Clustering

Emma Yu and Jaya Zenchenko

WiDS Austin, Jan 23, 2017

Slides adapted from David Blei, David Sontag, Piyush Rai, Jure Leskove, Anand Rajaraman, Shimon Ullman and Elena Marchiori

What is clustering?

- Given a set of data points, group them into clusters so that:
  - points within each cluster are similar to each other
  - points from different clusters are dissimilar

- Requires data, but no labels

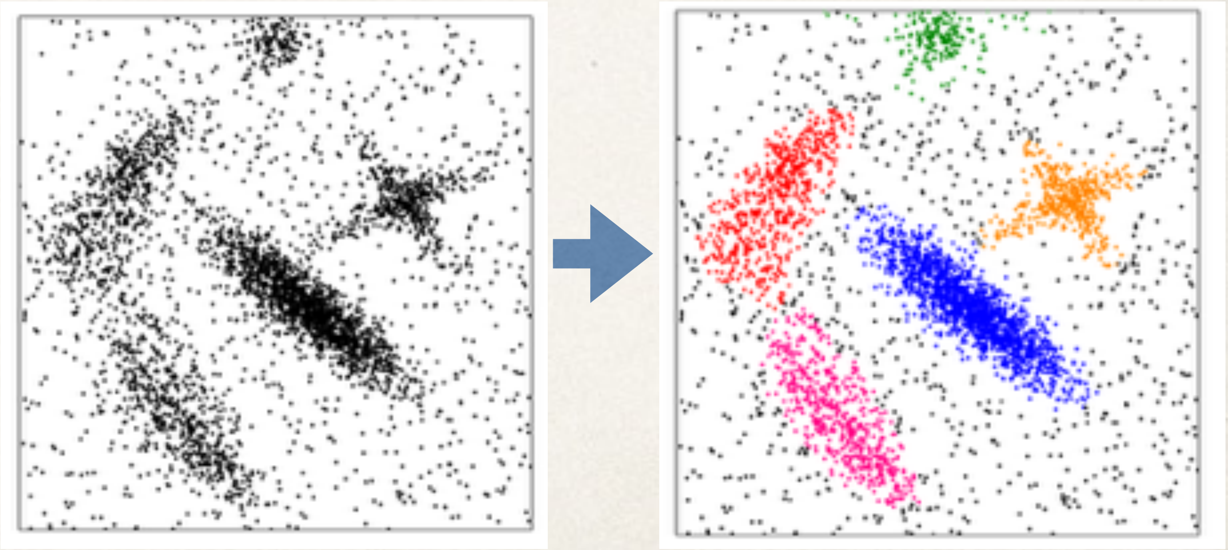


Image credit: http://people.sissa.it/~laio/Research/Res_clustering.php

Why would we want to do this?

- Group customers according to purchase histories
- Group search results according to topics
- Detect regions of images
- Clustering gene expression data

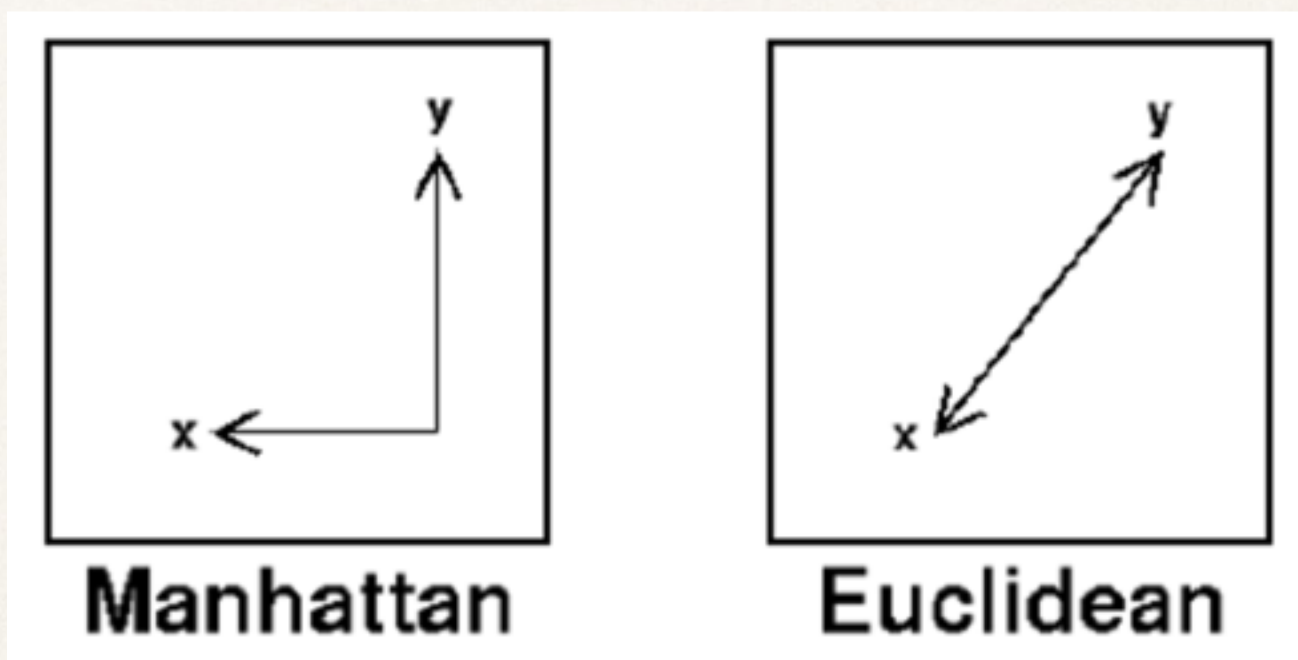
K=2	K=3	K=10	Original

4%	8%	17%	

what does similar mean, exactly?

what does similar mean, exactly?

- It depends on how we define the distance/similarity
- Euclidean distance, Manhattan (L1) distance

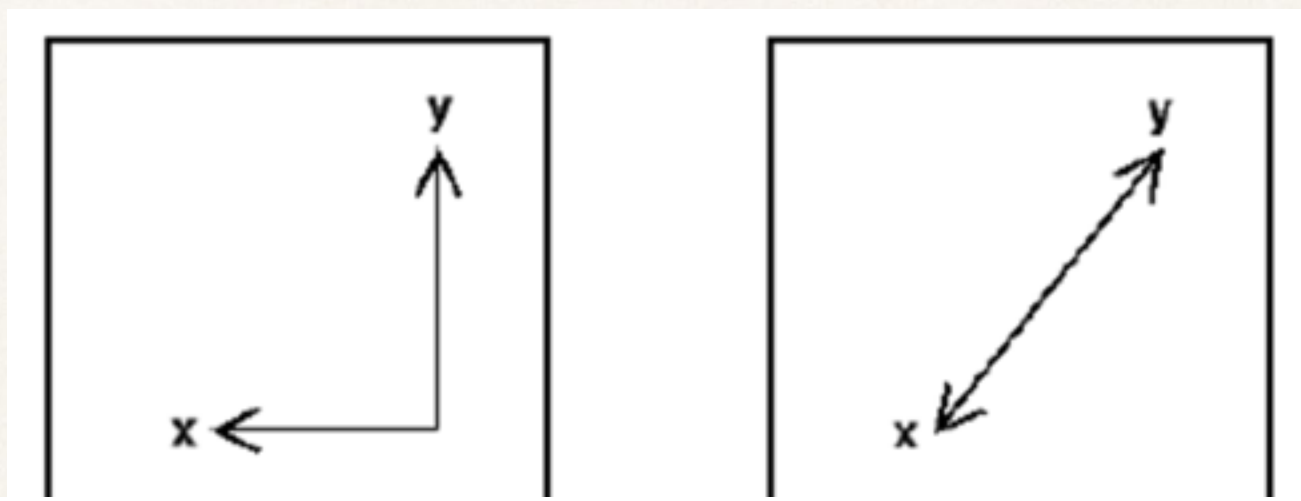


$$\sum_{i=1} |a_i - b_i|.$$

$$\sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}.$$

what does similar mean, exactly?

- It depends on how we define the distance/similarity
- Euclidean distance, Manhattan (L1) distance



They are special cases of **Minkowski distance:**

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^p \right)^{\frac{1}{p}}$$

- Jaccard distance

$$\mathbf{d}_J(A,B) = 1 - \mathsf{JS}(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}.$$

- Cosine distance

$$\mathbf{d}_{\cos}(a,b) = 1 - \frac{\langle a, b \rangle}{\|a\| \, \|b\|} = 1 - \frac{\sum_{i=1}^{d} a_i b_i}{\|a\| \, \|b\|}$$

How do we do it?

- Partition algorithms: determine all clusters at once.
- Hierarchical algorithms (agglomerative): find successive clusters using previously established clusters.
  - Do not need the number of clusters as an input, and can be viewed at different levels of granularities with different k

K-means

- A partition algorithm
- Basic idea: to describe each cluster by its mean value
- Goal: find the assignment of k clusters that minimizes the sum of square distance of cluster members to their cluster centers.

1. Initialization
   - Data are $\mathbf{x}_{1:N}$
   - Choose initial cluster means $\mathbf{m}_{1:k}$ (same dimension as data).

1. Initialization
   - Data are $\mathbf{x}_{1:N}$
   - Choose initial cluster means $\mathbf{m}_{1:k}$ (same dimension as data).
2. Repeat
   1. Assign each data point to its closest mean

$$z_n = \arg \min_{i \in \{1, \ldots, k\}} d(\mathbf{x}_n, \mathbf{m}_i)$$

1. Initialization
   - Data are $\mathbf{x}_{1:N}$
   - Choose initial cluster means $\mathbf{m}_{1:k}$ (same dimension as data).

2. Repeat
   1. Assign each data point to its closest mean

$$z_n = \arg \min_{i \in \{1,\ldots,k\}} d(\mathbf{x}_n, \mathbf{m}_i)$$

   2. Compute each cluster mean to be the coordinate-wise average over data points assigned to that cluster,

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{\{n : z_n = k\}} \mathbf{x}_n$$

1. Initialization
   - Data are $\mathbf{x}_{1:N}$
   - Choose initial cluster means $\mathbf{m}_{1:k}$ (same dimension as data).

2. Repeat
   1. Assign each data point to its closest mean

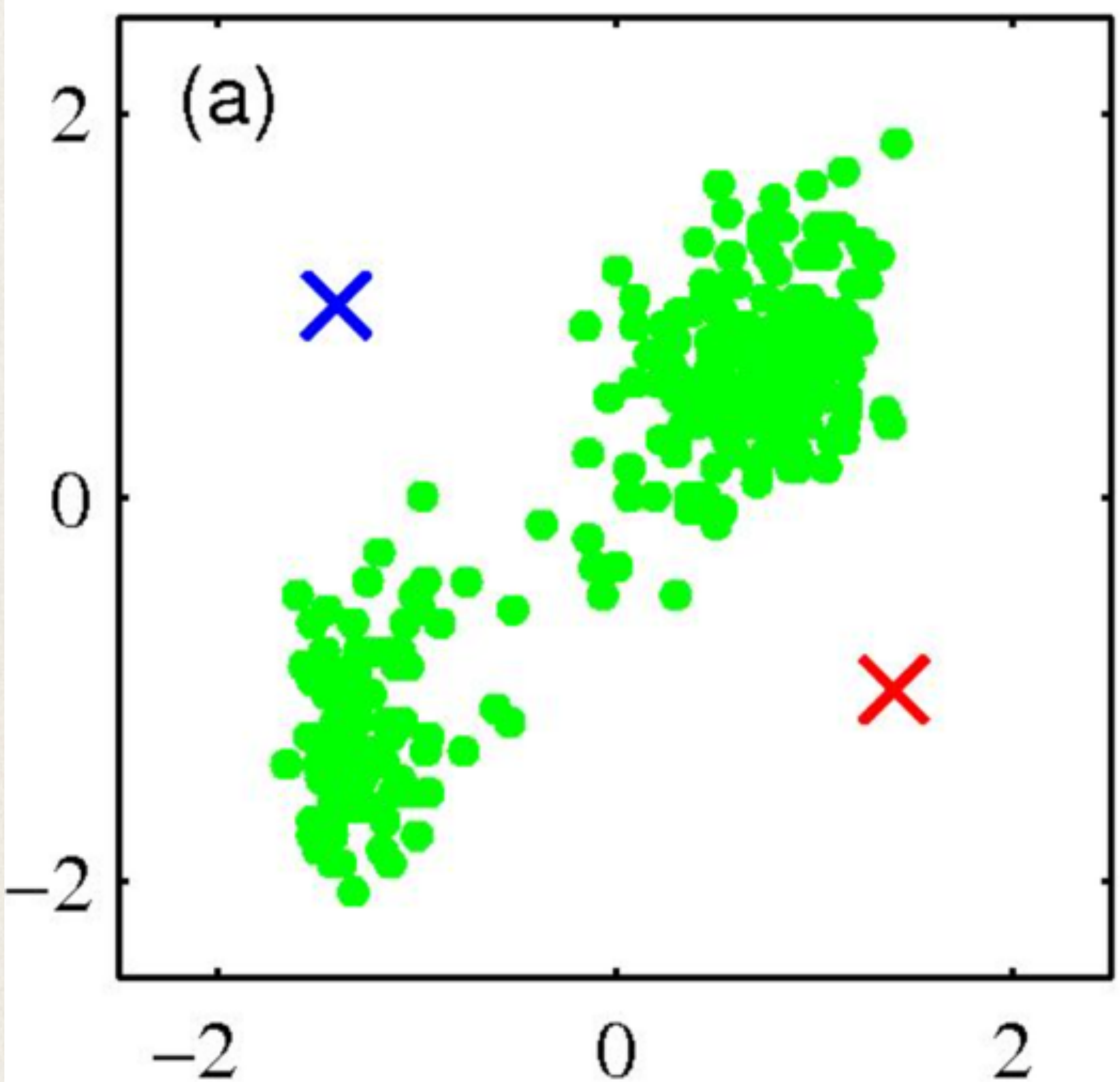
$$z_n = \arg \min_{i \in \{1,\ldots,k\}} d(\mathbf{x}_n, \mathbf{m}_i)$$

   2. Compute each cluster mean to be the coordinate-wise average over data points assigned to that cluster,

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{\{n : z_n = k\}} \mathbf{x}_n$$

3. Until assignments $\mathbf{z}_{1:N}$ do not change

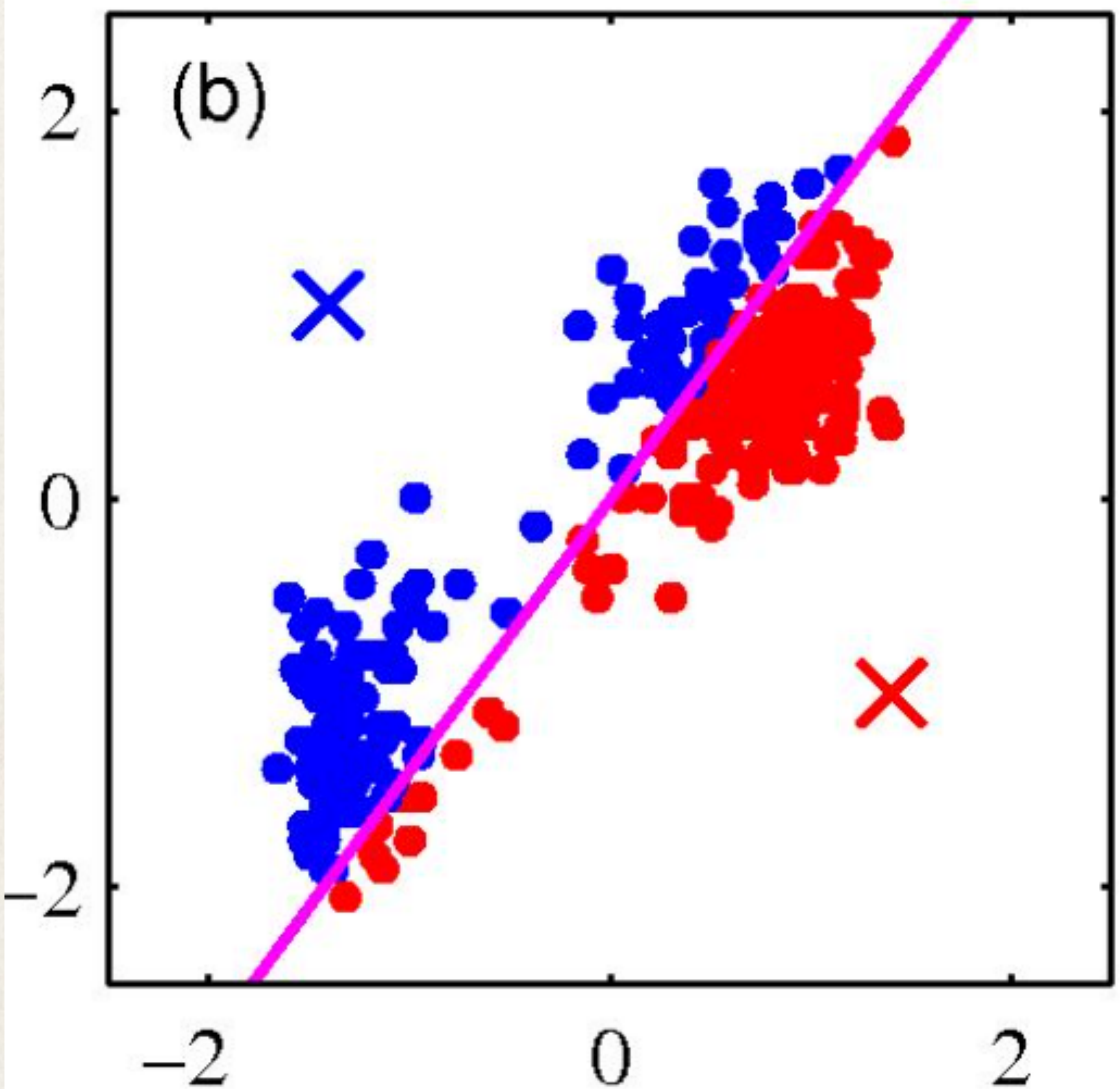
K-means — an example



- Pick K random points as cluster centers (means)

Shown here for K=2

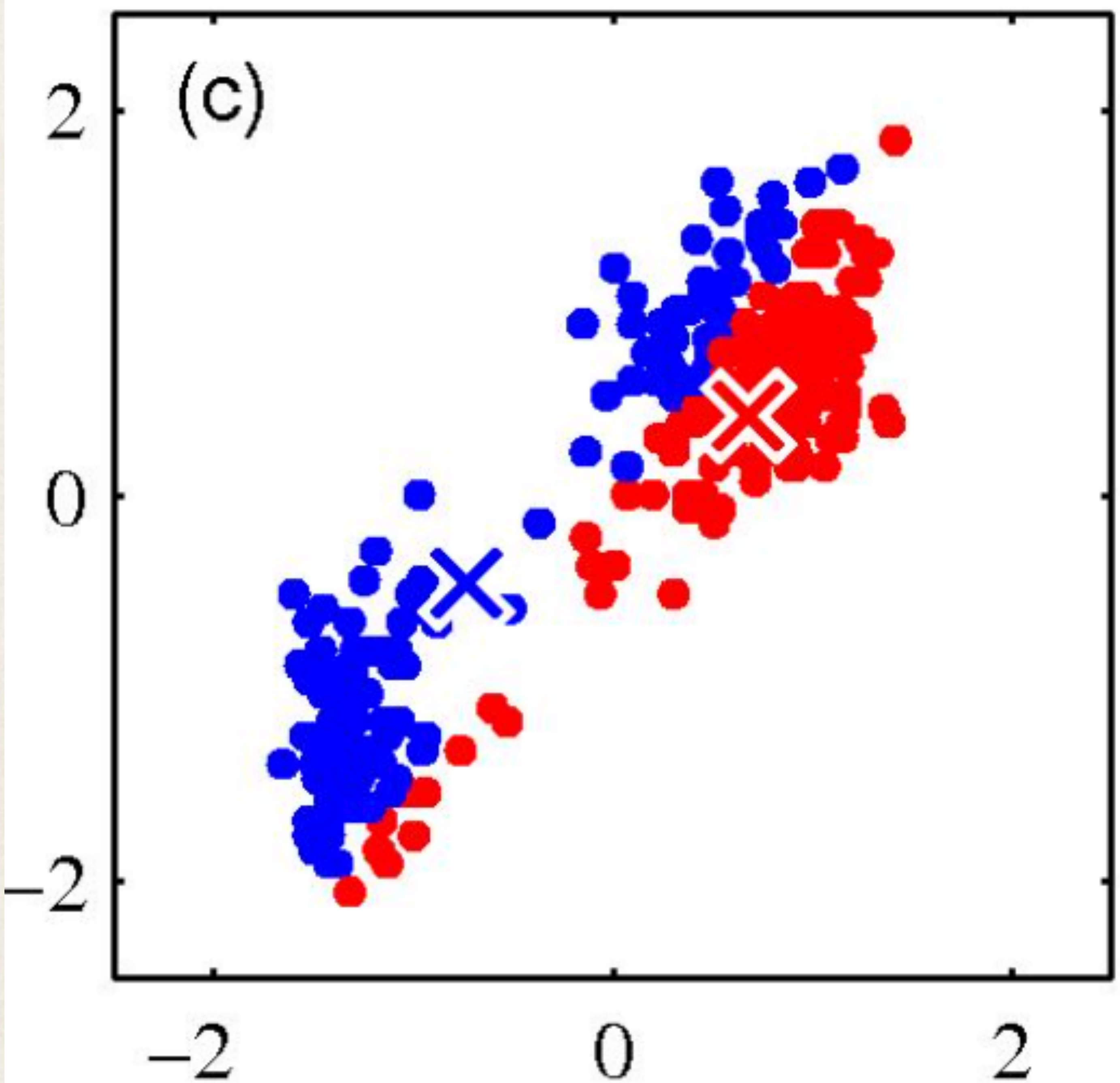
K-means — an example



Iterative Step 1

- Assign data points to closest cluster center

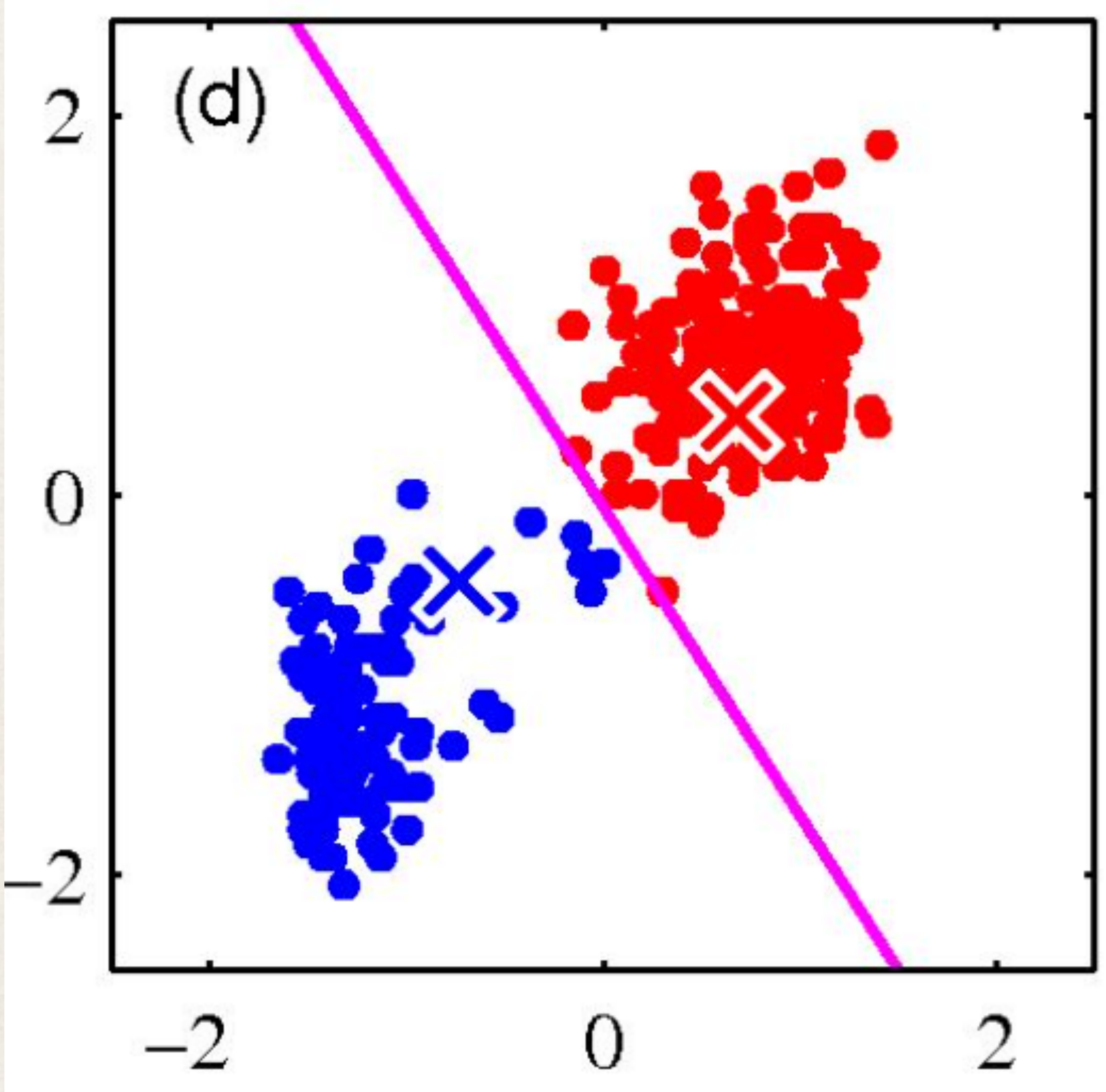
K-means — an example



Iterative Step 2

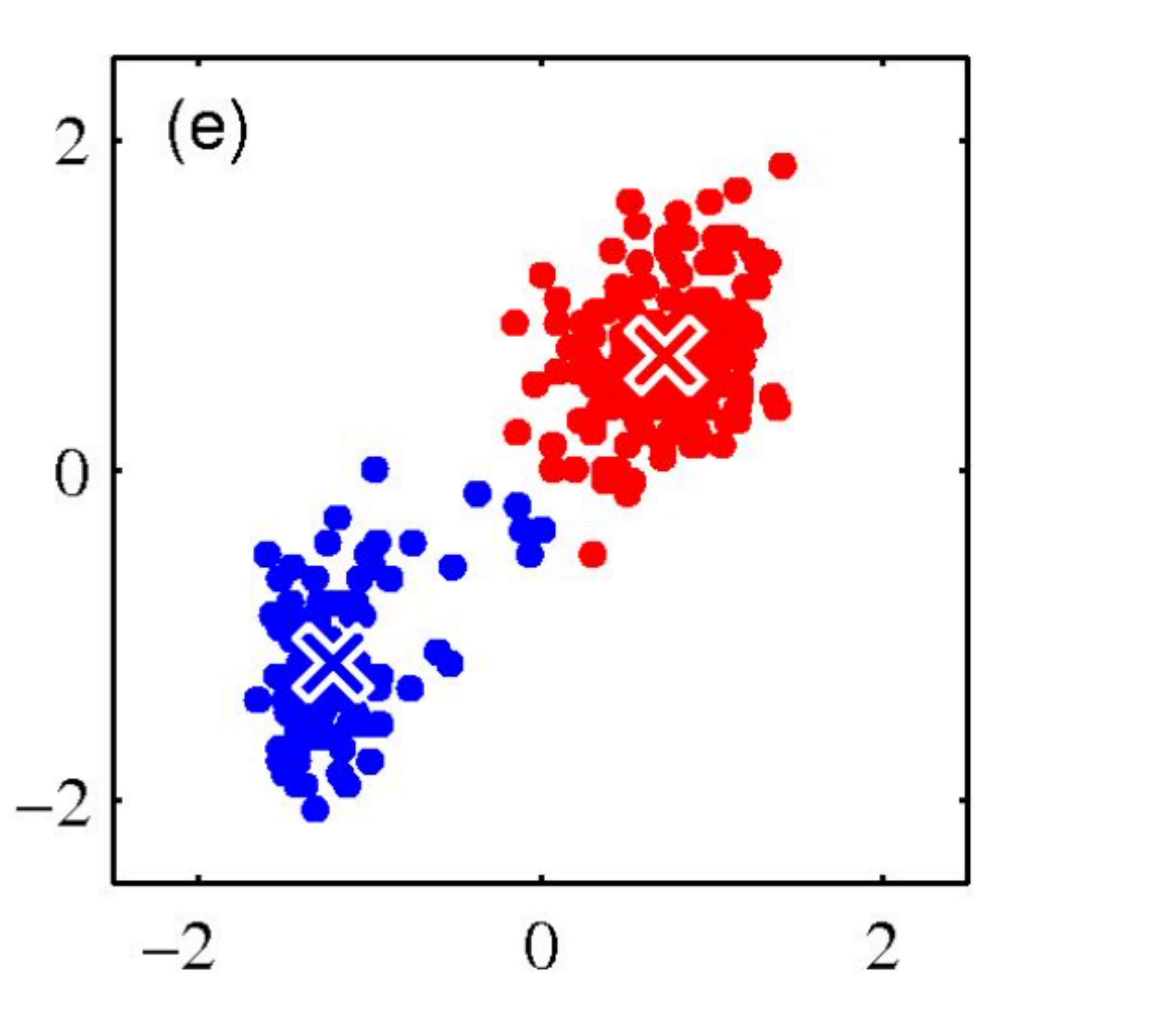
- Change the cluster center to the average of the assigned points

K-means — an example

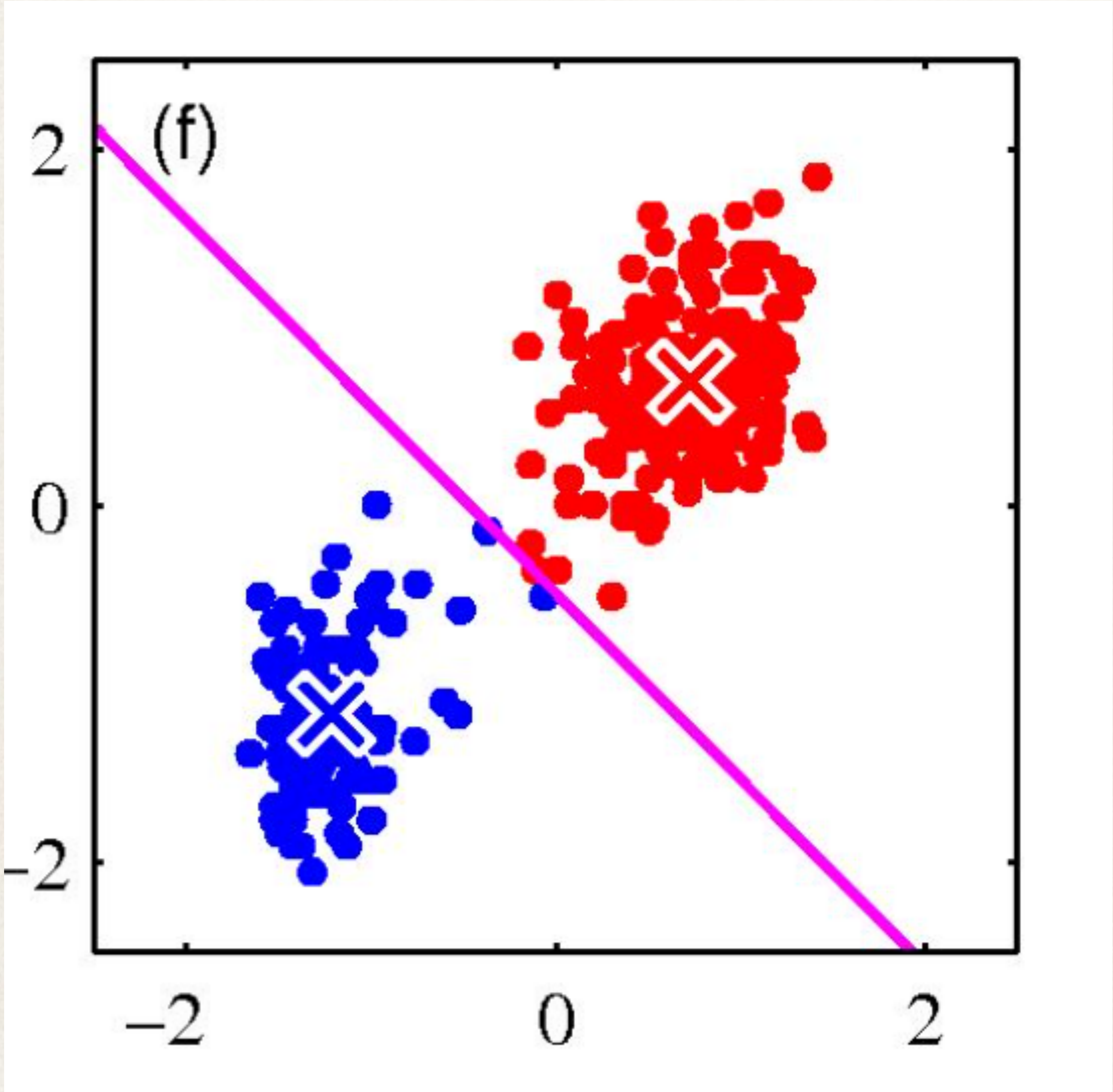


- Repeat until convergence

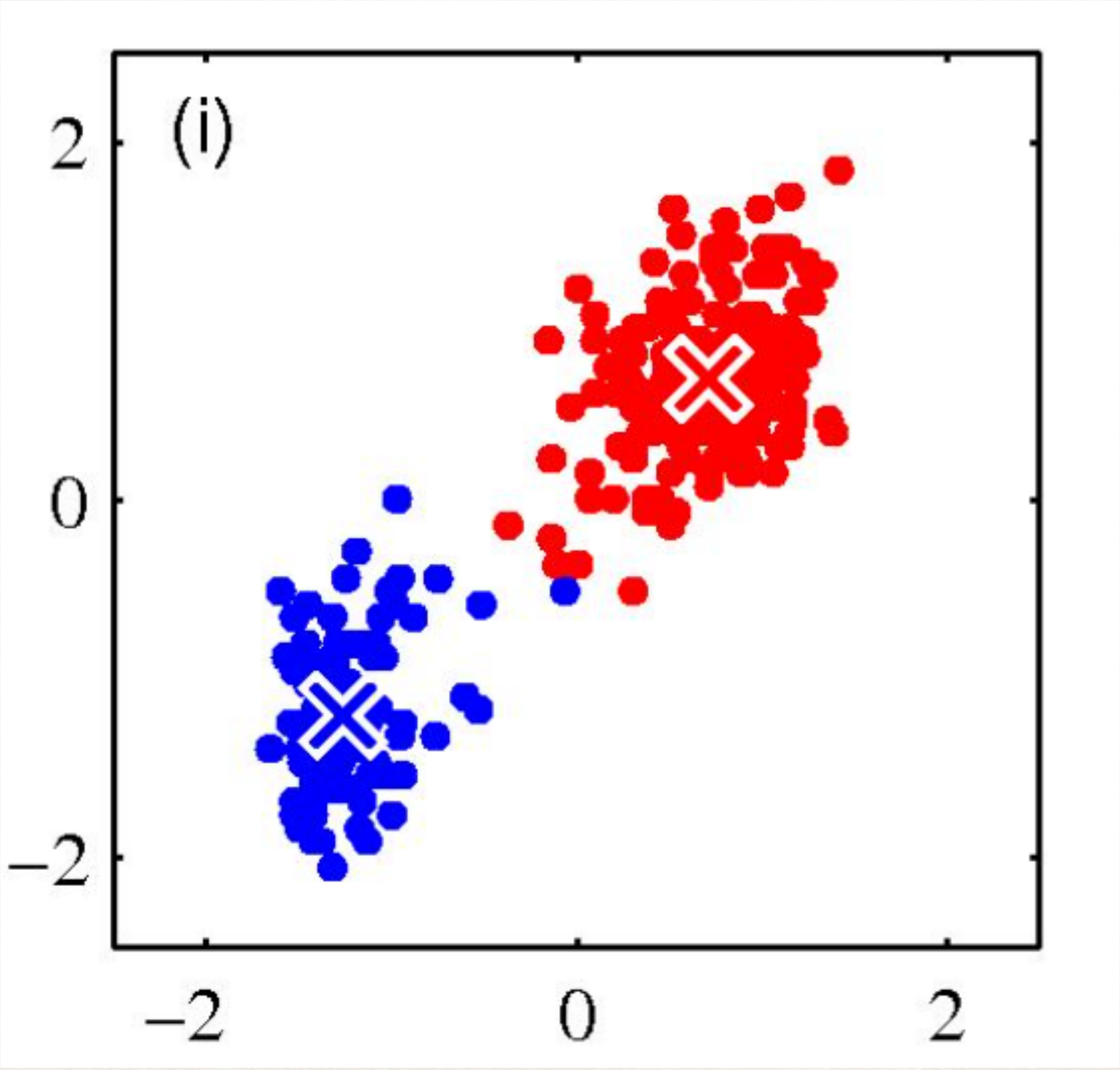
K-means — an example



K-means — an example



K-means — an example



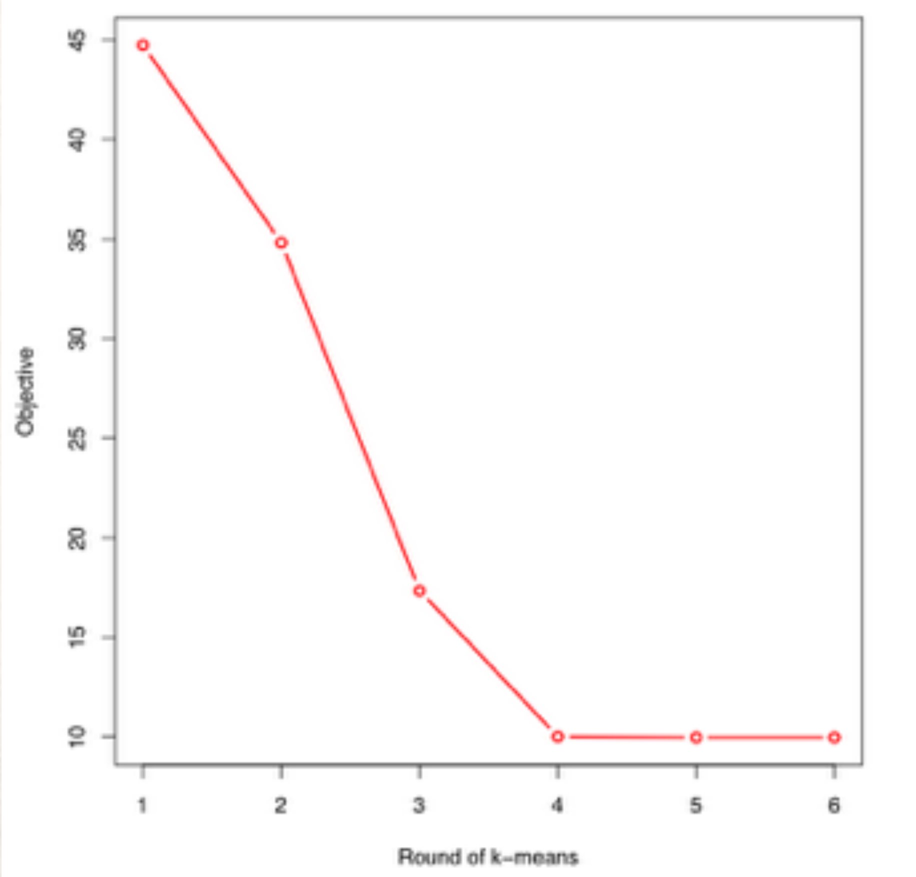
Coordinate descent

$$F(z_{1:N}, \mathbf{m}_{1:k}) = \frac{1}{2} \sum_{n=1}^{N} ||\mathbf{x}_n - \mathbf{m}_{z_n}||^2$$

- Holding the means fixed, assigning each point to its closest mean minimizes F with respect to $z_{1:N}$.
- Holding the assignments fixed, computing the centroids of each cluster minimizes F with respect to $\mathbf{m}_{1:k}$.

When to stop?

- No (or minimum) re-assignment of data points to different clusters, or
- no (or minimum) change of centroids, or
- minimum decrease in the sum of squared error

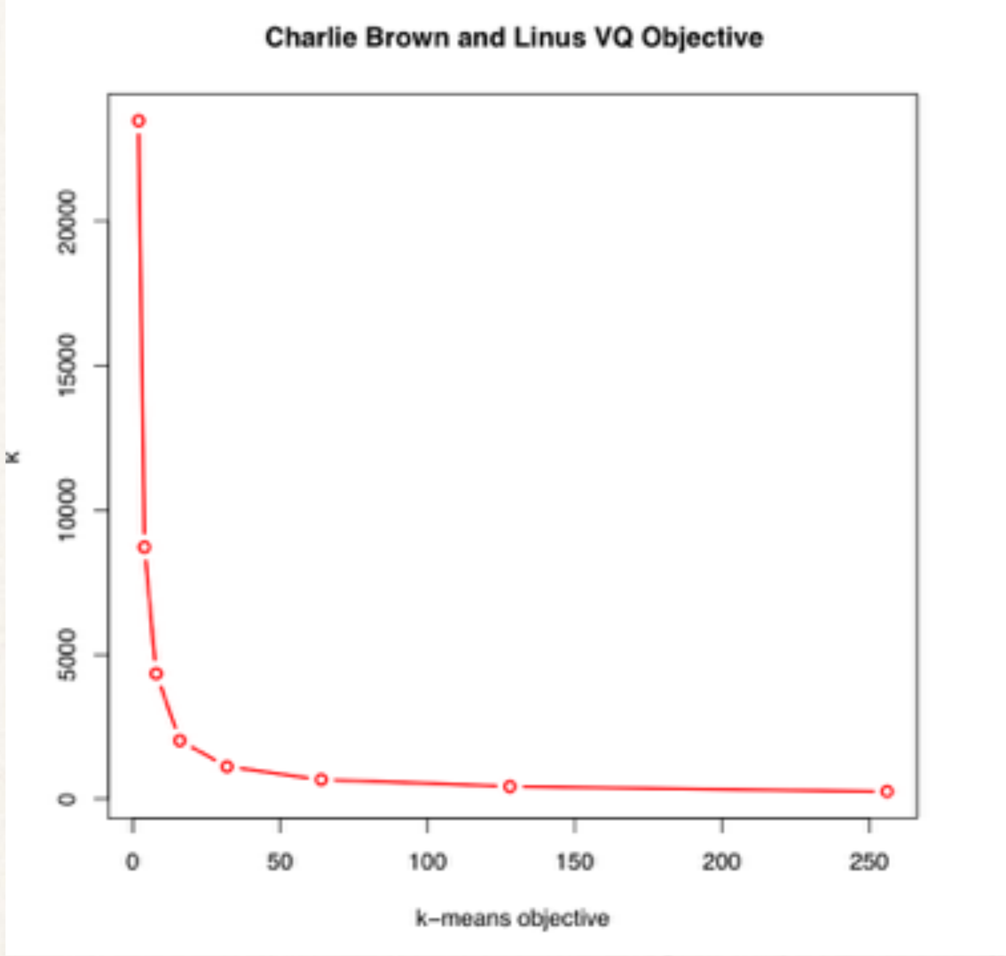


Knobs to turn:

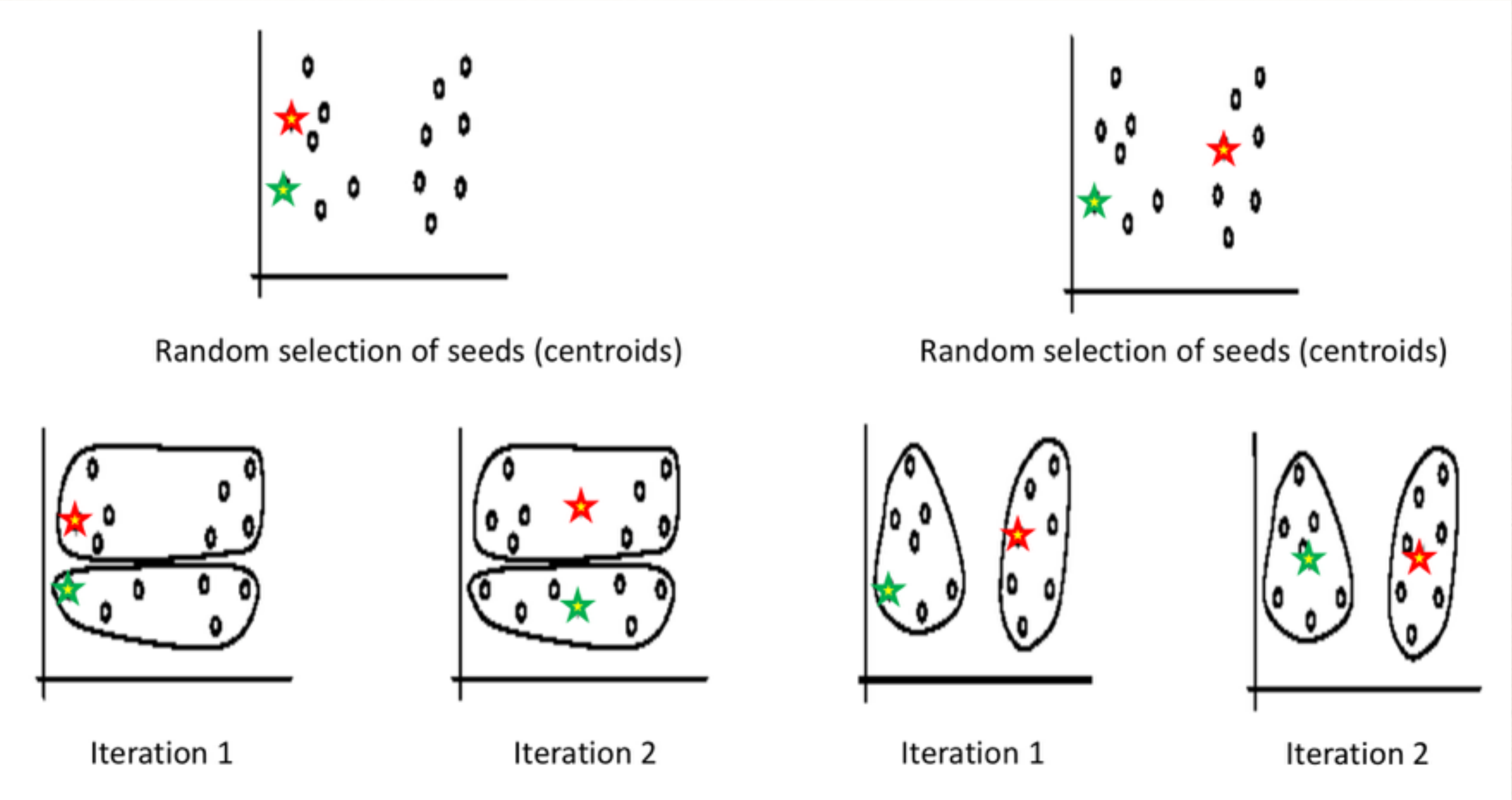
- a distance measure between data points
- number of clusters k
- initial assignment of data to clusters

How to choose K?

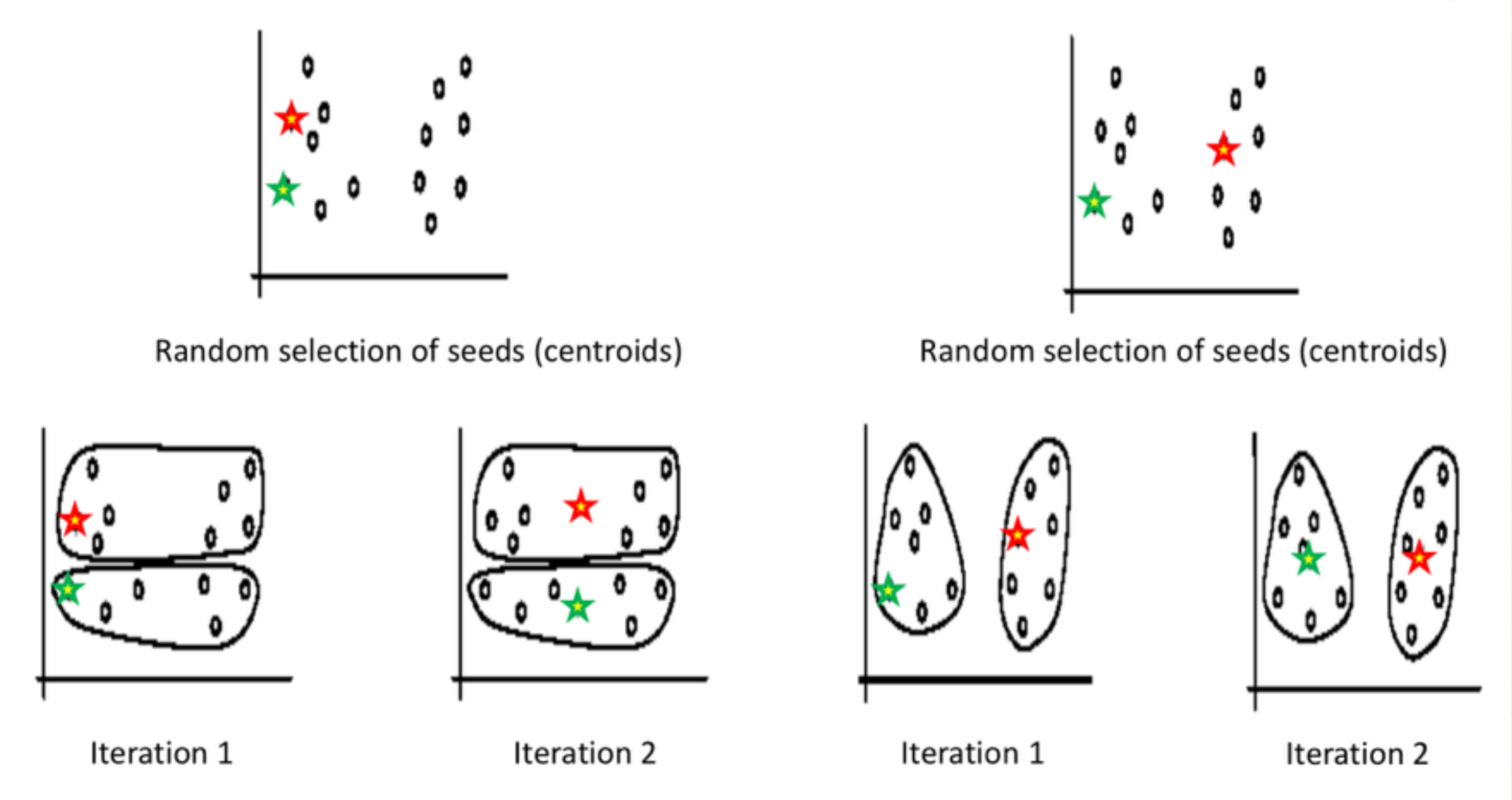
- Base on the required quality in the image compressing case
- Base on limits in real life applications
- Heuristic: a kink in the objective function



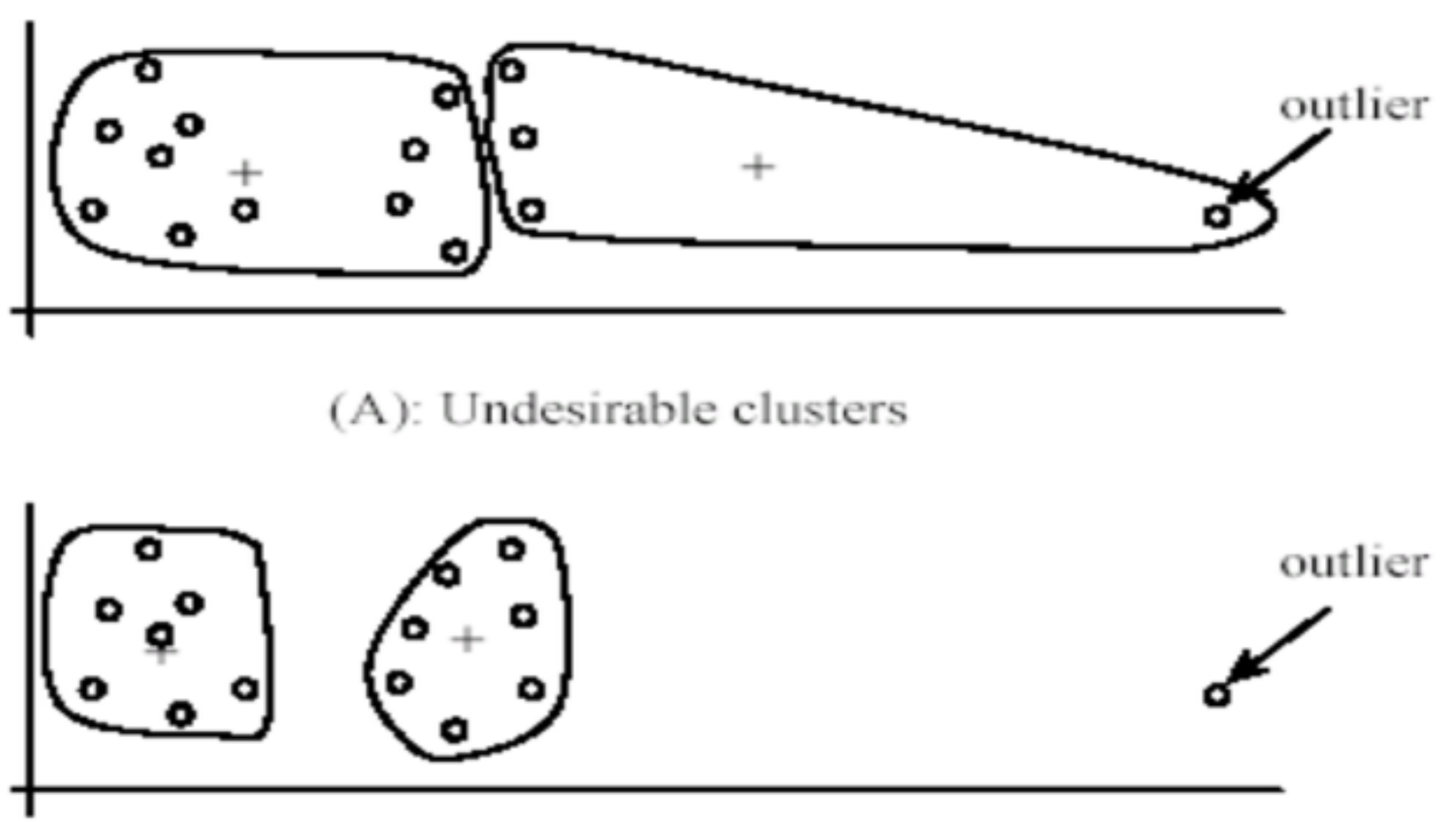
- K-mean algorithm finds the local minimum and is sensitive to initialization



- K-mean algorithm finds the local minimum and is sensitive to initialization
    - Try multiple initiations and choose the best result



- k-mean algorithm is sensitive to outliers

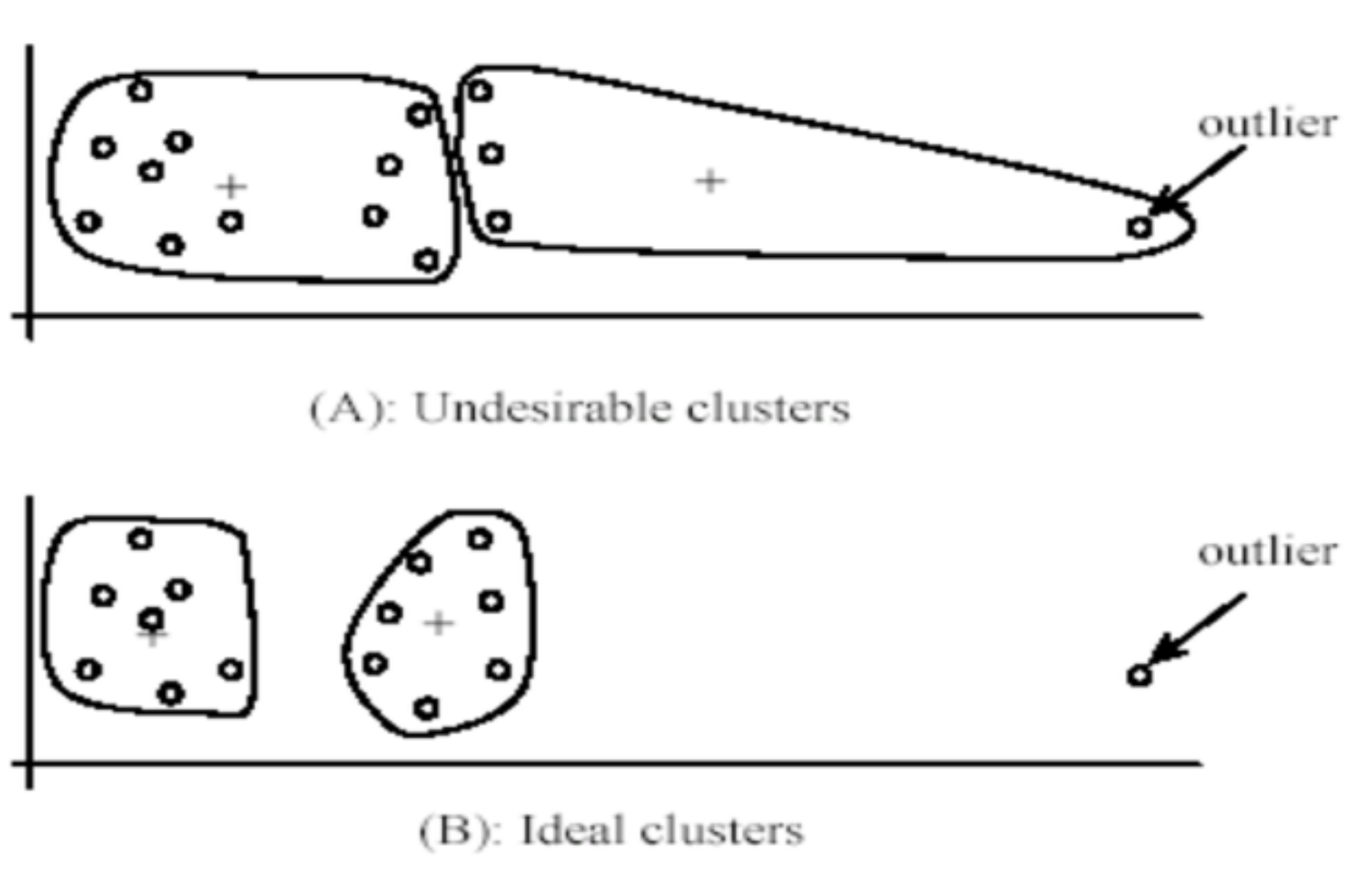


(A): Undesirable clusters

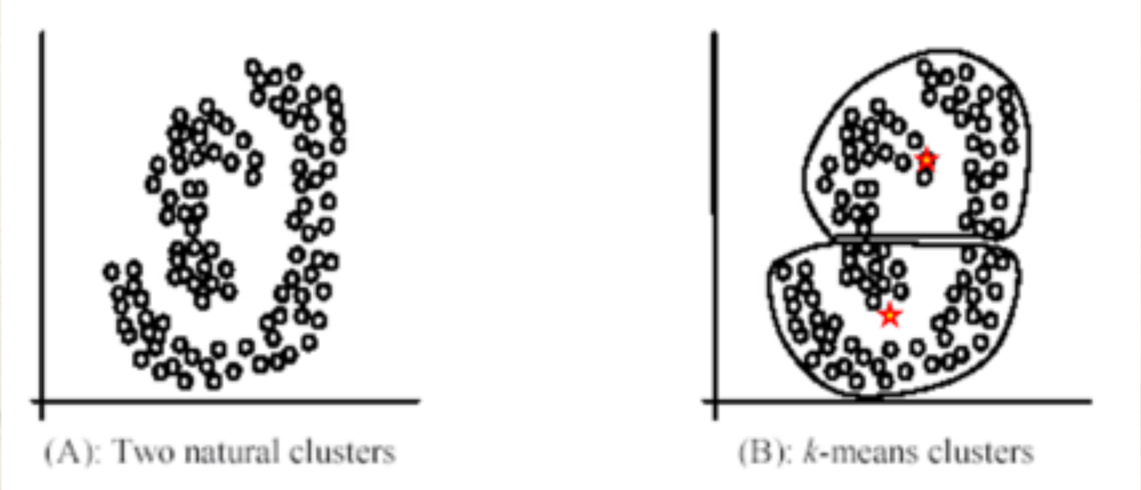
(B): Ideal clusters

- k-mean algorithm is sensitive to outliers

  - Remove outliers, resample, or use medians



- K-means is efficient to compute - O(tkn) (t is the number of iteration)
- K-means requires hard assignment - a point either belongs to a cluster or not. Other methods such as gaussian mixture models and fuzzy k-means allow one to assign a point to multiple clusters with certain probabilities
- K-means works well with round shaped clusters of roughly equal sizes and densities



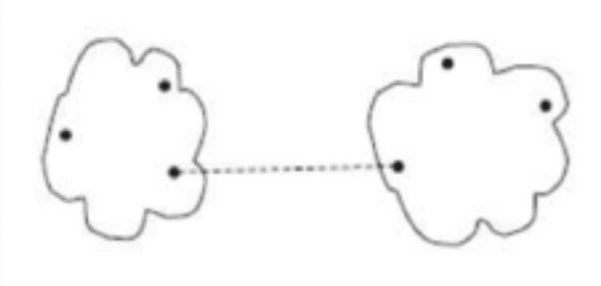
Hierarchical clustering

- Basic idea: Initially, each point is a cluster by itself. Repeatedly combine the two "nearest" clusters into one.
- It only requires a measure of similarity between groups of data points
- It's up to the user to choose a "natural" clustering from the merging sequence

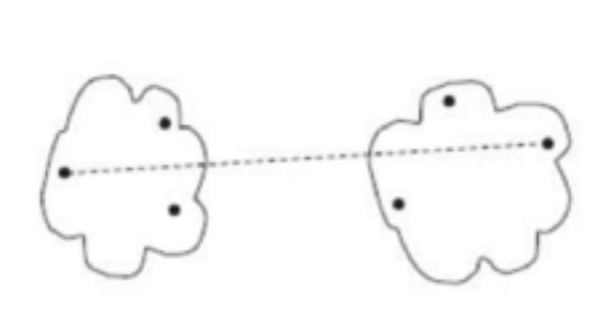
How to define “closest” for clusters?

How to define “closest” for clusters?

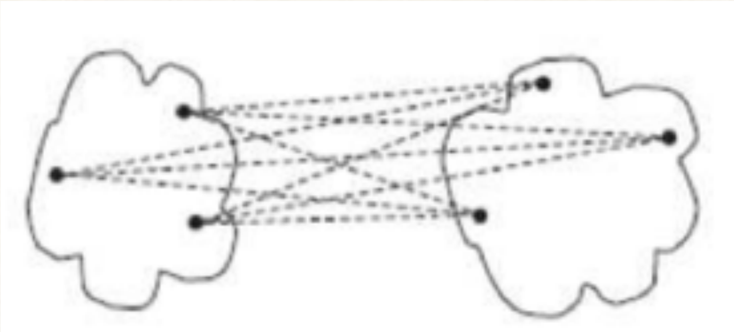
- Closest pair: single - link clustering



- Farthest pair: complete-link clustering



- Average similarity between groups: group average clustering

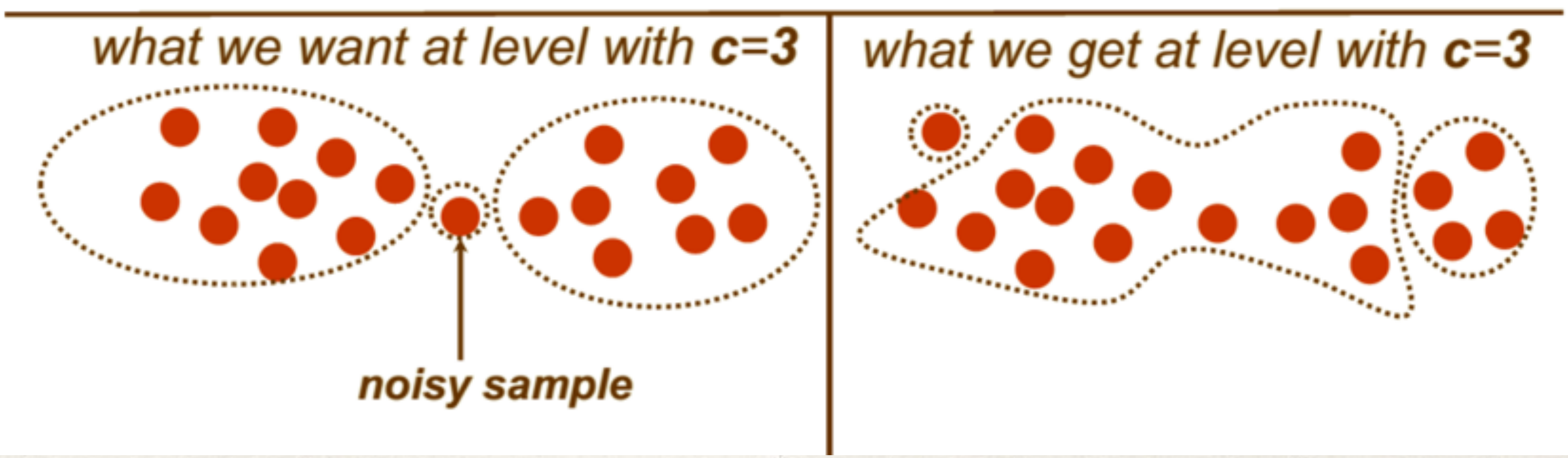


Single linkage

- Agglomerative clustering with minimum distance

$$d_{\min}(D_i, D_j) = \min_{x \in D_i, y \in D_j} \| x - y \|$$

- generates minimum spanning tree
- encourages growth of elongated clusters
- disadvantage: very sensitive to noise

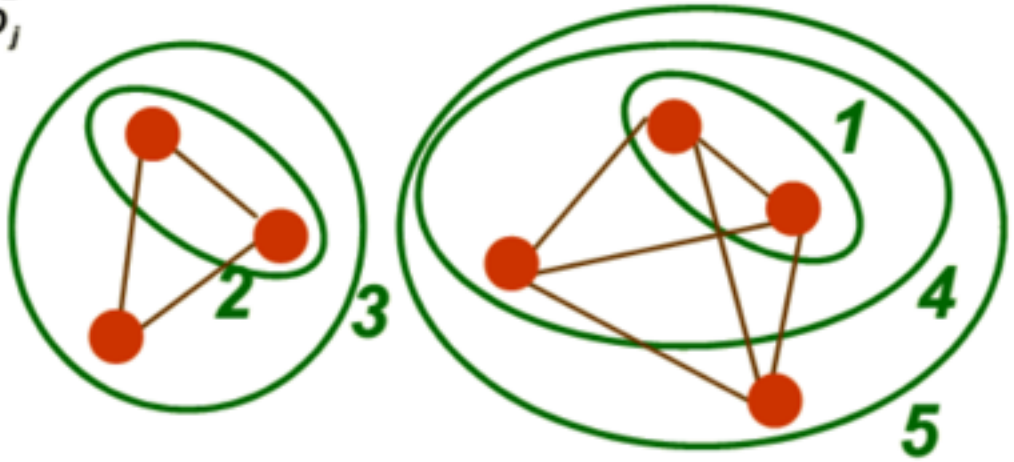


Complete linkage

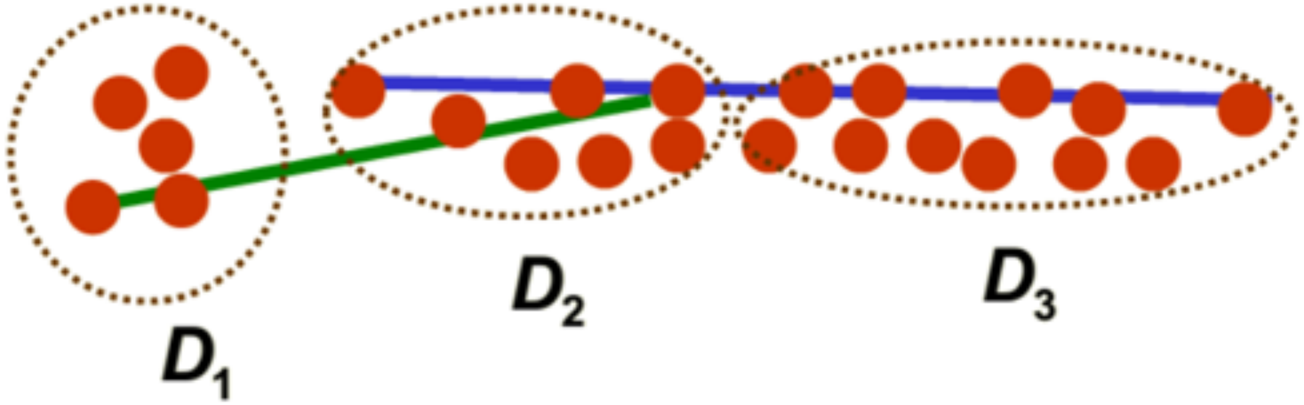
- Agglomerative clustering with maximum distance

$$d_{\max}(D_i, D_j) = \max_{x \in D_i, y \in D_j} \| x - y \|$$

- encourages compact clusters



- Does not work well if elongated clusters present

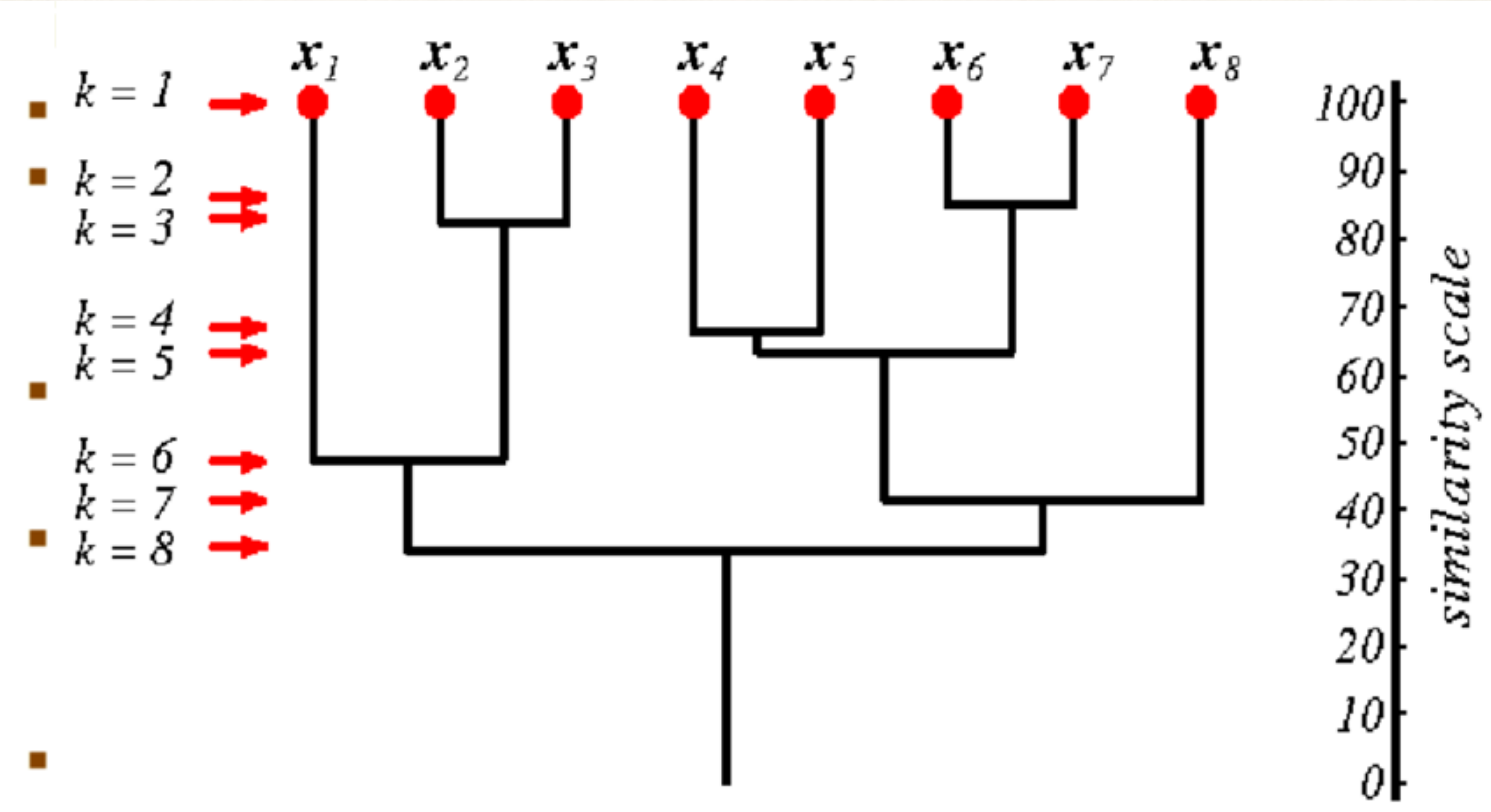


  - $d_{\max}(D_1, D_2) < d_{\max}(D_2, D_3)$
  - thus D_1 and D_2 are merged instead of D_2 and D_3

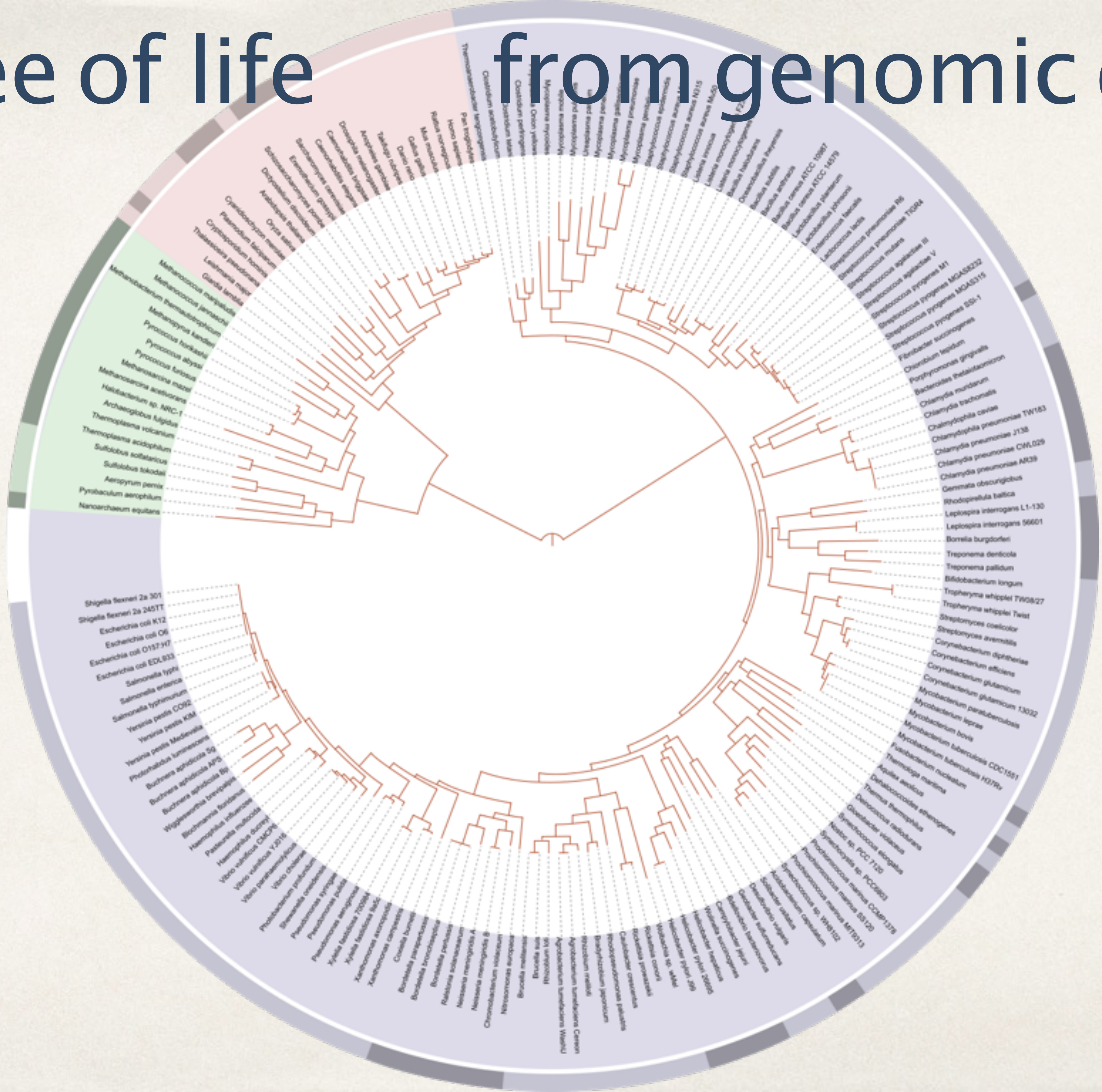
When to stop?

- Stop when we have k clusters
- Stop when cohesion of the cluster resulting from the best merger falls below a threshold
- Stop when there is a sudden jump in the cohesion value

Visualizing with a dendrogram

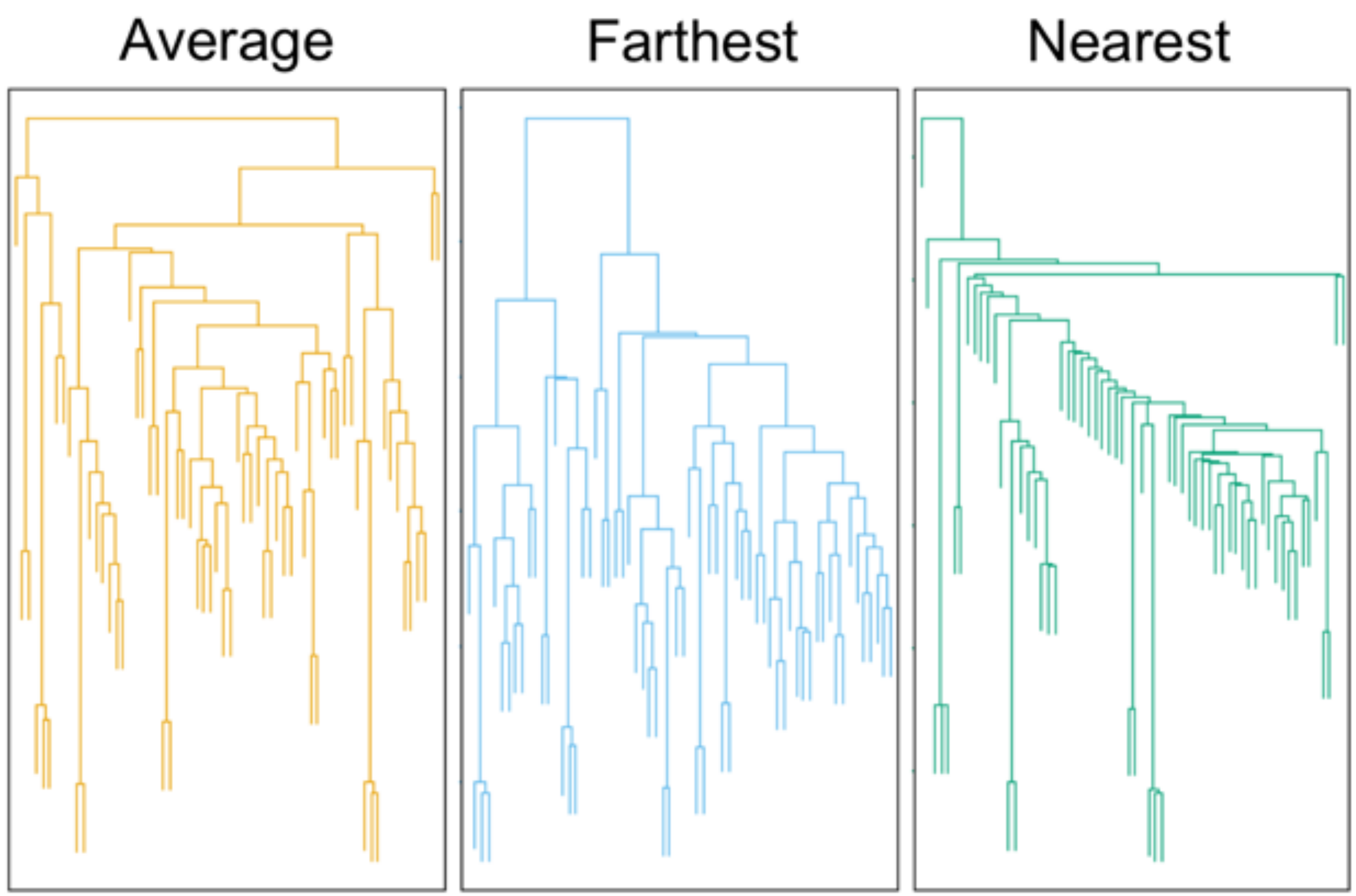


Tree of life from genomic data



Caveats

- Different decisions about group similarities can lead to vastly different dendrograms



Mouse tumor data from [Hastie *et al.*]

- Be cautious! The algorithm imposes a hierarchical structure on the data, even when it's not appropriate
- It's quite expensive to compute pairwise distances between each pair of cluster (naive implementation $O(N^3)$, with a priority queue $O(N^2 \log N)$)

Demo time!

Useful materials

- Distance metrics:

- http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Distance_Metrics_Overview.htm

- http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html

- https://www.cs.utah.edu/~jeffp/teaching/cs5955/L7-Distances.pdf