_{k=1}^{K} \sum_{C(x^{(i)})=k} \sum_{C(x^{(j)})=k} d(x^{(i)}, x^{(j)})^2$$

- $W(C)$ characterizes the extent to which observations assigned to the same cluster tend to be close to one another

$$\sum_{k=1}^{K} \left[\sum_{C(x^{(i)})=k} \left(\underbrace{\sum_{C(x^{(j)})=k} d(x^{(i)}, x^{(j)})^2}_{\text{WGSS}} + \underbrace{\sum_{C(x^{(j)})\neq k} d(x^{(i)}, x^{(j)})^2}_{\text{BGSS}} \right) \right]$$

$$= \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{n} d(x^{(i)}, x^{(j)})^2}_{\text{constant}}$$

K-means clustering

- K (fixed!) Clusters are obtained by minimizing some loss function
- Natural loss function given by within-groups sum of squares (WGSS):

$$W(C) = \sum_{k=1}^{K} \sum_{C(x^{(i)})=k} \sum_{C(x^{(j)})=k} d(x^{(i)}, x^{(j)})^2$$

- $W(C)$ characterizes the extent to which observations assigned to the same cluster tend to be close to one another
- K-means clustering: $d(x^{(i)}, x^{(j)})^2 = \|x^{(i)} - x^{(j)}\|_2^2$
- Then WGSS becomes: $W(C) = \sum_{k=1}^{K} 2N_k \sum_{C(x^{(i)})=k} \|x^{(i)} - \mu_k\|_2^2$

Claim: $$\sum_{C(x^{(i)})=k} \; \sum_{C(x^{(j)})=k} \|x^{(i)}-x^{(j)}\|_2^2 = 2n_k \sum_{C(x^{(i)})=k} \|x^{(i)}-\mu_k\|_2^2$$

Proof:
$$= \sum_{\substack{C(x^{(i)})=C(x^{(j)})=k \\ x^{(i)} \neq x^{(j)}}} \|x^{(i)}-\mu_k+\mu_k-x^{(j)}\|_2^2$$

$$= \sum \|x^{(i)}-\mu_k\|_2^2 + \|x^{(j)}-\mu_k\|_2^2 + 2(x^{(i)}-\mu_k)^T(\mu_k-x^{(j)})$$

$$= 2(n_k-1) \sum_{C(x^{(i)})=k} \|x^{(i)}-\mu_k\|_2^2$$

$$+ 2 \sum (x^{(i)}-\mu_k)^T(\mu_k-x^{(j)})$$

$$-$$

$$= 2 \sum_{C(x^{(i)})=k} \|x^{(i)}-\mu_k\|_2^2$$

$$= \sum_{C(x^{(i)})=k} \; \sum_{C(x^{(j)})=k} (x^{(i)}-\mu_k)^T(x^{(j)}-\mu_k) = 0$$

K-means clustering

- K (fixed!) Clusters are obtained by minimizing some loss function
- Natural loss function given by within-groups sum of squares (WGSS):

$$W(C) = \sum_{k=1}^{K} \sum_{C(x^{(i)})=k} \sum_{C(x^{(j)})=k} d(x^{(i)}, x^{(j)})^2$$

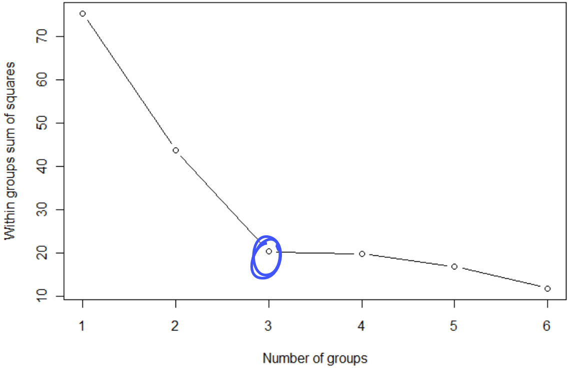
- $W(C)$ characterizes the extent to which observations assigned to the same cluster tend to be close to one another
- K-means clustering: $d(x^{(i)}, x^{(j)})^2 = \|x^{(i)} - x^{(j)}\|_2^2$
- Then WGSS becomes: $W(C) = \sum_{k=1}^{K} 2N_k \sum_{C(x^{(i)})=k} \|x^{(i)} - \mu_k\|_2^2$
- Exact solution computationally infeasible
  - Use greedy algorithm
  - Use random restarts to avoid local optima

- Leads to spherical shaped clusters of similar radii

Choosing the number of clusters

- Run K-means clustering for several number of groups K
- Plot WGSS versus the number of groups
- Choose number of groups after the last big drop of the curve

Example:



Partitioning around medoids (PAM)

- K-Means: Cluster centers μ_k can be arbitrary points in space

  $\Rightarrow$ very sensitive to outliers!

Partitioning around medoids (PAM)

- K-Means: Cluster centers μ_k can be arbitrary points in space

  $\Rightarrow$ very sensitive to outliers!

- Robust alternative: Partitioning around medoids (PAM)
  - Cluster center must be an observation ("medoid")
  - More robust against outliers
  - Also gives a representative object for each cluster (e.g., for easy interpretation)

Gaussian mixture model

- Assume underlying statistical model:

$$P(x) = \sum_{k=1}^{K} P(\text{cluster } k) P(x \mid \text{cluster } k),$$

where $X \mid \text{cluster } k \sim \mathcal{N}(\mu_k, \Sigma_k)$

Particular example
$\Sigma_k = \sigma^2 I$

- Sample x is assigned to cluster k that maximizes $P(\text{cluster } k \mid x)$

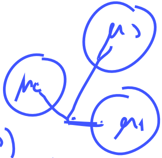
Gaussian mixture model

- Assume underlying statistical model:

$$P(x) = \sum_{k=1}^{K} \underbrace{P(\text{cluster } k)}_{p_k} P(x \mid \text{cluster } k),$$

where $X \mid \text{cluster } k \sim \mathcal{N}(\mu_k, \Sigma_k)$ $\quad \Sigma_k = \sigma^2 I$

- Sample x is assigned to cluster k that maximizes $P(\text{cluster } k \mid x)$
- Estimating $P(\text{cluster } k)$, μ_k and Σ_k by maximum likelihood estimation is difficult (leads to a non-convex optimization problem)
- Parameter estimates are usually found using the Expectation-Maximization (EM) algorithm



E-step: $P(\text{cluster } k \mid x^{(i)}) = \dfrac{p_k P(x^{(i)} \mid \text{cluster } k)}{P(x^{(i)})}$

M-step: $p_k = \frac{1}{n} \sum_{i=1}^{n} P(\text{cluster } k \mid x^{(i)})$

$\mu_k = \sum_{i=1}^{n} x^{(i)} \dfrac{P(\text{cluster } k \mid x^{(i)})}{\sum_{i=1}^{n} P(\text{cluster } k \mid x^{(i)})}$

$P(1 \mid x^{(i)}) > P(2 \mid x^{(i)}) > P(3 \mid x^{(i)})$

Gaussian mixture model

- Assume underlying statistical model:

$$P(x) = \sum_{k=1}^{K} P(\text{cluster } k) P(x \mid \text{cluster } k),$$

where $X \mid \text{cluster } k \sim \mathcal{N}(\mu_k, \Sigma_k)$

- Sample x is assigned to cluster k that maximizes $P(\text{cluster } k \mid x)$
- Estimating $P(\text{cluster } k)$, μ_k and Σ_k by maximum likelihood estimation is difficult (leads to a non-convex optimization problem)
- Parameter estimates are usually found using the Expectation-Maximization (EM) algorithm
- Number of clusters is found for example by maximizing the Bayesian information criterion

$$\text{BIC} = \log\text{-likelihood} - \frac{\log(n)}{2} \cdot (\#\text{ of parameters})$$

Clustering

- Find groups, so that elements within cluster are very similar and elements between clusters are very different
- **Examples:**
  - Find customer groups to adjust advertisement
  - Find subtypes of diseases to fine-tune treatment
- N samples, k clusters: k^N possible assignments
  - E.g., $N = 100$, $k = 3$: $3^{100} = 5 * 10^{47}$!!

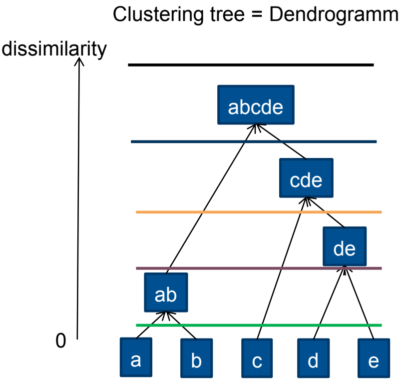
    $\Rightarrow$ impossible to search through all assignments

We will discuss:

- k-means clustering
- Gaussian mixture models
- Hierarchical clustering
- DBSCAN

Hierarchical clustering

- Agglomerative clustering: Build up clusters from individual observations
- Divisive clustering: Start with whole group of observations and split off clusters



Advantage of hierarchical clustering:

- Solve clustering for all possible numbers of cluster $1, 2, \ldots, n$ at once
- Choose desired number of clusters later

Examples of dissimilarity measures between samples

- Euclidean distance (i.e., ℓ_2 - norm)

$$d(x^{(i)}, x^{(j)}) = \sqrt{(x_1^{(i)} - x_1^{(j)})^2 + (x_2^{(i)} - x_2^{(j)})^2 + \cdots + (x_p^{(i)} - x_p^{(j)})^2}$$

- Manhattan distance (i.e., ℓ_1 - norm)

$$d(x^{(i)}, x^{(j)}) = |x_1^{(i)} - x_1^{(j)}| + |x_2^{(i)} - x_2^{(j)}| + \cdots + |x_p^{(i)} - x_p^{(j)}|$$

- Maximum distance (i.e., ℓ_∞ - norm)

$$d(x^{(i)}, x^{(j)}) = \max_{k=1,\ldots,p} |x_k^{(i)} - x_k^{(j)}|$$

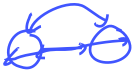
- or more flexible dissimilarity satisfying

$$d(x^{(i)}, x^{(j)}) \geq 0, \; d(x^{(i)}, x^{(i)}) = 0, \; d(x^{(i)}, x^{(j)}) = d(x^{(j)}, x^{(i)})$$

Examples of dissimilarity measures between clusters

- single linkage (i.e., minimum distance)

$$d(C_r, C_s) = \min_{x^{(i)} \in C_r, x^{(j)} \in C_s} d(x^{(i)}, x^{(j)})$$



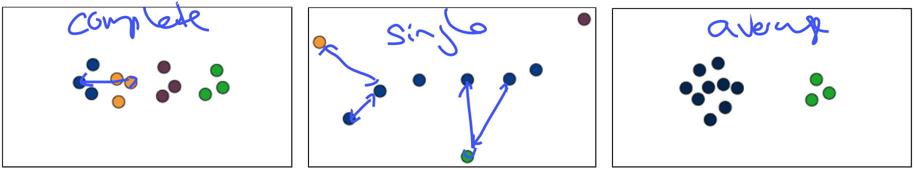
- complete linkage (i.e., maximum distance)

$$d(C_r, C_s) = \max_{x^{(i)} \in C_r, x^{(j)} \in C_s} d(x^{(i)}, x^{(j)})$$

- average linkage (i.e., average distance)

$$d(C_r, C_s) = \frac{1}{n_r} \frac{1}{n_s} \sum_{x^{(i)} \in C_r} \sum_{x^{(j)} \in C_s} d(x^{(i)}, x^{(j)})$$

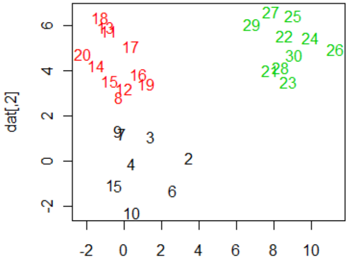
How do the resulting clusters look like? Which one is which?

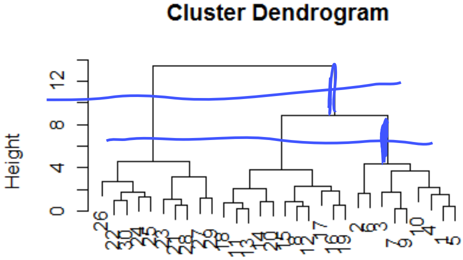


Choosing the number of clusters

- No strict rule
- Find the largest vertical "drop" in the tree

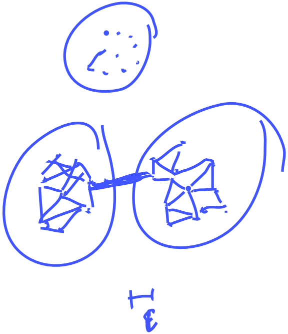
Example:

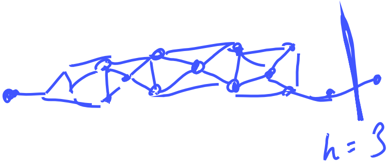




DBSCAN

ε: distance between points to be connected
k: core strength



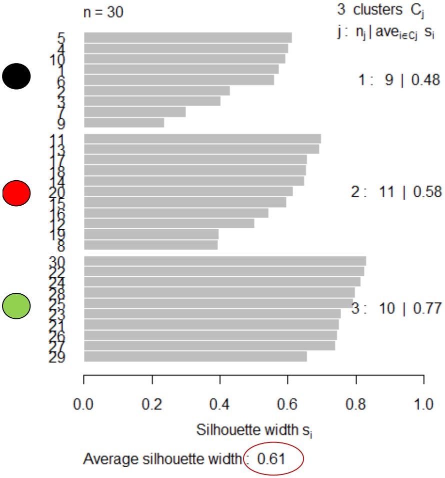


Quality of clustering: Silhouette plot

Compute for each sample $x^{(i)}$:

- $a(x^{(i)})$ = average dissimilarity between $x^{(i)}$ and all other points in its cluster
- $b(x^{(i)})$ = average dissimilarity between $x^{(i)}$ and the closest cluster it does not belong to
- $S(x^{(i)}) \in [-1, 1]$ with

$$S(x^{(i)}) = \frac{(b(x^{(i)}) - a(x^{(i)}))}{\max(a(x^{(i)}), b(x^{(i)}))}$$

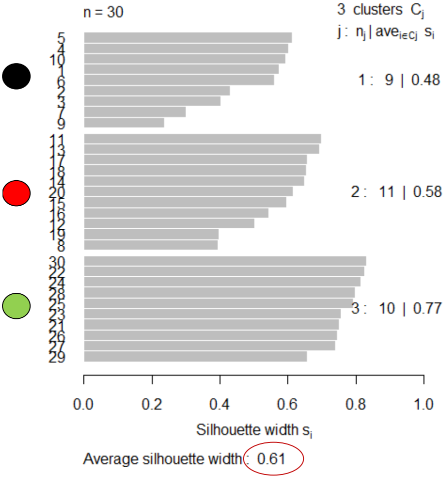


Quality of clustering: Silhouette plot

Compute for each sample $x^{(i)}$:

- $a(x^{(i)})$ = average dissimilarity between $x^{(i)}$ and all other points in its cluster
- $b(x^{(i)})$ = average dissimilarity between $x^{(i)}$ and the closest cluster it does not belong to
- $S(x^{(i)}) \in [-1, 1]$ with

$$S(x^{(i)}) = \frac{(b(x^{(i)}) - a(x^{(i)}))}{\max(a(x^{(i)}), b(x^{(i)}))}$$



Note: $S(x^{(i)})$ large: well clustered; $S(x^{(i)})$ small: badly clustered; $S(x^{(i)}) < 0$: assigned to wrong cluster

References

Chapter 14 in

- T. Hastie, R. Tibshirani, & J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.