Foundation of Data Science
Lecture 7, Module 2
Fall 2022

Rumi Chunara, PhD

Two techniques

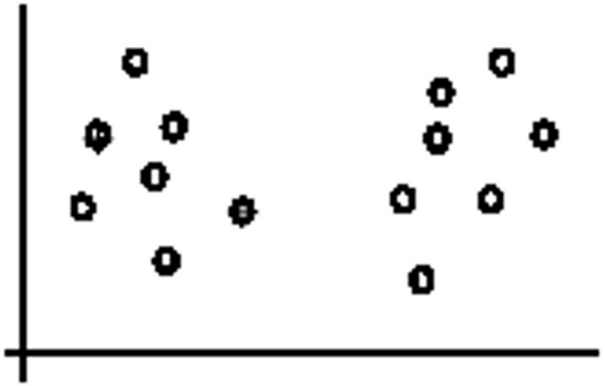
K-Means (partitioning)

- Clusters are defined by a center point
- # of groupings chosen in advance
- Each object belongs to the cluster in which it has the minimum distance to the cluster center
- Generally cheaper, but not stable

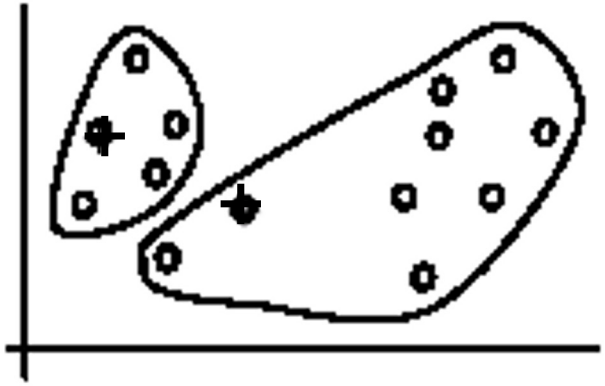
Hierarchical

- Clusters are arranged in a nested taxonomy
- # of groupings can be chosen a posteriori
- Stable but computationally expensive

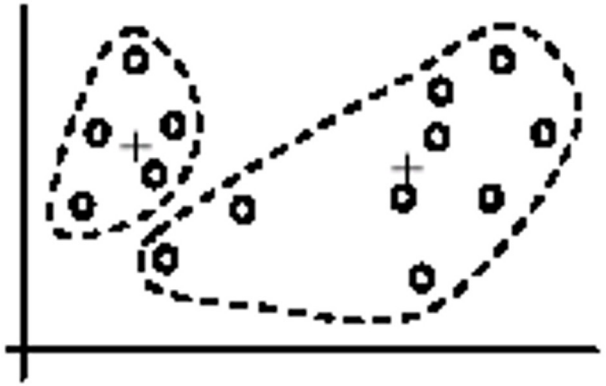
K-means: Lloyd's Algorithm



(A). Random selection of k centers



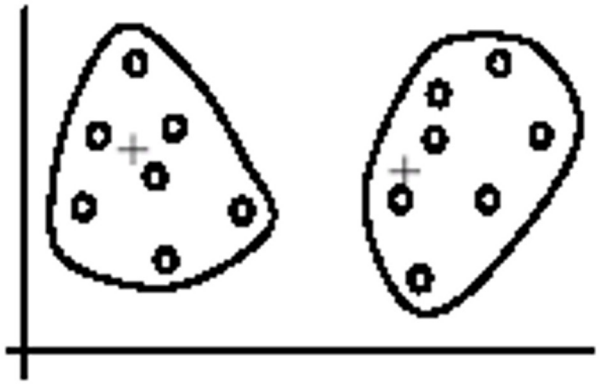
Iteration 1: (B). Cluster assignment



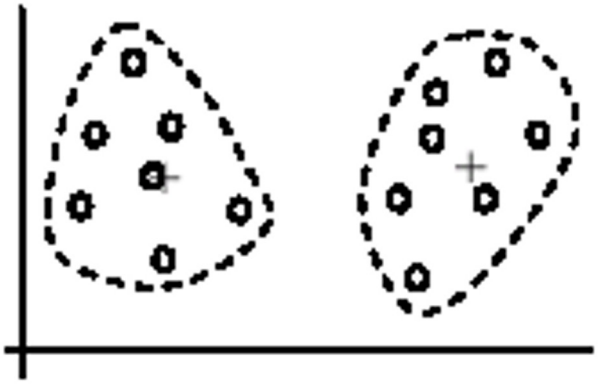
(C). Re-compute centroids

https://en.wikipedia.org/wiki/Lloyd%27s_algorithm

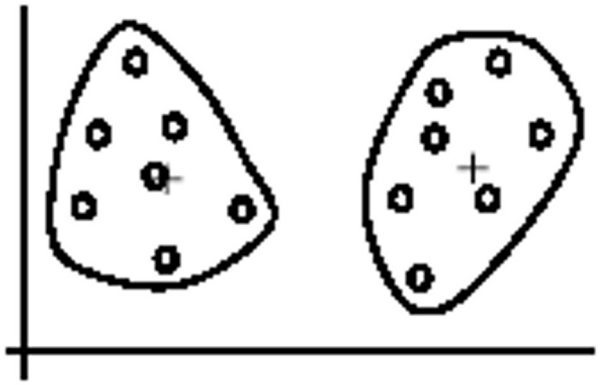
K-means: Lloyd's Algorithm



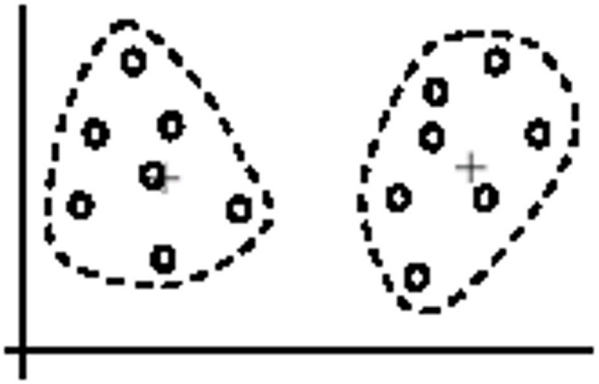
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

https://en.wikipedia.org/wiki/Lloyd%27s_algorithm

K-means clustering

Iterate (stopping criteria):

- For fixed number of iterations
- Until no change in assignments
- Until small change in quality

K-means Properties

- It is a greedy algorithm with random setup – **solution is not optimal** and varies significantly with different initial points

- Very simple convergence proofs
- **Performance is O(nk) per iteration** -- not bad and can be heuristically improved
  - n = total features in the dataset, k = number clusters

K-means Properties

- It is a greedy algorithm with random setup – **solution is not optimal** and varies significantly with different initial points

- Very simple convergence proofs
- **Performance is O(nk) per iteration** -- not bad and can be heuristically improved
  - n = total features in the dataset, k = number clusters

- *Elkan, Charles. "Using the triangle inequality to accelerate k-means." Proceedings of the 20th international conference on Machine Learning (ICML-03). 2003.*

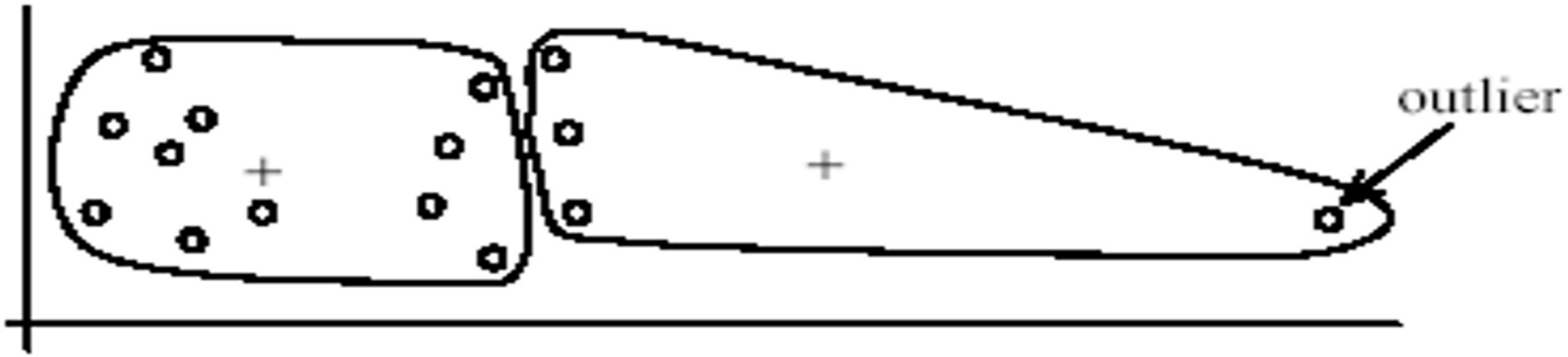
K-means: Advantages

- Simple: easy to understand and to implement
- Efficient: the time complexity is $O(tkn)$ where:
  - n is the number of data points
  - k is the number of clusters
  - t is the number of iterations
  - If k and t are small (they usually are), k-means is roughly linear on n
- *K*-means is the most popular clustering algorithm
- Note that k-means terminates at a local optimum. Finding the global optimum is NP-hard

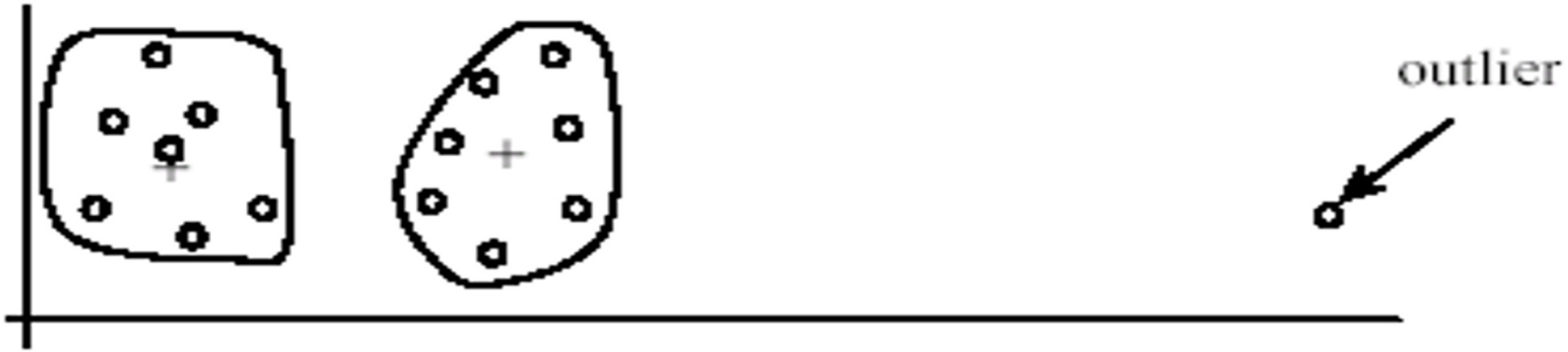
K-means: Disadvantages

- The algorithm is only applicable if the mean is defined
  - For categorical data, use *k*-mode - the centroid is represented by most frequent values
- The user needs to specify *k*
- The algorithm is very sensitive to the initial seeds
- The algorithm is sensitive to **outliers**
  - Data points that are very far away from others
  - Errors in the data recording
  - Special data points with very different values
- It is important to scale the data

Weaknesses of *K*-means: Outliers



(A): Undesirable clusters



(B): Ideal clusters

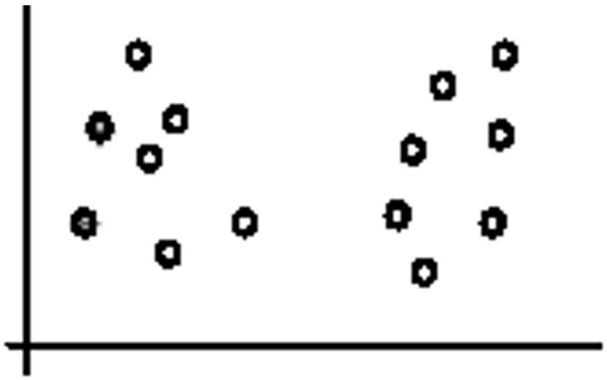
Dealing with outliers

- One method: remove those data points in the clustering process that are too far from the centroids
  - Monitor these outliers over a few iterations and then decide whether to remove them

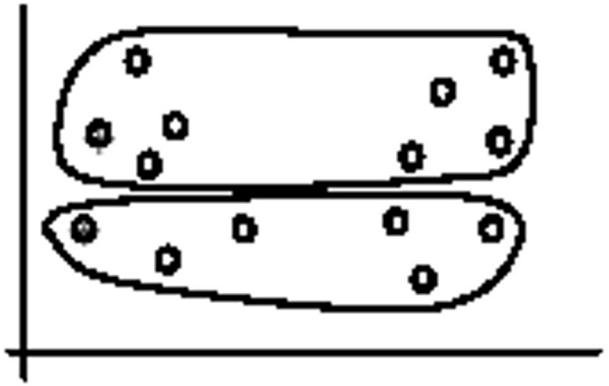
- Another method: perform random sampling -- the chance of selecting an outlier is very small
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification (each cluster is a class)

Initial Seeds

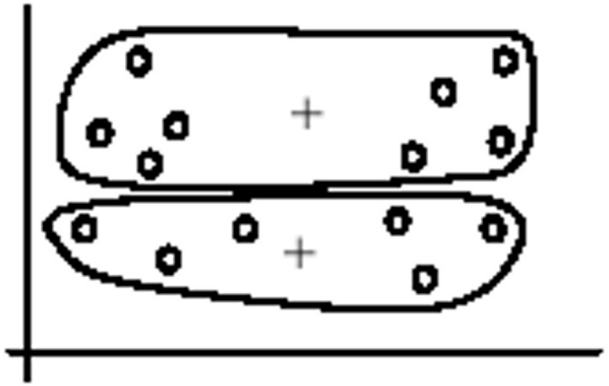
- The algorithm is sensitive to initial seeds.



(A). Random selection of seeds (centroids)



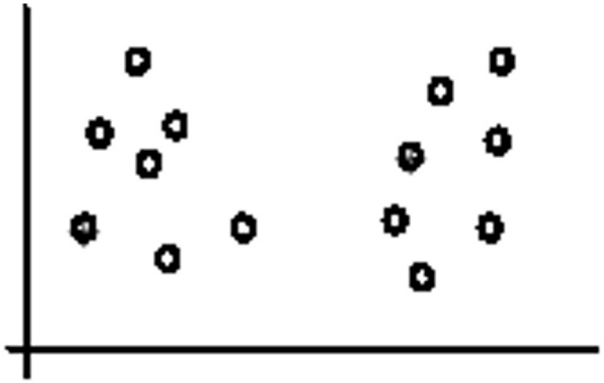
(B). Iteration 1



(C). Iteration 2

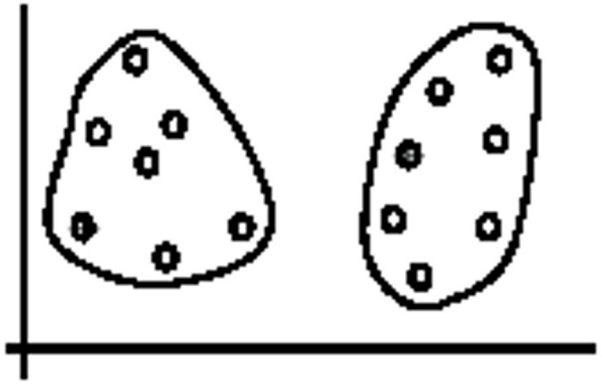
Initial Seeds

- If we use different seeds: good results

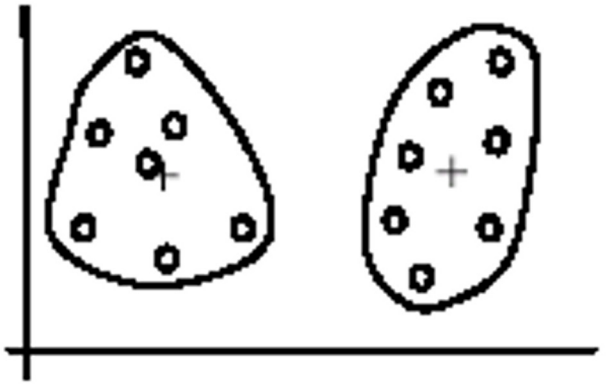


There are methods to help choose good seeds!

(A). Random selection of k seeds (centroids)



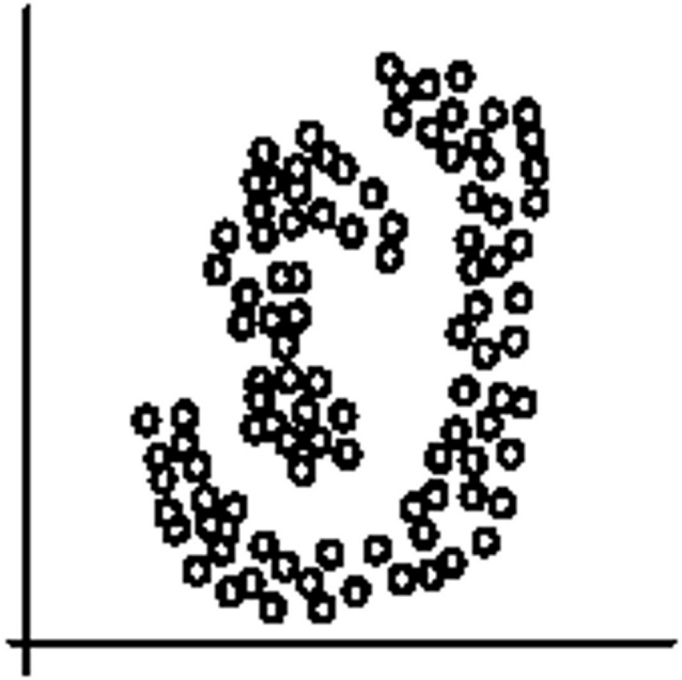
(B). Iteration 1



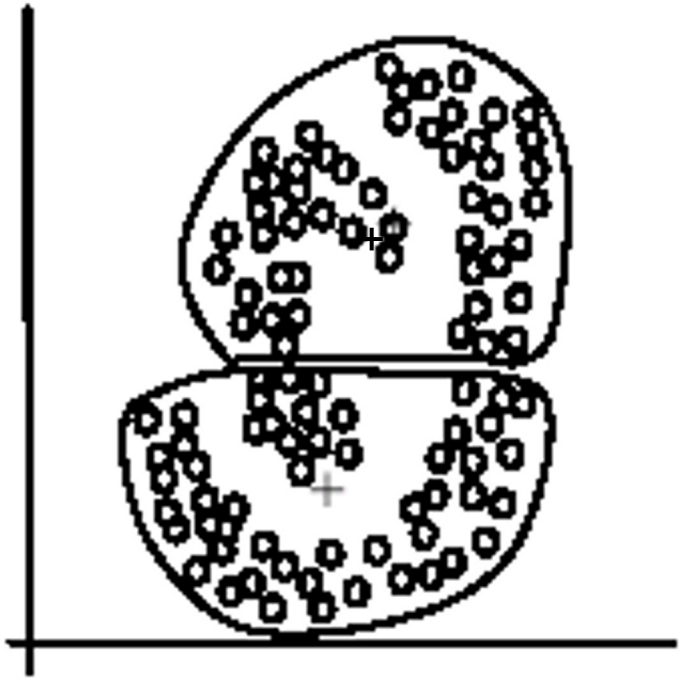
(C). Iteration 2

Initial Seeds

- The *k*-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres)



(A): Two natural clusters



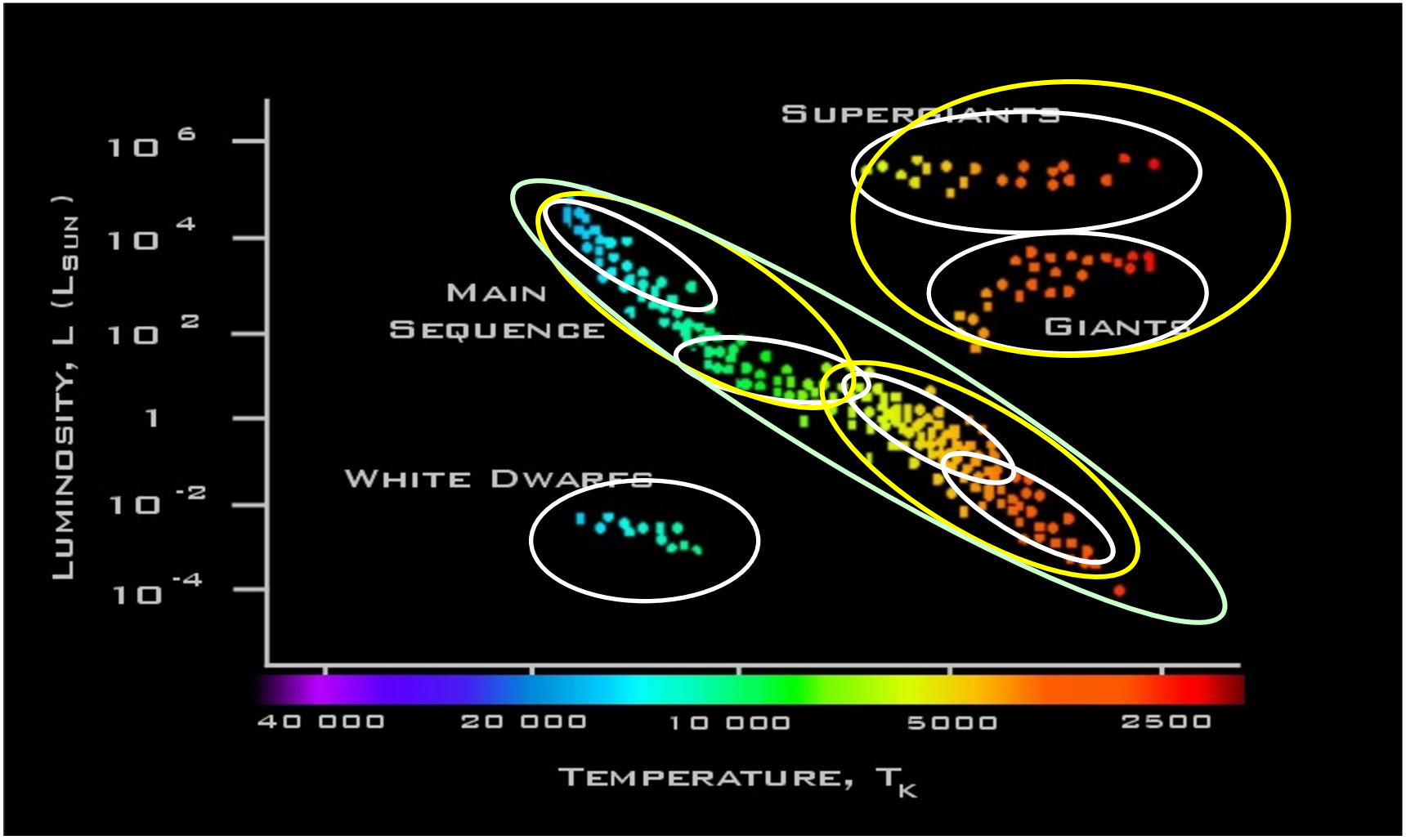
(B): *k*-means clusters

K-means: summary

- Despite disadvantages, *k*-means is still the most popular algorithm due to its simplicity and efficiency
- No clear evidence that any other clustering algorithm performs better in general
  - Although they may be more suitable for some specific types of data or applications
- Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

Cluster Hierarchy

Clusters can be embedded in other clusters. Hierarchical clustering attempts to uncover this embedding.



Generic algorithm

This is the generic algorithm for agglomerative clustering:

1. Compute all pairwise similarities (what is the cost?)

2. Place each instance into its own cluster (n clusters at first)

3. Merge the two most similar clusters into one
   - Replace two clusters into the new cluster
   - Recompute intercluster similarity scores

4. Repeat until there are only k clusters left

Example

Imagine a group of students filling out a questionnaire about their Data Science interests.

You want to to verify how they are clustered interest-wise.

You start by computing the pairwise similarity for their answers.

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

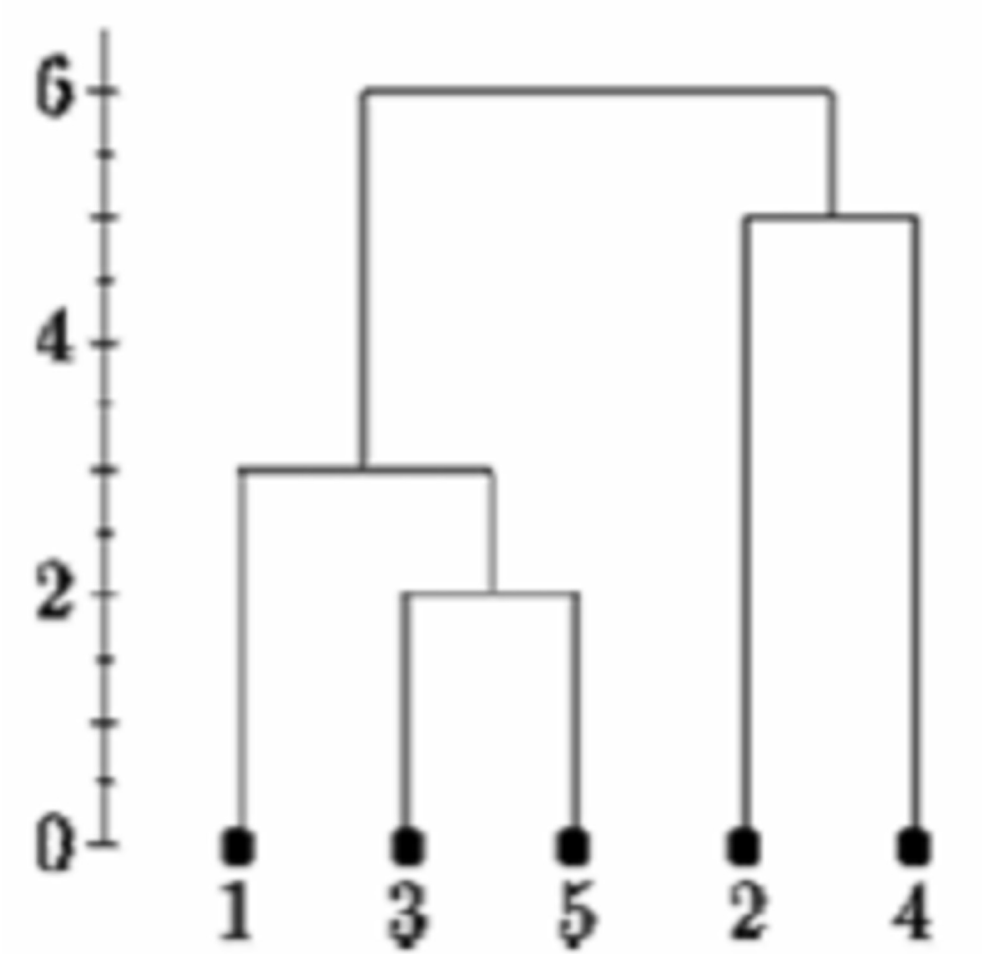
Note that students 3 and 5 are the closest in terms of interest!

Dendrogram

Hierarchical clustering gives us the opportunity to plot the nested nature of the clusters with a dendrogram

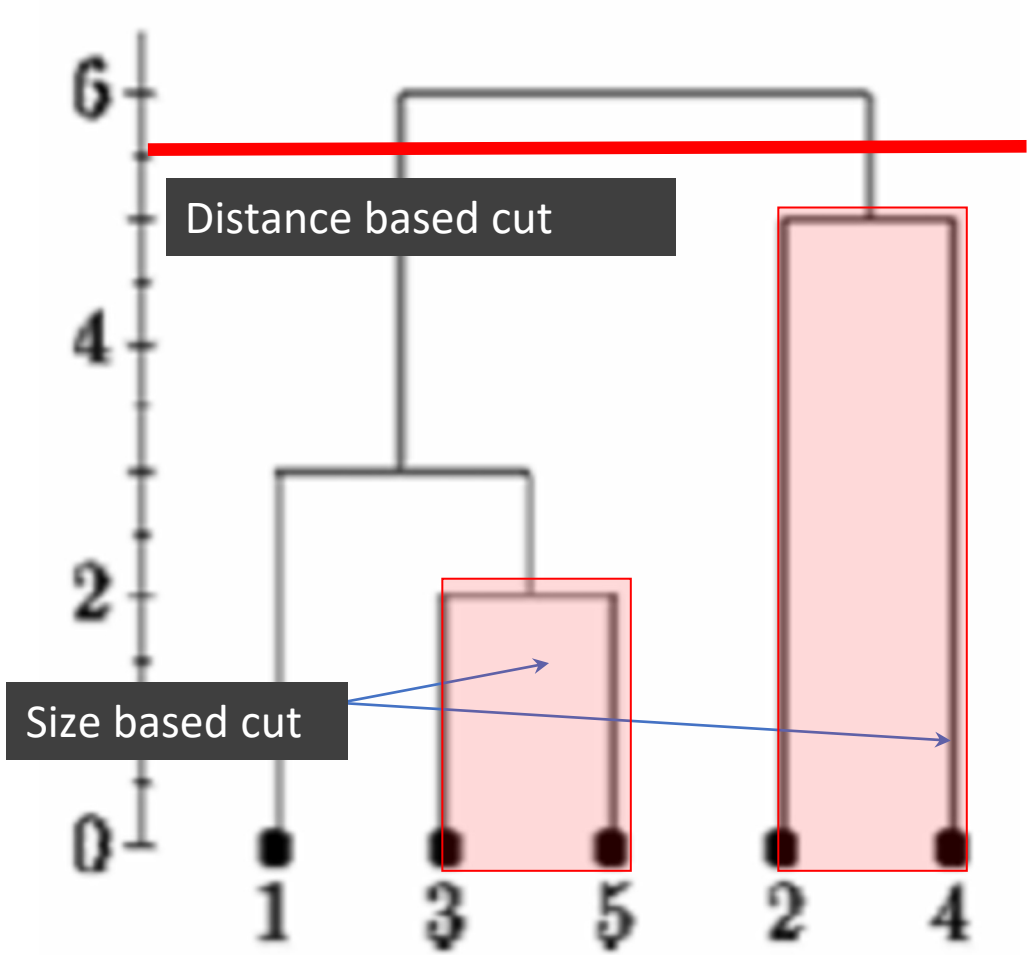
Single linkage
d(1, “35”) =
min(d(1, 3), d(1,5))

There are other ways of determining cluster distances!



Using the Dendrogram

We can use the dendrogram to cut the space into *k* clusters based on distance or size



Size based: note that we have 2 clusters with the same size (2), and one singleton!

How to choose a clustering algorithm

- Clustering research has a long history. A vast collection of algorithms are available
  - We only introduced some main algorithms

- Choosing the "best" algorithm is a challenge
  - Every algorithm has limitations and works well with certain data distributions
  - It is very hard, if not impossible, to know what distribution the application data follows. The data may not fully follow any "ideal" structure or distribution required by the algorithms
  - One also needs to decide how to standardize the data, to choose a suitable distance function and to select other parameter values

How to choose a clustering algorithm

- Due to these complexities, the common practice is to:
  - run several algorithms using different distance functions and parameter settings
  - carefully analyze and compare the results

- The interpretation of the results must be based on insight into the meaning of the original data together with knowledge of the algorithms used

- Clustering is highly application dependent and to a certain extent subjective (domain/task priorities)

Clustering Evaluation

- Not trivial!
- Evaluation should not take the absolute values of the cluster labels into account
  - Rather, the clustering strategy should define separations of the data similar to some ground truth set or satisfying assumption
- Getting ground truth data to evaluate clusters is challenging
  - How would you try to create that?