

Machine Learning Lecture 7

Clustering

Qian Ma

Sun Yat-sen University

Spring Semester, 2020



Acknowledgement

- A large part of slides in this lecture are originally from
 - Prof. Andrew Ng (Stanford University)
 - Prof. Shuai Li (Shanghai Jiao Tong University)



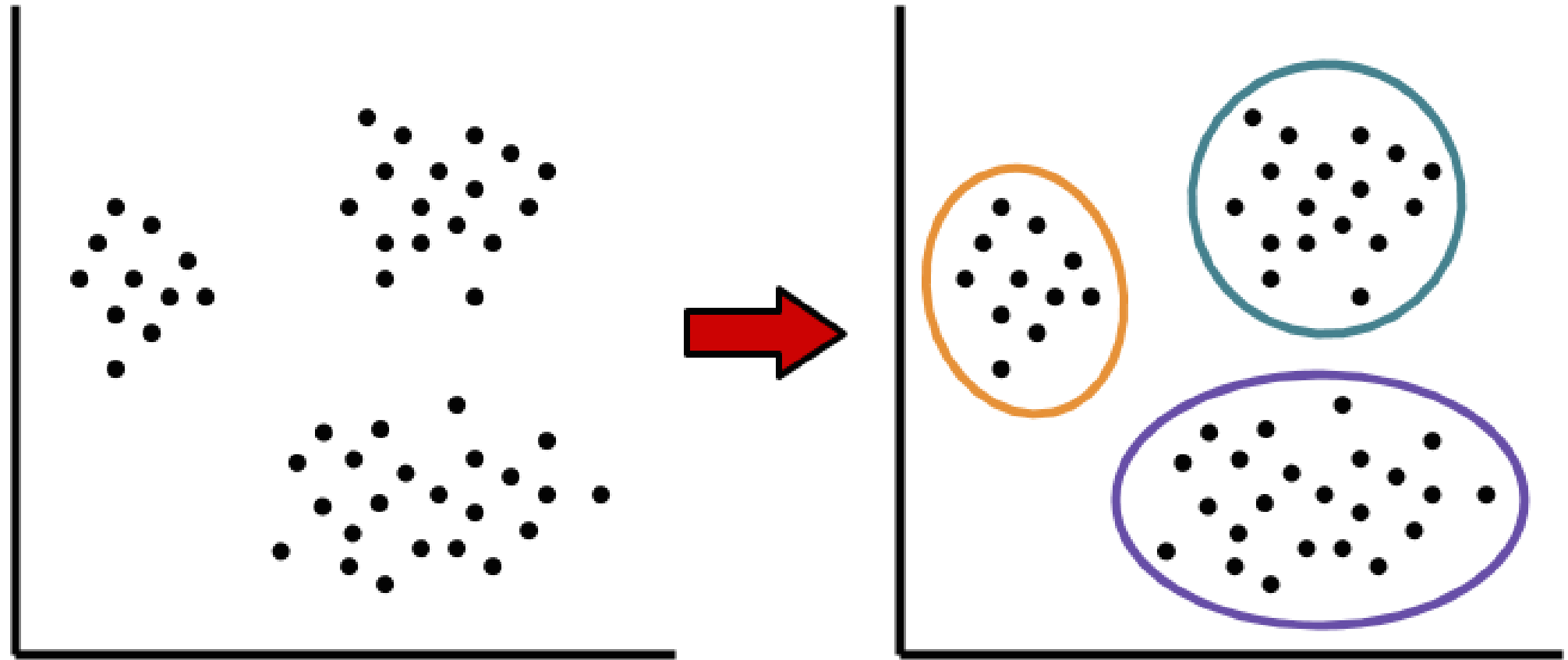
Prof. Andrew Ng
Stanford University



Prof. Shuai Li
Shanghai Jiao Tong University

Unsupervised Learning

Unsupervised learning example



Supervised Learning	Unsupervised Learning
Input data is labelled	Input data is unlabeled
Uses training dataset	Uses just input dataset
Used for prediction	Used for analysis
Classification and regression	Clustering, density estimation and dimensionality reduction

Clustering

Clustering

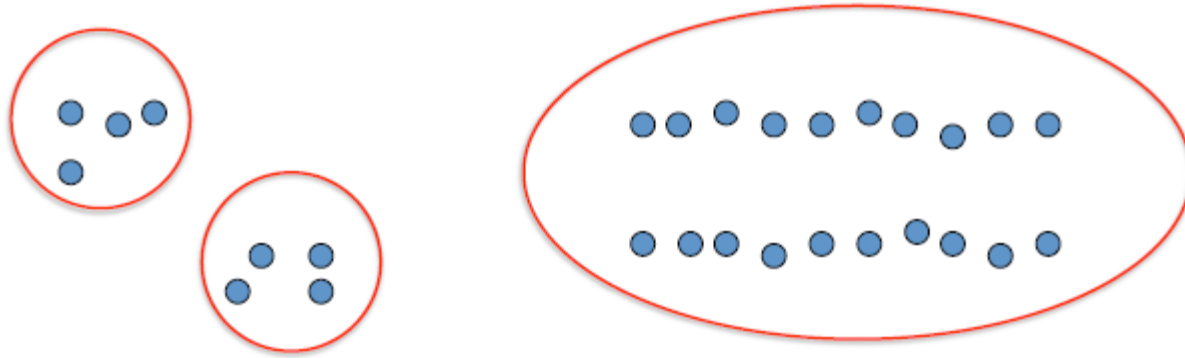
- Unsupervised learning
- Requires data, but no labels
- Detect patterns e.g. in
 - Group emails or search results
 - Customer shopping patterns
 - Regions of images
- Useful when don't know what you're looking for
- But: can get gibberish

Clustering (cont.)

- Goal: Automatically segment data into groups of similar points
- Question: When and why would we want to do this?
- Useful for:
 - Automatically organizing data
 - Understanding hidden structure in some data
 - Representing high-dimensional data in a low-dimensional space
- Examples: Cluster
 - customers according to purchase histories
 - genes according to expression profile
 - search results according to topic
 - Facebook users according to interests
 - a museum catalog according to image similarity

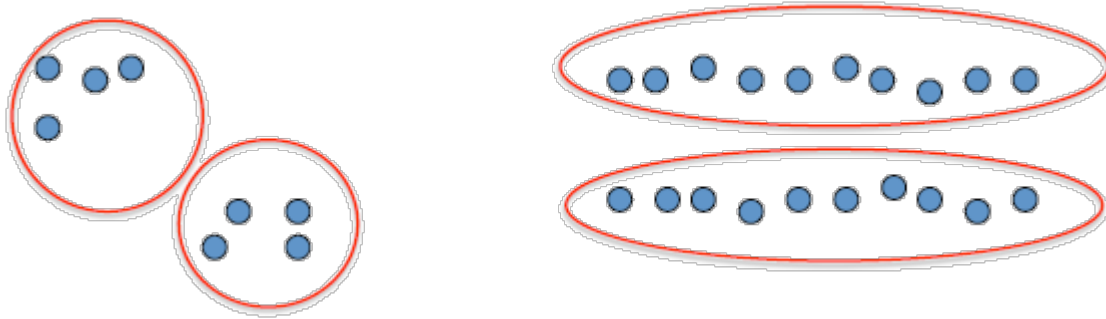
Intuition

- Basic idea: group together similar instances
- Example: 2D point patterns



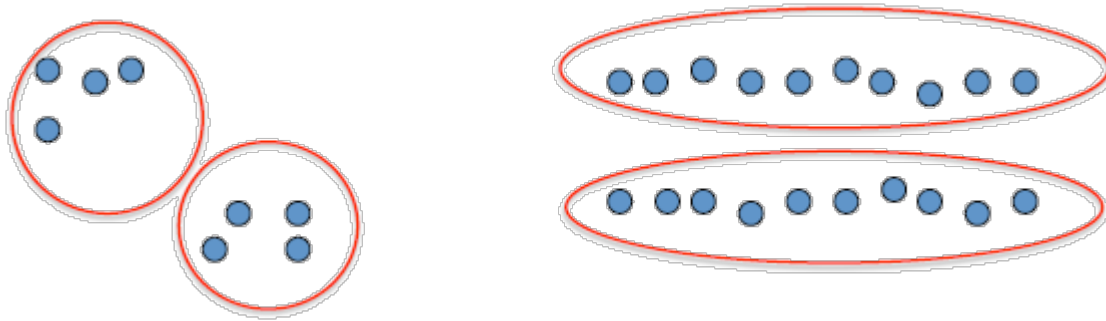
Intuition (cont.)

- Basic idea: group together similar instances
- Example: 2D point patterns



Intuition (cont.)

- Basic idea: group together similar instances
- Example: 2D point patterns



- What could “similar” mean?
 - – One option: small Euclidean distance (squared)
 - – Clustering results are crucially dependent on the measure of similarity (or **distance**) between “points” to be clustered

Set-up

- Given the data: $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$

- Each data point \mathbf{x} is d -dimensional:

$$\mathbf{x}_i = (x_{i,1}, \dots, x_{i,d})$$

- Define a distance function between data:

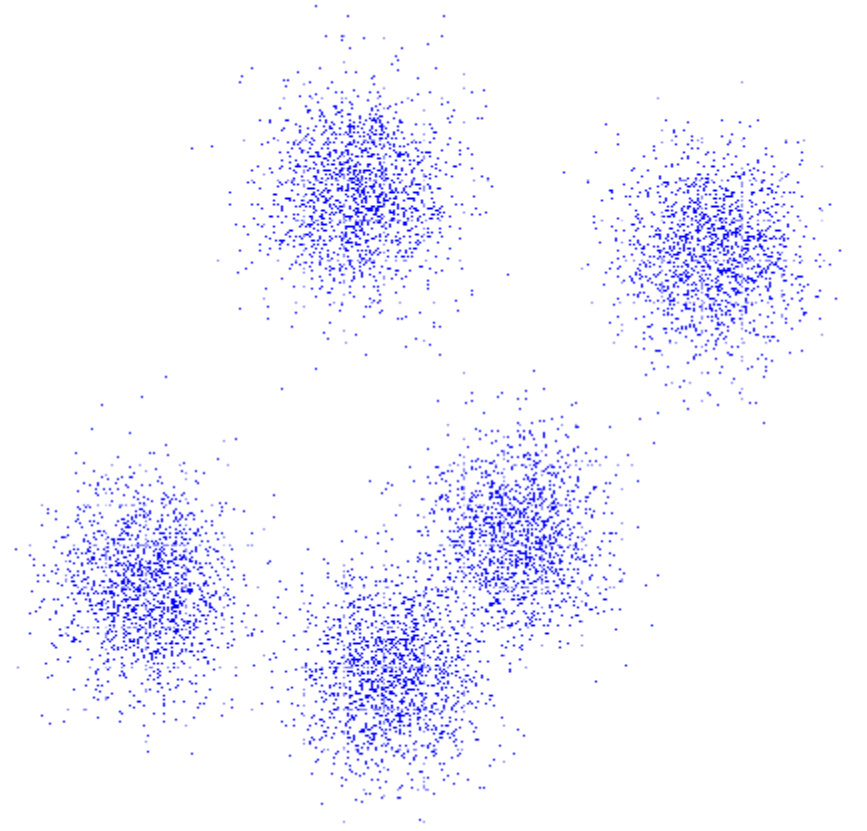
$$d(\mathbf{x}_n, \mathbf{x}_m).$$

- Goal: segment the data into K groups

K-Means

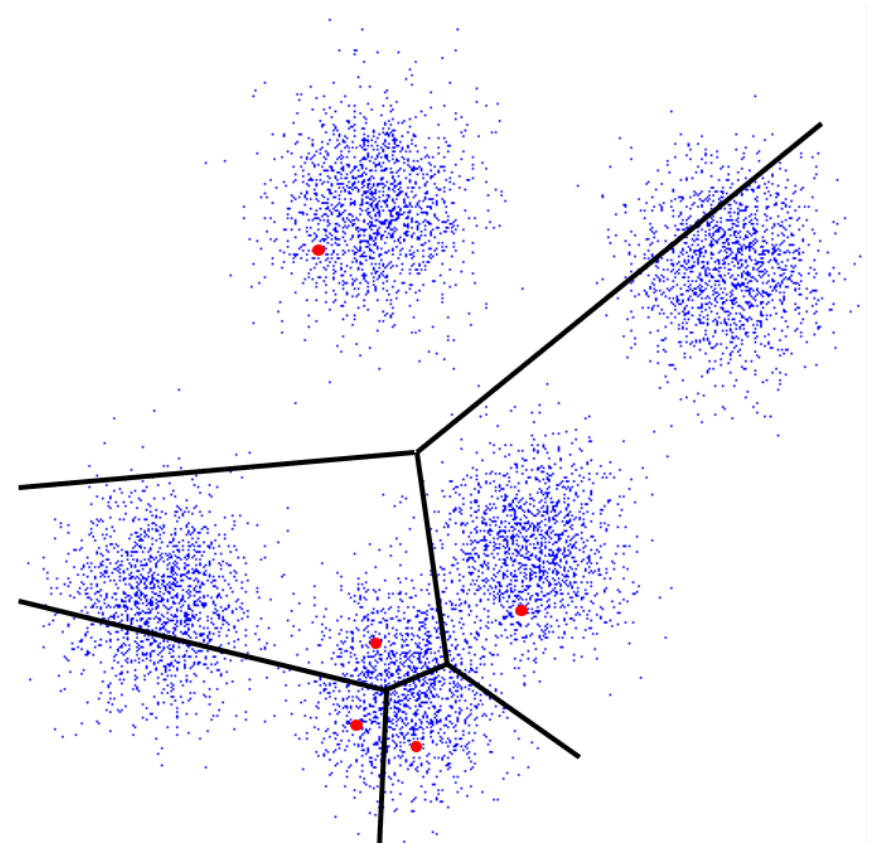
K-Means

- An iterative clustering algorithm
- **Initialize:** Pick K **random** points as cluster centers
- **Alternate:**
 - Assign data points to closest cluster center
 - Change the cluster center to the average of its assigned points
- **Stop:** when no points' assignments change



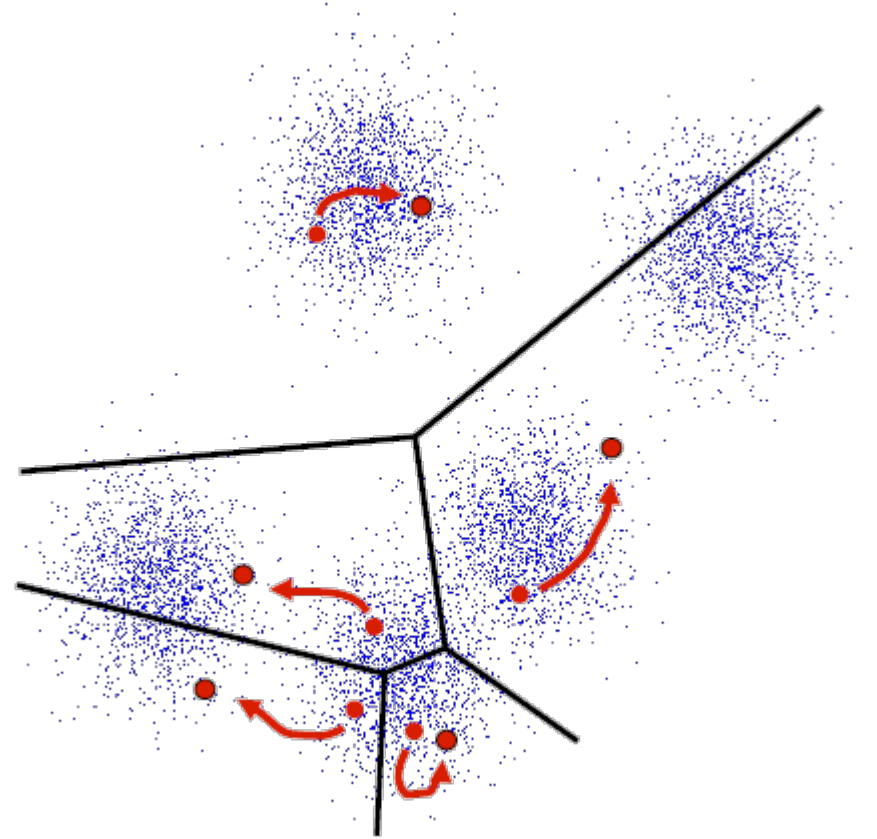
K-Means (cont.)

- An iterative clustering algorithm
- **Initialize**: Pick K random points as cluster centers
- **Alternate**:
 - Assign data points to closest cluster center
 - Change the cluster center to the average of its assigned points
- **Stop**: when no points' assignments change



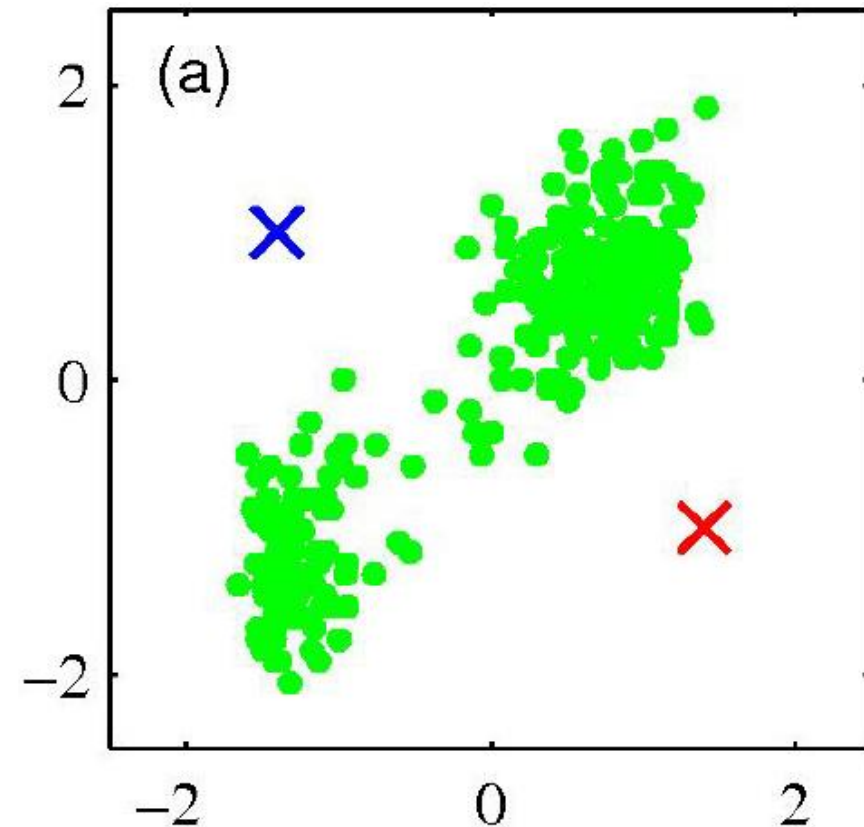
K-Means (cont.)

- An iterative clustering algorithm
- **Initialize**: Pick K random points as cluster centers
- **Alternate**:
 - Assign data points to closest cluster center
 - Change the cluster center to the average of its assigned points
- **Stop**: when no points' assignments change



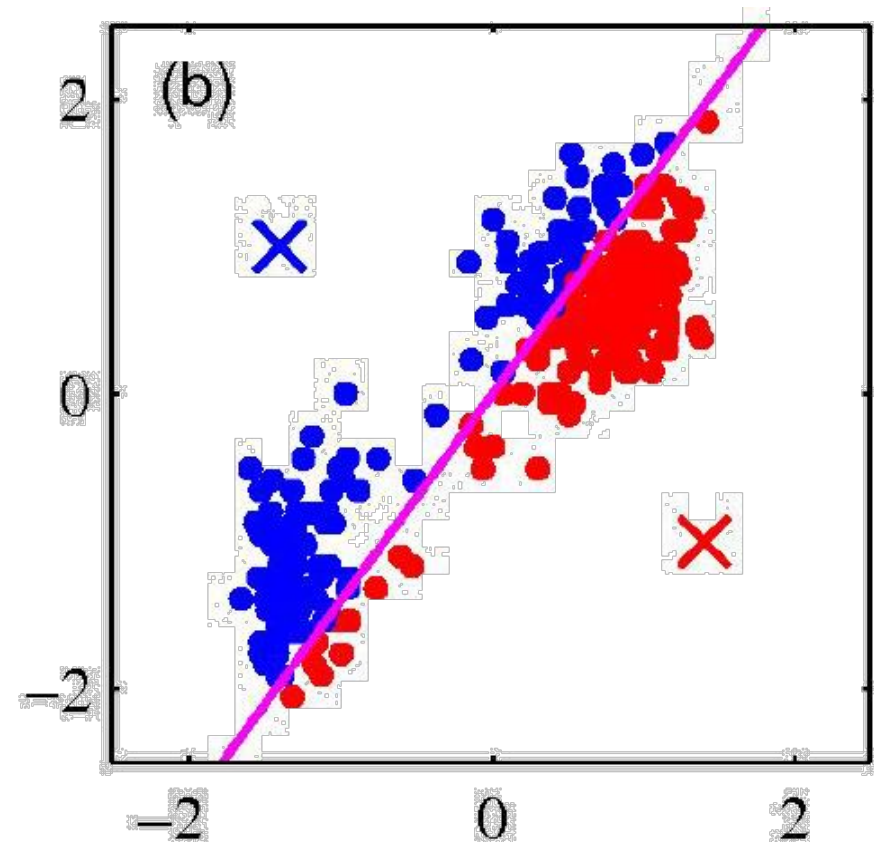
Example

- Pick K random points as cluster centers (means)
- Shown here for $K = 2$



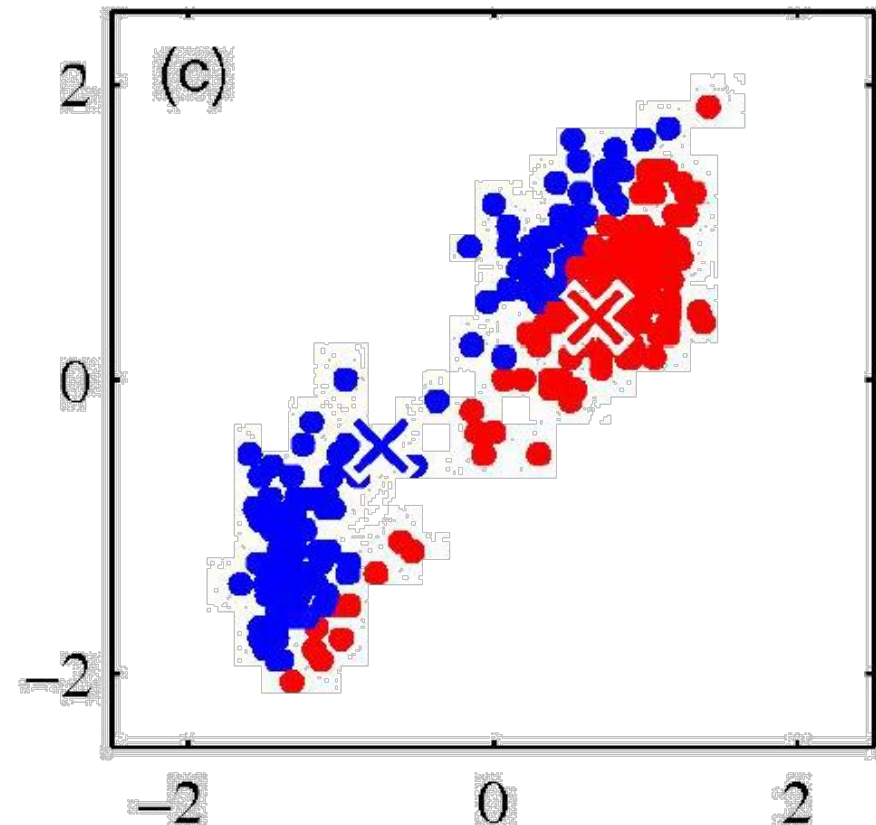
Example (cont.)

- Iterative step 1
- Assign data points to closest cluster center



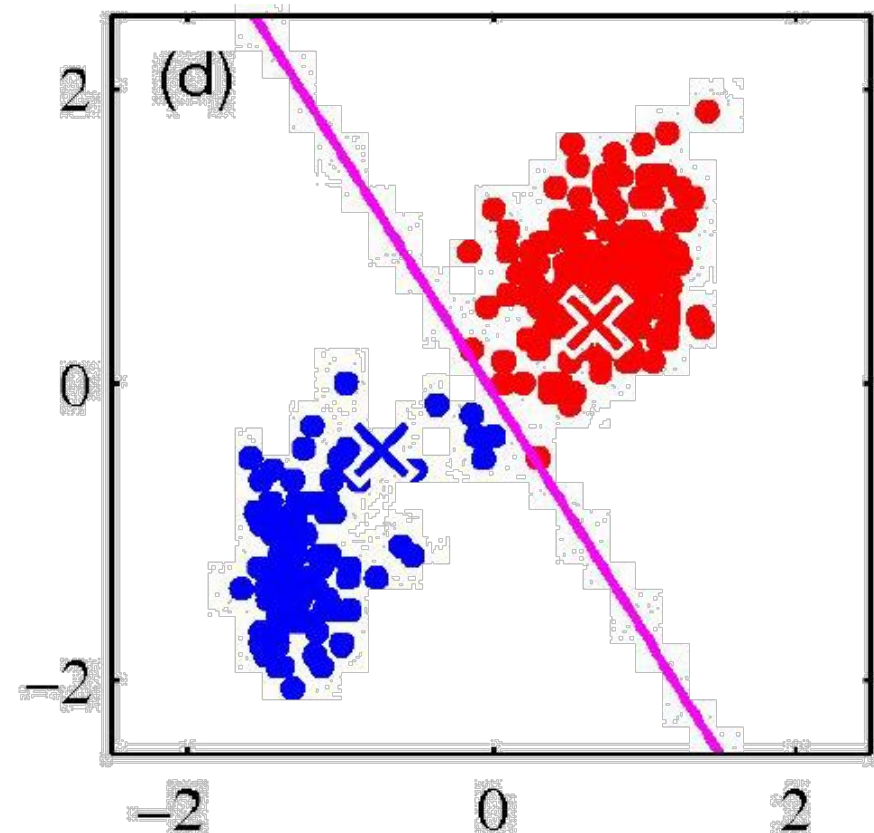
Example (cont.)

- Iterative step 2
- Change the cluster center to the average of the assigned points



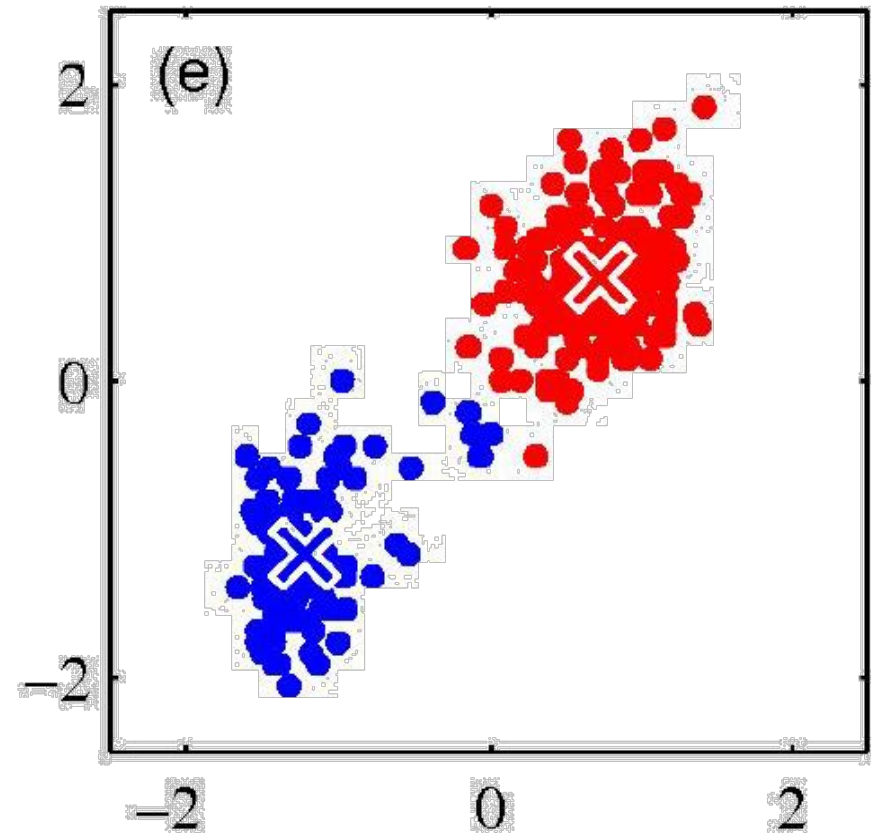
Example (cont.)

- Repeat until convergence
- Convergence means that the differences of the center positions in two continuous loops is smaller than a threshold



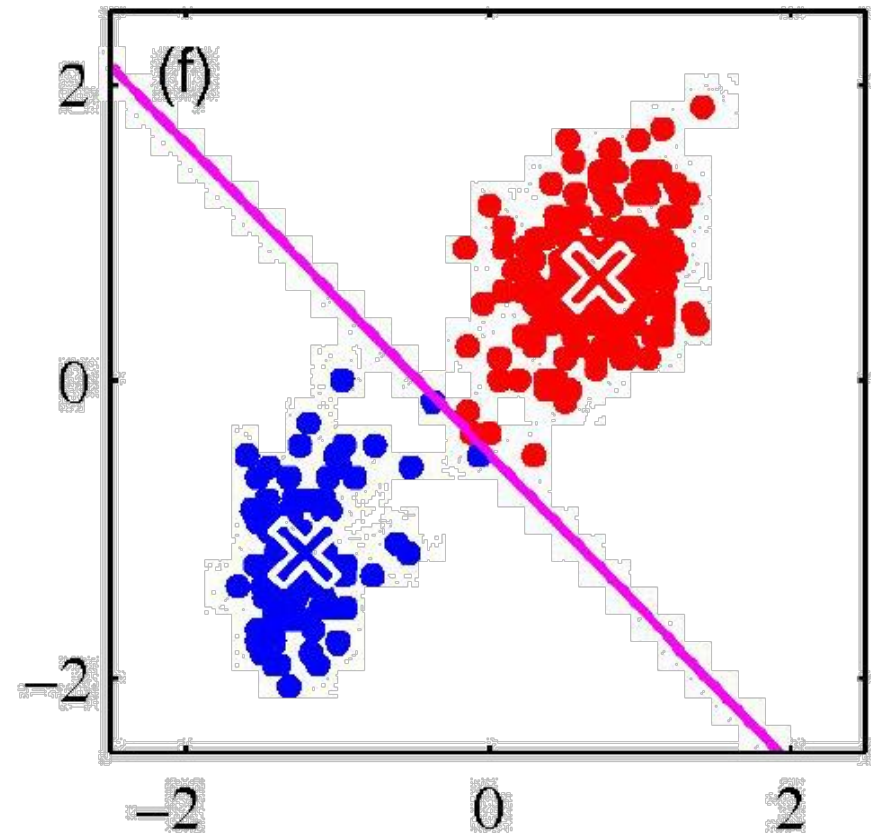
Example (cont.)

- Repeat until convergence
- Convergence means that the differences of the center positions in two continuous loops is smaller than a threshold



Example (cont.)

- Repeat until convergence
- Convergence means that the differences of the center positions in two continuous loops is smaller than a threshold



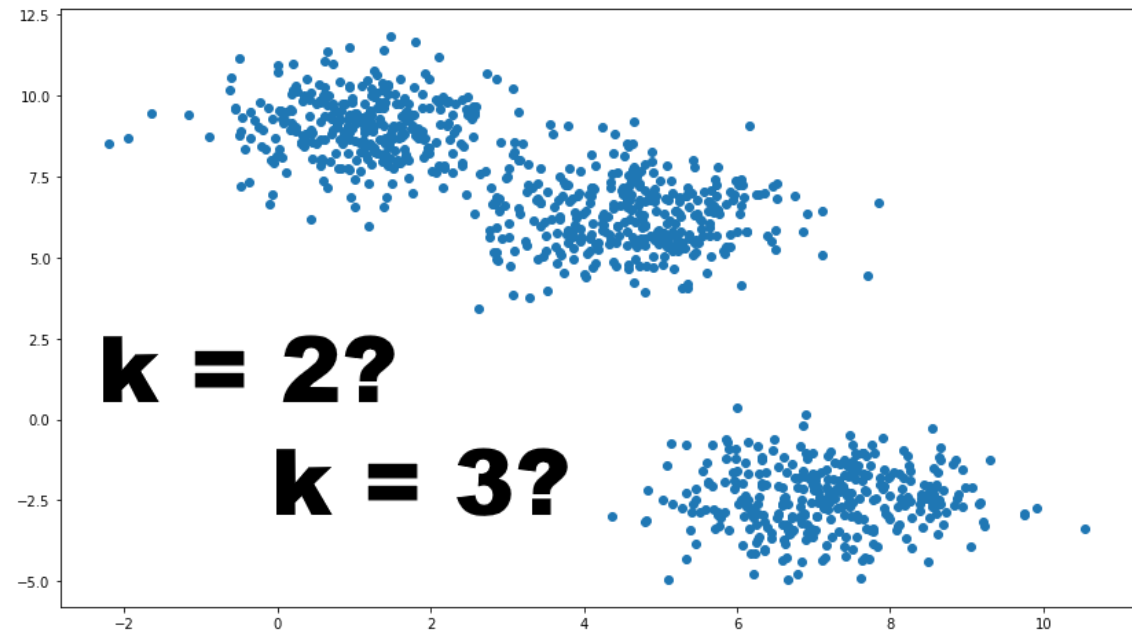
Remained Questions in K-Means

Remained questions in K-means

- Although the workflow of K-means is straight forward, there are some important questions that need to be discussed
- How to choose the hyper-parameter K ?
- How to initialize?

How to choose K?

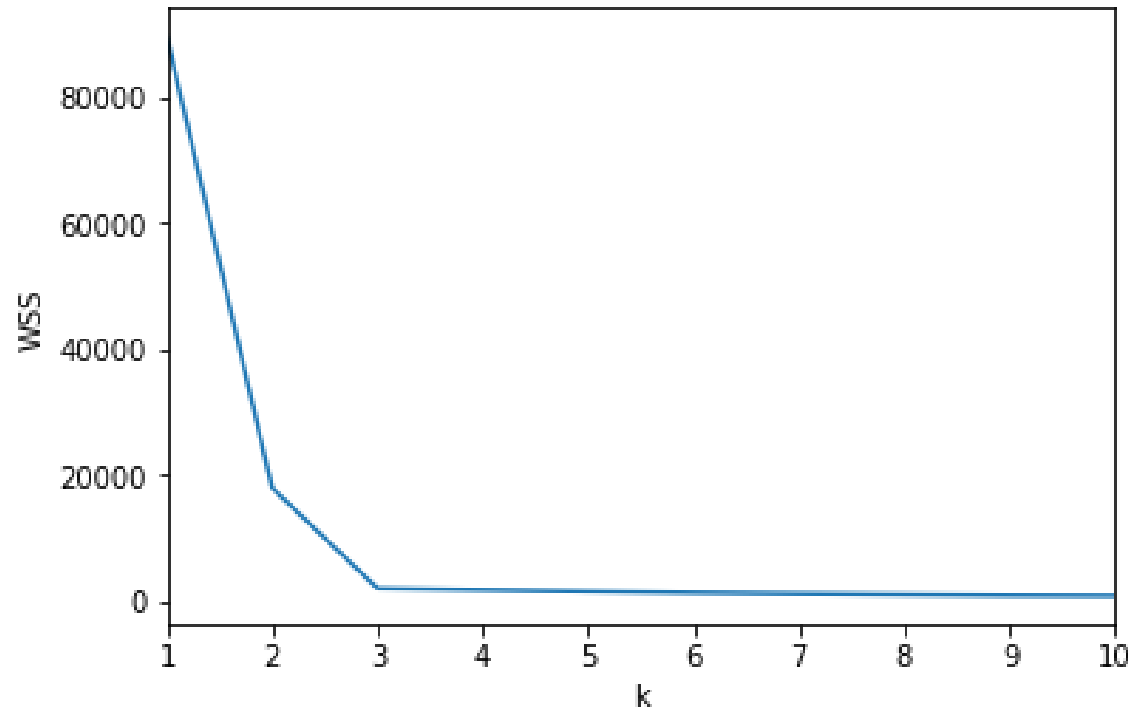
- K is the most important hyper-parameter in K-means which strongly affects its performance. In some situation, it's not an easy task to find the proper K
- The solution includes:
 - The elbow method
 - The silhouette method



The elbow method

- Calculate the **Within-Cluster-Sum of Squared Errors (WSS)** for different values of K , and choose the K **for which WSS stops dropping significantly**. In the plot of WSS-versus-k, this is visible as an elbow

- Example: $K = 3$



The silhouette method

- The problem of the elbow method is that in many situations the most suitable K cannot be unambiguously identified. So we need the silhouette method
- The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation). The range of the silhouette value is between +1 and -1. A high value is desirable and indicates that the point is placed in the correct cluster

The silhouette method (cont.)

- For each data point $i \in C_k$, let $a(i)$ be its mean distance to all other points in the same cluster

$$a(i) = \frac{1}{|C_k| - 1} \sum_{j \in C_k, i \neq j} d(i, j)$$

- And let $b(i)$ be the smallest mean distance to other clusters

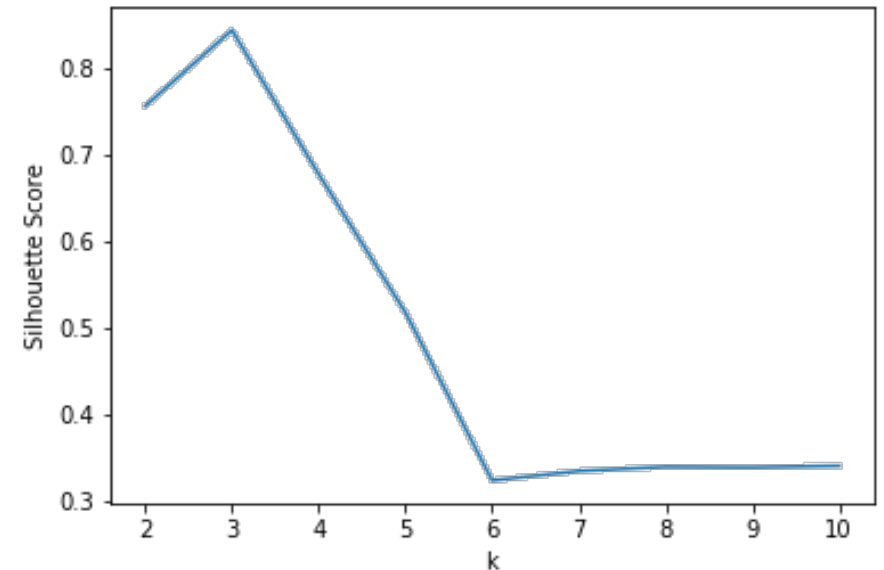
$$b(i) = \min_{l \neq k} \frac{1}{|C_l|} \sum_{j \in C_l} d(i, j)$$

The silhouette method (cont.)

- The silhouette value of $i \in C_k$ is defined as:

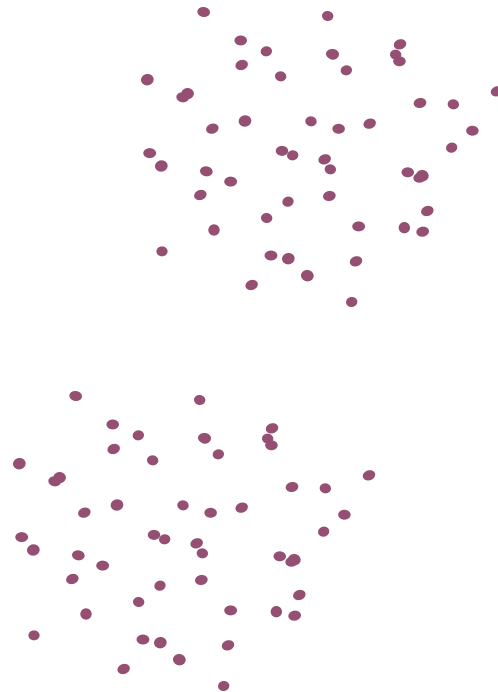
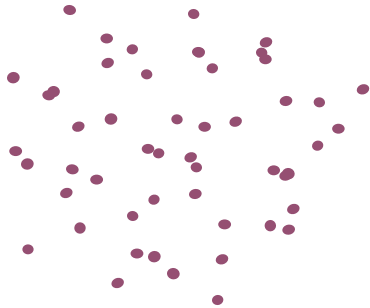
$$s(i) = \begin{cases} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, & \text{if } |C_k| > 1 \\ 0, & \text{if } |C_k| = 1 \end{cases}$$

- The silhouette score is the **average** of $s(i)$ among all data
- Choose the k with the maximal silhouette score



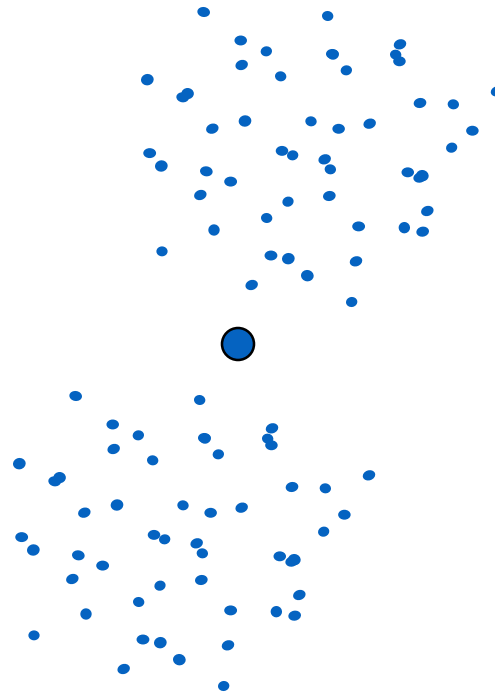
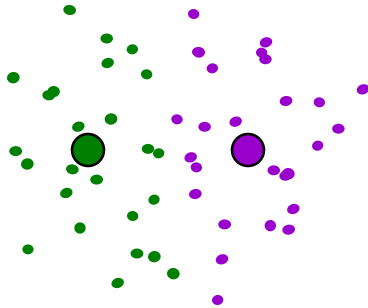
How to initialize center positions?

- The positions of the centers in the stage of initialization are also very important in K-means algorithms. In some situations it can produce totally different clustering results
- Example:



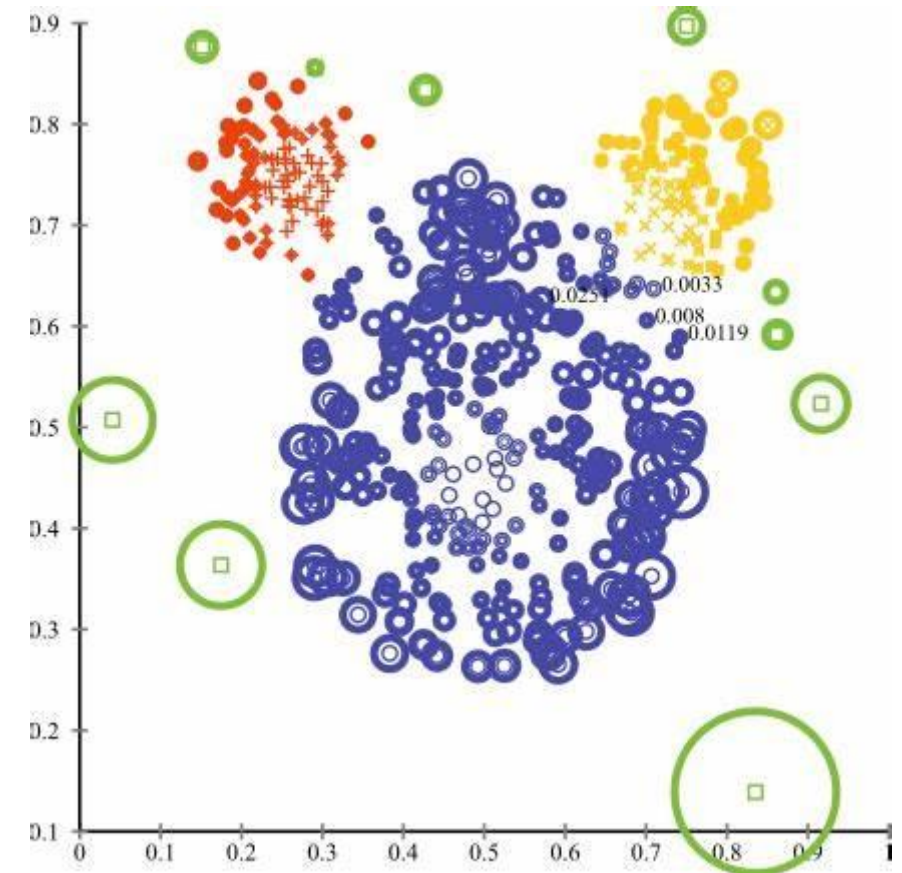
How to initialize center positions? (cont.)

- The positions of the centers in the stage of initialization are also very important in K-means algorithms. In some situations it can produce totally different clustering results
- Example:



A possible solution

- Pick one point at random, then $K - 1$ other points, each as far away as possible from the previous points
 - OK, as long as there are no *outliers* (points that are far from any reasonable cluster)



K-means++

1. The **first** centroid is chosen uniformly at random from the data points that we want to cluster. This is similar to what we do in K-Means, but instead of randomly picking all the centroids, we just pick one centroid here
2. Next, we compute the distance d_x is the nearest distance from data point x to the centroids that have already been chosen
3. Then, choose the new cluster center from the data points with the probability of x being proportional to d_x^2
4. We then repeat steps 2 and 3 until K clusters have been chosen

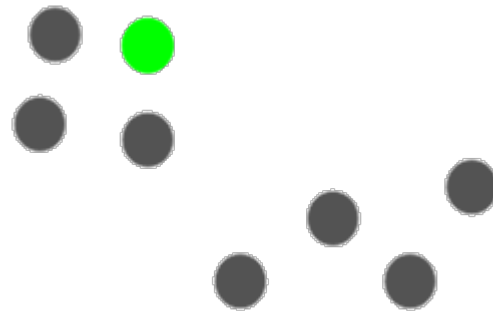
Example

- Suppose we have the following points and we want to make 3 clusters here:



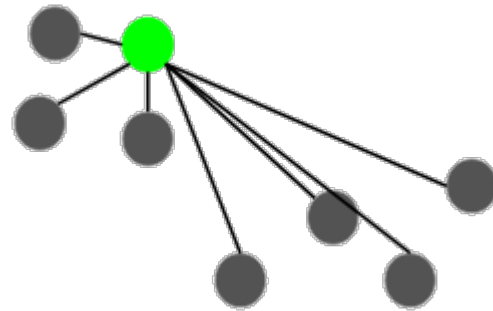
Example (cont.)

- First step is to randomly pick a data point as a cluster centroid:



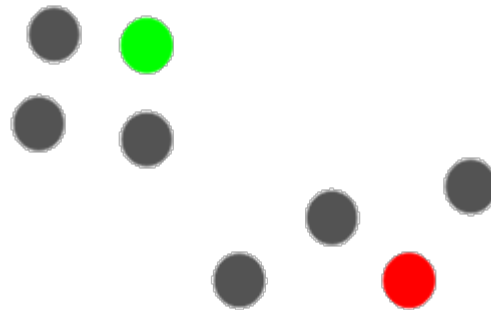
Example (cont.)

- Calculate the distance d_x of each data point with this centroid:



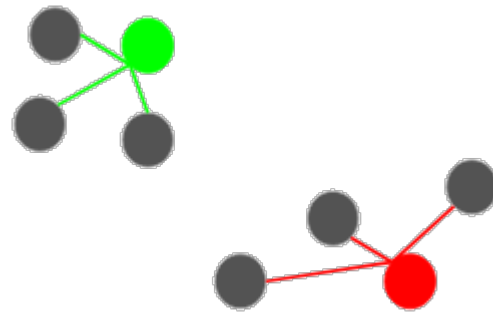
Example (cont.)

- The next centroid will be sampled with the probability proportional to d_x^2
- Say the sampled is the red one



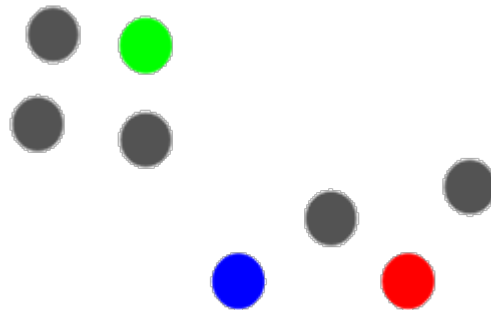
Example (cont.)

- To select the last centroid, compute d_x , which is the distance to its closest centroid



Example (cont.)

- Sample the one with the probability proportional to d_x^2
- Say, the blue one



Properties of K-Means

How to measure the performance

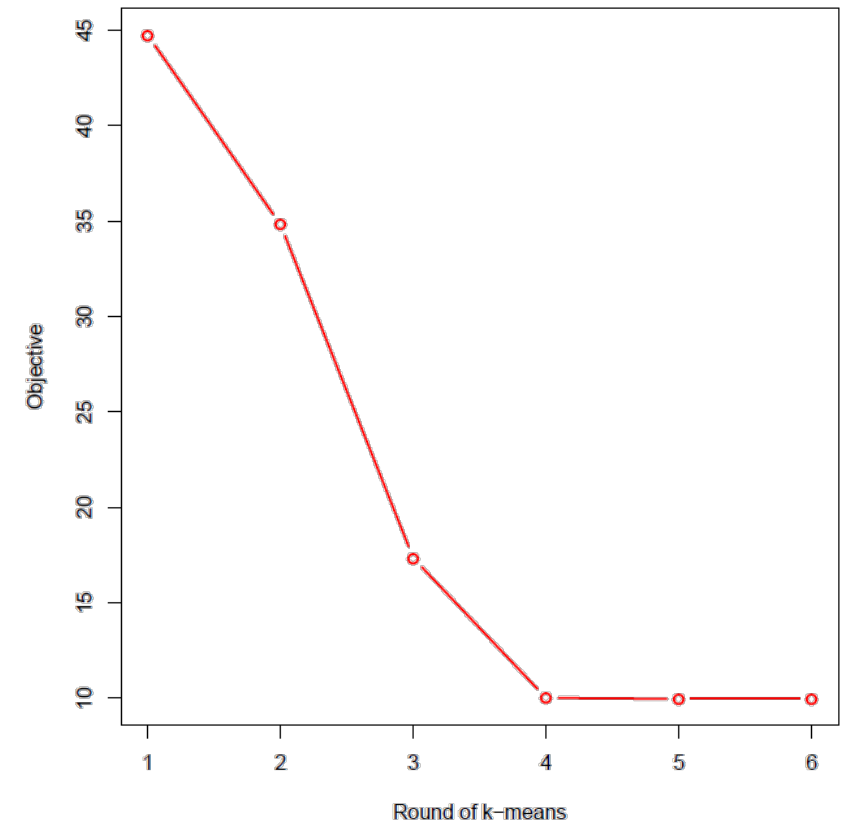
- K-means can be evaluated by the sum of distance from points to corresponding centers, or the WSS

number of clusters number of cases centroid for cluster j

case i

objective function $\leftarrow J = \sum_{j=1}^k \sum_{i=1}^n \underbrace{\|x_i^{(j)} - c_j\|}_{\text{Distance function}}^2$

- The loss will approach zero when increase K



Properties of the K-means algorithm

- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
 1. Assign data points to closest cluster center
 $O(KN)$ time
 2. Change the cluster center to the average of its assigned points
 $O(N)$

Distance

- Distance is of crucial importance in K-means. So what kind of properties should the distance measure have?
- **Symmetric**
 - $D(A, B) = D(B, A)$
- **Positivity, and self-similarity**
 - $D(A, B) \geq 0$, and $D(A, B) = 0$ iff $A = B$
- **Triangle inequality**
 - $D(A, B) + D(B, C) \geq D(A, C)$

Convergence of K-means

Objective

$$\min_{\mu} \min_C \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2$$

1. Fix μ , optimize C :

$$\min_C \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2 = \min_c \sum_i^n |x_i - \mu_{x_i}|^2$$

Step 1 of kmeans

2. Fix C , optimize μ :

$$\min_{\mu} \sum_{i=1}^k \sum_{x \in C_i} |x - \mu_i|^2$$

- Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

Step 2 of kmeans

Not guaranteed to converge to optimal

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

Application: Segmentation

- Goal of segmentation is to partition an image into regions each of which has reasonably homogenous visual appearance
- Cluster the colors

