K-Means

References

- Emily Fox & Carlos Guestrin Machine Learning Specialization University of Washington https://www.coursera.org/learn/ml-clustering-and-retrieval/

- **Cluster Analysis in Data Mining** University of Illinois at Urbana-Champaign https://www.coursera.org/learn/cluster-analysis

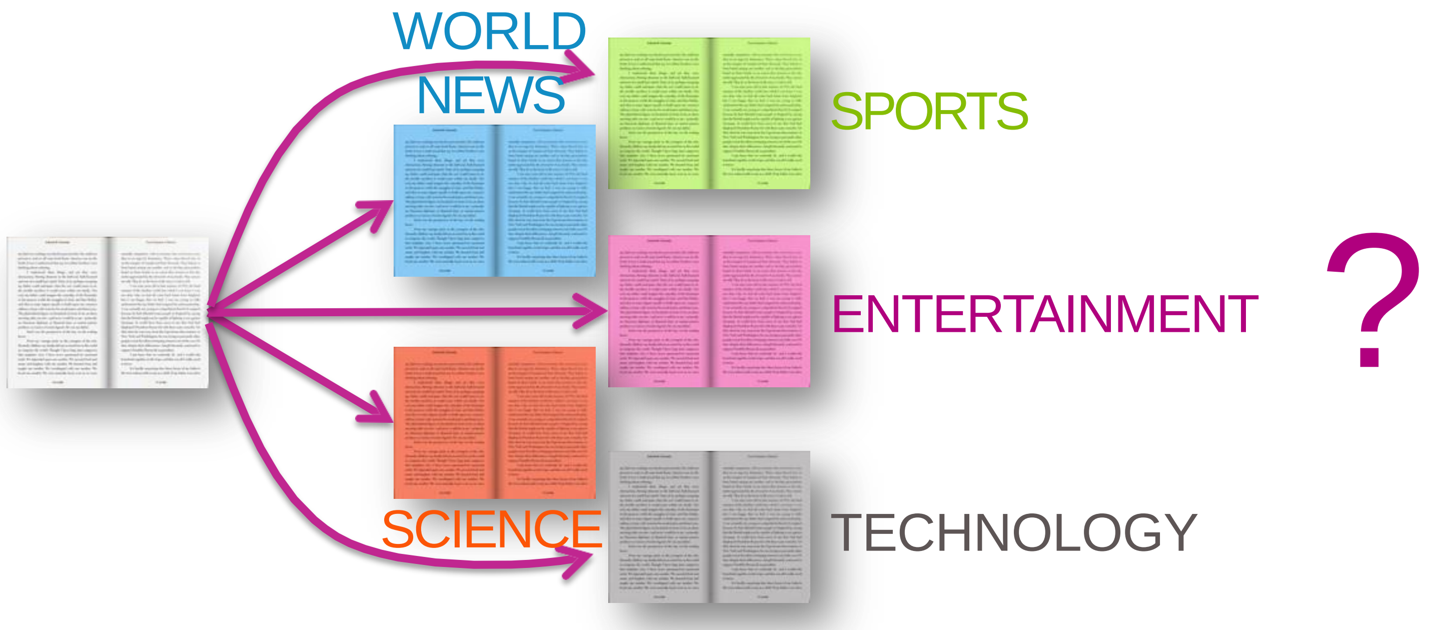
- Dr. Maggie Mashali, GUC.

What if the labels are known?

Training set of labeled docs

https://www.coursera.org/learn/ml-clustering-and-retrieval/

Multiclass classification problem



Example of
supervised learning

https://www.coursera.org/learn/ml-clustering-and-retrieval/

Clustering

No labels provided

…uncover cluster structure from input alone

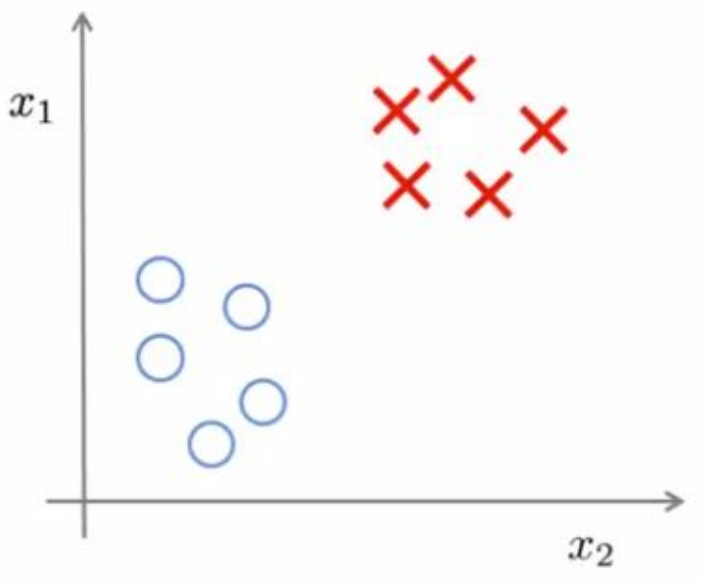
Input: docs as vectors x_i Output: cluster labels z_i

An unsupervised learning task

https://www.coursera.org/learn/ml-clustering-and-retrieval/

Supervised vs. Unsupervised Learning

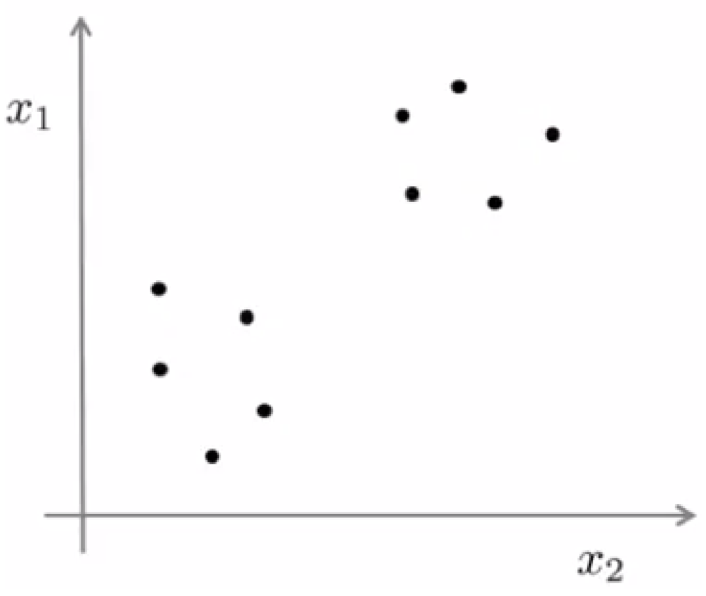
Supervised Learning



Training set:

$$\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),(x^{(3)},y^{(3)}),\ldots,(x^{(m)},y^{(m)})\}$$

Unsupervised Learning



Training set:

$$\{x^{(1)},x^{(2)},x^{(3)},\ldots,x^{(m)}\}$$

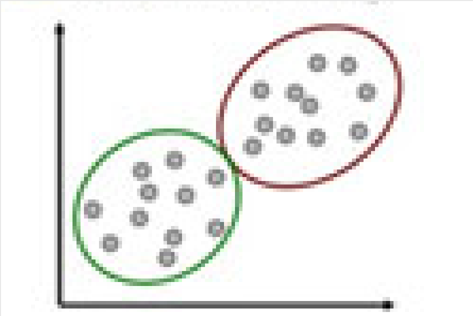
Unsupervised Learning

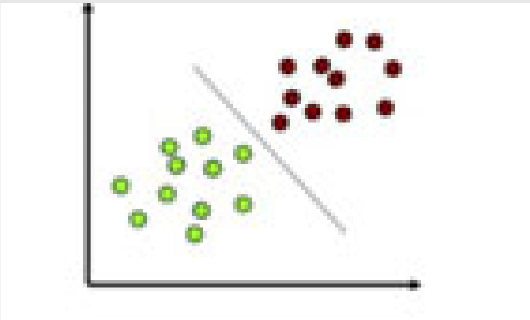
- **Goal:**

Finding structure within the data, usually by dividing it into Clusters

- But mind the difference:

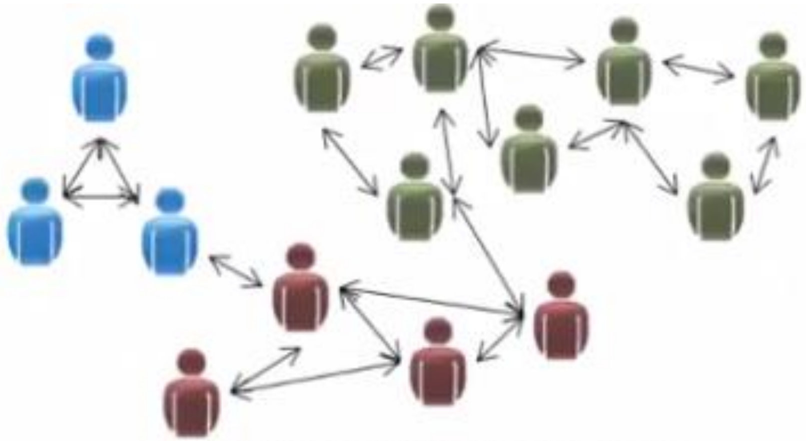
Clustering	Classification
• Data is not labeled • Group points that are "close" to each other • Identify structure or patterns in data • Unsupervised learning	• Labeled data points • Want a "rule" that assigns labels to new points • Supervised learning





Clustering: Use cases

Market segmentation



Social network analysis

Organize computing clusters

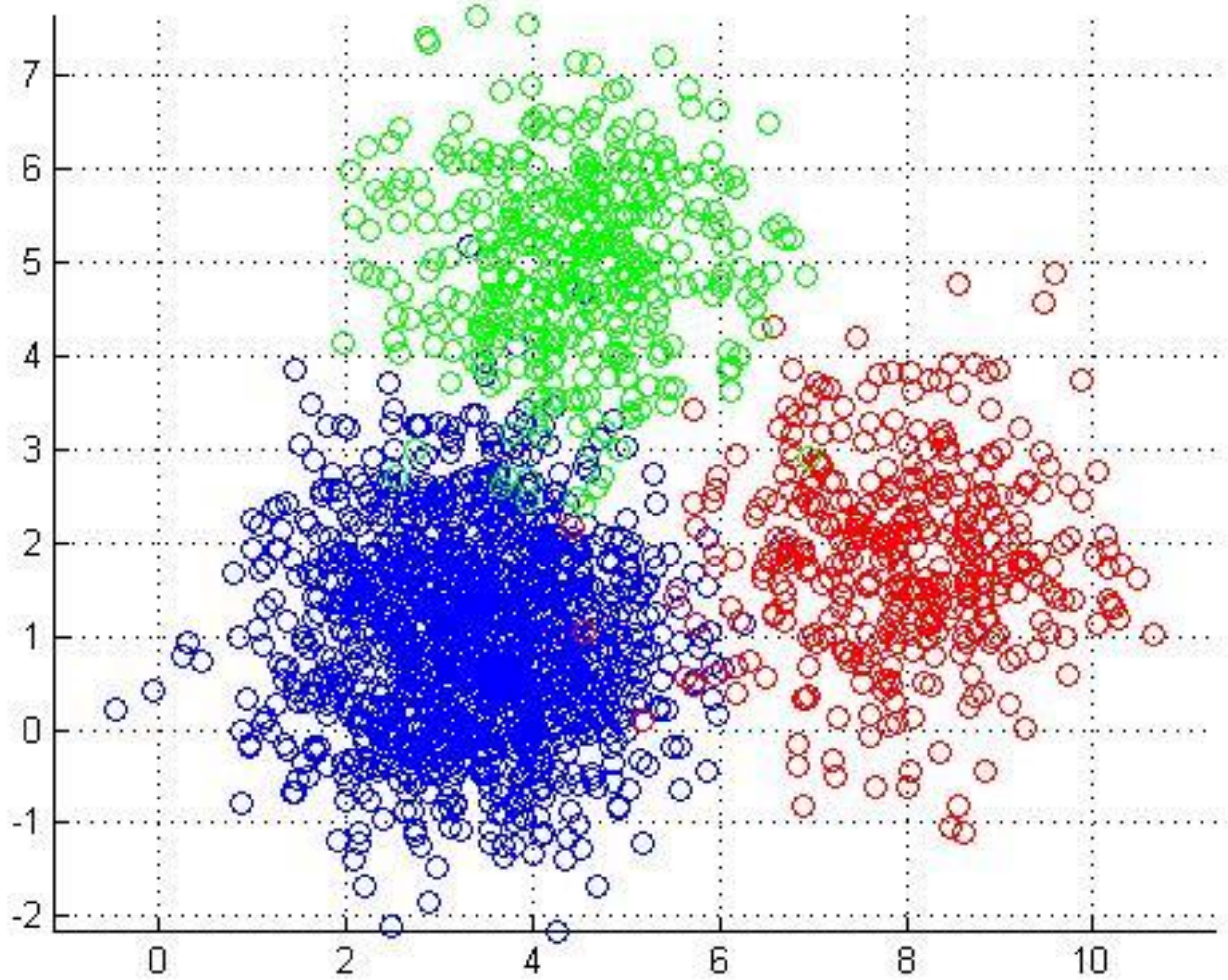
Astronomical data analysis

Applications of Clustering

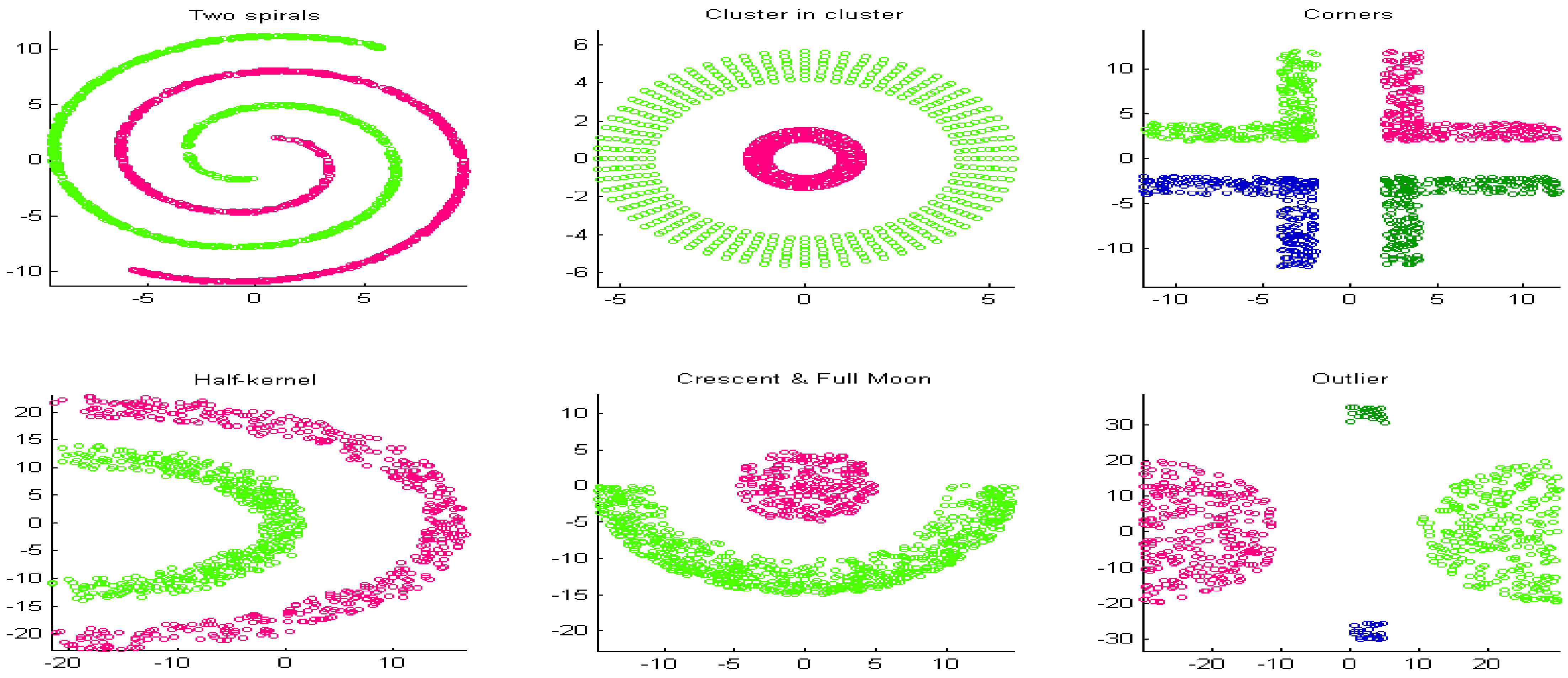
- Data summarization, compression, and reduction
  - Examples: Image processing or vector quantization
- Collaborative filtering, recommendation systems, or customer segmentation
  - Finding like-minded users or similar products
- Dynamic trend detection
  - Clustering stream data and detecting trends and patterns
- Multimedia data analysis, biological data analysis, and social network analysis
  - Example: Clustering images or video/audio clips, gene/protein sequences, etc.
- A key intermediate step for other data mining tasks
  - Generating a compact summary of data for classification, pattern discovery, and hypothesis generation and testing
- Outlier detection: Outliers - those “far away” from any cluster

https://www.coursera.org/learn/cluster-analysis/

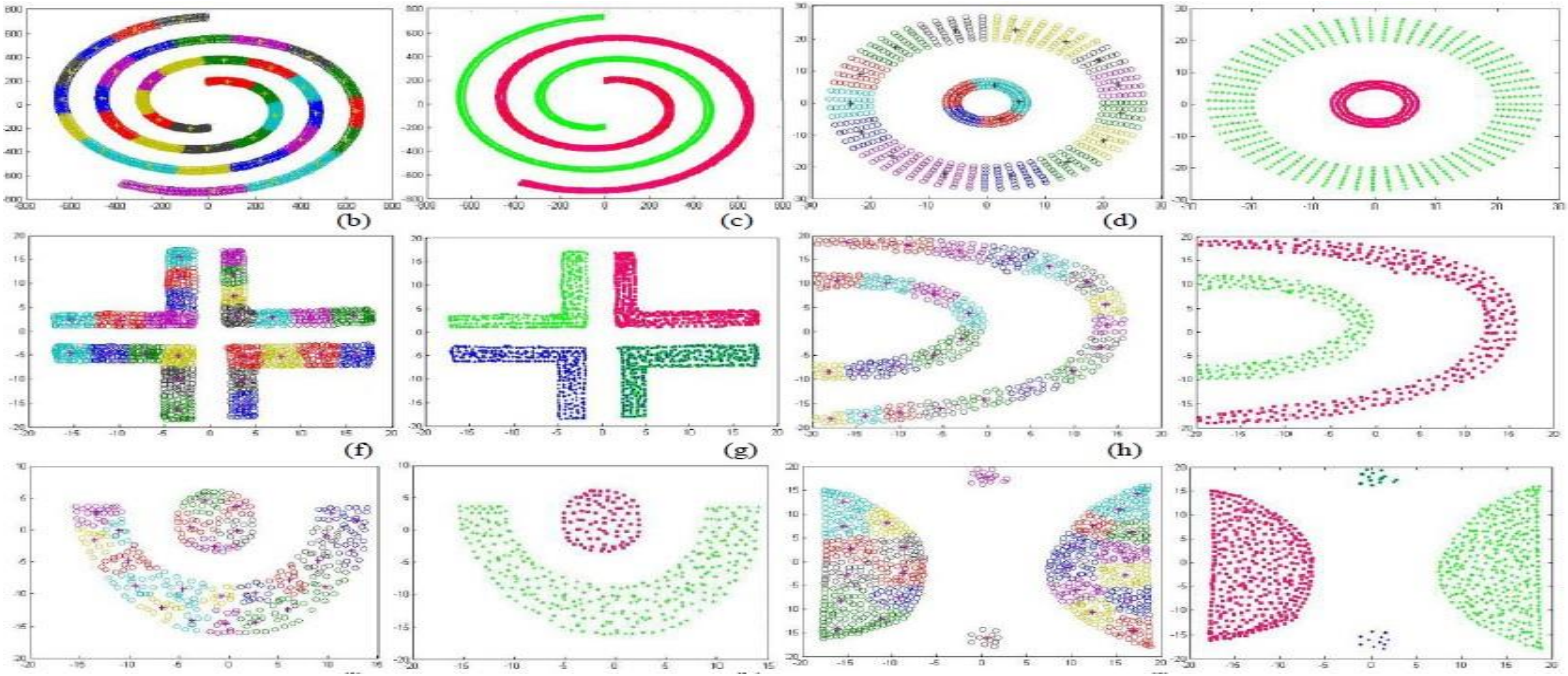
Clustering Easy?



(challenging!) clusters to discover…



(challenging!) clusters to discover…



Types of clustering paradigms

- Representative-based
- Hierarchical
- Density-based
- Graph-based
- Spectral clustering.

Clustering

- Clustering is the task of partitioning the data points into natural groups called clusters, such that points within a group are very similar, whereas points between different groups are as dissimilar as possible.
- $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$
- $C_i = \{x_j | x_j \in C_i\}$
- $\boldsymbol{\cup_{i=1}^{k} C_i}$**=D** → **noise point** $\notin \boldsymbol{\cup_{i=1}^{k} C_i}$
- $\boldsymbol{C_i} \cap \boldsymbol{C_j} = \emptyset$ → **disjoint cluster no overlapping**
- Clustering is an unsupervised learning approach since it does not require a separate training dataset to learn the model parameters.

Clustering Different Types of Data

- Numerical data
- Categorical data (including binary data)
  - Discrete data, no natural order (e.g., sex, race, zip-code, and market-basket)
- Text data: Popular in social media, Web, and social networks
  - Features: High-dimensional, sparse, value corresponding to word frequencies
- Multimedia data: Image, audio, video (e.g., on Flickr, YouTube)
  - Multi-modal (often combined with text data)
- Time-series data: Sensor data, stock markets, temporal tracking, forecasting, etc.
- Sequence data: Weblogs, biological sequences, system command sequences
- Stream data

Representative-based Clustering

Representative-based Clustering

- K-means
- Expectation-Maximization (EM) algorithms

Representative-based Clustering

- K-means is a greedy algorithm that minimizes the squared distance of points from their respective cluster means, and it performs hard clustering, that is, each point is assigned to only one cluster.
- We also show how kernel K-means can be used for nonlinear clusters.
- EM generalizes K-means by modeling the data as a mixture of normal distributions, and it finds the cluster parameters (the mean and covariance matrix) by maximizing the likelihood of the data. It is a soft clustering approach, that is, instead of making a hard assignment, it returns the probability that a point belongs to each cluster.

Kmeans

- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Portioning problem

- Problem definition: Given K, find a partition of K clusters that optimizes the chosen partitioning criterion
- A brute-force or exhaustive algorithm for finding a good clustering is simply to
  - generate all possible partitions of n points into k clusters
  - evaluate clusters
  - Choose the best clusters
- However, this is clearly infeasilbe, since there are $O(k_n/k!)$ clusterings of n points into k groups.
- Global optimal: Needs to exhaustively enumerate all partitions
- Heuristic methods (i.e., greedy algorithms): K-Means, K-Medians, K-Medoids, etc.

Example - K-means in one dimension

- Consider the following one-dimensional data. Assume that we want to cluster the data into k = 2 groups. {2, 3,4,10, 11,12,20,25,30}.

1) initial centroids **μ1 = 2** and **μ2 = 4**.

2) Loop until convergence

A. first iteration

a) cluster assignment, assigning each point to the closest mean:

C1 = {2, 3} C2 = {4, 10,11,12,20,25,30}

b) centroid update, update the means

$$\mu 1 = \frac{2+3}{2} = \frac{5}{2} = 2.5$$

$$\mu 2 = \frac{4+10+11+12+20+25+30}{7} = \frac{112}{7} = 16$$

B. **Second iteration**

a) assigning each point to the closest mean:

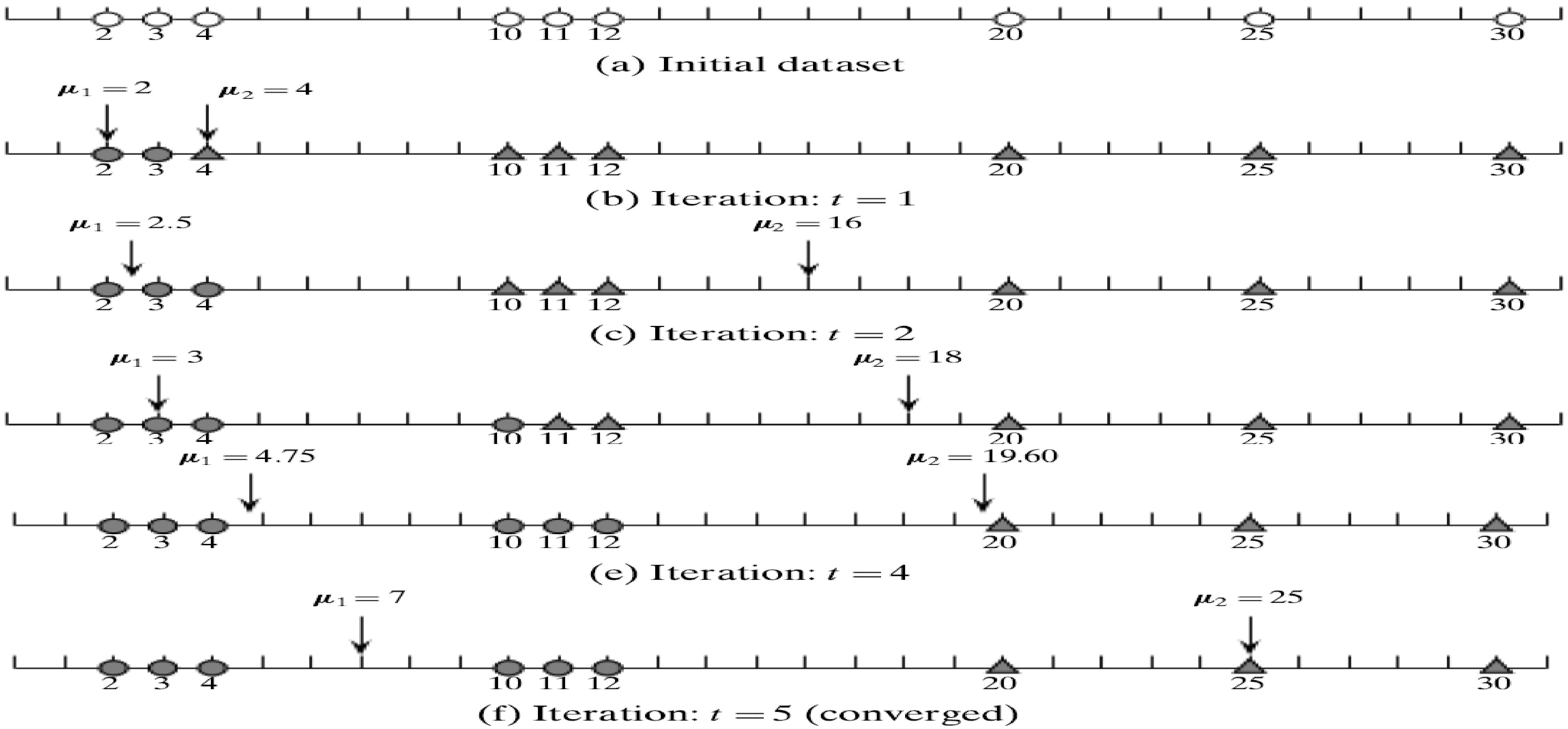
C1 = {2, 3,4} C2 = {10,11,12,20,25,30}

b) update the means

$$\mu 1 = \frac{2+3+4}{3} = \frac{9}{3} = 3$$

$$\mu 2 = \frac{10+11+12+20+25+30}{7} = \frac{108}{6} = 18$$

Example - K-means in one dimension



C1 = {2, 3,4, 10, 11, 12} C2 = {20,25,30} with representatives μ1 = 7 and μ2 = 25.

K-means Algorithm

- Each cluster is represented by the center of the cluster
- Given K, the number of clusters, the K-Means clustering algorithm is outlined as follows
- Randomly generating k points as initial centroids
- Repeat
  - Cluster assignment: Form K clusters by assigning each point to its closest centroid
  - Centroid update : Re-compute the centroids (i.e., mean point) of each cluster
- Until convergence criterion is satisfied

K-means Numerical Example(1)

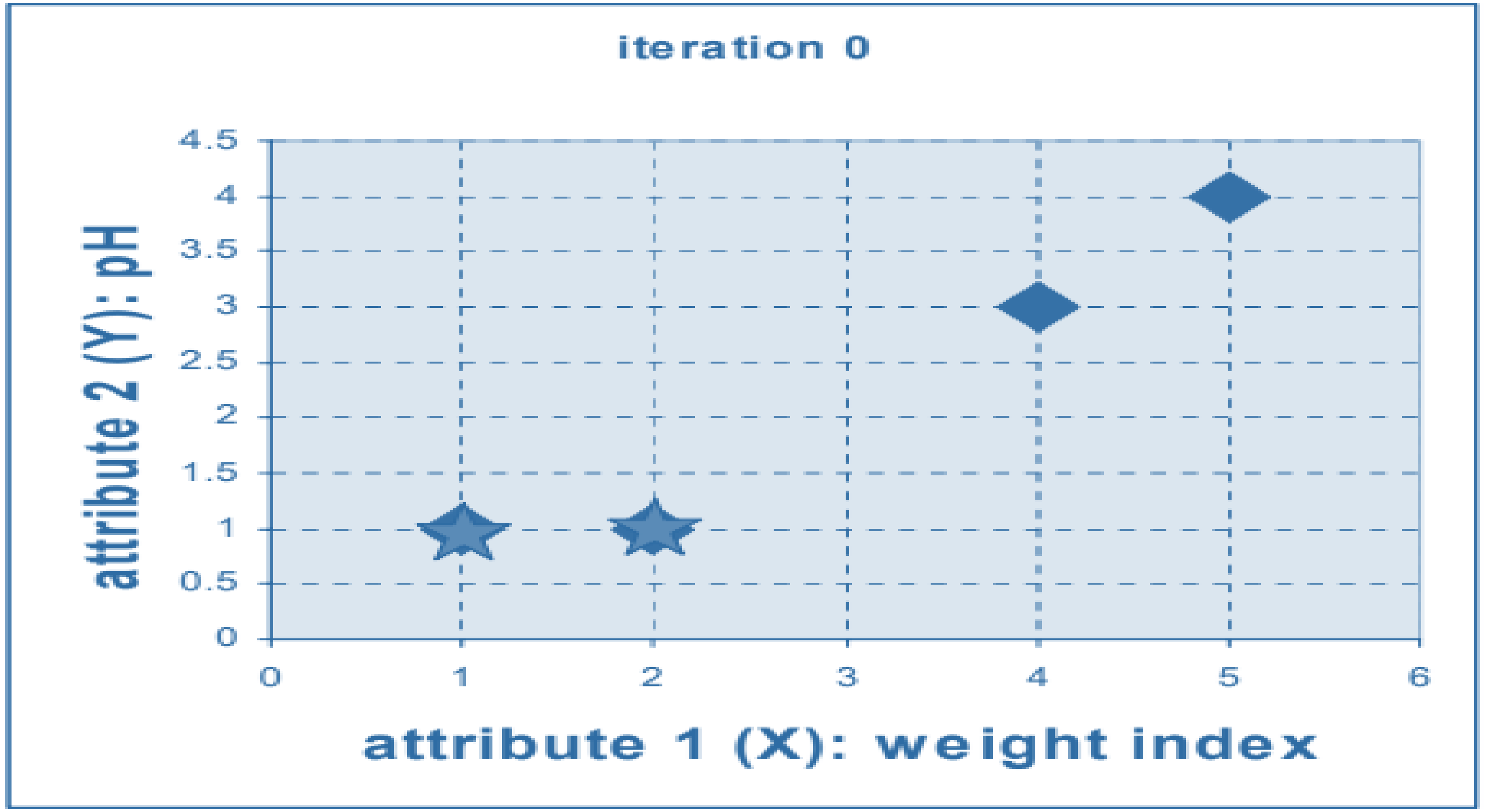
- Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. Our goal is to group these objects into K=2 group of medicine based on the two features (pH and weight index).

Object	Attribute 1 (x1): weight index	Attribute 2 (x2): pH
A	1	1
B	2	1
C	4	3
D	5	4

http://people.revoledu.com/kardi/tutorial/kMean/index.html

K-means Numerical Example(2)

- Each medicine represents one point with two features (X, Y) that we can represent it as coordinate in a feature space as shown in the figure.



Object	weight index	pH
A	1	1
B	2	1
C	4	3
D	5	4

K-means Numerical Example(3)

- Initial value of centroids: **μ1** = (1,1) and **μ2** = (2,1)
- **μ1** = (1,1) with Medicine C
- $=\sqrt{(4-1)^2+(3-1)^2}$
- $=\sqrt{(3)^2+(2)^2}$
- $= \sqrt{9+4}$
- $\sqrt{13}$ =3.61

Object	weight index	pH
A	1	1
B	2	1
C	4	3
D	5	4

K-means Numerical Example(3)

Cluster assignment

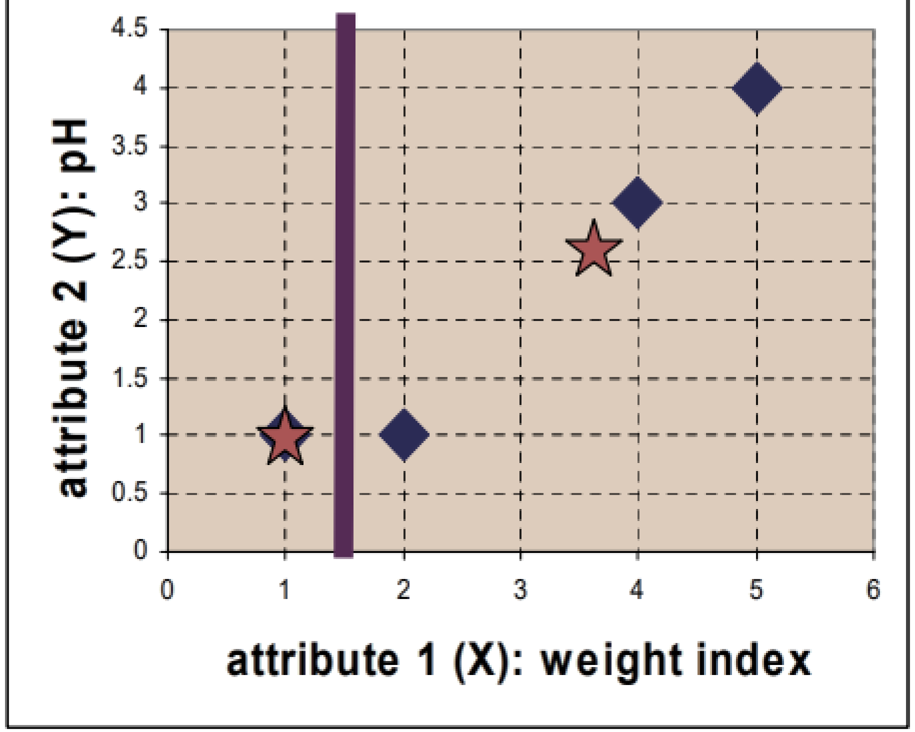
	μ1	μ2	Cluster
A	0	1	1
B	1	0	2
C	3.605551	2.828427	2
D	5	4.242641	2

Object	weight index	pH
A	1	1
B	2	1
C	4	3
D	5	4

Update Centroid

$\mu 1 = (1,1)$

$$\mu 2 = \frac{2+4+5}{3}, \frac{1+3+4}{3} = \frac{11}{3}, \frac{8}{3}$$



K-means another example

- Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:
- A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). The distance matrix based on the Euclidean distance is given below

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

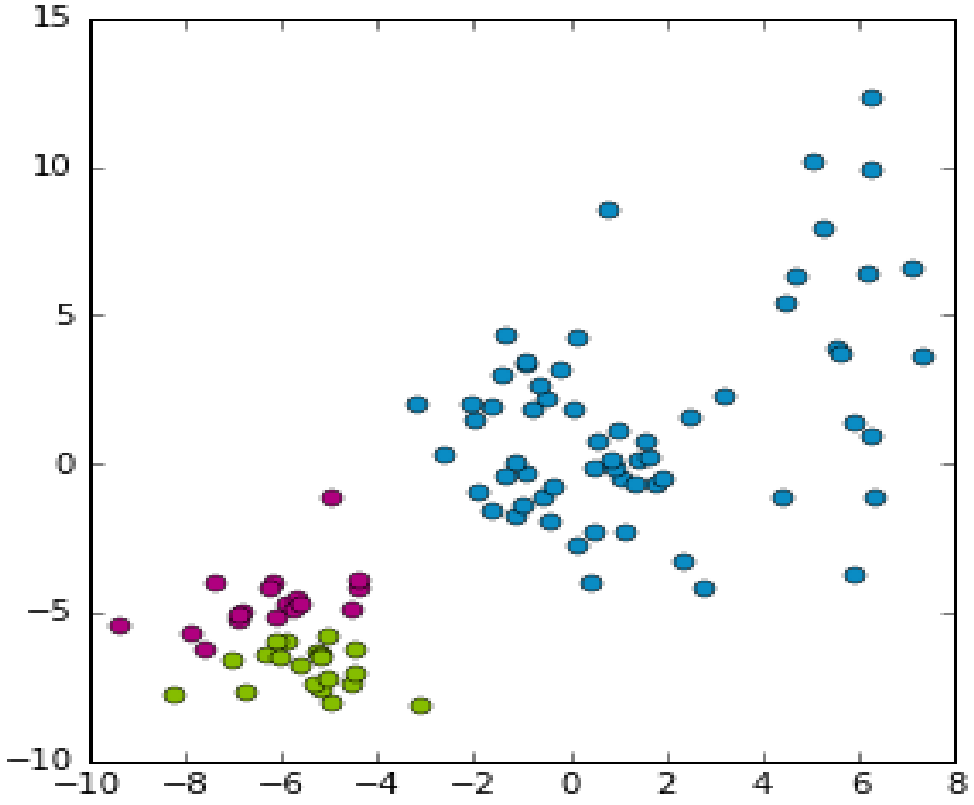
https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput695/F07/exercises/Exercises695Clus-solution.pdf

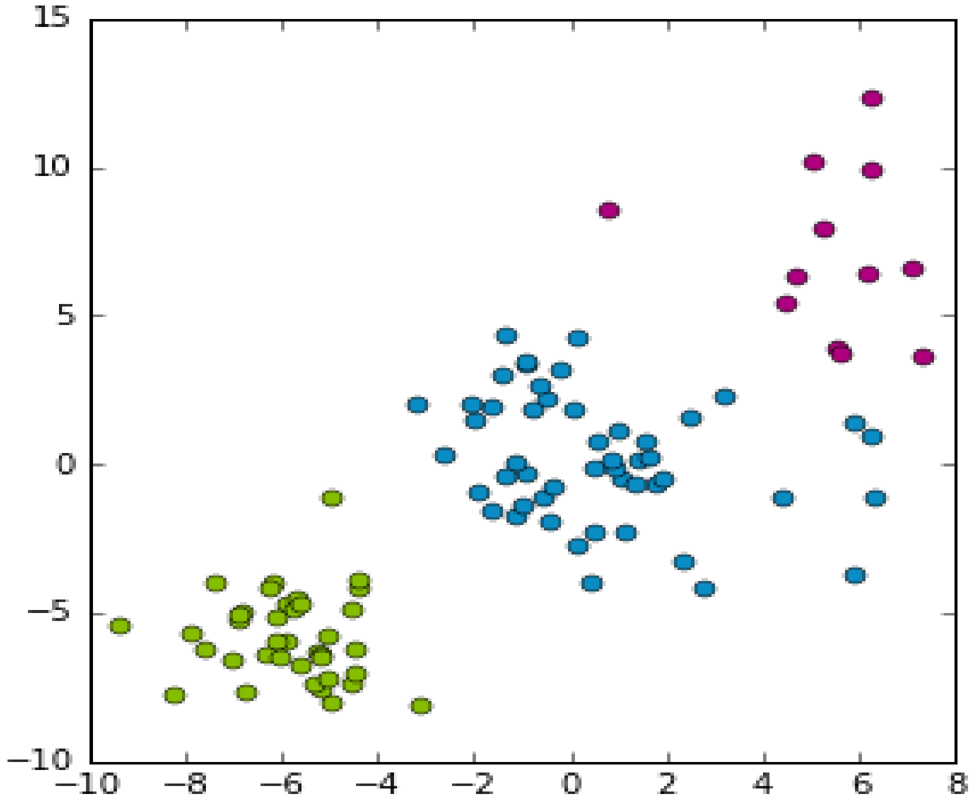
K-means

https://erogol.com/how-k-means-clustering-works/

Assessing quality of the clustering

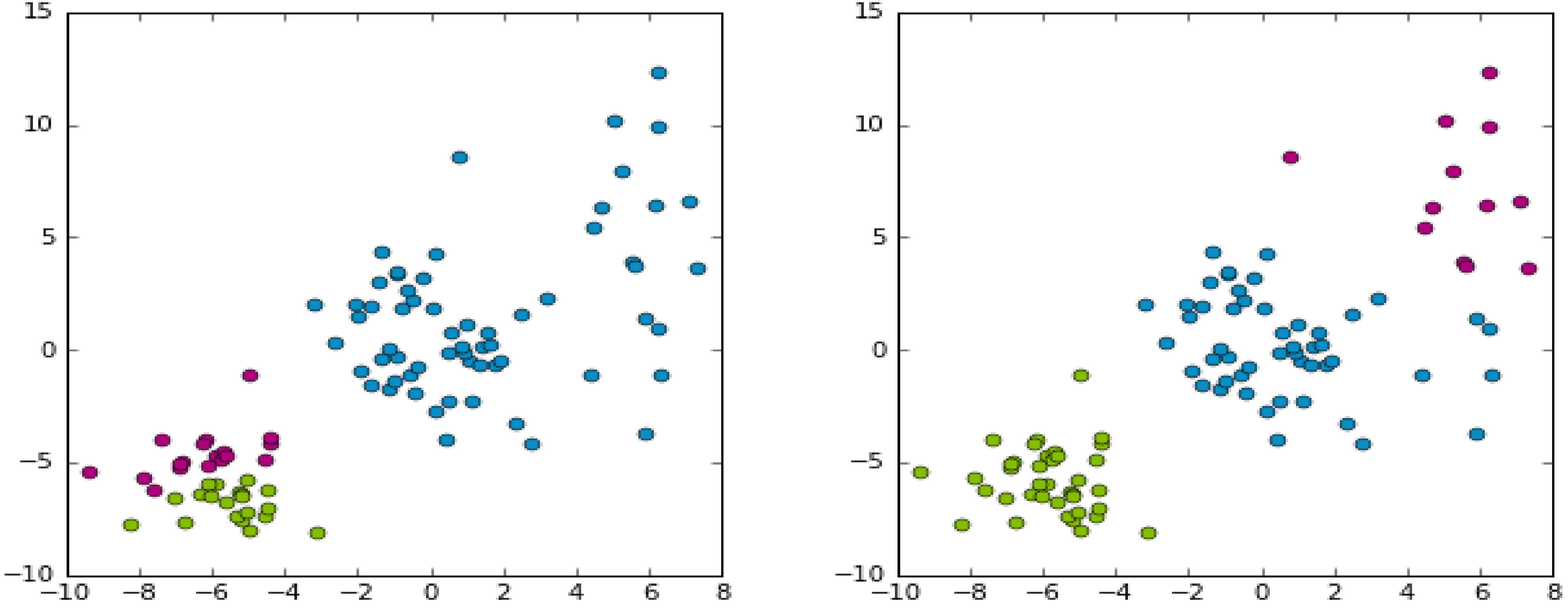
- Which cluster is **better**?





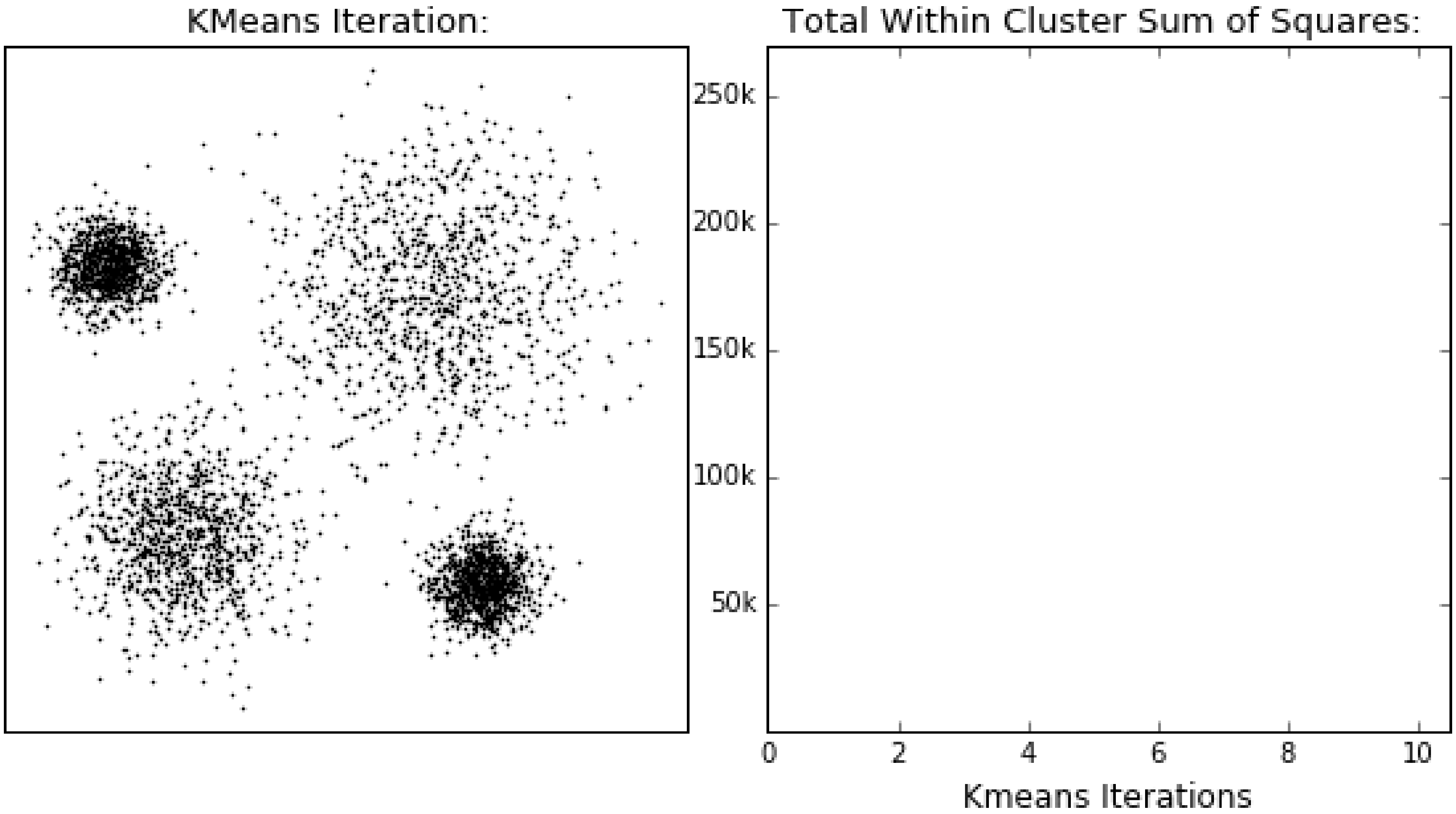
SSE

- k-means is trying to minimize the sum of squared distances:



$$\sum_{j=1}^{k} \sum_{i:z_i=j} ||\mu_j - \mathbf{x}_i||_2^2$$

Clustering: K-means Algorithm



https://giphy.com/gifs/spot-dashee87githubio-scikit-12vVAGkaqHUqCQ

Clustering: K-means Algorithm

Input:

- K (number of clusters)
- Training set $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$ (here we drop $x_0 = 1$)

Algorithm:

- Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$
- Repeat

{ for i=1 to m

$c^{(i)}$=index (from 1 to K) of cluster centroid closest to $x^{(i)}$

Calculated as : $\min_k \left\| x^{(i)} - \mu_K \right\|^2$

for k=1 to K

μ_K= average of points assigned to cluster k

}

Formalization

Representative-based Clustering

Given a dataset with n points in a d-dimensional space, $\boldsymbol{D} = \{\boldsymbol{x}_i\}_{i=1}^n$, and given the number of desired clusters k, the goal of representative-based clustering is to partition the dataset into k groups or clusters, which is called a *clustering* and is denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$.

For each cluster C_i there exists a representative point that summarizes the cluster, a common choice being the mean (also called the *centroid*) $\boldsymbol{\mu}_i$ of all points in the cluster, that is,

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x_j \in C_i} \boldsymbol{x}_j$$

where $n_i = |C_i|$ is the number of points in cluster C_i.

A brute-force or exhaustive algorithm for finding a good clustering is simply to generate all possible partitions of n points into k clusters, evaluate some optimization score for each of them, and retain the clustering that yields the best score. However, this is clearly infeasilbe, since there are $O(k^n/k!)$ clusterings of n points into k groups.

Data Mining and Machine Learning: Fundamental Concepts and Algorithms - dataminingbook.info (rpi.edu)

K-means Algorithm: Objective

The *sum of squared errors* scoring function is defined as

$$SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

The goal is to find the clustering that minimizes the SSE score:

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \{SSE(\mathcal{C})\}$$

K-means employs a greedy iterative approach to find a clustering that minimizes the SSE objective. As such it can converge to a local optima instead of a globally optimal clustering.

K-means Algorithm: Objective

K-means initializes the cluster means by randomly generating k points in the data space. Each iteration of K-means consists of two steps: (1) cluster assignment, and (2) centroid update.

Given the k cluster means, in the cluster assignment step, each point $\boldsymbol{x}_j \in \boldsymbol{D}$ is assigned to the closest mean, which induces a clustering, with each cluster C_i comprising points that are closer to $\boldsymbol{\mu}_i$ than any other cluster mean. That is, each point $\boldsymbol{x}_j$ is assigned to cluster C_{j^*}, where

$$j^* = \arg\min_{i=1}^{k} \left\{ \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|^2 \right\}$$

Given a set of clusters C_i, $i = 1, \ldots, k$, in the centroid update step, new mean values are computed for each cluster from the points in C_i.

The cluster assignment and centroid update steps are carried out iteratively until we reach a fixed point or local minima.

K-Means Algorithm

K-means $(\boldsymbol{D}, k, \epsilon)$:

$t = 0$

Randomly initialize k centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t \in \mathbb{R}^d$

repeat

  $t \leftarrow t + 1$

  $C_j \leftarrow \emptyset$ for all $j = 1, \cdots, k$

  `// Cluster Assignment Step`

  foreach $\boldsymbol{x}_j \in \boldsymbol{D}$ **do**

    $j^* \leftarrow \arg\min_i \left\{ \|\boldsymbol{x}_j - \boldsymbol{\mu}_i^t\|^2 \right\}$ `// Assign` $\boldsymbol{x}_j$ `to closest centroid`

    $C_{j^*} \leftarrow C_{j^*} \cup \{\boldsymbol{x}_j\}$

  `// Centroid Update Step`

  foreach $i = 1$ *to* k **do**

    $\boldsymbol{\mu}_i^t \leftarrow \frac{1}{|C_i|} \sum_{\boldsymbol{x}_j \in C_i} \boldsymbol{x}_j$

until $\sum_{i=1}^k \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

Data Mining and Machine Learning: Fundamental Concepts and Algorithms - dataminingbook.info (rpi.edu)

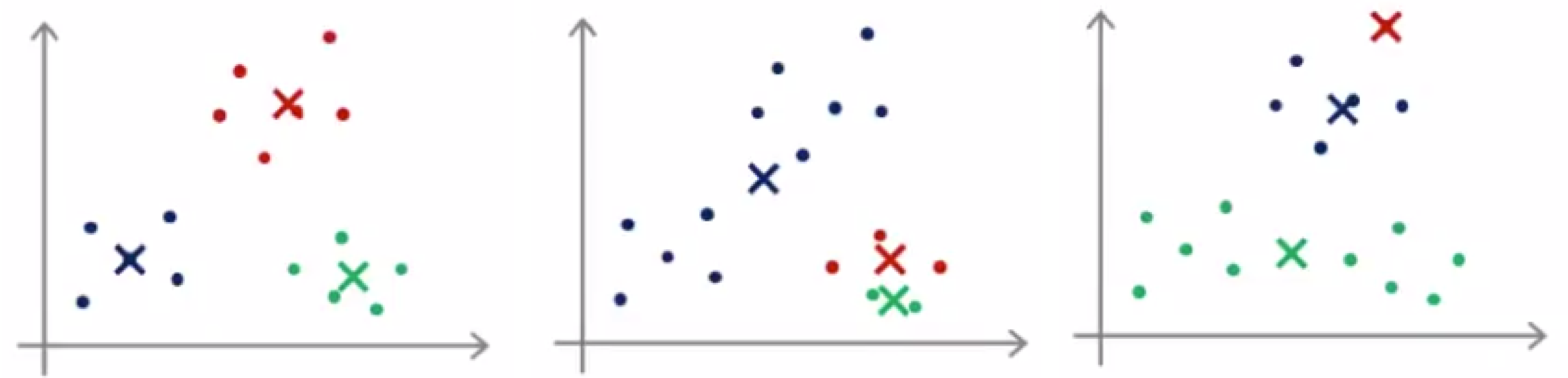
Limitation

K-means Algorithm: Random Initialization

How to randomly initialize cluster centroids?

1. Choose K<m
2. Randomly pick K training examples
3. Set $\mu_1, \ldots, \mu_K$ equal to these K examples

Seems correct, but will it always work?



No, because K-means can get stuck at different local optimas

K-means Algorithm: Random Initialization

Solution:

Instead of initializing K-means once and hoping that it works;

- ➢ Initialize and run K-means many times, and use the solution that gives best local or global optima as possible.

So we do the following:

for i=1 to 100

{ Randomly initialize K-means

Run K-means to get $c^{(1)}, \ldots c^{(m)}, \mu_1, \ldots, \mu_K$

Compute cost function (distortion) $J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K\right)$

}

- ➢ **Pick clustering that gave lowest cost** $J\left(c^{(1)}, \ldots, c^{(m)}, \mu_1, \ldots, \mu_K\right)$

Dependent on initial values.

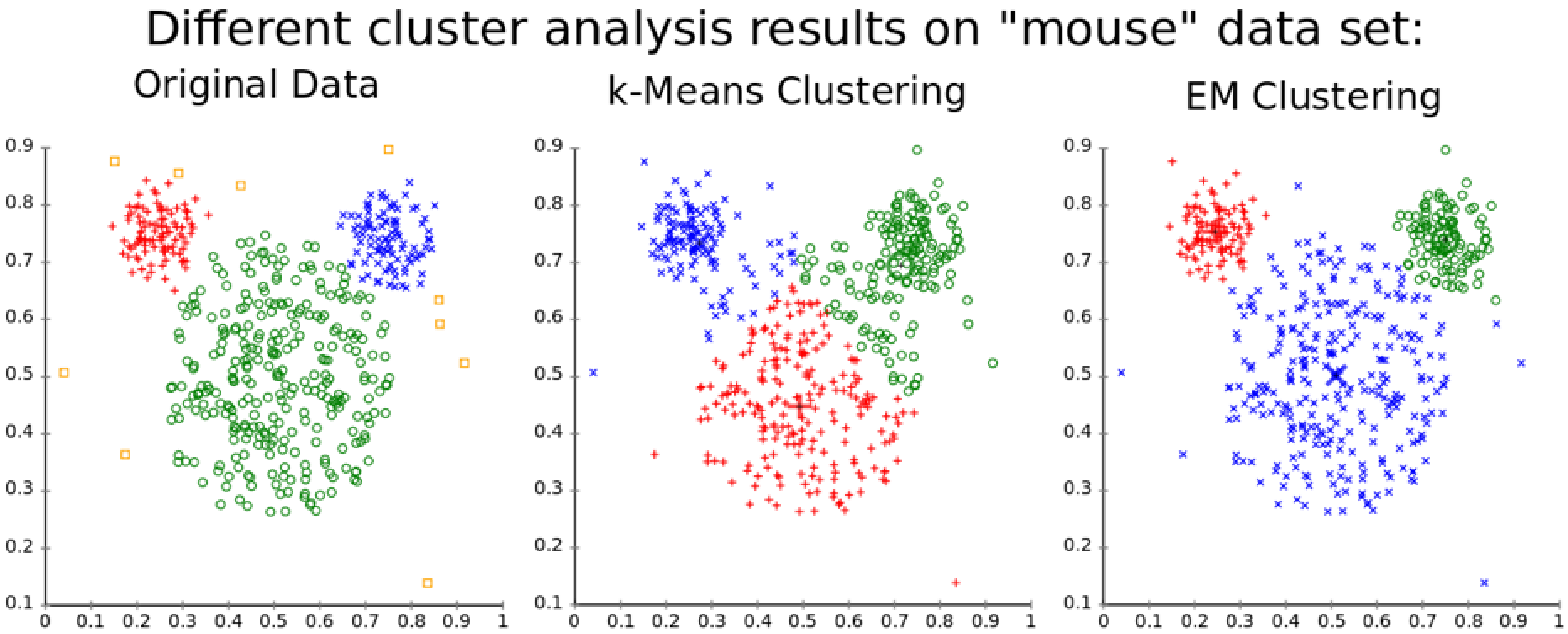
- running k-means several times with different initial values and picking the best result. As increases, you need advanced versions of k-means to pick better values of the initial centroids (called k-means seeding or kmeans++)

- A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm by M. Emre Celebi, Hassan A. Kingravi, Patricio A. Vela.

Smart initialization with k-means++

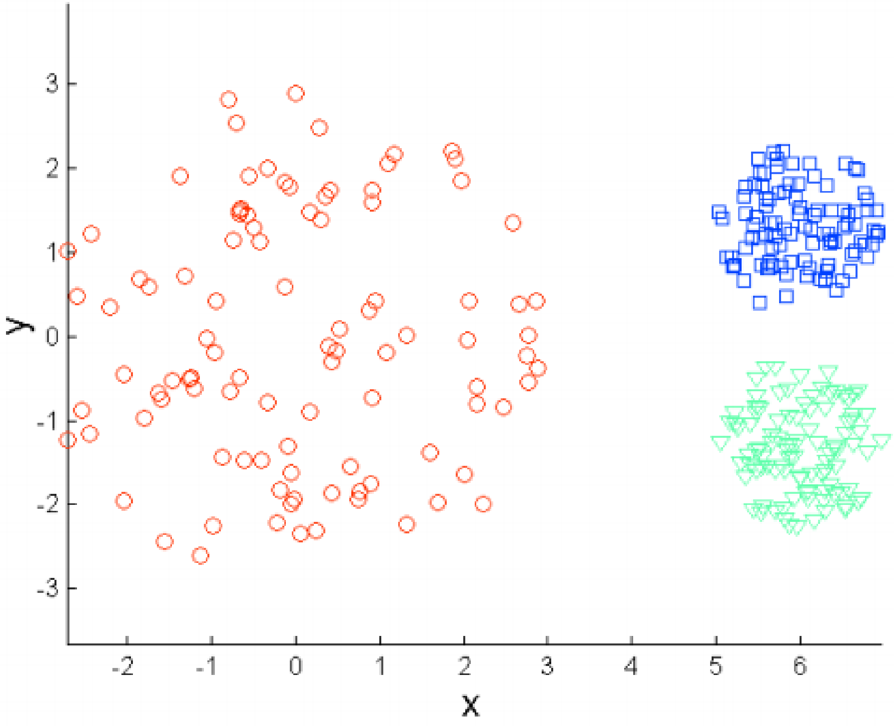
- Initialization of k-means algorithm is critical to quality of local optima found
- Smart initialization:
    - Choose first cluster center uniformly at random from data points
    - For each obs x, compute distance d(x) to nearest cluster center
    - Choose new cluster center from amongst data points(the furthest data points).
    - Repeat Steps 2 and 3 until k centers have been chosen
- Cons: Computationally costly relative to random initialization, but the subsequent k-means often converges more rapidly
- Pros: Tends to improve quality of local optimum and lower runtime.
- Used in sk-learn library

Spherical and equally sized clusters

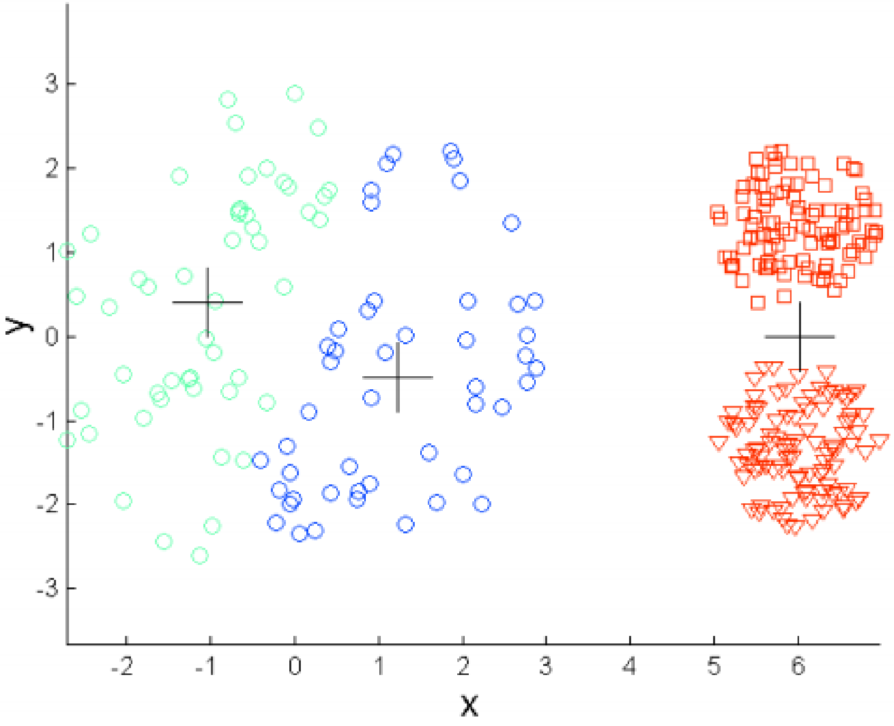


https://www.inovex.de/blog/disadvantages-of-k-means-clustering

Spherical and equally sized clusters



Original Points

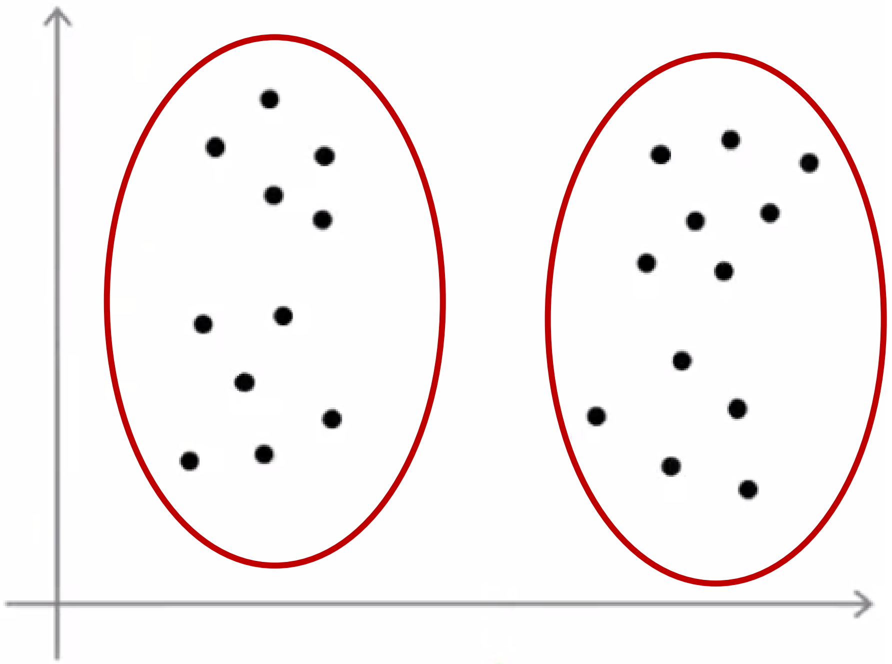


K-means (3 Clusters)

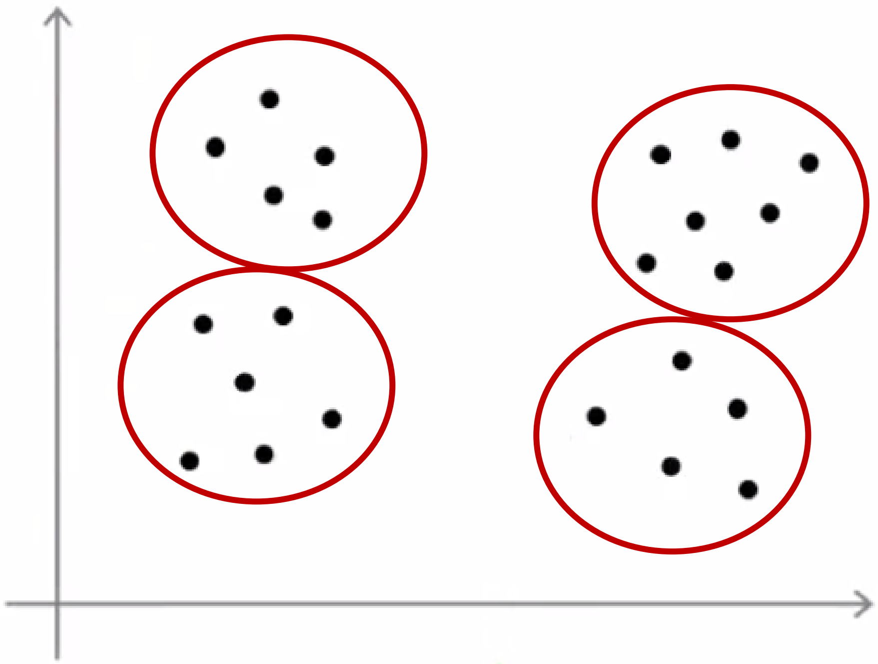
[Microsoft PowerPoint - ClusteringAnalysis.pptx (wmich.edu)]

K-means Algorithm: Choosing number of clusters

K=2



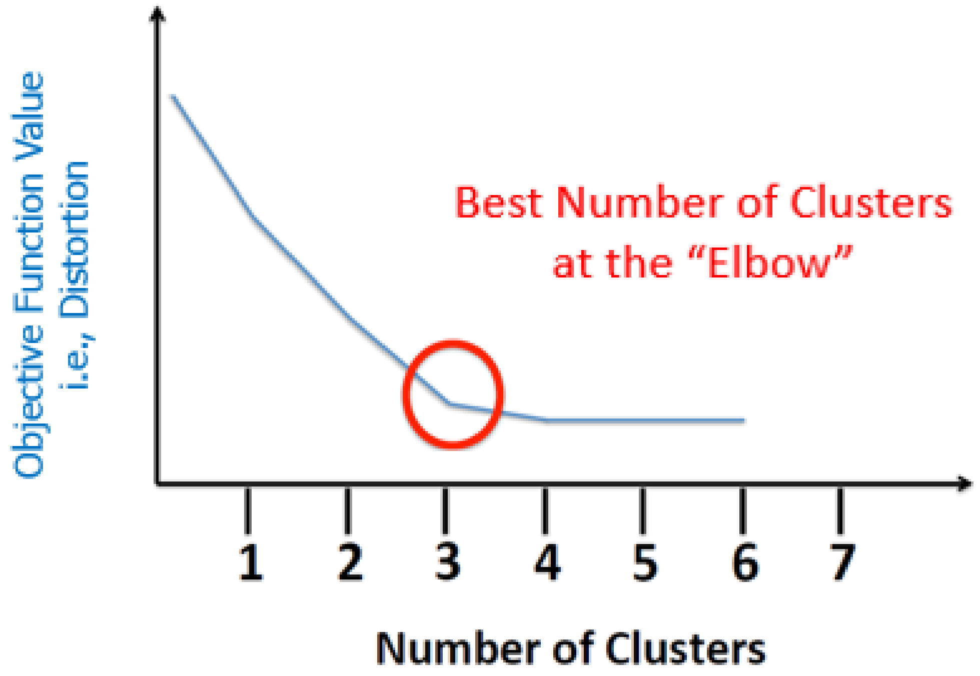
K=4



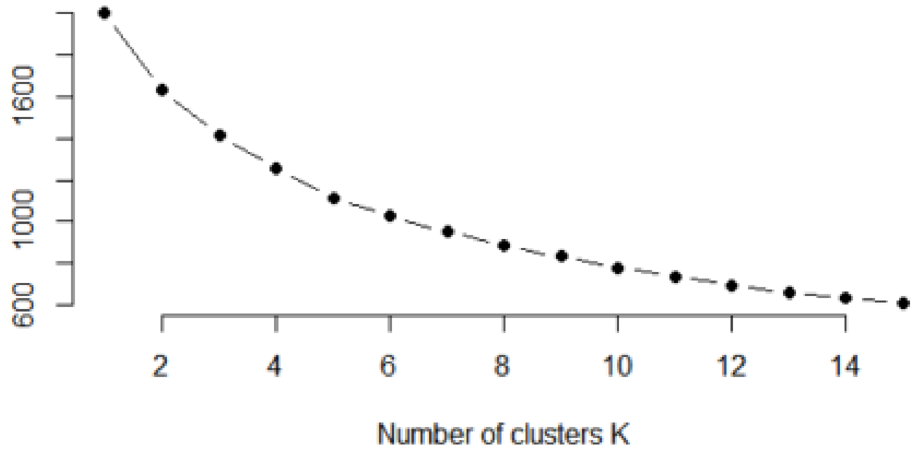
K-means Algorithm: Choosing number of clusters

One possible way is using Elbow Method

- Plotting cost function J verses number of clusters

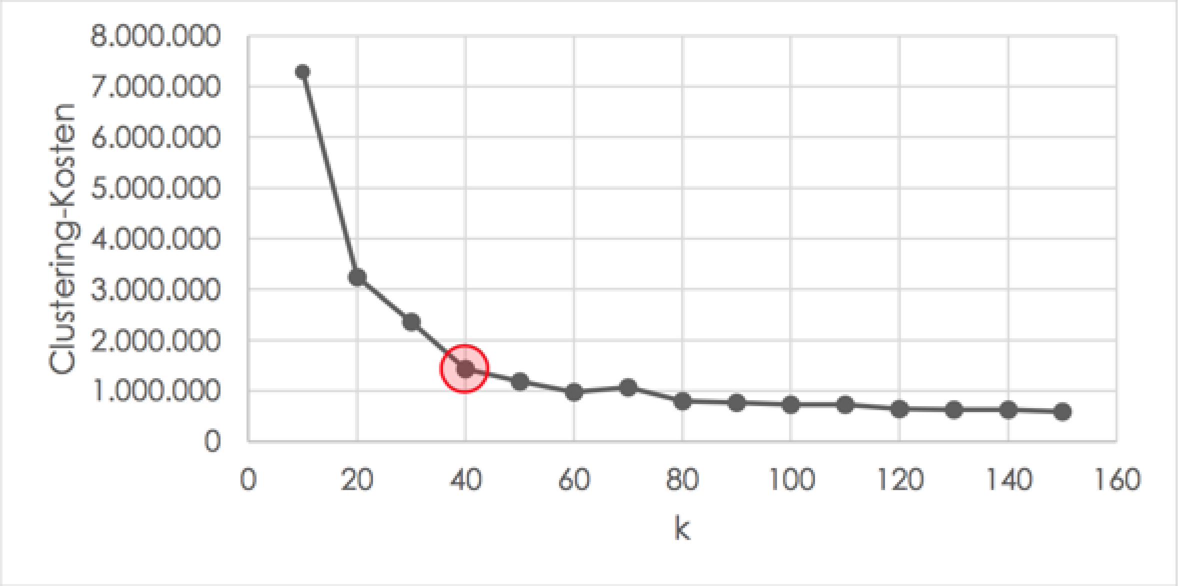


But, you may not always find an "elbow"…



Choosing an appropriate k

- Using the elbow method we run k-means clustering for a range of values of k. (e.g. 1 to 150). For each value of k we then compute the sum of squared errors (SSE) and add both into a line plot. Illustration 1 shows an exemplary curve of a range of values of k and the corresponding SSE.



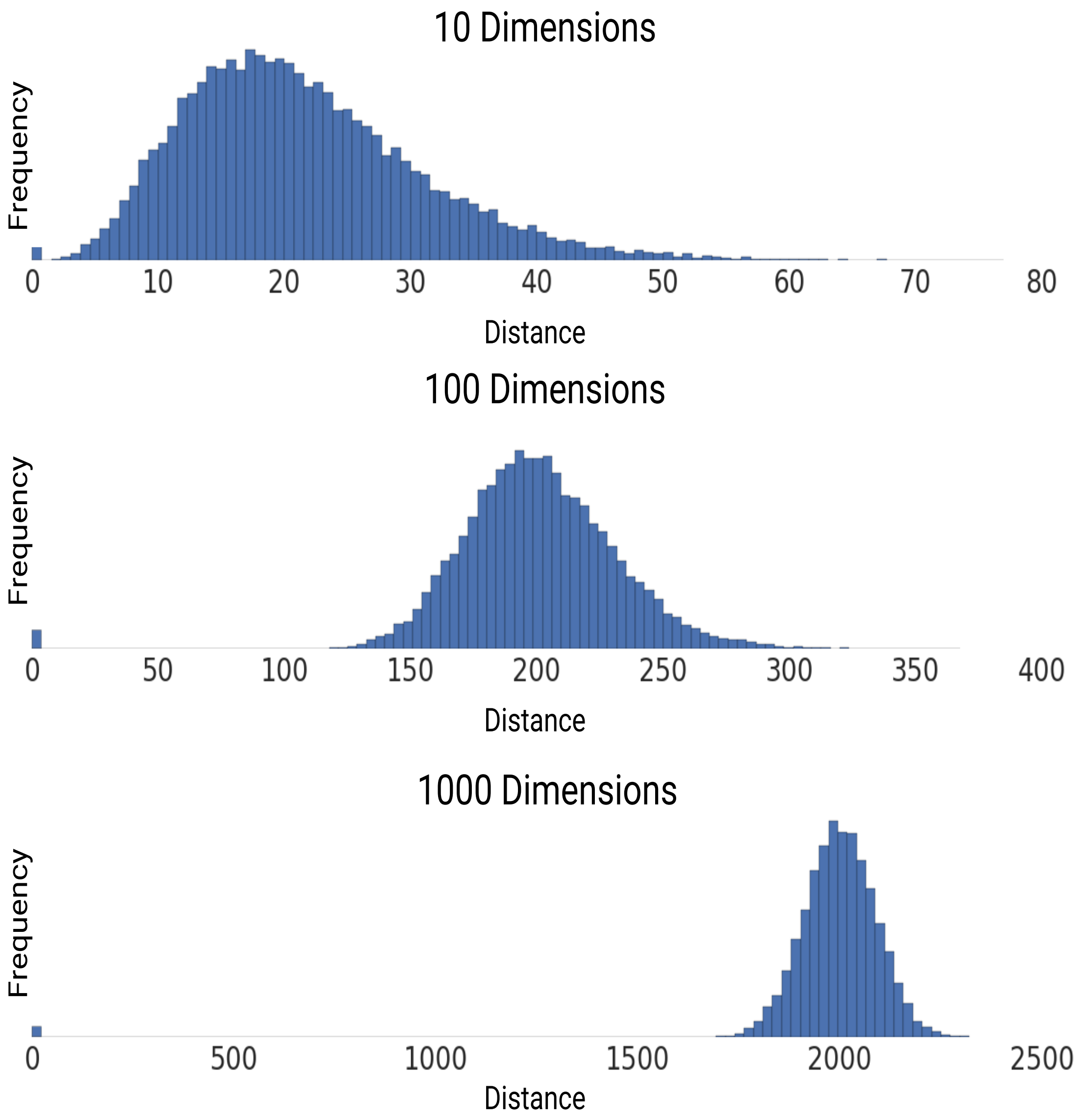
https://www.inovex.de/blog/disadvantages-of-k-means-clustering

Clustering outliers

- Centroids can be dragged by outliers, or outliers might get their own cluster instead of being ignored. Consider removing or clipping outliers before clustering.

Curse of Dimensionality

- These plots show how the ratio of the standard deviation to the mean of distance between examples decreases as the number of dimensions increases. This convergence means k-means becomes less effective at distinguishing between examples. This negative consequence of high-dimensional data is called the curse of dimensionality.



https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

Voronoi tessellation