HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Introduction to
Machine Learning and Data Mining

IT3190

Lecture: Clustering

ONE LOVE. ONE FUTURE.

Contents

- Lecture 1: Introduction to Machine Learning & Data Mining
- Lecture 2: Data crawling and pre-processing
- Lecture 3: Linear regression
- Lecture 4: Decision tree and Random forest
- Lecture 5: Neural networks
- Lecture 6: Support vector machines
- Lecture 7: Performance evaluation
- **Lecture 8+9: Unsupervised learning - Clustering**
- Lecture 10: Probabilistic models
- Lecture 11: Basics of data mining
- Lecture 12: Association rule mining
- Lecture 13: Regularization and advanced topics

Unsupervised learning (clustering)

- Introduction to Clustering
- K-means
- Hirarchical clustering
- Some practices

Introduction to Clustering

- Overview
  - Basic learning problems
  - Clustering
  - Evaluation of clustering
- Clustering methods
  - Partition method
  - Hierarchical method
  - Density-based method
- Requirements and applications
  - Requirements for clustering methods
  - Applications

Basic learning problems

- Lean a function/model from a given dataset
- Supervised learning
  - $S = \{\{x_1, x_2, \ldots, x_N\}; \{y_1, y_2, \ldots, y_N\}\}$
  - Each training instance has a label/response
  - Target: $y_i \cong f(x_i)$ for every i. Predict classes or values of future data.
- Unsupervised learning
  - $S = \{x_1, x_2, \ldots, x_N\}$
  - No label/response for training instances
  - Target: $y = f(x)$. Detect hidden structures in data.

Clustering

- Clustering problem
  - Unsupervised learning
  - Input: a training set without any label.
  - Output: clusters of the training instances
- A cluster
  - Consists of similar instances in some senses.
  - Two clusters should be different from each other.

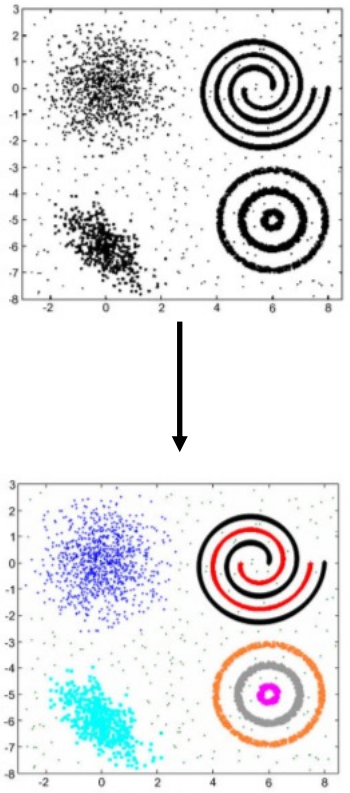
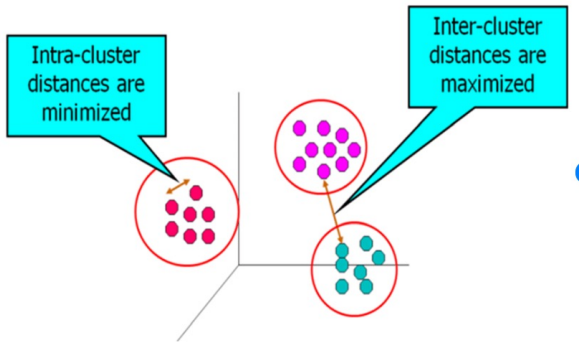


Figure 1: Clustering

Source: *Santosh K., Deepanwita D. and Sanjay K.S.. (2015). Black Hole Algorithm and Its Applications. Studies in CI. 575. 147-170.*

Clustering

- Inter-cluster distance
  - Distance/difference between any two clusters should be large.
- Intra-cluster distance
  - Distance/difference between instances inside a cluster should be small.



Source: *https://www.researchgate.net/figure/Intra-cluster-distances-vs-inter-cluster-distances_fig22_336111538*

- **Evaluation of clustering**
  - Sum-of-Squared error
    - Sum of squared distances between instances and cluster centers

$$E(C) = \sum_{i=1}^{k} \sum_{o \in C_i} d(o, cen_i)^2$$

  - Silhouette value
    - $a(o_i)$: distance $o_i \rightarrow o_j$ same cluster
    - $b(o_i)$: distance $o_i \rightarrow o_j$ different cluster
    - $sil(o_i) \in [-1, 1]$, $sil(o_i) \approx 1$: o_i is well classified

$$a(o_i) = \frac{1}{|C_A| - 1} \sum_{o_j \in C_A, o_j \neq o_i} d(o_i, o_j)$$

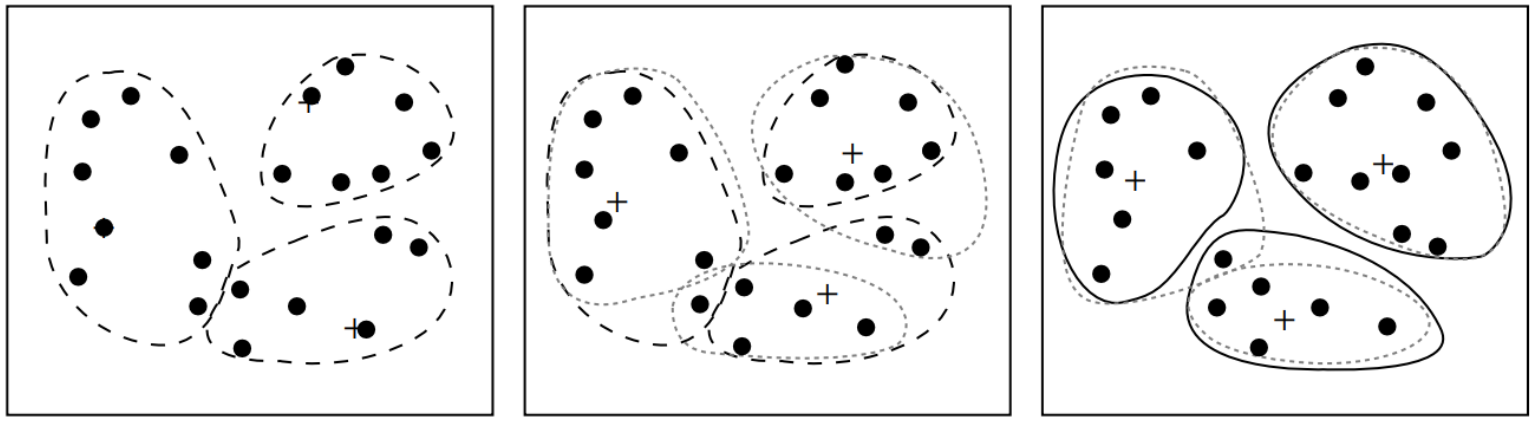
$$b(o_i) = min_{C_B \neq C_A} \frac{1}{|C_B|} \sum_{o_j \in C_B} d(o_i, o_j)$$

$$sil(o_i) = \frac{b(o_i) - a(o_i)}{max\{a(o_i), b(o_i)\}}$$

  - Extrinsic methods:
    - Comparing to a manual clustering (ground of truth)

Partition method (Centroid-based)

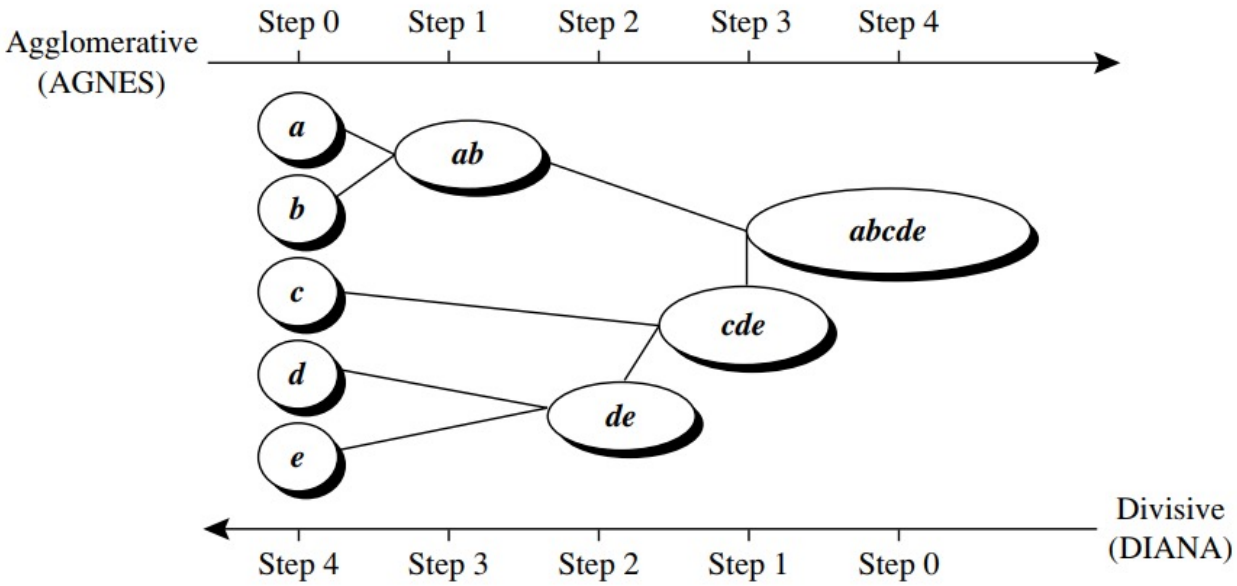
- Example of algorithm: K-means



(a) Initial clustering (b) Iterate (c) Final clustering

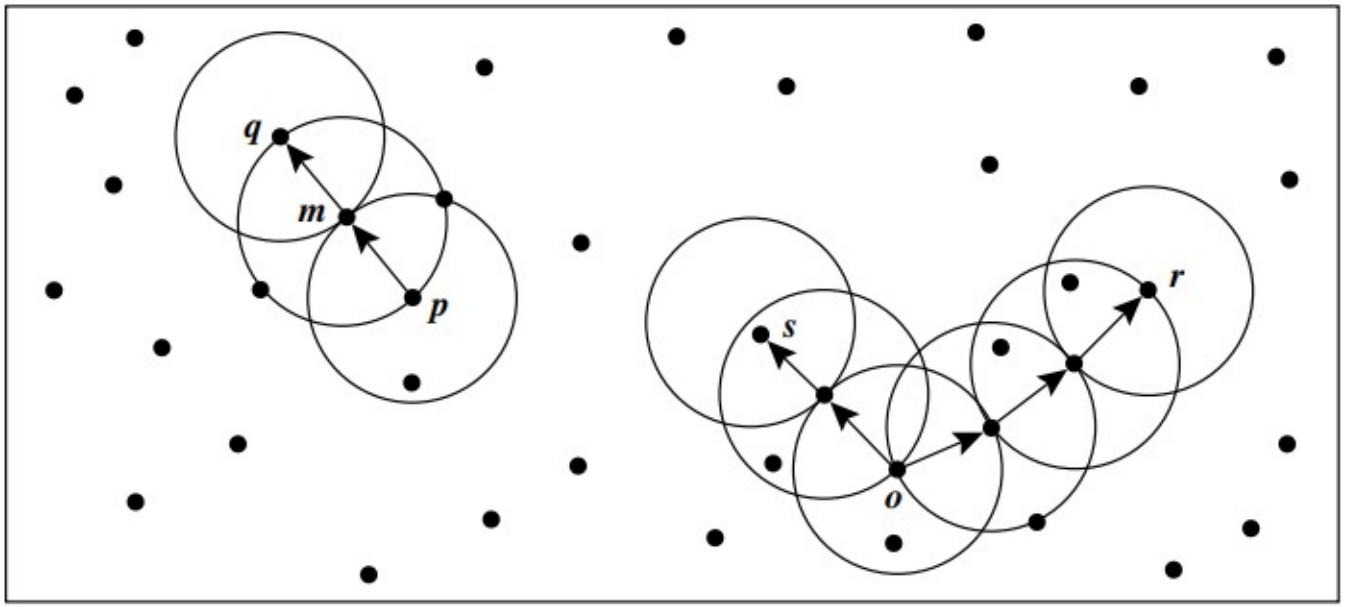
Hierarchical method (Connectivity-based)

- Example of algorithm: BIRCH



Density-based method

- Example of algorithm: DBSCAN



Requirements for clustering methods

- Scalability: Work with small/large databases
- Ability to deal with different types of attributes: Work with numeric, categorical, mixtures or complex data types (graphs, images, text,...).
- Discovery of clusters with arbitrary shape: spherical, not spherical, intersecting data...
- Ability to deal with noisy data: datasets contain outliers, missing, unknown, or erroneous data.

Introduction to Clustering

Applications

- Market segmentation
- Social network analysis
- Wireless network analysis or Network traffic classification
- Image compression
- Data processing and feature weighing
- Life science and healthcare
- …

K-means

- K-means
  - Overview
  - Algorithm
  - Key ingredients
  - Some problems
- K-means++
  - Overview
  - Algorithm
  - Evaluation
- Online K-means
  - Serializing K-means
  - Algorithm

K-means - Overview

- K-means was first introduced by Lloyd in 1957.
- K-means is the most popular method for clustering, which is partition-based.
- Data representation:
  - $D = \{x_1, x_2, \ldots, x_r\}$, each instance x_i is a vector in the n-dimensional Euclidean space.
- K-means partitions D into k clusters:
  - Each cluster has a central point which is called centroid.
  - K is a pre-specified constant.

K-means - Algorithmn

- Input: training data *D*, number of clusters *k*, distance measure *dist*(x,y).

1. Initialization: select randomly *k* initial centroids $C = \{c_1, \ldots, c_k\}$
2. For each instance $x \in D$
   - Assign *x* to the cluster with nearest centroid
3. For each cluster $i \in \{1, \ldots, k\}$:
   - Recompute the centroid of cluster *i* from its instances
4. Repeat Steps 2 and 3 until *C* no longer changes.

K-means - Algorithmn

- Assign *x* to the cluster with nearest centroid
  - Compute distance from *x* to each centroid: $dist(x, cen_i)$
  - Assign *x* to cluster of closest centroid (C_i):

$$\text{j} = \underset{i \in \{1,2,\ldots k\}}{arg\,min}\, dist(x, cen_i)$$

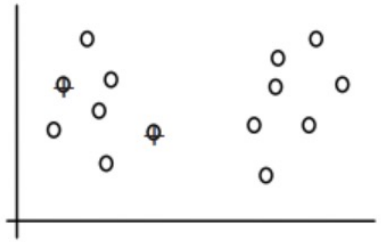
$$C_j \leftarrow x$$

- Recompute the centroid of cluster *i* from its instances

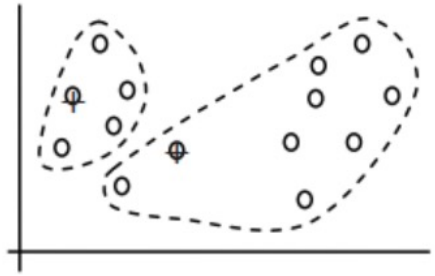
$$cen_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

  - Sum of instances: adding the corresponding dimensions of the vectors (vector addition)

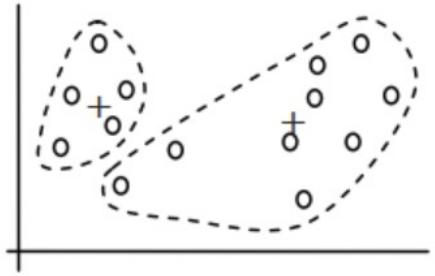
K-means - Algorithmn



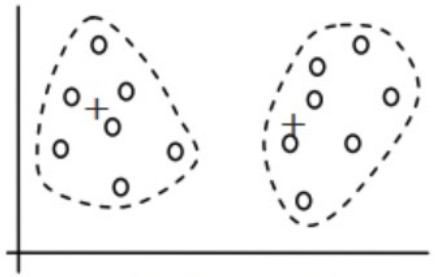
(A). Random selection of k seeds (or centroids)



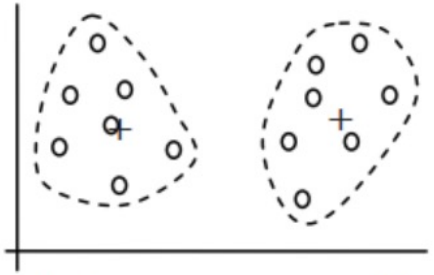
Iteration 1: (B). Cluster assignment



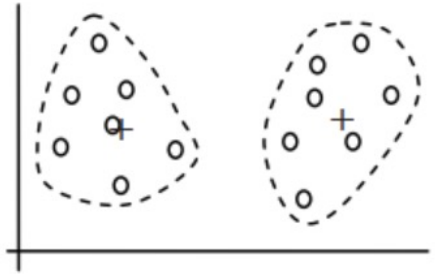
(C). Re-compute centroids



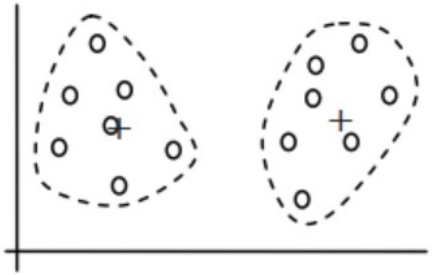
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

K-means - Key ingredients

- The algorithm converges if:
  - Very few instances are reassigned to new clusters, or
  - The centroids do not change significantly, or
  - The sum of squared error does not change significantly

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, cen_i)^2$$

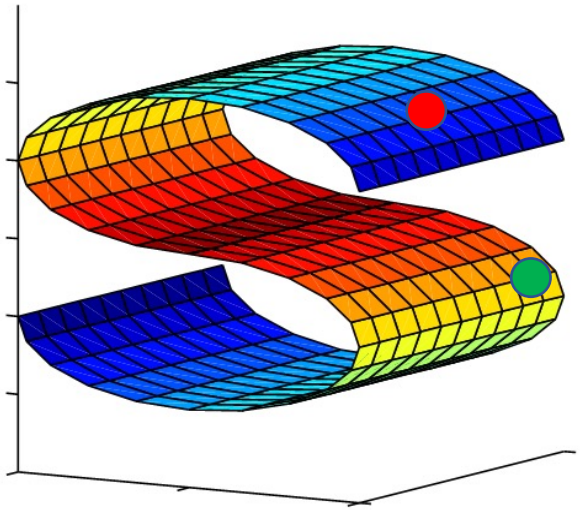
where cen_i is the centroid of cluster C_i.

K-means - Key ingredients

- Distance measures
  - Manhattan (L1-norm): $d(x, z) = \sum_{i=1}^{n} |x_i - z_i|$
  - Euclid (L2-norm): $d(x, z) = \sqrt{\sum_{i=1}^{n} (x_i - z_i)^2}$
  - Hamming distance: $d(x, z) = \sum_{i=1}^{n} Difference\ (x_i, z_i)$
  - Other measures are possible.

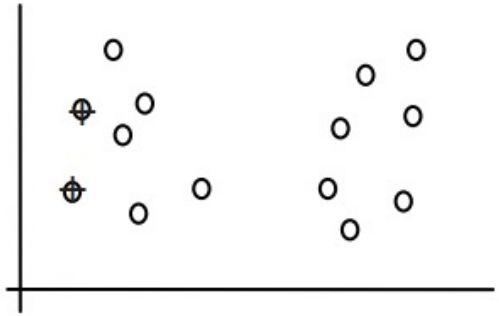
K-means - Key ingredients

- Distance measures
  - Feature
    - Each measure provides a view on data
    - Which distance is good?
  - Similarity measures can be used
    - Similarity between two objects

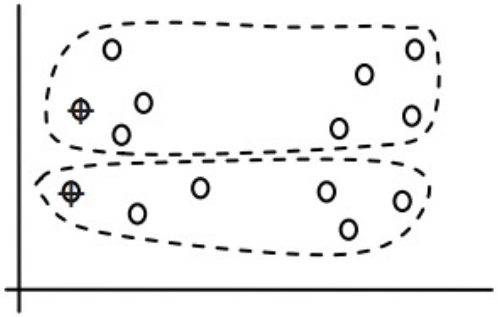


K-means – Some problems

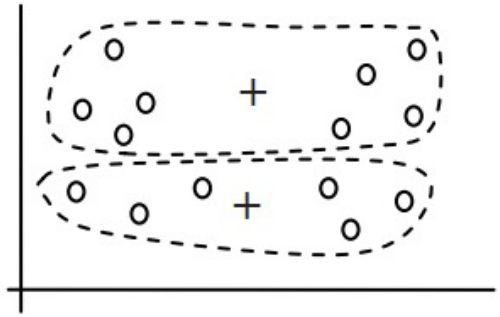
- Quality of K-means depends much on the initial centroids



(A). Random selection of seeds (centroids)



(B). Iteration 1



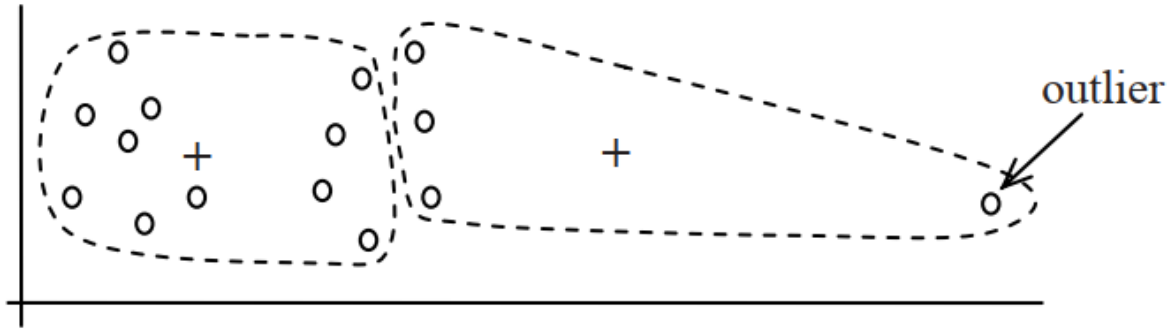
(C). Iteration 2

K-means – Some problems

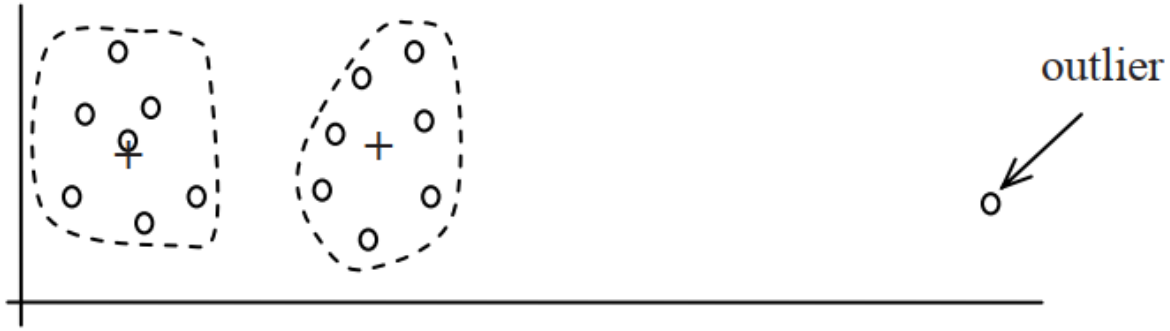
- **Solutions**
  - Repeat K-means many times. Each time initialize a different set of centroids.
  - After learning, we combine results from those runs to obtain a unified clustering.
  - K-means++ (next session)

K-means – Some problems

- Some outliers in data can affect quality of K-means



(A): Undesirable clusters



(B): Ideal clusters

K-means – Some problems

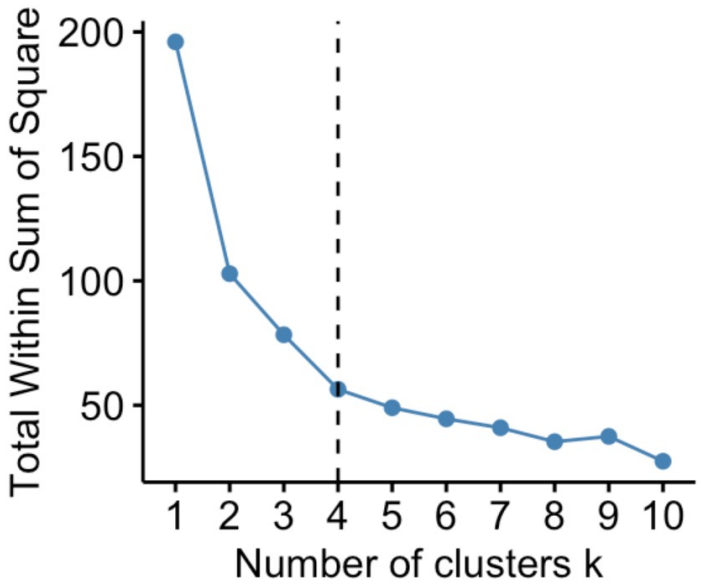
- **2.2. Solutions**
  - Outlier removal
    - Remove instances that are far from the centroids
    - Removal can be done a priori or when learning clusters.
  - Random sampling
    - Cluster a random sample *S* instead of the whole training data
    - *S* contains fewer noises/outliers than the original training data.
    - After learning, the remaining data will be assigned to the learned clusters.

K-means – Some problems

- K-means requires the user to specify the number of clusters
- Optimal number of clusters is subjective and depends on the distance measures

K-means – Some problems

- Elbow method
  - Run K-means different values of k
  - For each k, calculate the total within-cluster sum of square (S).
  - Plot the curve of S according to the number of clusters k.
  - The location of a bend (elbow) in the plot.



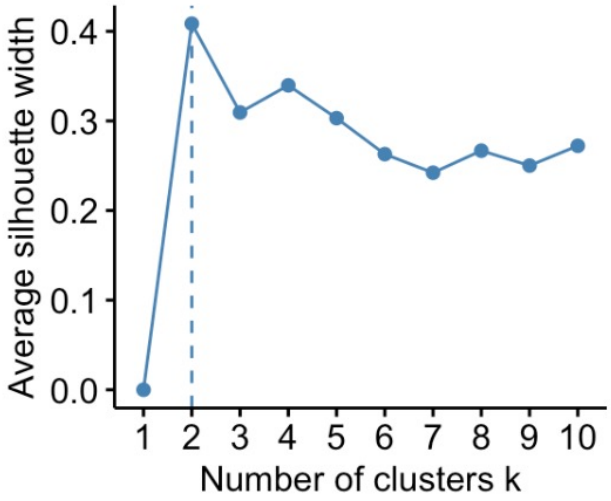
Source: *https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/*

K-means – Some problems

- Average silhouette method

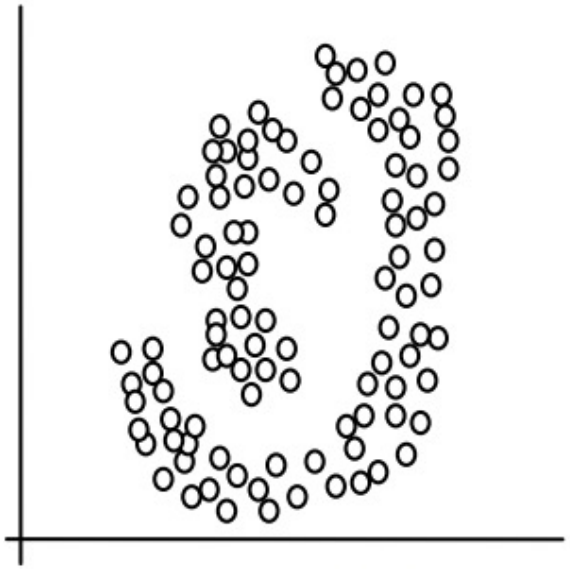
$$SC(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

  - *a(i)*: average distance between sample *i* and other samples in the same cluster.
  - *b(i)*: minimum average distance between sample *i* and samples in other clusters.
  - *SC* $\in$ [-1, 1], *SC* getting closer to 1 indicates good cluster of *i*.

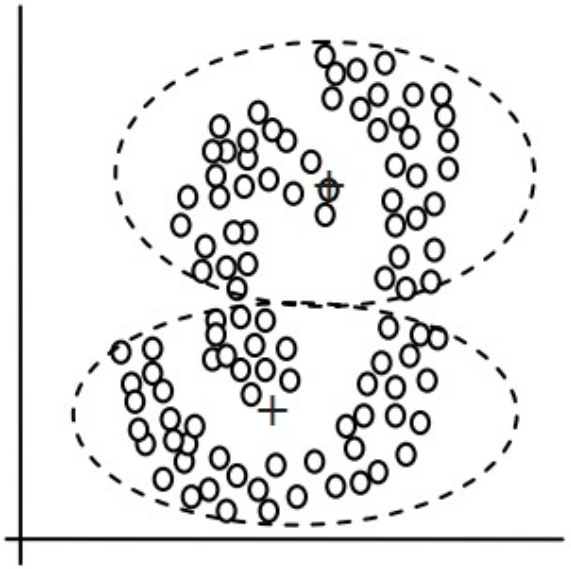


K-means – Some problems

- Data with varying shapes



(A): Two natural clusters



(B): *k*-means clusters

Source: *https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/*

K-means++

- **K-means++ idea**
  - Problem on K-means initialization: Quality of K-means depends much on the initial centroids
  - Initial centroids that are far away from one another.
    - This increases the chances of centroids that lie in different clusters.
    - Since centroids are picked up from the data points, each centroid has some data points associated with it at the end.

K-means++

Algorithmns

- Update initialization step of K-means:
  - 1. Choose an initial centroid c_1 uniformly at random from X .
  - 2. Choose the next centroid c_i:
    - Select $c_i = x' \in X$ with probability $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$
    - D(x) denote the shortest distance from x to the closest centroid
  - 3. Repeat Step 2 until a total of k centers has been chosen.

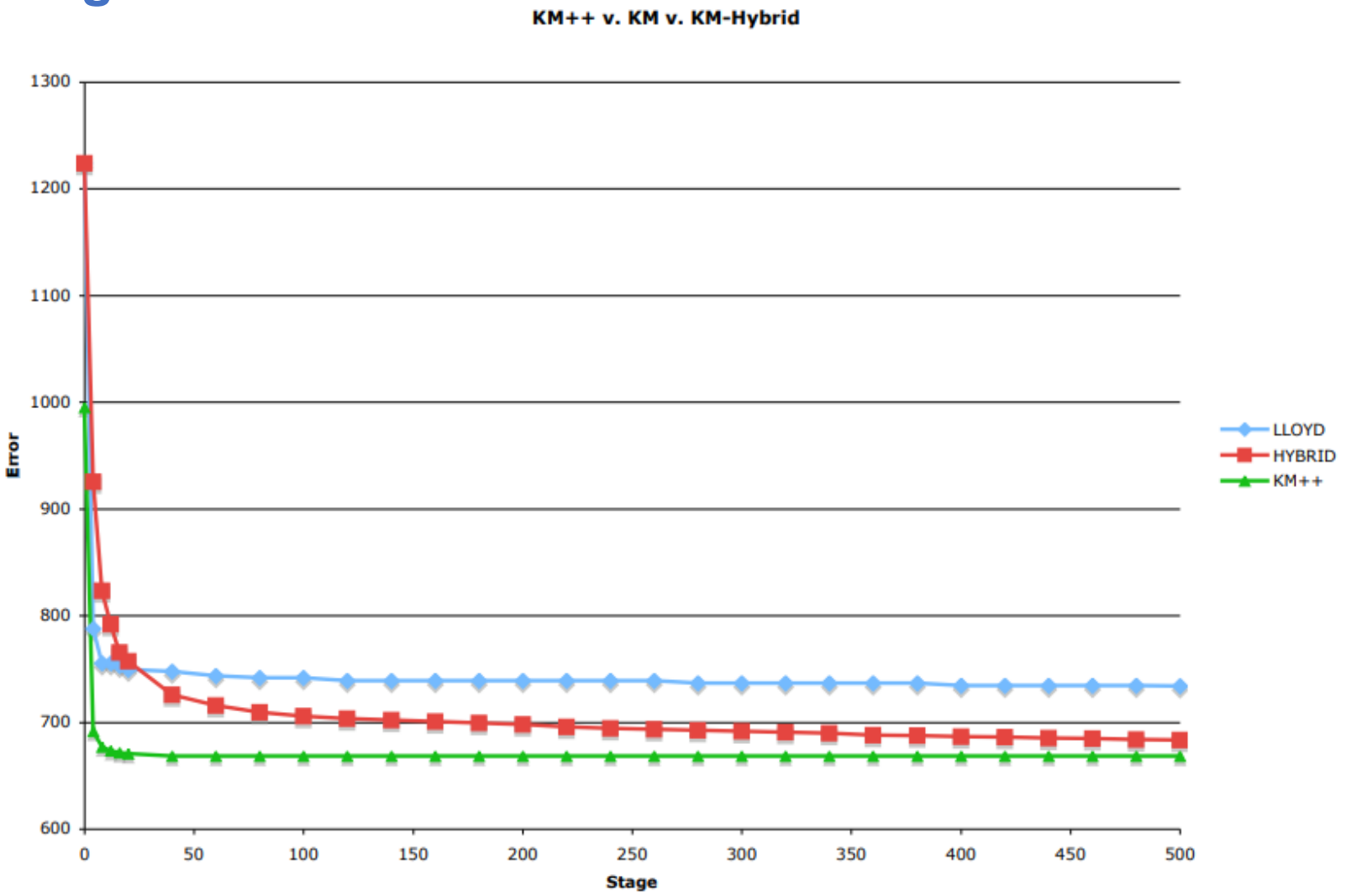
K-means++

Algorithm interpretation

- $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ is directly proportional to data point distance to the nearest centroid.
- The point having maximum distance is most likely to be selected
- This increase the chances of picking up centroids that lie in different clusters
- The centroids are selected with some degree of randomness

K-means++

Convergence



Source: *https://theory.stanford.edu/~sergei/slides/BATS-Means.pdf*

Clustering quality

k	Average ϕ		Minimum ϕ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	$1.365 \cdot 10^5$	8.47%	$1.174 \cdot 10^5$	0.93%	0.12	46.72%
25	$4.233 \cdot 10^4$	99.96%	$1.914 \cdot 10^4$	99.92%	0.90	87.79%
50	$7.750 \cdot 10^3$	99.81%	$1.474 \cdot 10^1$	0.53%	2.04	−1.62%

Experimental results on the Norm25 dataset (n = 1000, d = 15).

k	Average ϕ		Minimum ϕ		Average T	
	k-means	k-means++	k-means	k-means++	k-means	k-means++
10	$3.698 \cdot 10^4$	49.43%	$3.684 \cdot 10^4$	54.59%	2.36	69.00%
25	$3.288 \cdot 10^4$	88.76%	$3.280 \cdot 10^4$	89.58%	7.36	79.84%
50	$3.183 \cdot 10^4$	95.35%	$2.384 \cdot 10^4$	94.30%	12.20	75.76%

Experimental results on the Spam dataset (n = 4601, d = 58).

Source: *Arthur, D.; Vassilvitskii, S. (2007). K-means++: the advantages of careful seeding.*

Online K-means

Idea of serialization

- K-means:
  - K-means requires all training data for each iteration.
  - It cannot work with big datasets.
  - It cannot work with stream data where data come in sequence.
- Sum of squared error function:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, cen_i)^2 = \sum_{x \in D} dist(x, cen(\mathrm{x}))^2$$

  - *cen(x)* is the centroid of cluster containing *x*
  - K-means finds clusters that minimizes error value

Online K-means

- **1.2. Minimizing loss function**
  - K-means is an optimization problem with sum of squared error
    - Rewrite Loss function from Error function:

$$Q_{k-means} = \sum_{i=1}^{M} (x_i - w(x_i))^2$$

    - where $w(x_i)$ is the nearest centroid to x_i, *with M* instances $x_1, x_2, \ldots, x_M$
  - Q can be minimized by using Gradient descent:

$$w_{t+1} = w_t + \gamma_t \sum_{i=1}^{M} [x_t - w_t(x_i)]$$

    - where γ_t is a small constant, often called learning rate.

Online K-means

Stochastic gradient

- Note that each iteration of K-means requires the full gradient:
  - $Q'_t = \sum_{i=1}^{M} [x_i - w_t(x_i)]$
  - Which requires all training data.
- Online K-means minimizes Q using stochastic gradient:
  - Use only information of a traning instance *x* from the whole gradient Q'. $x_t - w_t(x_t)$:
    - $w_{t+1} = w_t + \gamma_t(x_t - w_t(x_t))$

Online K-means

Algorithms

- Initialize *k* centroids randomly.
- Update the centroids as an instance comes
  - At iteration *t*, take an instance x_t.
  - Find the nearest centroid w_t to x_t
  - Update w_t as follows:

$$w_{t+1} = w_t + \gamma_t(x_t - w_t(x_t))$$

Online K-means

Learning rate

- Learning rates are positive constants and satisfy:

$$\sum_{t=1}^{\infty} \gamma_t = \infty; \sum_{t=1}^{\infty} {\gamma_t}^2 < \infty$$

- A popular choice of learning rate:

$$\gamma_t = (t + \tau)^{-\kappa}$$

  - τ, κ are possitive constants.
- $\kappa \in (0.5, 1]$ is called forgeting rate.
  - Large κ means that the algorithm remembers the past longer, and new observation plays less important role as t grows.

Online K-means

Properties

- Online K-means is an online learning version of K-means.
- It follows the methodology from online learning and stochastic gradient.
  - One instance is used to update the available clusters at each iteration.
- Online K-means helps to cluster big/stream data.

Hirarchical clustering

Algorithms

- Two approaches
- Agglomerative algorithm

Cluster distance

- Single-Link Method
- Complete-Link Method
- Other methods

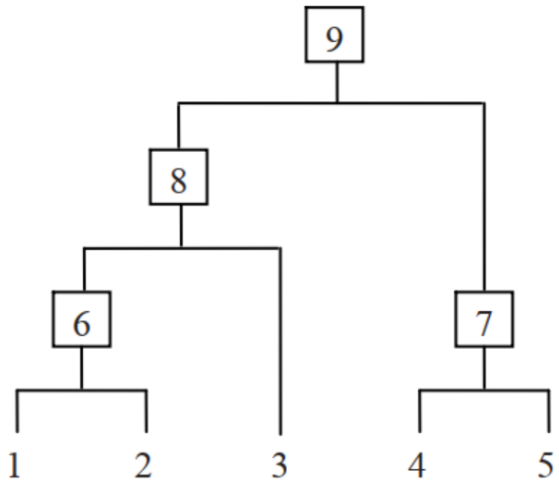
Strengths and Weaknesses

- Strengths
- Weaknesses

Hirarchical clustering

Two approaches

- Agglomerative (bottom up) clustering:
  - Build the dendrogram (tree) from the bottom level
  - Merge the most similar pair of clusters
  - Continue until all points are merged into one cluster
- Divisive (top down) clustering:
  - Start with all data points in one cluster (top level)
  - Split a cluster into smaller ones based on some criterion
  - Continue until each cluster is a single point.



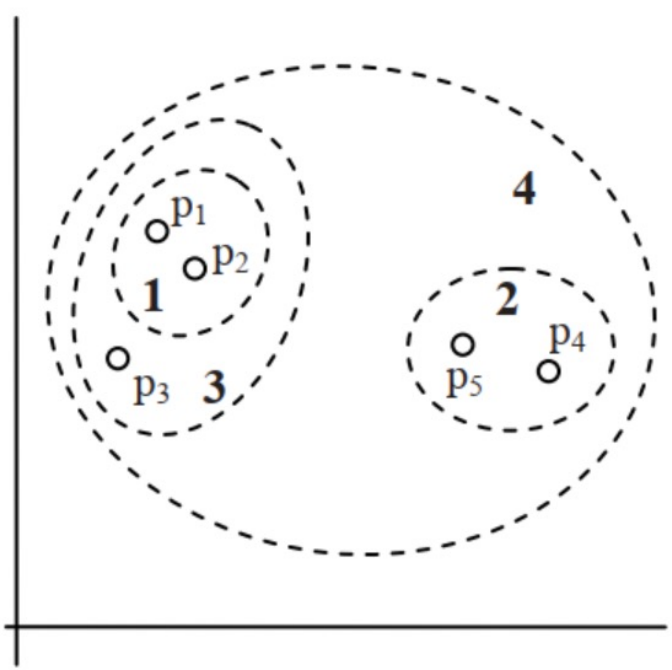
Hirarchical clustering

Agglomerative Algorithm

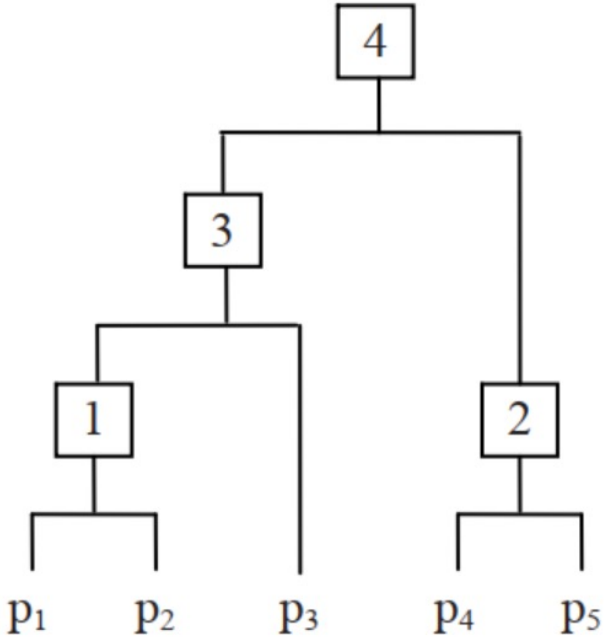
1. Make each data point in the data set D a cluster,
2. Compute all pair-wise distances of $x_1, x_2, \ldots, x_n \in D$
3. Repeat
    4. Find two clusters that are nearest to each other
    5. Merge the two clusters form a new cluster c
    6. Compute the distance from c to all other clusters
7. Until there is only one cluster left

Hirarchical clustering

Agglomerative Algorithm



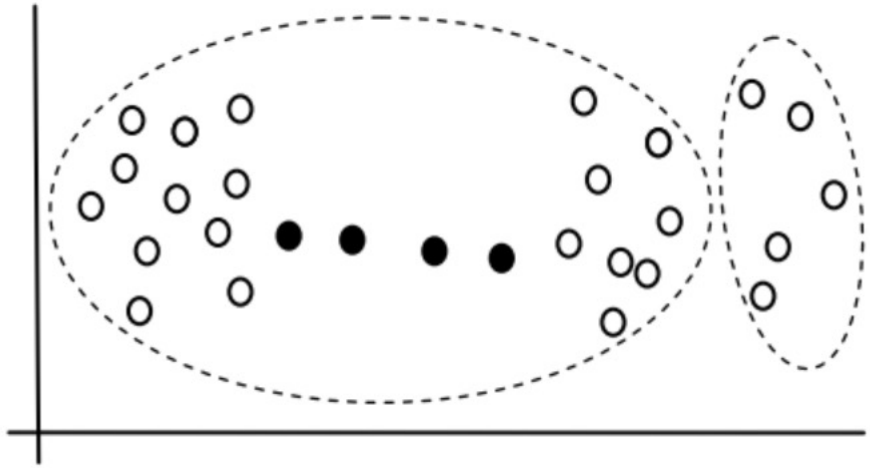
(A). Nested clusters



(B) Dendrogram

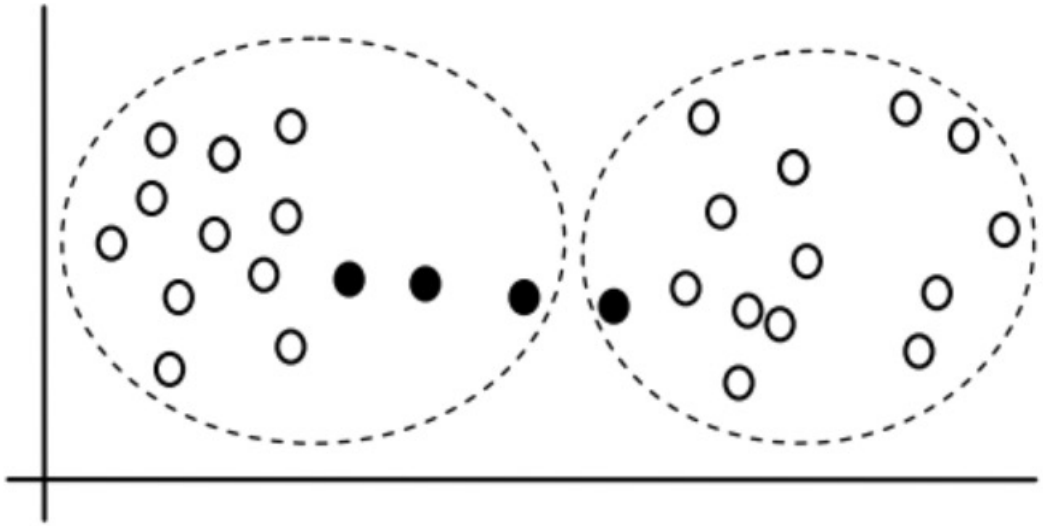
Single-Link Method

- Cluster distance: the distance between two closest points in the two clusters
- Merge the two clusters with the smallest minimum pair-wise distance.
- Can be sensitive to noise in the data (have chain effect)



Complete-Link Method

- Cluster distance: the distance between two furthest points in the two clusters
- Merge the two clusters with the smallest maximum pair-wise distance.
- Can be sensitive to noise (does not have the problem of chain effects)



Other Methods

- Average-Link Method: The distance between two cluster is the average distance of all pair-wise distances between two clusters.
- Centroid method: The distance between two clusters is the distance between their centroids.
- Ward's method: The distance between two clusters is the increase in the sum of squared error (distances) from that of two clusters to that of one merged cluster.

Hirarchical clustering

Strengths

- Can explore clusters at any level of detail
- Resulting hierarchy can be useful
  - Cluster hierarchy may represent a topic hierarchy in the documents.
- It can also find clusters of arbitrary shapes
  - E.g., using the single-link method.

Hirarchical clustering

Weaknesses

- The single-link method may suffer from the chain effect
- The complete-link method is sensitive to outliers
- Computation complexities and space requirements
  - They are at least quadratic

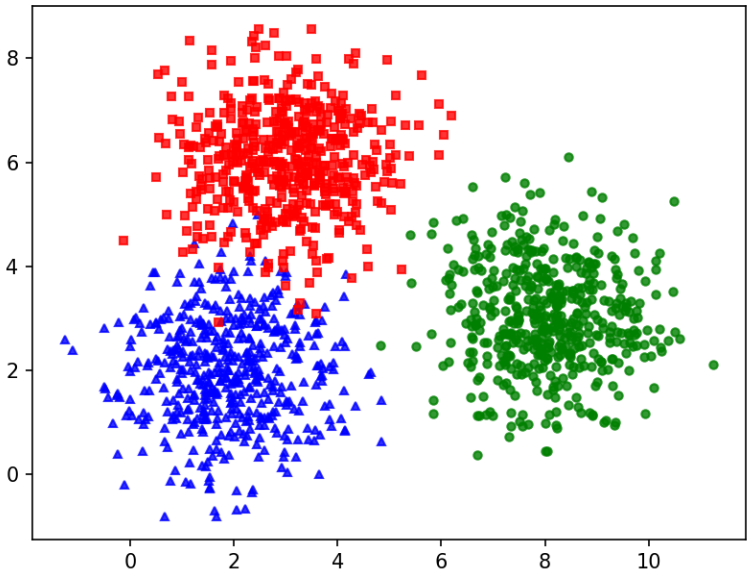
Some practices

Partition clustering (K-means)

Hirarchical clustering

Some practices

- Context
  - Data: synthetic data, 3 clusters, 500 2D points for each cluster
  - Source: https://machinelearningcoban.com/2017/01/01/kmeans/ (data, K-means implementation)
  - Content: Implement K-means using scilearn library and evaluate clustering result



Some practices - K-means

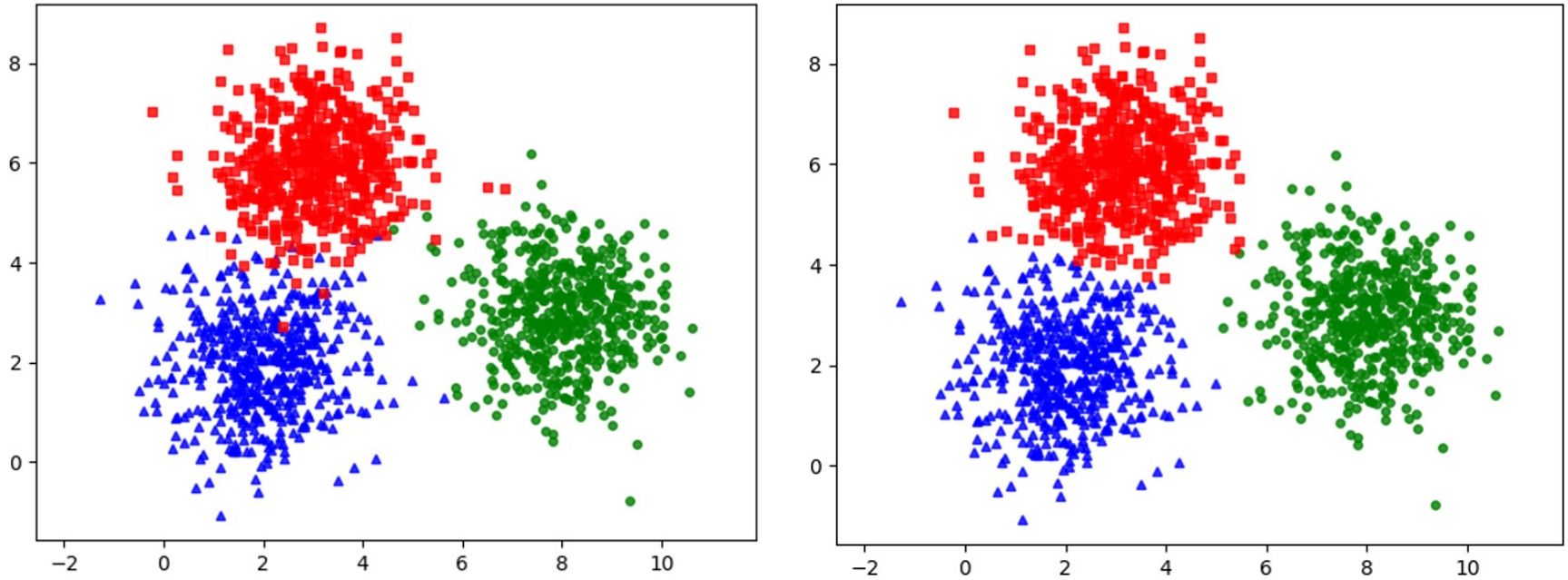
- Using a library to work with K-means

```
from sklearn.cluster import Kmeans

kmeans = KMeans(n_clusters=3)
kmeans.fit(data)
labels = kmeans.predict(data)
kmeans_display(data, labels)
```

Some practices - K-means

- Using a library to work with K-means



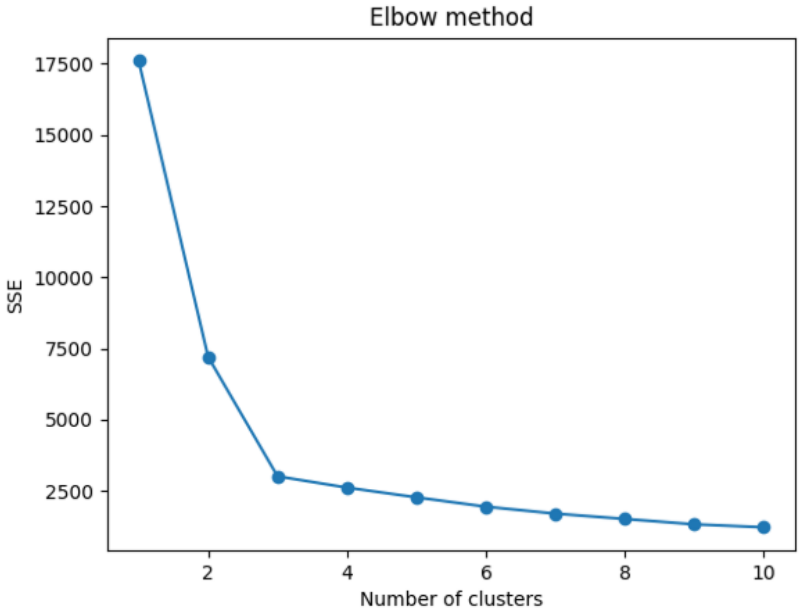
Clusters from original data (left) and found by K-means (right)

Some practices - K-means

- Evaluate clustering result and find number of clusters (SSE)

```
sse = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(data)
    sse.append(kmeans.inertia_)

plt.plot(range(1,11), sse,
marker='o')
plt.title('Elbow method')
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.show()
```



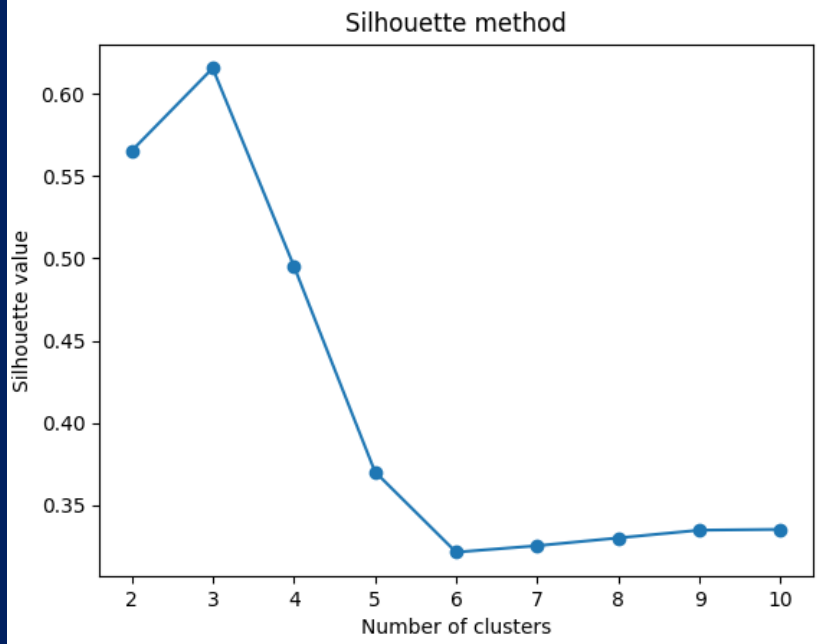
Some practices - K-means

- Evaluate clustering result and find number of clusters (silhouette)

```
from sklearn.metrics import
silhouette_score

silhouette = []
for i in range(2, 11):
    kmeans = KMeans(n_clusters=i,
random_state=0)
    kmeans.fit(data)
    score = silhouette_score(data,
kmeans.labels_)
    silhouette.append(score)

plt.plot(range(2,11), silhouette,
marker='o')
plt.title('Silhouette method')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette value')
plt.show()
```



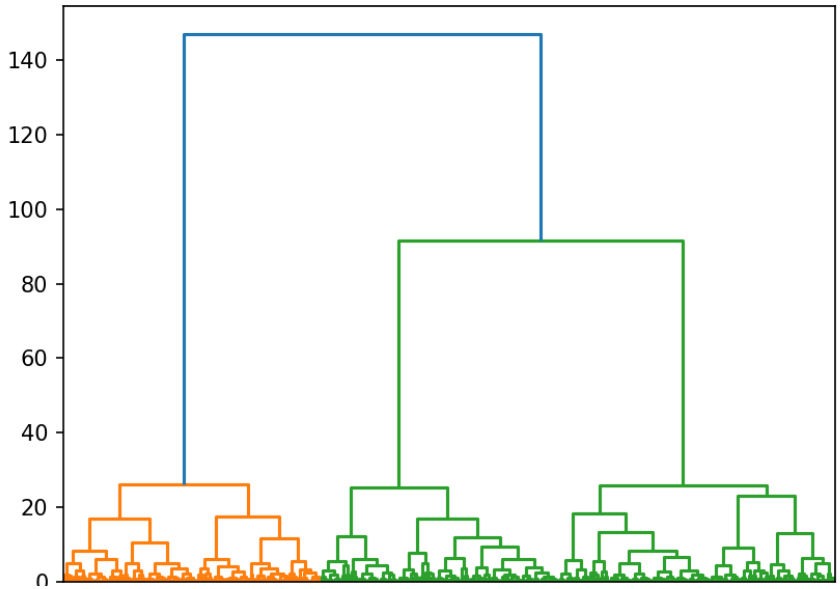
Some practices - Hirarchical clustering

- Evaluate clustering result and find number of clusters

```
from scipy.cluster.hierarchy
import dendrogram, linkage

linkage_data = linkage(data,
method='complete',
metric='euclidean’)

dendrogram(linkage_data)
plt.show()
```



Some practices - Hirarchical clustering

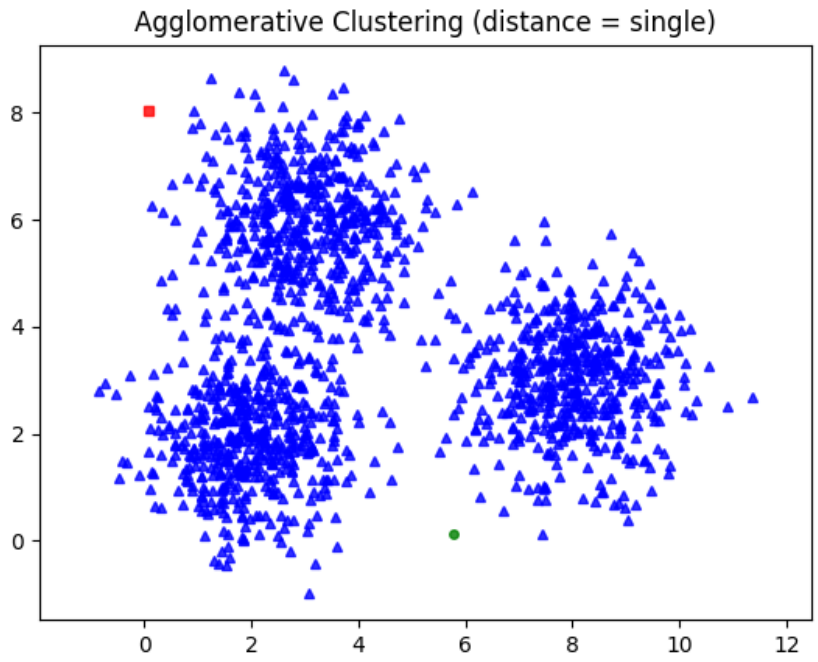
- Implementation of Hirarchical clustering (using library)

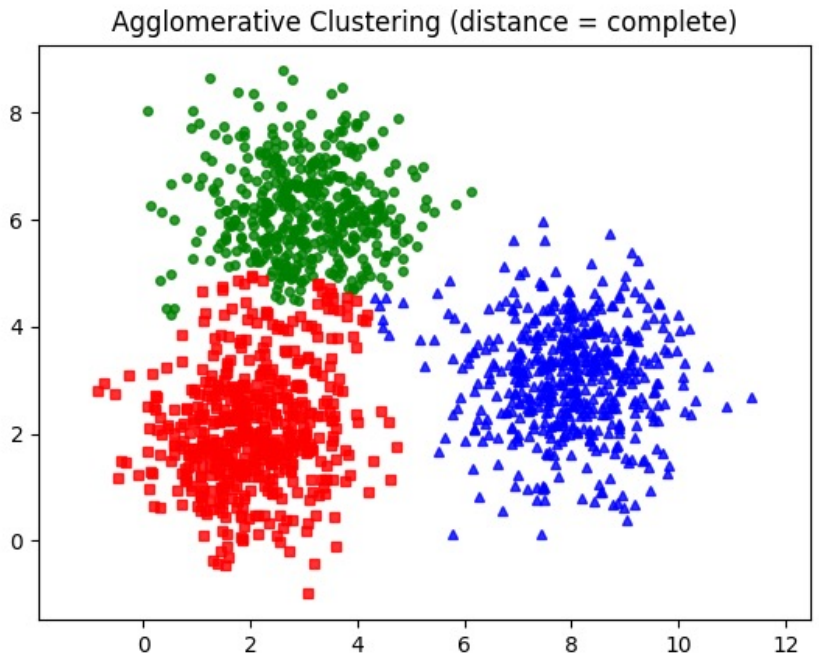
```
from sklearn.cluster import AgglomerativeClustering

for distance in ['single', 'complete', 'average', 'ward']:
    hierarchical_cluster = AgglomerativeClustering(n_clusters=3,\
        affinity='euclidean', linkage=distance)
    labels = hierarchical_cluster.fit_predict(data)
    cluster_display(data, labels, "Agglomerative Clustering \
        (distance = " + distance +')')
```

Some practices - Hirarchical clustering

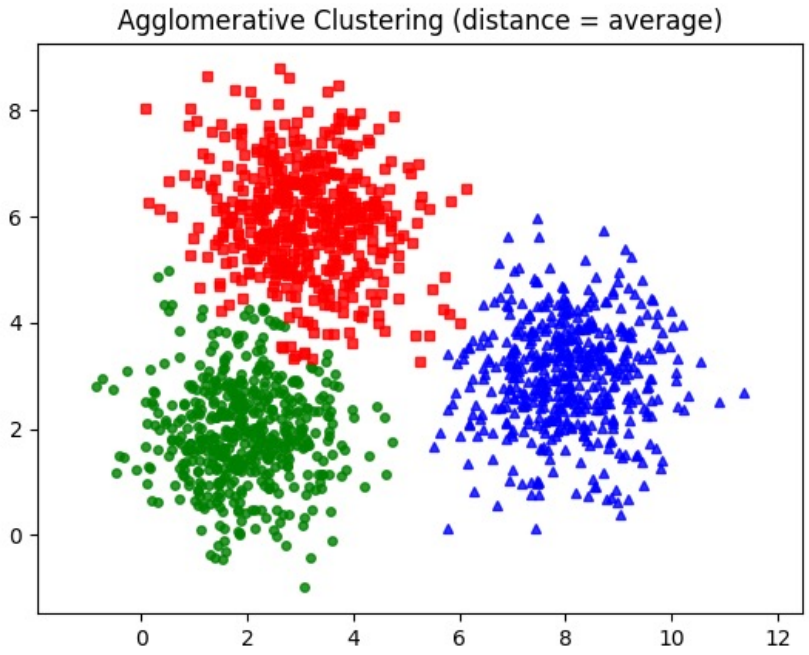
- Implementation of Hirarchical clustering (result)

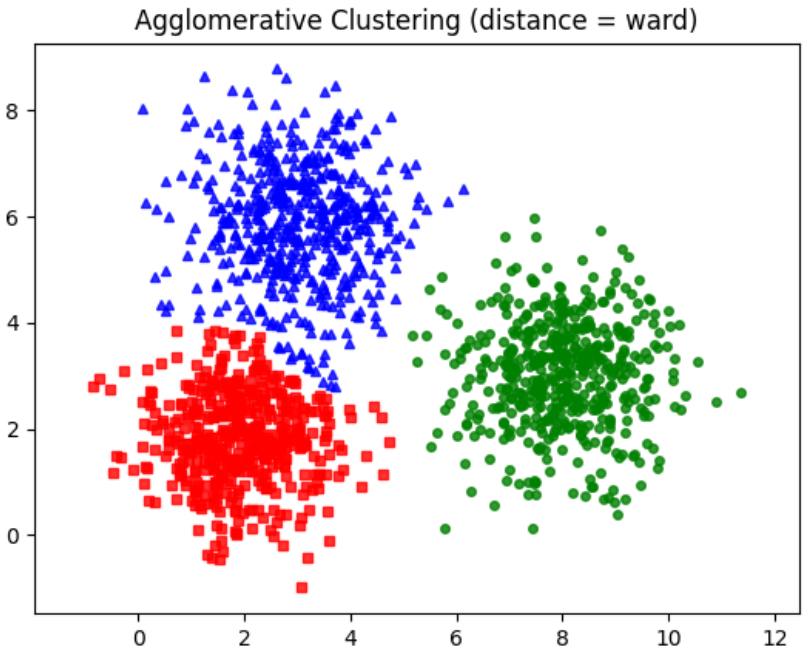




Some practices - Hirarchical clustering

- Implementation of Hirarchical clustering (result)





HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !