Clustering: *K*-Means

Qinliang Su （苏勤亮）

Sun Yat-sen University

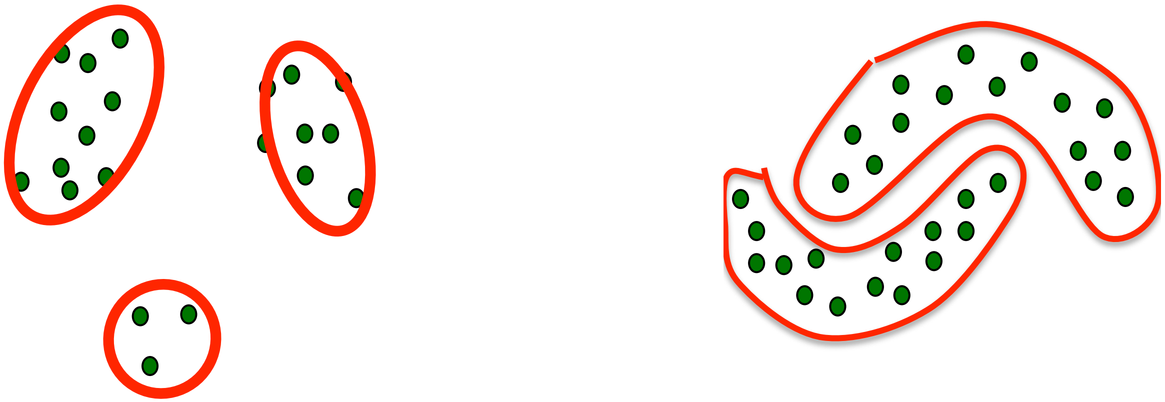
suqliang@mail.sysu.edu.cn

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 group them into different clusters

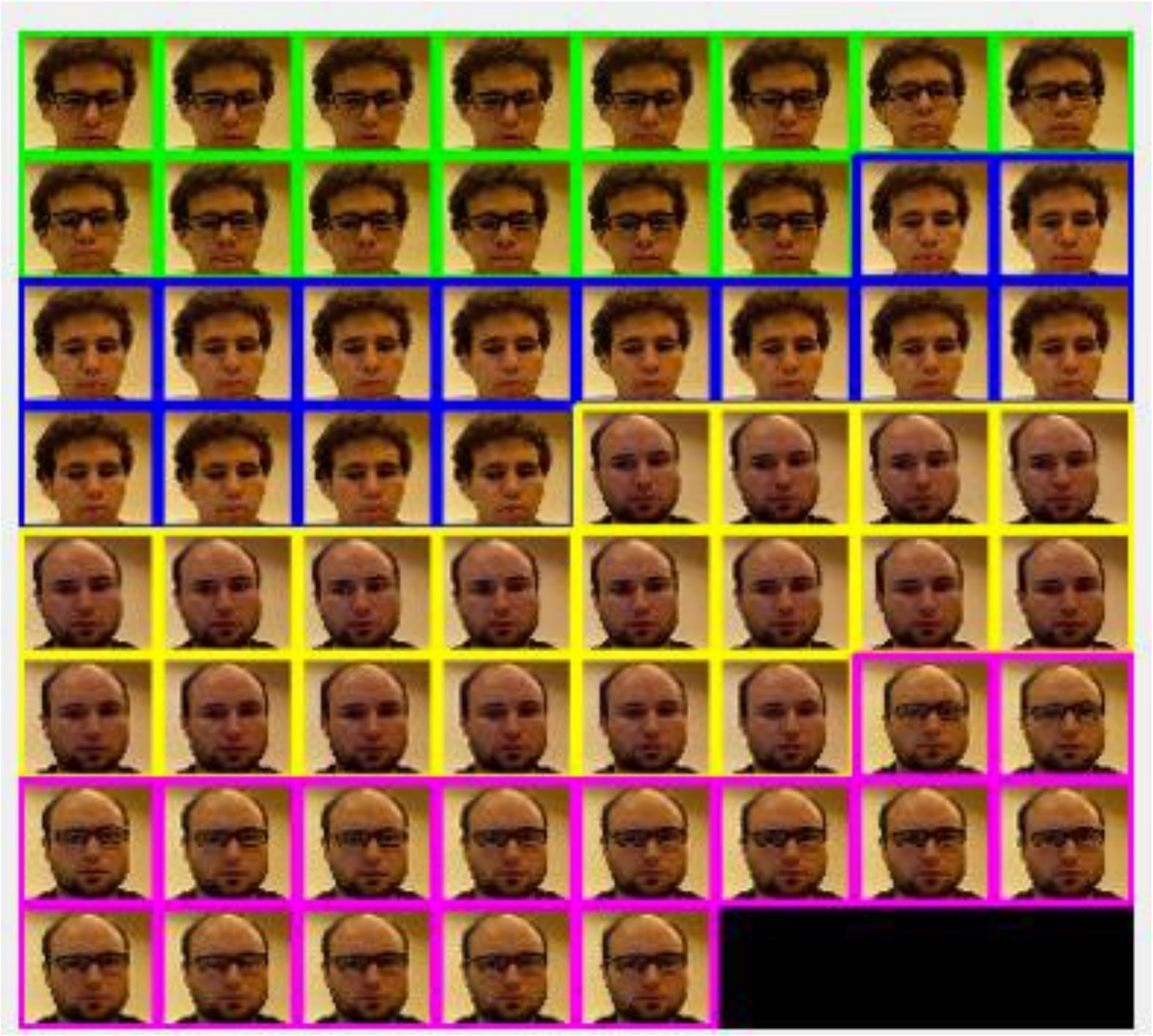


- The objective
  - High similarity for intra-class instances
  - Low similarity for inter-class instances

Similarity Criteria Matters

- Different similarity criteria could lead to different results

Similar or not?

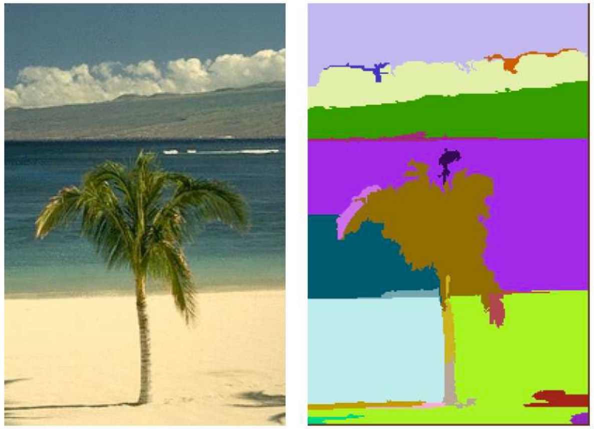


Criteria 1: Identity Criteria 2: Glasses

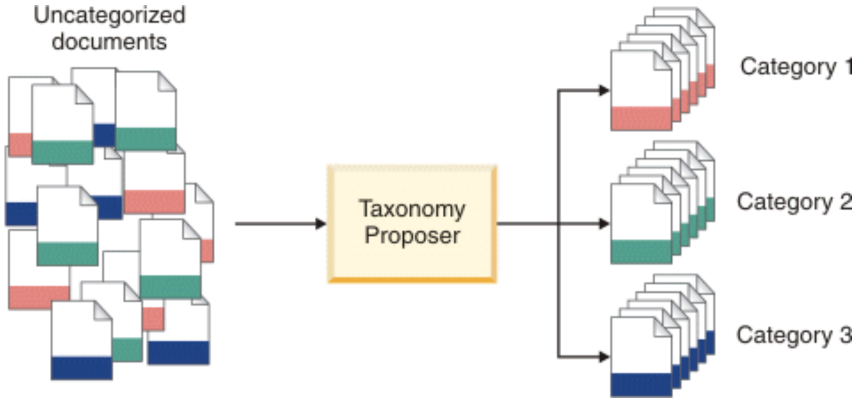
Real-world Applications

- Image grouping

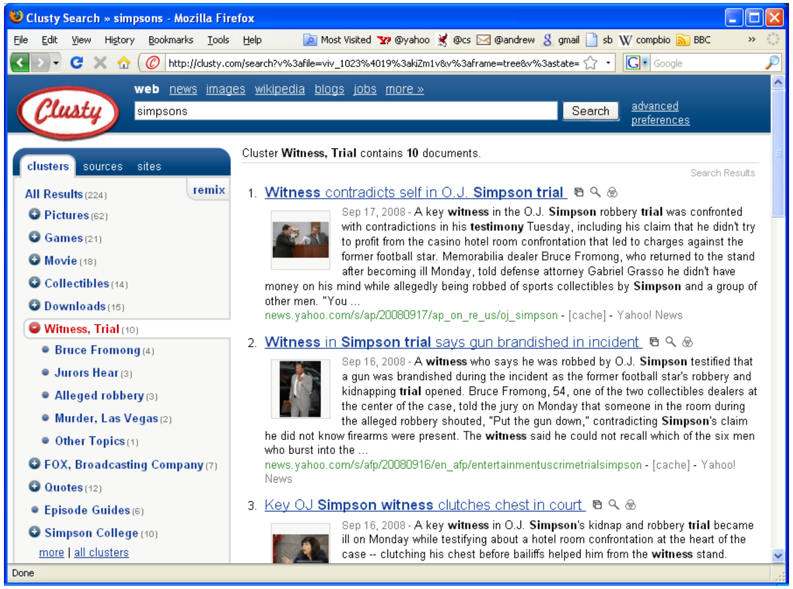
- Image segmentation



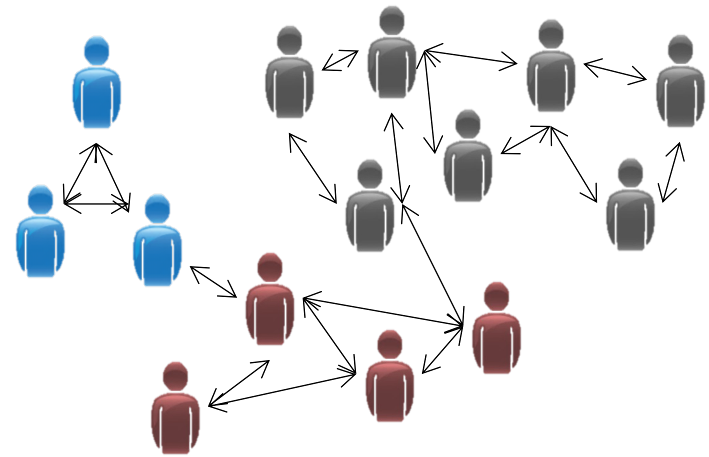
- Automatically group semantic-similar documents together



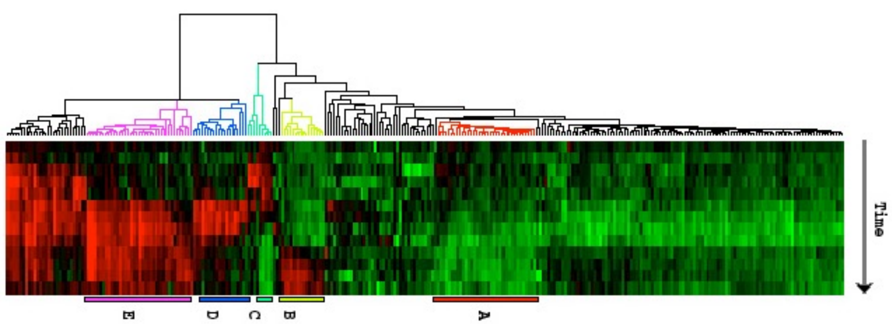
- Web-search result clustering



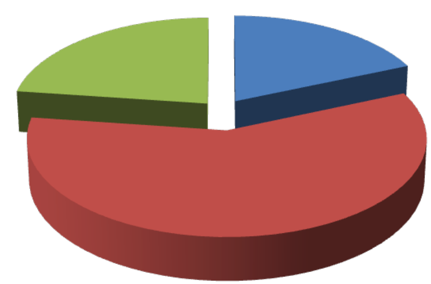
- Social network analysis



- Gene expression data clustering



- Market segmentation

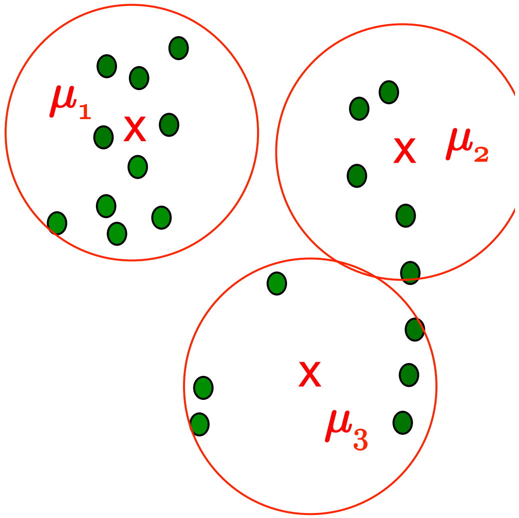


Outline

- Introduction to Clustering
- K-Means

K-Means Algorithm

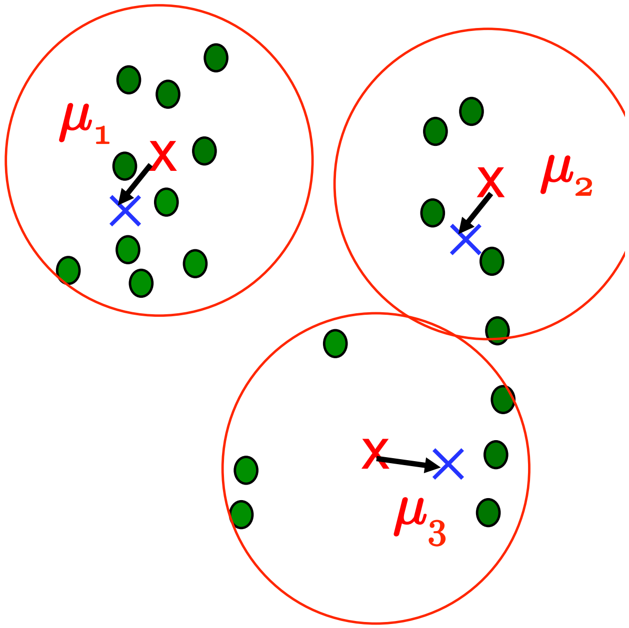
- Designate K centers $\boldsymbol{\mu}_k$ for $k = 1, \cdots, K$, and then evaluate the distance between every data $\boldsymbol{x}^{(n)}$ and all centers $\boldsymbol{\mu}_k$



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

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

- Updating the centers using the mean of samples within a cluster



Two questions

1) What does the algorithm really do?
2) Is the algorithm guaranteed to converge?

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

- Repeating the assignment and center updating steps above

Convergence Guarantee

- Defining an objective, which is the summation of all distances between a data instance and its corresponding center

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

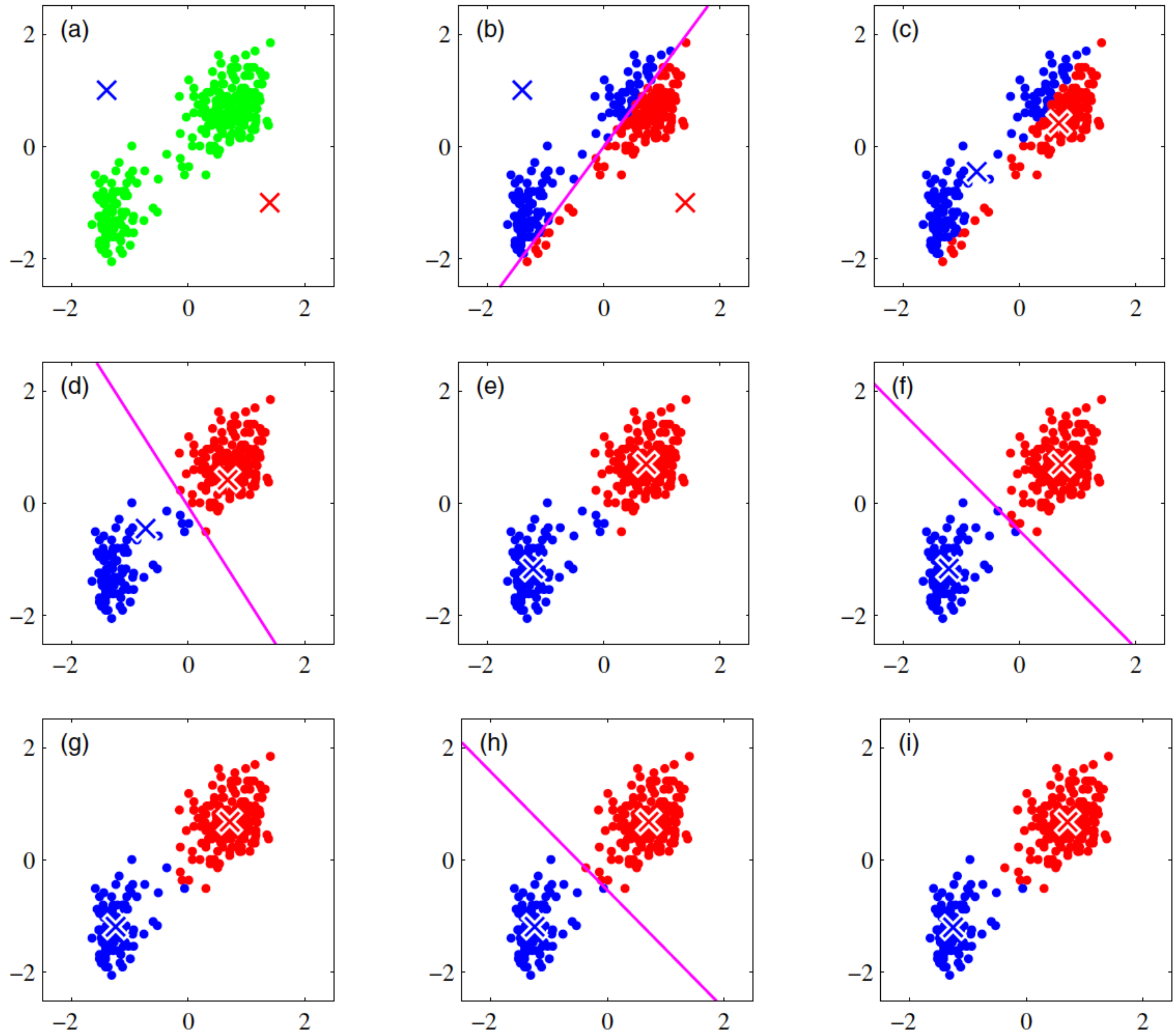
- *K*-means can be recovered from the following optimization by updating $\boldsymbol{r}_n$ and $\boldsymbol{\mu}_k$ *in an alternative way*

$$\min_{\boldsymbol{r}_n, \boldsymbol{\mu}_k} J$$

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

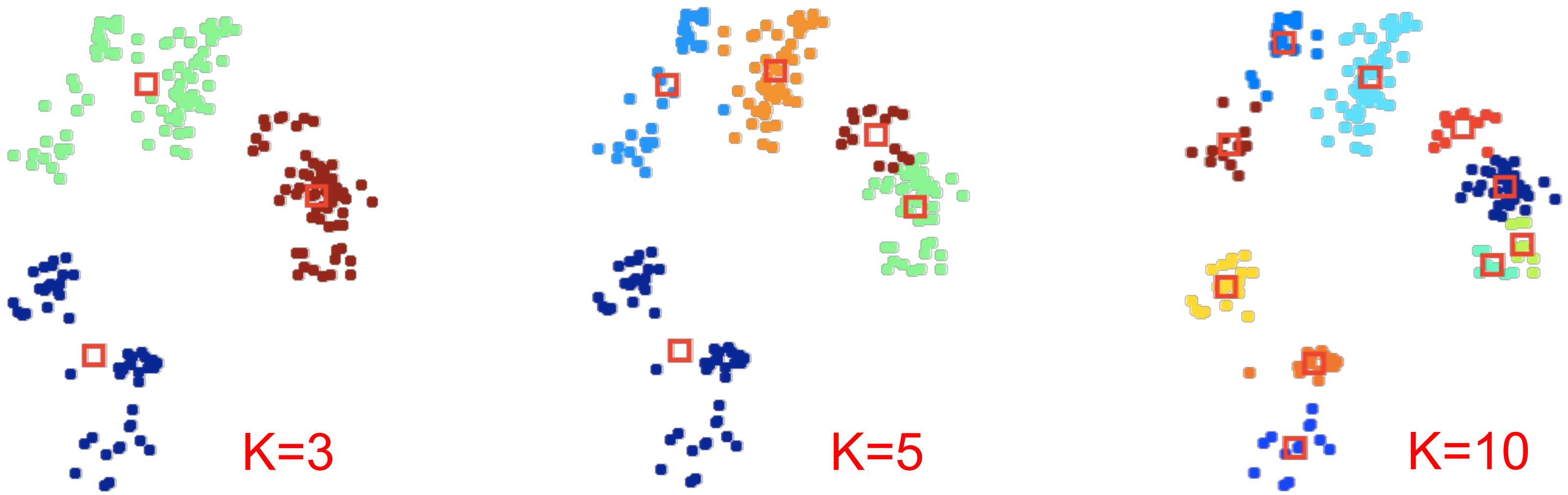
where $\boldsymbol{r}_n \triangleq [r_{n1}, r_{n2}, \cdots, r_{nK}]$ is required to be a one-hot vector

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

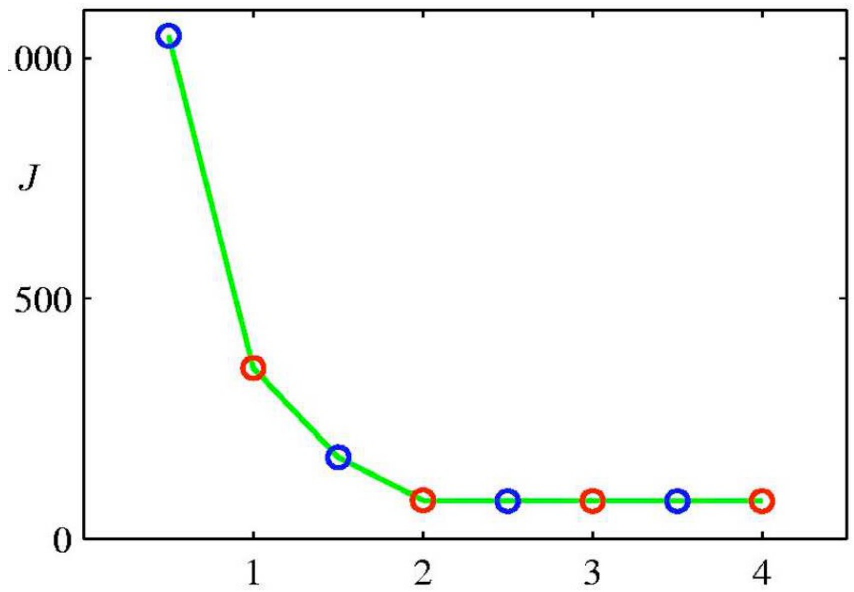

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)

Issue: Number of Clusters

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



- Distance J is determined to decrease as the number of clusters K increases. Thus, K cannot be determined by minimizing J

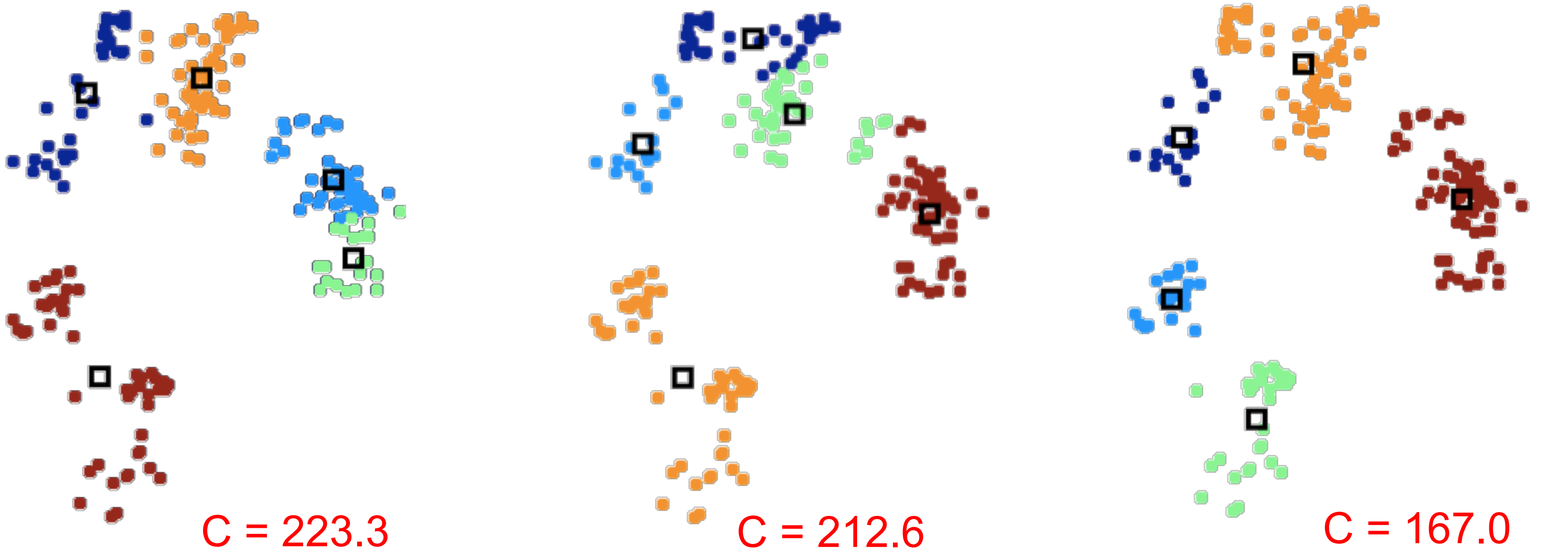


1) One possible method is to choose the elbow point (here $K = 2$)

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

Issue: Initialization

- Performance of *K*-means also highly depends on the initial centers



1) Random method

- Randomly choosing data instances as the initialization
- Issue: may choose nearby instances

2) Distance-based method

- Start with one random data instance
- Choose the point that is furthest 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

Issue: Hard Assignment

- Hard assignment

  A data instance belongs to a cluster or not deterministically, that is, $\boldsymbol{r}_n$ is required to be a one-hot vector

- Soft *K*-means

  β controls sharpness of the distribution

  Instead of assigning $\boldsymbol{x}^{(n)}$ to a cluster deterministically, soft *K*-means assign the cluster in a soft way

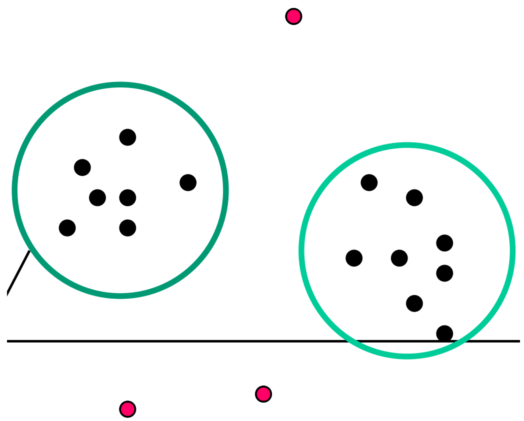
$$r_{nk} = \frac{e^{-\beta \left\|\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_k\right\|^2}}{\sum_{i=1}^{K} e^{-\beta \left\|\boldsymbol{x}^{(n)} - \boldsymbol{\mu}_i\right\|^2}}$$

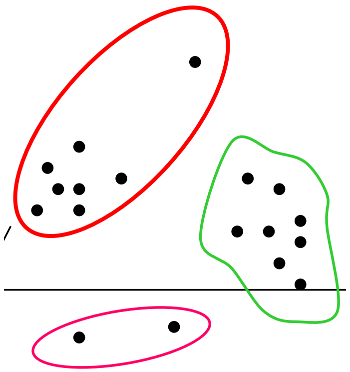
$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_{n=1}^{N} r_{nk}\, \boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}}$$

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 implies the boundary can only be globular. When clusters have irregular shapes, the performance is poor