Clustering: *K*-Means

DCS310

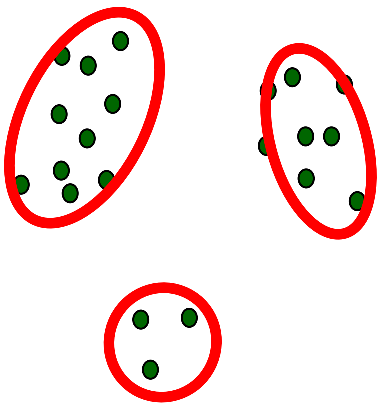
Sun Yat-sen University

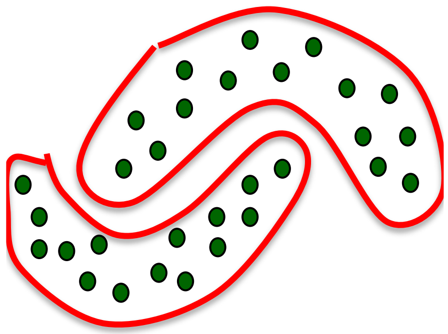
Outline

- Introduction to Clustering
- *K*-Means

What is Clustering?

- Given a set of data instances $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, clustering is about how to partition them into different groups



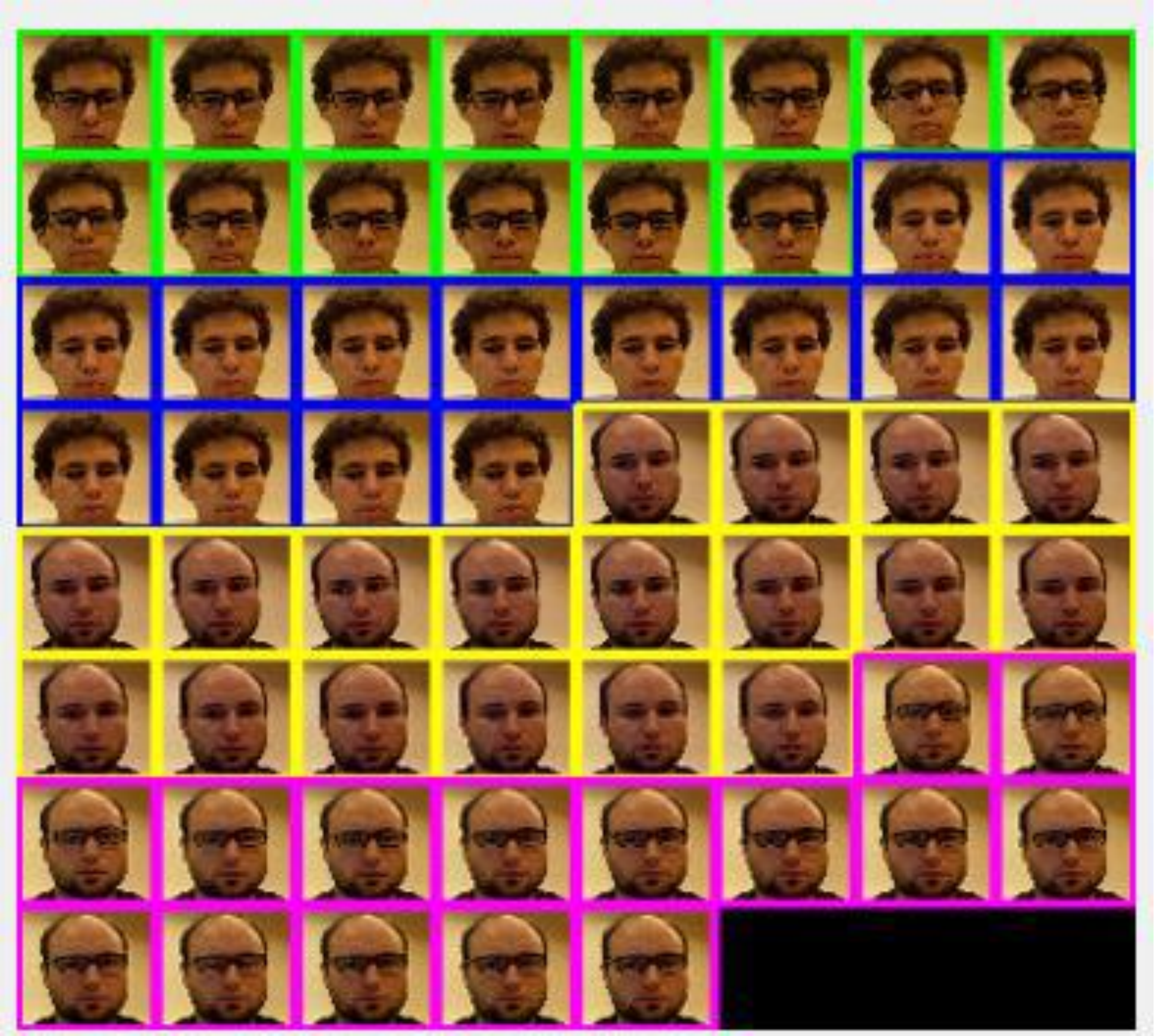


- The Objective
  - High similarity for intra-class instances
  - Low similarity for inter-class instances

Similarity Criteria Matters

- Different similarity criteria could lead to different results

Similar or not?

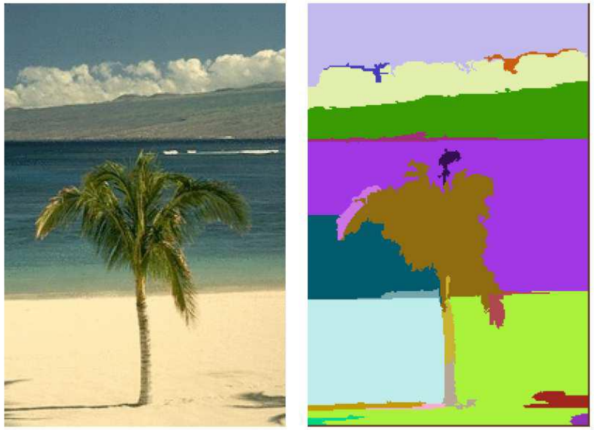


Criteria 1: Identity   Criteria 2: Glasses

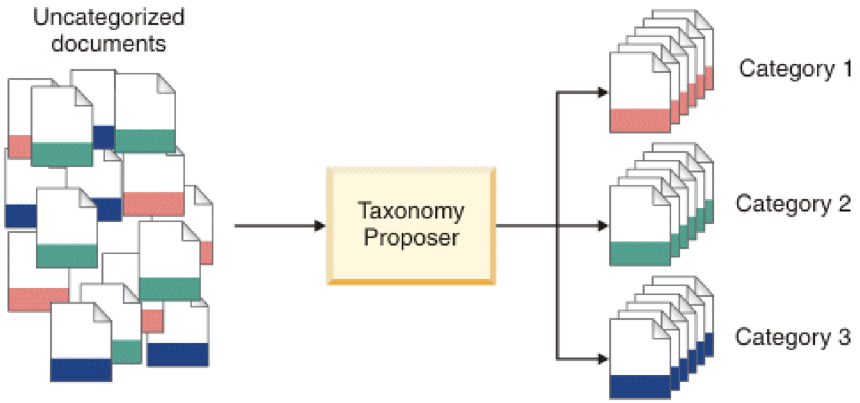
Real-world Applications

- Image grouping

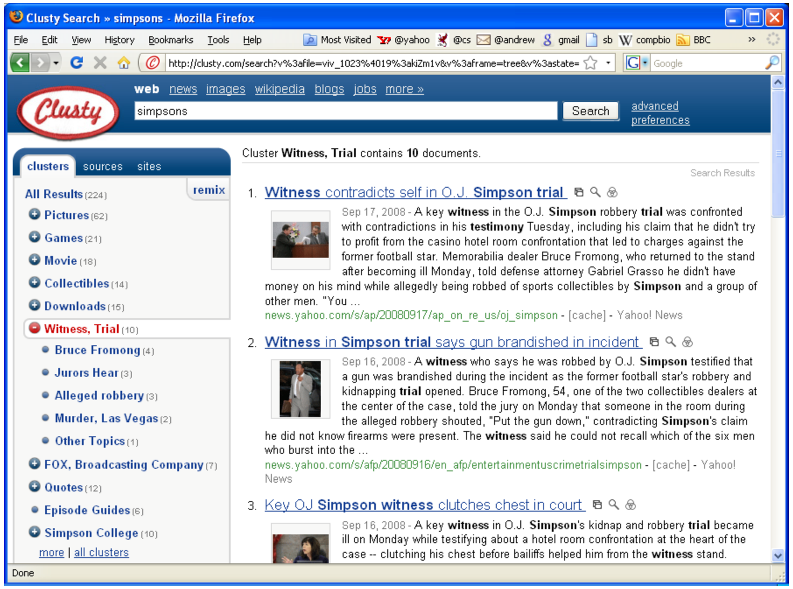
- Image segmentation



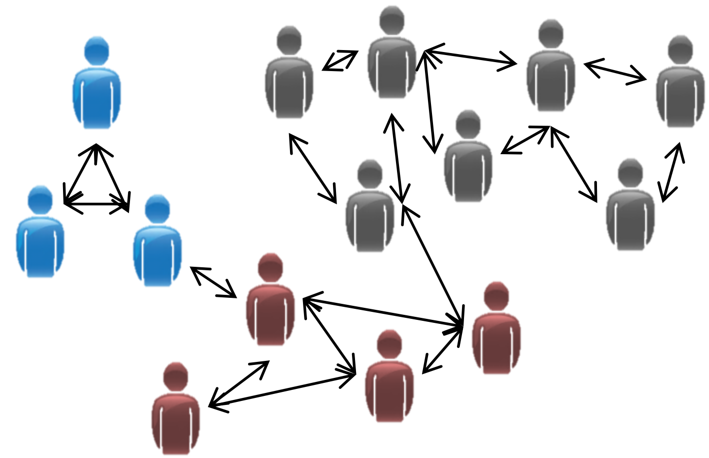
- Automatically group semantic-similar documents together



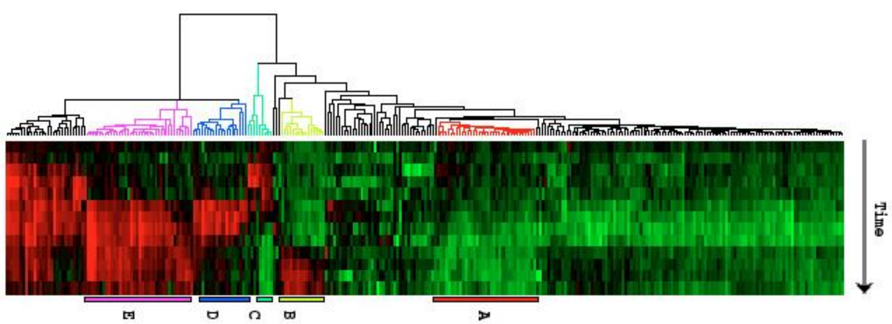
- Web-search result clustering



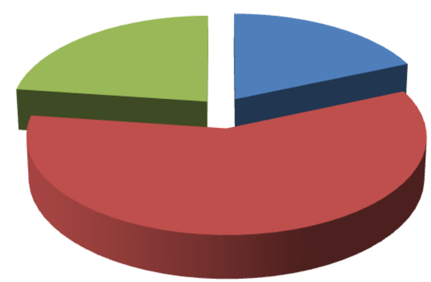
- Social network analysis



- Gene expression data clustering



- Market segmentation

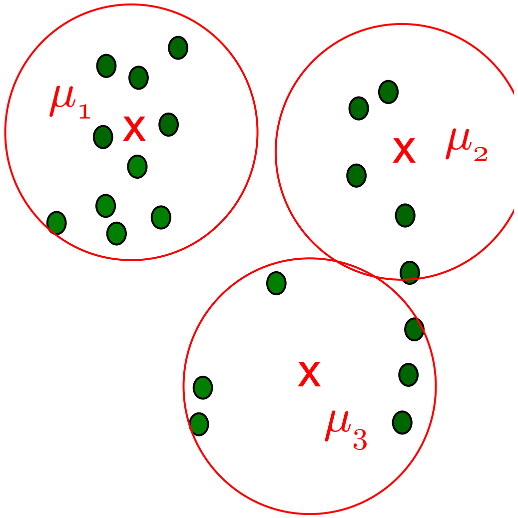


Outline

- Introduction to Clustering
- *K*-Means

K-Means Algorithm

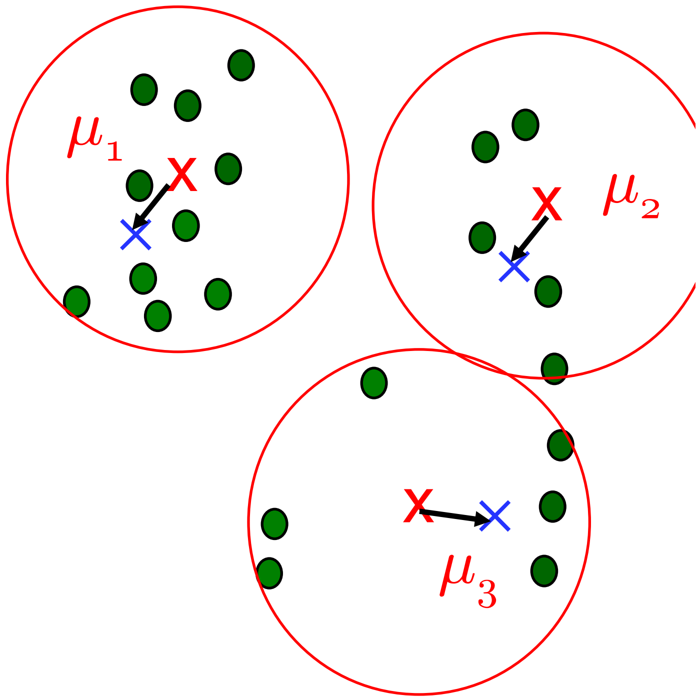
- Randomly initialize K centers $\boldsymbol{\mu}_k$ for $k = 1, \cdots, K$, and then evaluate the distance between data $\boldsymbol{x}^{(n)}$ and the centers $\boldsymbol{\mu}_k$



- Data $\boldsymbol{x}^{(n)}$ is assigned to the cluster k with the smallest distance

$$r_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_j \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_j \right\|^2 \\ 0, & \text{otherwise} \end{cases}$$

- Updating the centers with the average of data in every cluster



$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{n=1}^{N} r_{nk}\, \boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

- Repeating the assignment and center updating process above

Convergence Guarantee

- The total distance between the data and their corresponding centers

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| \boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k \right\|^2$$

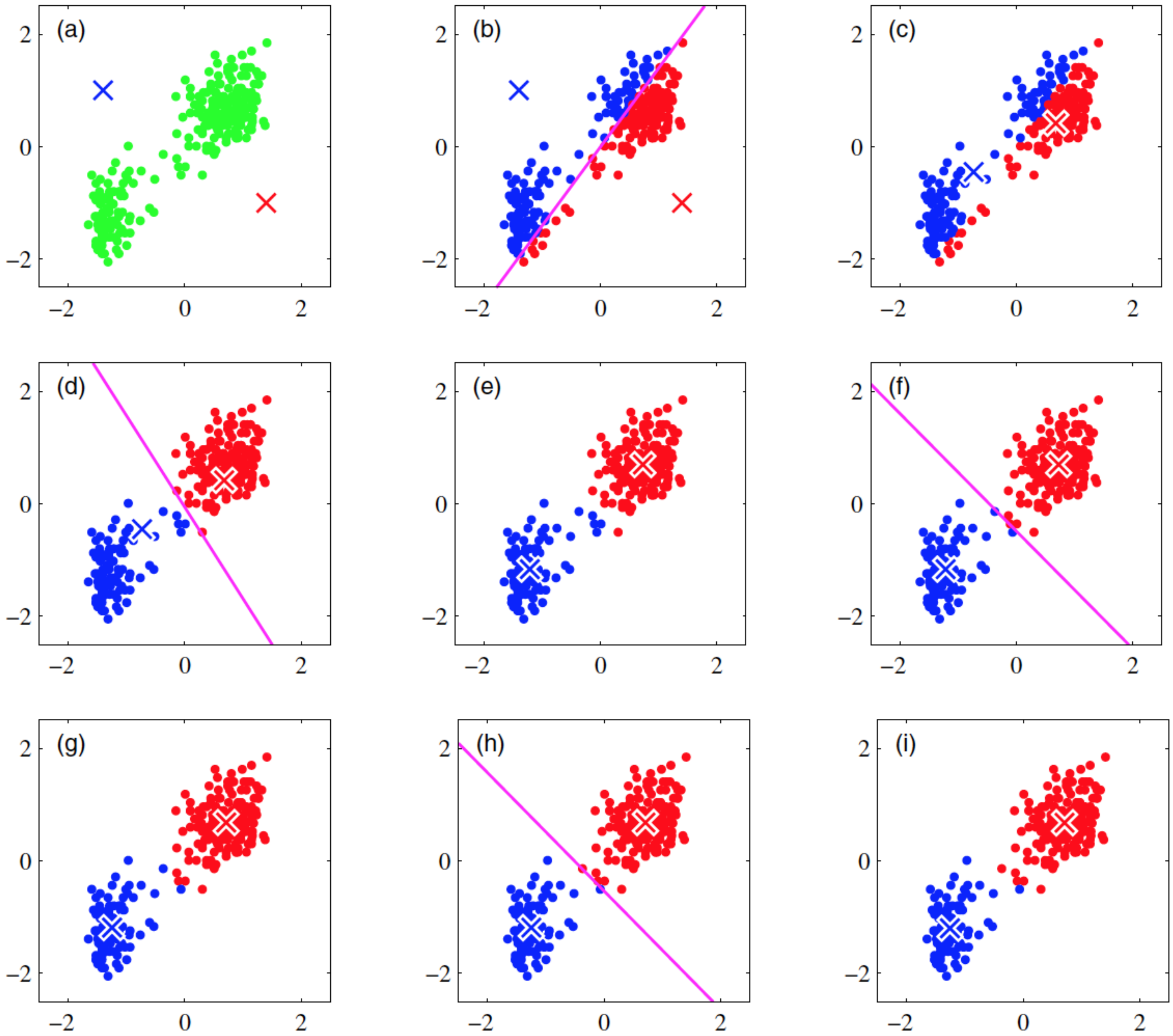
- It can be shown that the *k*-means algorithm can be recovered from optimizing $\boldsymbol{r}_n$ and $\boldsymbol{\mu}_k$ of the below problem *alternatively*

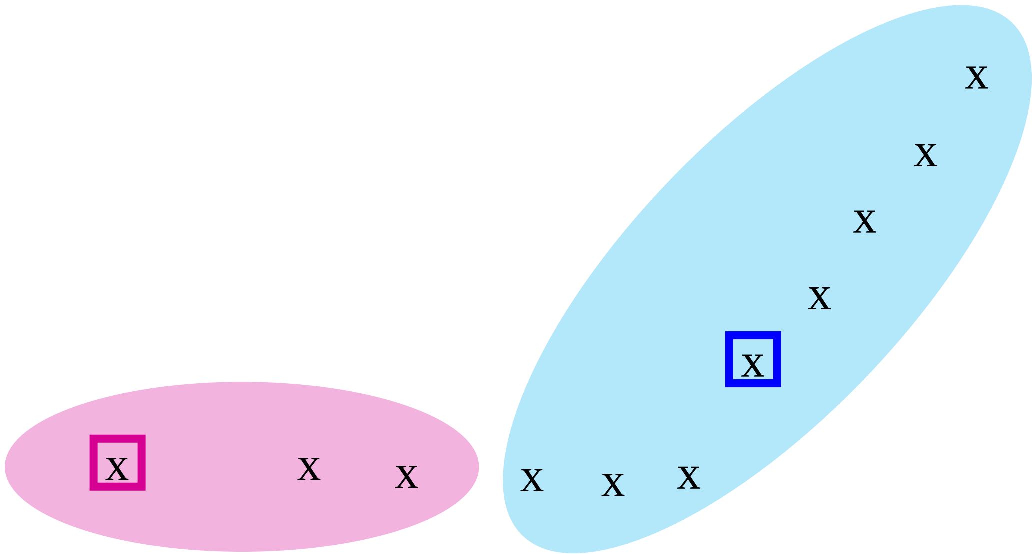
$$\min_{\boldsymbol{r}_{n,} \boldsymbol{\mu}_k} \ J$$

$$s.t. \ \boldsymbol{r}_n \in \text{onehot vector} \quad \forall \, n \,\&\, k$$

where $\boldsymbol{r}_n \triangleq [r_{n1}, r_{n2}, \cdots, r_{nK}]$ is required to be a one-hot vector

- The total distance J decreases *monotonically*, thus the *K*-means algorithm is guaranteed to converge

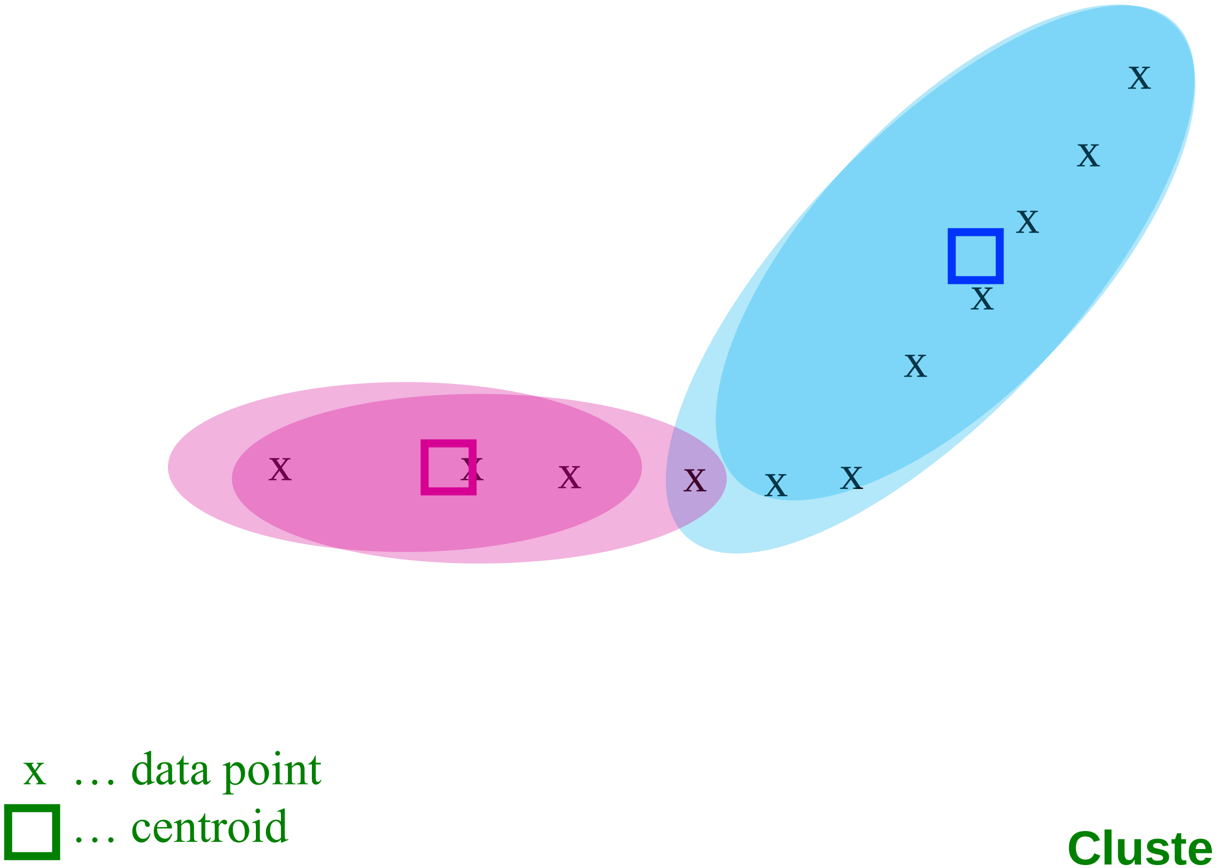

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)



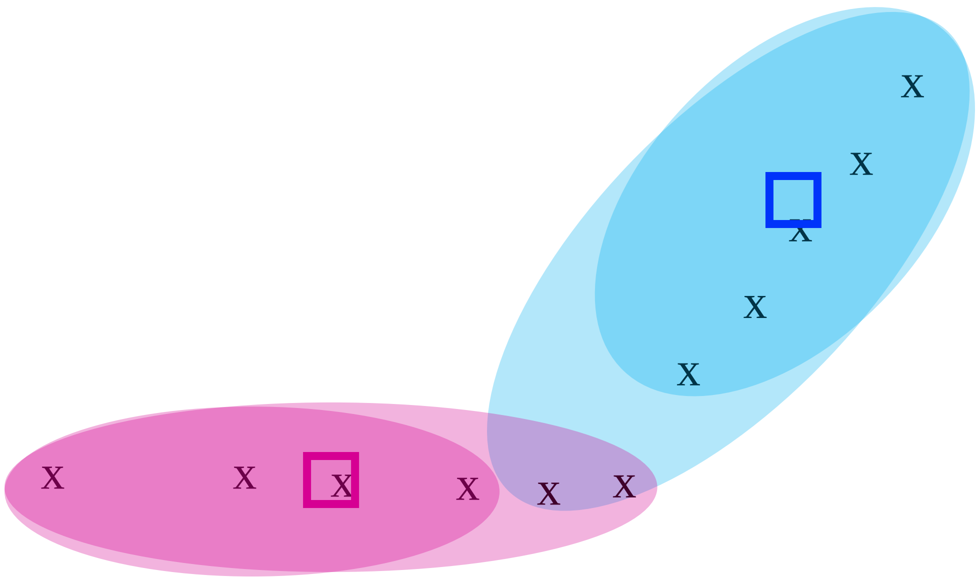
x … data point

… centroid

Clusters after round 1



Clusters after round 2



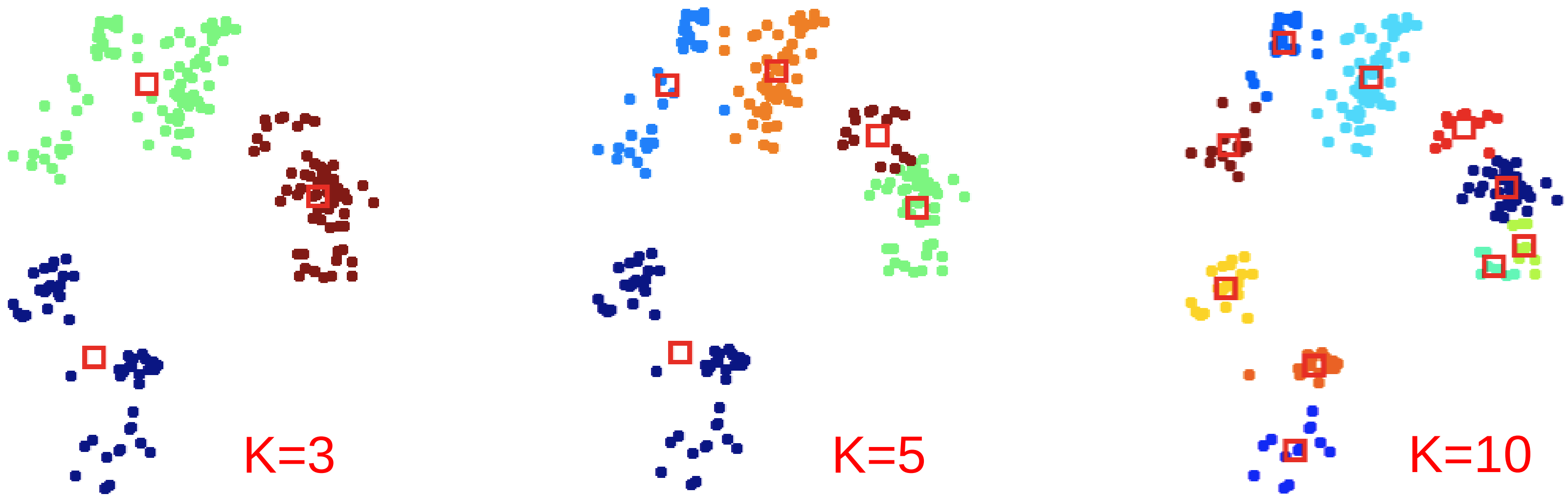
x … data point

… centroid

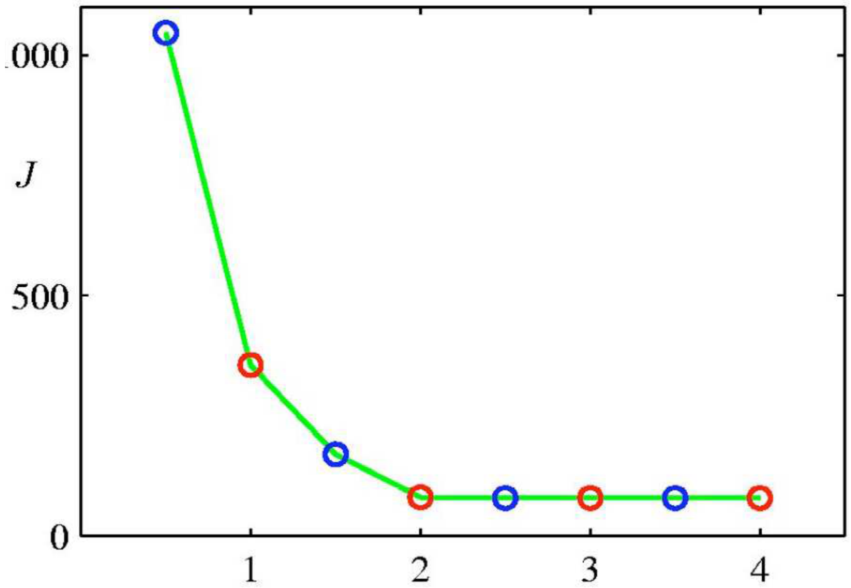
Clusters at the end

Issues: Number of Clusters

- How to set the value for K is extremely important to the final clustering result



- The distance J decreases as the number of clusters K increases. Thus, we cannot determine K by seeking the minimum of J

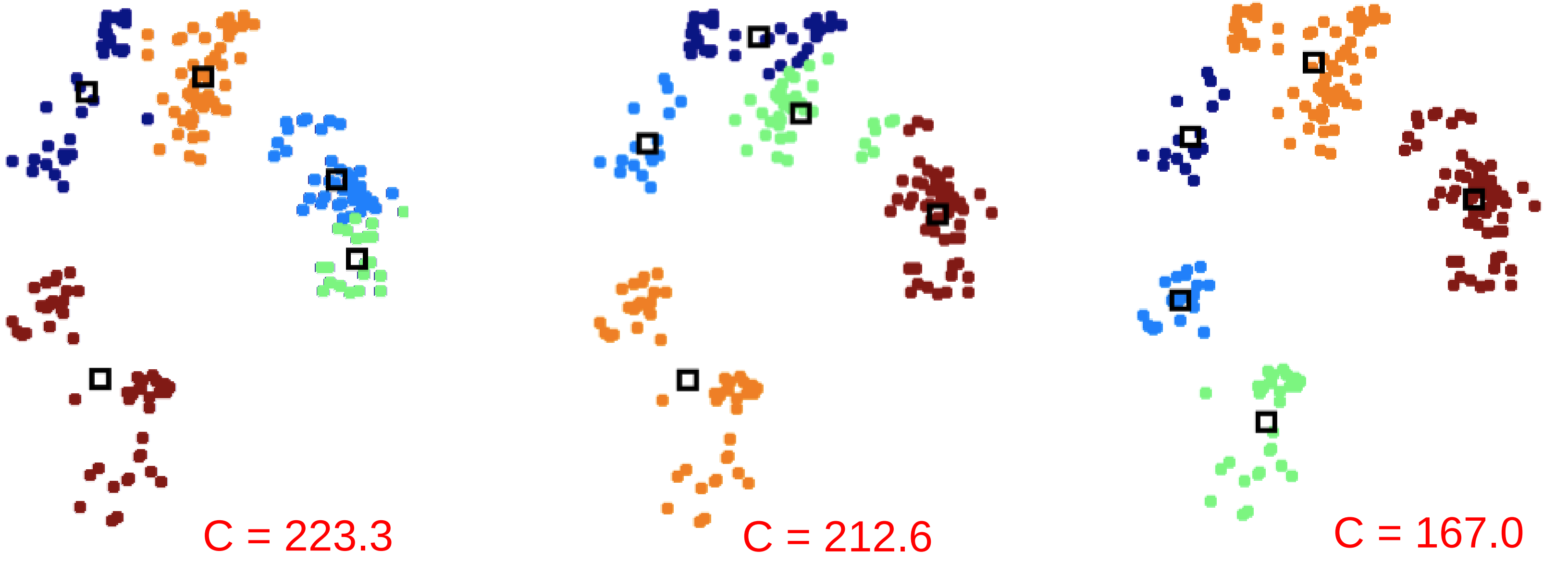


1) One possible method is to choose the elbow point (here $K = 2$)

2) Another possible method is to determine the best K value according to the performance of downstream applications

Issues: Initialization

- The performance of K-means also highly depends on the positions of initial centers



1) Random method

- Choose the data instance randomly
- Issue: may choose nearby instances

2) Distance-based method

- Start with one random data instance
- Choose the point that is farthest to the existing centers
- Issue: may choose outliers

3) Random + Distance method

- Start with one random data instance
- Choose the next center randomly from the remaining instances that is far away from existing centers

Issues: Hard Assignment

- Hard assignment

  A point either belongs to a cluster or not at all, that is, r_{nk} is equal to either 1 or 0

- Soft *K*-means

  Instead of assigning $\boldsymbol{x}^{(n)}$ to a cluster k in a hard fashion, soft *K*-means assigns it in a soft way

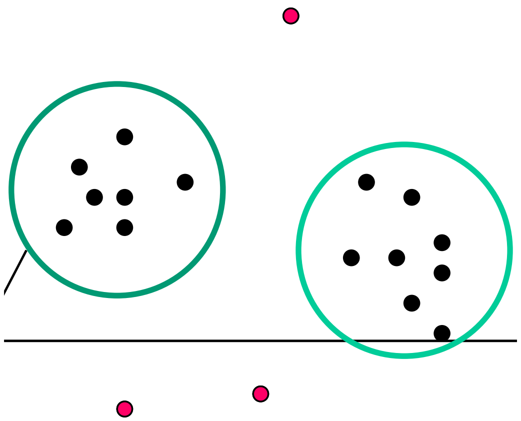
$$r_{nk} = \frac{e^{-\beta\left\|\boldsymbol{x}^{(n)}-\boldsymbol{\mu}_k\right\|^2}}{\sum_{i=1}^{K} e^{-\beta\left\|\boldsymbol{x}^{(n)}-\boldsymbol{\mu}_i\right\|^2}}$$

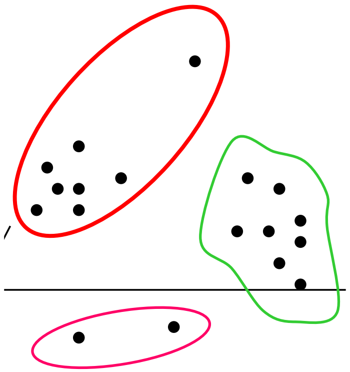
$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{n=1}^{N} r_{nk}\,\boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

  r_{nk} can be interpreted as the probability that data $\boldsymbol{x}^{(n)}$ belongs to the cluster k

Issues: Others

- Sensitive to outliers





- Round shape

  The Euclidean distance determines the boundary is globular. When different clusters have irregular shapes, the performance is poor