

LECTURE 8: CLUSTERING ALGORITHMS

STAT 545: INTRO. TO COMPUTATIONAL STATISTICS

Vinayak Rao

Purdue University

September 12, 2018

CLUSTERING

Given a large dataset, group data points into 'clusters'.

Data points in the same cluster are similar in some sense.

E.g. cluster students scores (to decide grade)

Given a large dataset, group data points into 'clusters'.

Data points in the same cluster are similar in some sense.

E.g. cluster students scores (to decide grade)

Applications:

Compression/feature-extraction/exploration/visualization

- simpler representation of complex data

Image segmentation, community detectn, co-expressed genes

CLUSTERING (CONTD.)

We are given N data vectors $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ in \mathbb{R}^d .

Let c_i be the cluster assignment of observation \mathbf{x}_i :

$$c_i \in \{1 \dots K\}$$

Equivalently, we can use one-hot (or 1-of-K) encoding:

$$r_{ic} = \begin{cases} 1, & \text{if } c_i = c \\ 0, & \text{otherwise} \end{cases}$$

Observe: $r_{ic} \geq 0$ and $\sum_{c=1}^K r_{ic} = 1$ just like a probability vector.

However, r_{ic} is binary: we will relax this in later lectures.

CLUSTER PARAMETERS

Associate cluster i with parameter $\theta_i \in \mathbb{R}^d$ (cluster prototype).

Write $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_K\}$, $\mathbf{C} = \{c_1, \dots, c_N\}$ (or $\mathbf{R} = \{\mathbf{r}, \dots, \mathbf{r}_N\}$).

CLUSTER PARAMETERS

Associate cluster i with parameter $\theta_i \in \mathbb{R}^d$ (cluster prototype).

Write $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_K\}$, $\mathbf{C} = \{c_1, \dots, c_N\}$ (or $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_N\}$).

Problem: Given data $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ find $\boldsymbol{\theta}$ and \mathbf{C} .

Define a loss-function $L(\boldsymbol{\theta}, \mathbf{C})$, and minimize it.

CLUSTER PARAMETERS

Associate cluster i with parameter $\theta_i \in \mathbb{R}^d$ (cluster prototype).

Write $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_K\}$, $\mathbf{C} = \{c_1, \dots, c_N\}$ (or $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_N\}$).

Problem: Given data $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ find $\boldsymbol{\theta}$ and \mathbf{C} .

Define a loss-function $L(\boldsymbol{\theta}, \mathbf{C})$, and minimize it.

Start by defining a distance (or similarity measure) $d(\mathbf{x}, \boldsymbol{\theta})$:

$$d(\mathbf{x}, \boldsymbol{\theta}) = \sum_{j=1}^d (x_j - \theta_j)^2 \quad \text{Squared Euclidean or } L_2 \text{ dist.}$$

$$d(\mathbf{x}, \boldsymbol{\theta}) = \sum_{j=1}^d |x_j - \theta_j| \quad L_1 \text{ distance}$$

CLUSTERING LOSS FUNCTION

We want all members of a cluster to be close to the prototype.

$$\sum_{i \text{ s.t. } c_i=c} d(\mathbf{x}_i, \boldsymbol{\theta}_c) = \sum_{i=1}^N r_{ic} d(\mathbf{x}_i, \boldsymbol{\theta}_c) \text{ should be small for each } c.$$

Overall loss function:

$$\begin{aligned} L(\boldsymbol{\theta}, \mathbf{C}) &= \sum_{c=1}^K \sum_{i \text{ s.t. } c_i=c} d(\mathbf{x}_i, \boldsymbol{\theta}_c) \\ &= \sum_{c=1}^K \sum_{i=1}^N r_{ic} d(\mathbf{x}_i, \boldsymbol{\theta}_c) \end{aligned}$$

Optimize over both:

- cluster assignments (discrete)
- cluster parameters (continuous)

Minimizing $L(\boldsymbol{\theta}, \mathbf{C})$ is hard ($O(N^{DK+1})$).

Minimizing $L(\boldsymbol{\theta}, \mathbf{C})$ is hard ($O(N^{DK+1})$).

Instead, use heuristic greedy (local-search) algorithms.

When $d(\cdot, \cdot)$ is Euclidean, the most popular is Lloyd's algorithm.

Minimizing $L(\boldsymbol{\theta}, \mathbf{C})$ is hard ($O(N^{DK+1})$).

Instead, use heuristic greedy (local-search) algorithms.

When $d(\cdot, \cdot)$ is Euclidean, the most popular is Lloyd's algorithm.

If we had the cluster parameters $\boldsymbol{\theta}^*$, can we solve for \mathbf{C} ?

$$\mathbf{C}_{opt} = \operatorname{argmin} L(\boldsymbol{\theta}^*, \mathbf{C})$$

If we had the cluster assignments \mathbf{C}^* , can we solve for $\boldsymbol{\theta}$?

$$\boldsymbol{\theta}_{opt} = \operatorname{argmin} L(\boldsymbol{\theta}, \mathbf{C}^*)$$

Start with an initialization of the parameters, call it θ_0 .

Assign observations to nearest clusters, giving R_0 .

Repeat for i in 1 to N :

- Recalculate cluster means, θ_i
- Recalculate cluster assignments, R_i

Coordinate-descent.

Start with an initialization of the parameters, call it θ_0 .

Assign observations to nearest clusters, giving R_0 .

Repeat for i in 1 to N :

- Recalculate cluster means, θ_i
- Recalculate cluster assignments, R_i

Coordinate-descent.

Resulting algorithm has complexity $O(INKD)$

[demo]

Does this algorithm converge to a global minimum?

Does it converge at all?

What is the convergence criteria?

LIMITATIONS

Local optima: Sensitive to initialization.

Solution: Run many times and pick the best clustering.

Empty clusters.

Solution: discard them, or use heuristics to assign observations to them

Choosing K .

Solution: search over a set of K 's, penalizing larger values.

Requires circular clusters.

Solution: use some other method

Modify distance functions.

L_1 distance: k-medians

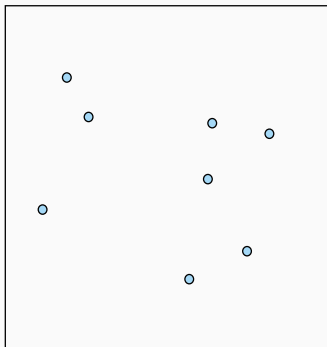
Modify the algorithm.

L_1 distance: k-medoids (exemplar-based)

HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

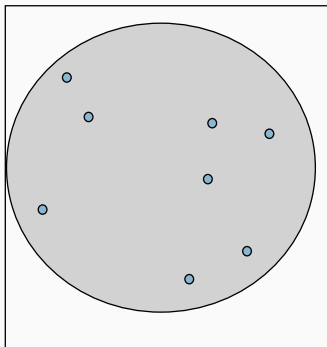
Often there is no clear best clustering.



HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

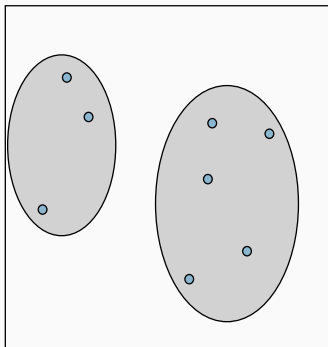
Often there is no clear best clustering.



HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

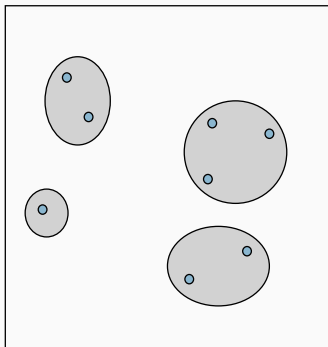
Often there is no clear best clustering.



HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

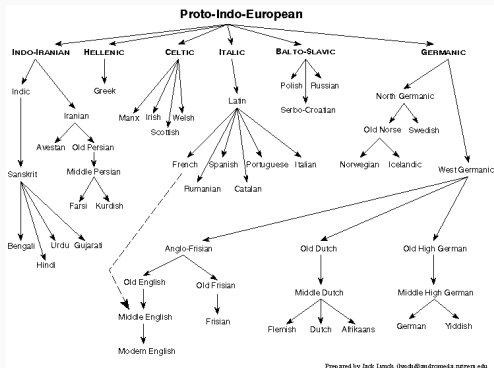
Often there is no clear best clustering.



HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

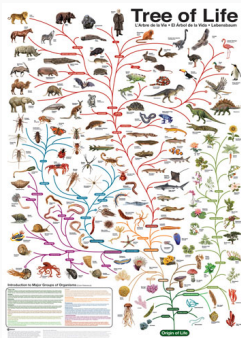
Often it is natural to view data as having a hierarchical structure



HIERARCHICAL CLUSTERING

k-means is a partitioning algorithm that assigns each observation to a unique cluster.

Often it is natural to view data as having a hierarchical structure



TWO APPROACHES:

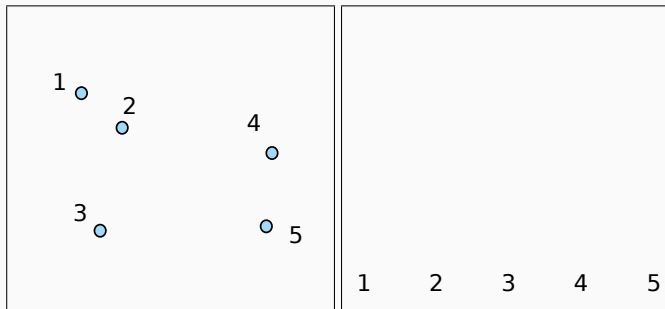
Top-down (divisive) clustering:

- Initialize all observations into a single cluster, and divide clusters sequentially.

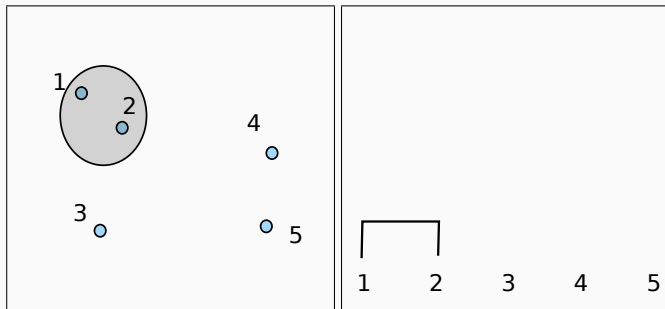
Bottom-up (agglomerative) clustering:

- Initialize each observation in its own cluster, and merge clusters sequentially.
- More flexible, and more common.

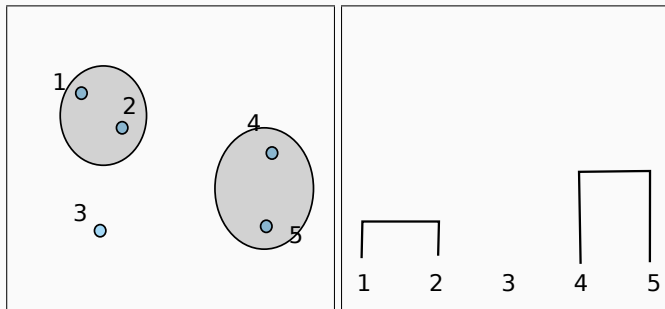
AGGLOMERATIVE CLUSTERING



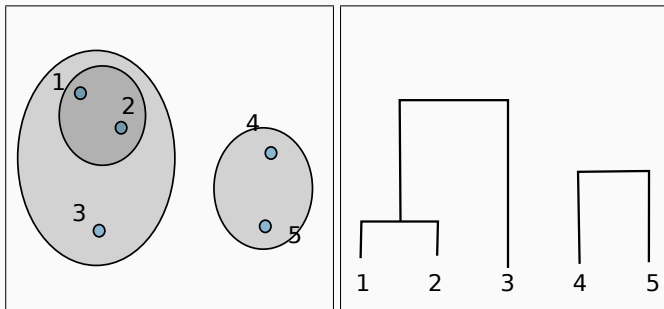
AGGLOMERATIVE CLUSTERING



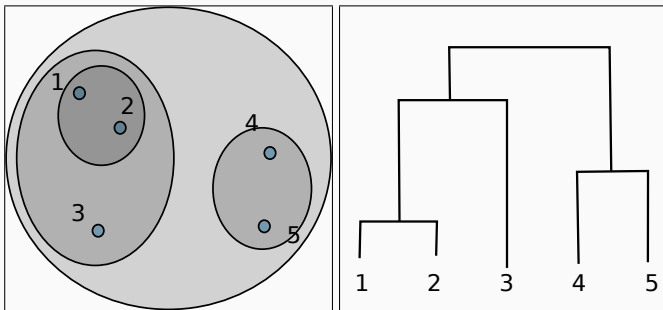
AGGLOMERATIVE CLUSTERING



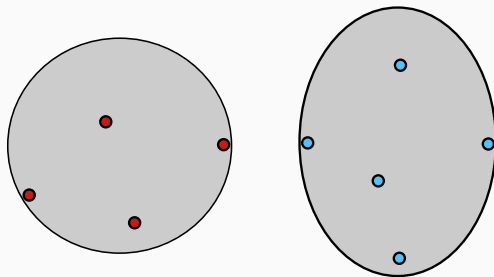
AGGLOMERATIVE CLUSTERING



AGGLOMERATIVE CLUSTERING



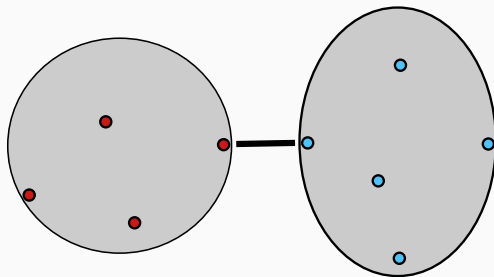
AGGLOMERATIVE CLUSTERING



Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

AGGLOMERATIVE CLUSTERING

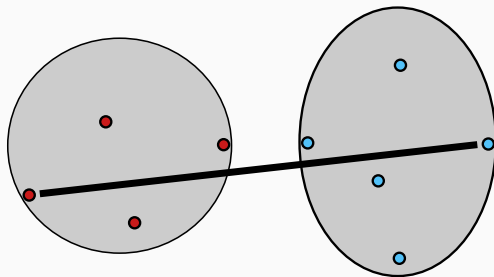


Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

- Single linkage: $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.

AGGLOMERATIVE CLUSTERING

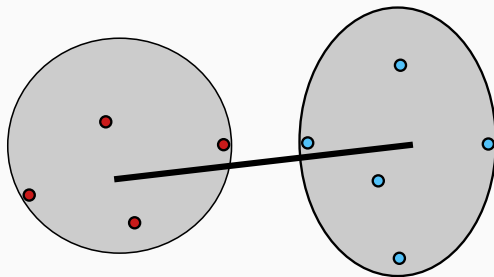


Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

- Single linkage: $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.
- Complete linkage: $d(A, B) = \max_{x \in A, y \in B} d(x, y)$.

AGGLOMERATIVE CLUSTERING

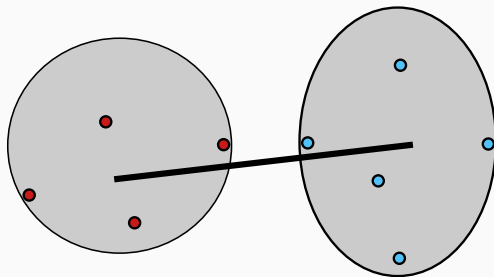


Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

- Single linkage: $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.
- Complete linkage: $d(A, B) = \max_{x \in A, y \in B} d(x, y)$.
- Centroid linkage: $d(A, B) = d(C_A, C_B)$ (C_A : centroid of A).

AGGLOMERATIVE CLUSTERING

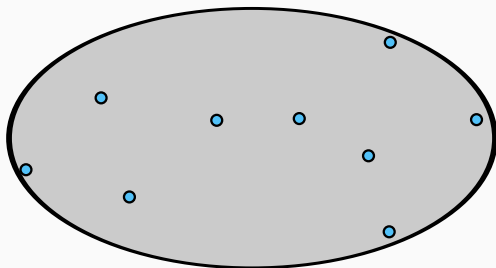


Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

- Single linkage: $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.
- Complete linkage: $d(A, B) = \max_{x \in A, y \in B} d(x, y)$.
- Centroid linkage: $d(A, B) = d(C_A, C_B)$ (C_A : centroid of A).
- Average linkage: $d(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y)$.

AGGLOMERATIVE CLUSTERING



Pick a distance function (e.g. Euclidean).

Pick a linkage criterion defining distance between two clusters:

- Single linkage: $d(A, B) = \min_{x \in A, y \in B} d(x, y)$.
- Complete linkage: $d(A, B) = \max_{x \in A, y \in B} d(x, y)$.
- Centroid linkage: $d(A, B) = d(C_A, C_B)$ (C_A : centroid of A).
- Average linkage: $d(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y)$.