Lab 9. Clustering

Intro to Machine Learning
Fall 2018, Innopolis University

Today’s plan

- K-means clustering
  - Runtime complexity
  - Some tricks to reduce it
- K-means++
  - What problem does it solve
- Hierarchical clustering
  - Runtime complexity
  - Specifics of different linkages
  - Top-down clustering
- HW explanation

K-means clustering

What’s k-means?

Why is it so popular?

Any limitations?

K-means clustering

Recall: **k-means** is trying to minimize the sum of squared distances:

$$\sum_{j=1}^{k} \sum_{i:z_i=j} ||\mu_j - \mathbf{x}_i||_2^2$$

K-means clustering

- Is fairly computationally expensive: It needs to access the data and recalculate distances of every data point with regard to its centroid at each iteration
    - But, it can live without the full pair-wise distance matrix (that can be, in fact, convenient)
- Also, it is sensitive to outliers…
    - A remedy: “K-medoids”

K-means. Runtime complexity

What is its runtime complexity?

K-means. Runtime complexity

```
O(n * K * I * d)

n : number of points

K : number of clusters

I : number of iterations

d : number of attributes
```

K-means. Runtime complexity

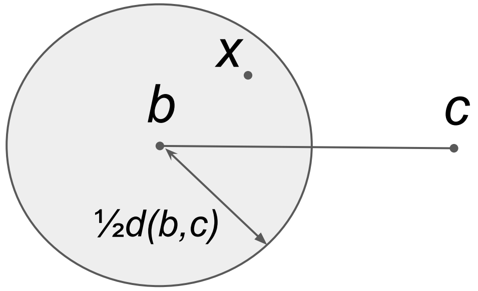
Can we do better?

K-means. Runtime complexity

Can we do better? - Yes! If we exploit geometrical properties
For example:

Lemma 1: Let x be a point and let b and c be centers. If $d(b,c) \geq 2d(x,b)$ then $d(x,c) \geq d(x,b)$.

That is, if we know inequality holds true, there’s no need to calculate **d(x,c)**



K-means++

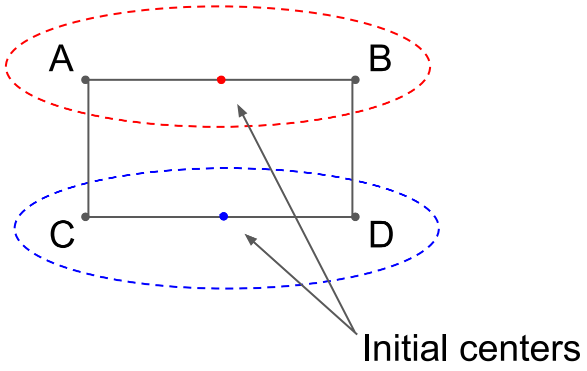
What’s the problem with k-mean?

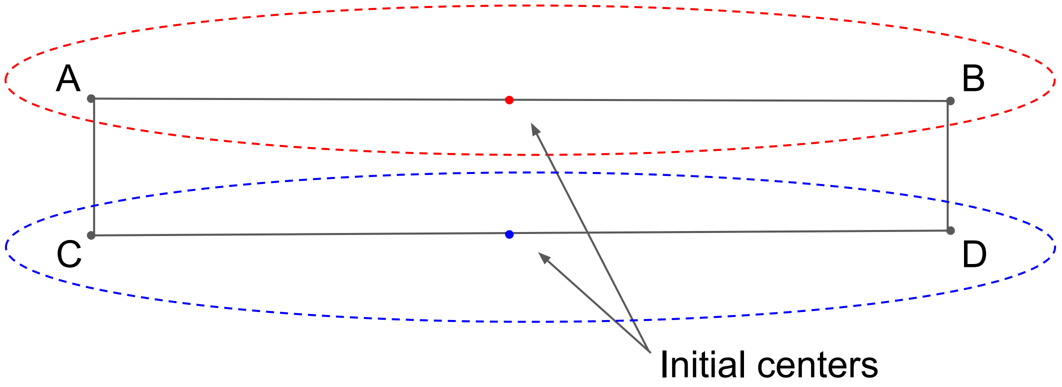
The approximation found can be ***arbitrarily bad*** with respect to the objective function compared to the optimal clustering

K-means++

What’s the problem with k-mean?

Consider the following situation:





K-means++

So, to fight this, we try to spread out the *k* initial cluster centers:

- The first cluster center is chosen uniformly at random from the data points that are being clustered,
- Each subsequent cluster center is chosen from the remaining data points with probability *proportional to its squared distance* from the point's closest existing cluster center.

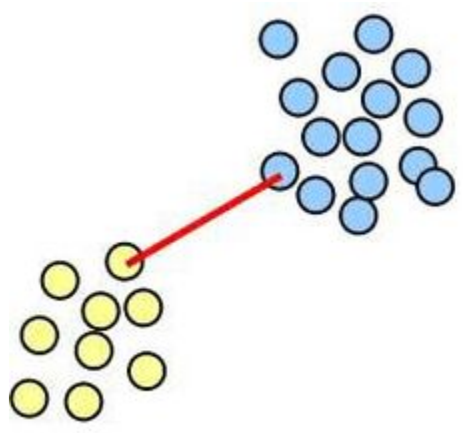
Hierarchical clustering

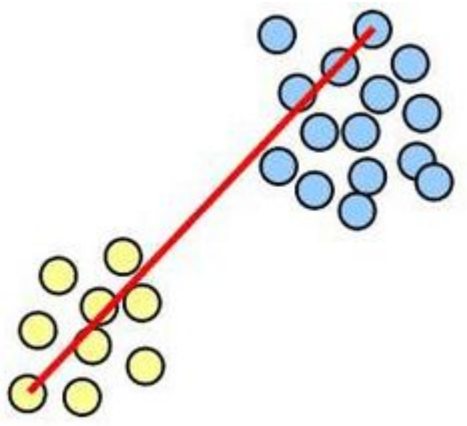
Top-down

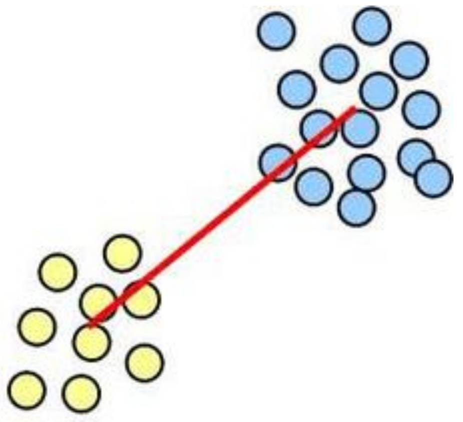
Bottom-up

Hierarchical clustering. Bottom Up

What is its runtime complexity?
Recall, each iteration we a combining two closest clusters. Closest with respect to chosen linkage:

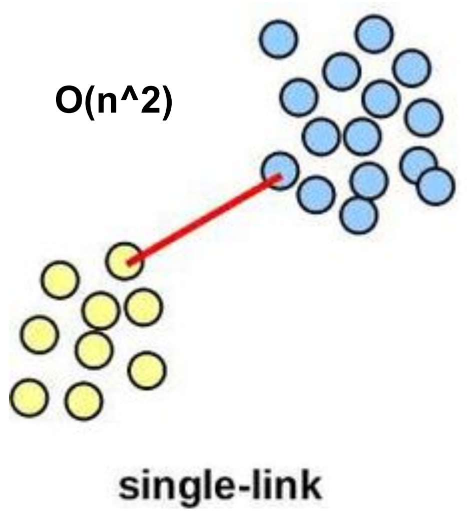

single-link

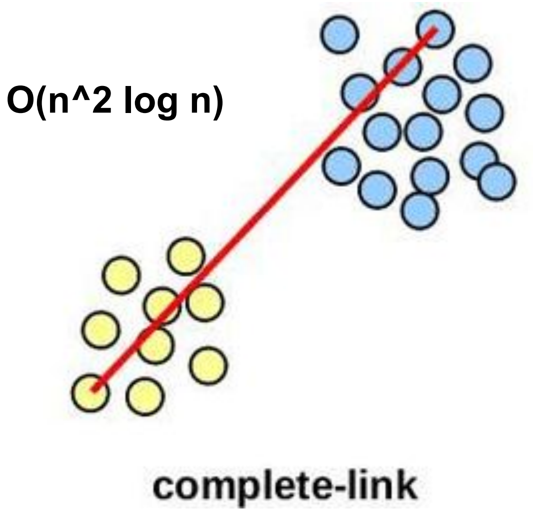

complete-link

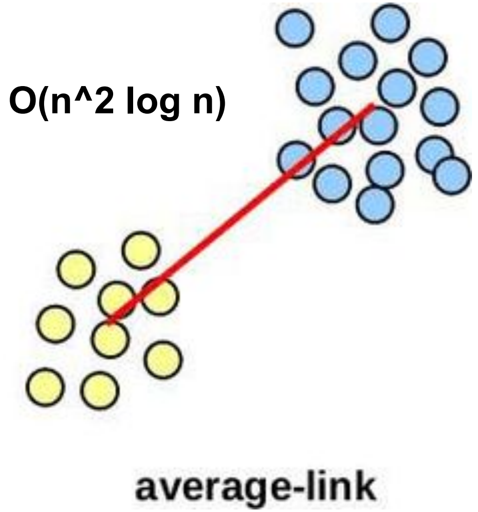

average-link

Hierarchical clustering. Bottom Up

What is its runtime complexity?
Recall, each iteration we a combining two closest clusters. Closest with respect to chosen linkage:







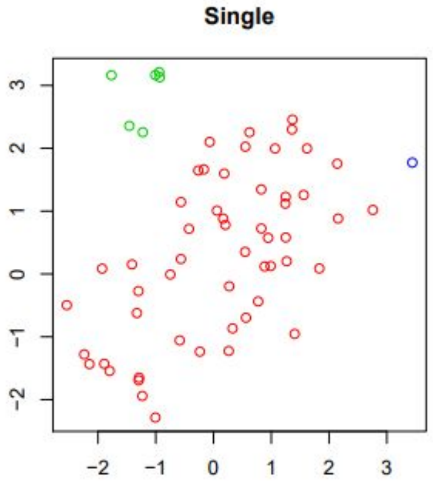
Limitations of different linkage measures

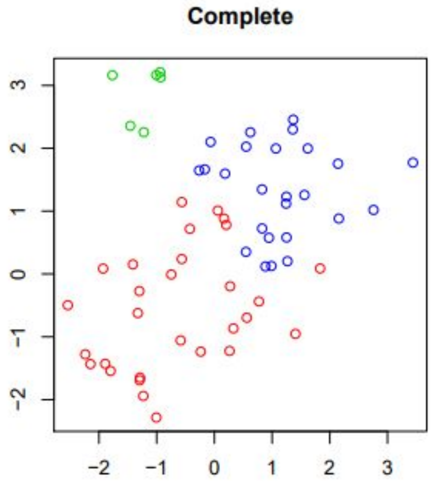
Single linkage suffers from chaining. In order to merge two groups, only need one pair of points to be close, irrespective of all others. Therefore clusters can be too spread out, and not compact enough

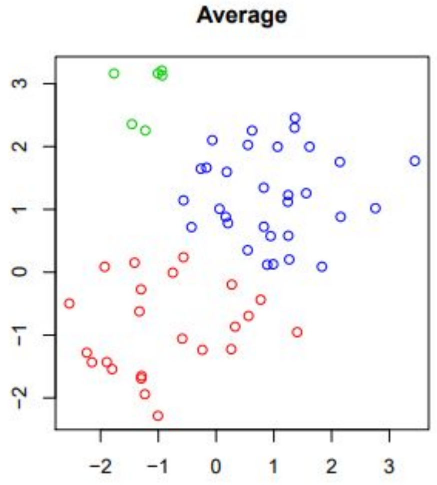
Complete linkage suffers from crowding. A point can be closer to points in other clusters than to points in its own cluster. Clusters are compact, but not far enough apart

Average linkage tries to strike a balance. It uses average pairwise dissimilarity, so clusters tend to be relatively compact and relatively far apart

Limitations







Hierarchical clustering. Top-down

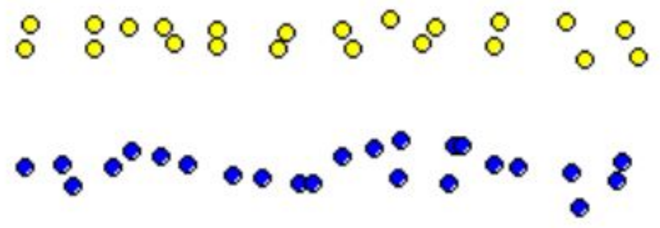
Top-down clustering is conceptually more complex than bottom-up clustering since we need a second, flat clustering algorithm as a “subroutine”.

It has the advantage of being *more efficient* - for a fixed number iterations, using an efficient flat algorithm like K-means, top-down algorithms are linear in the number of points and clusters. So they run much faster than bottom-up algorithms, which are at least quadratic.

Exercise

Which clustering method(s) is most likely to produce the following results at k = 2? Why it/they will work better where others will not? Your options:

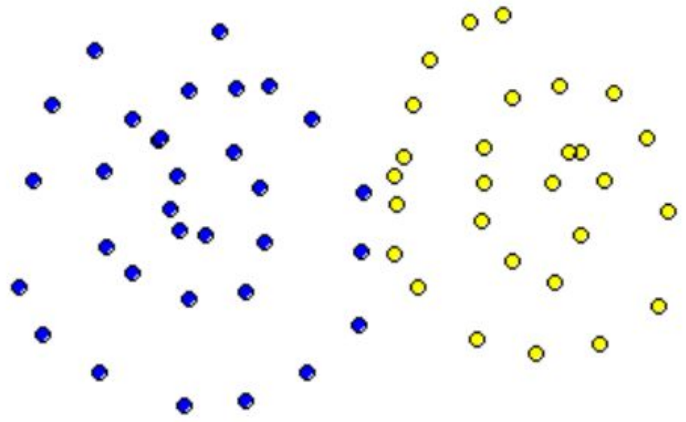
- Hierarchical clustering with single link
- Hierarchical clustering with complete link
- Hierarchical clustering with average link
- K-means



Exercise

Which clustering method(s) is most likely to produce the following results at k = 2? Why it/they will work better where others will not? Your options:

- Hierarchical clustering with single link
- Hierarchical clustering with complete link
- Hierarchical clustering with average link
- K-means



Practice

Download `Mall_Customers.csv` dataset and `hc.py` file from Moodle

Execute it using different linkage options, see the difference in resulting dendrograms

Looking at dendrogram, choose the number of clusters and explore the resulting clusters.

Homework

Implement bottom-up clustering using different linkage functions (single vs complete vs average). Template is provided.

Home Reading

https://nlp.stanford.edu/IR-book/completelink.html