Freie Universität Berlin

Lecture: Machine Learning for Data Science

Winter semester 2021/22

Lecture 10: Unsupervised learning – Clustering (Partitioning-based)

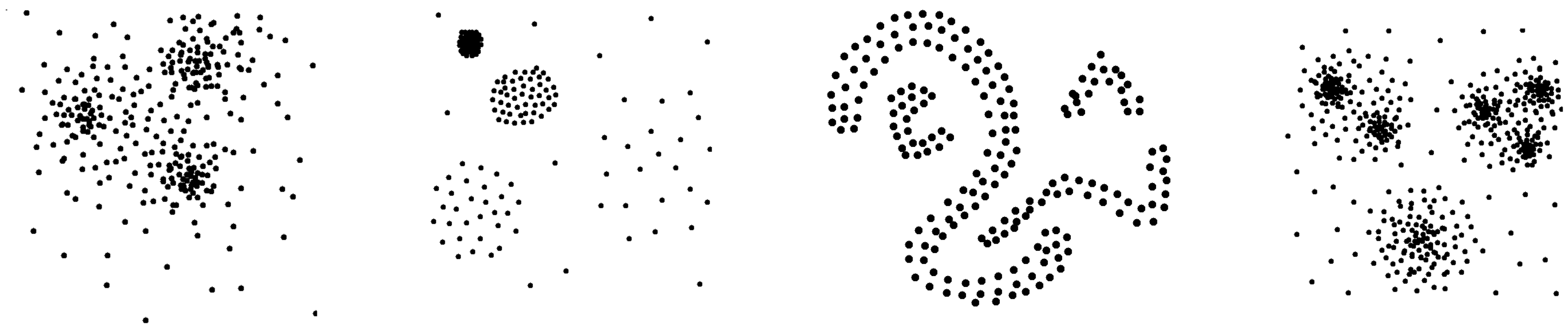
Prof. Dr. Eirini Ntoutsi

Outline

- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: k-Means
- Partitioning-based clustering: k-Medoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

What is cluster analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into the same clusters



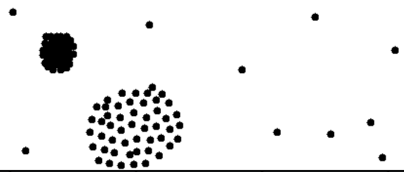
An unsupervised learning task

- Clustering is an unsupervised learning task
  - Given a set of measurements, observations, etc., the goal is to group the data into groups of similar data (th so called, clusters)
  - We are given a dataset as input which we want to cluster but there are no class labels (unlabeled dataset)
  - We don’t know how many clusters exist in the data
  - We don’t know the characteristics of the individual clusters

- In contrast to classification/regression, which are supervised learning tasks
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations (labeled dataset)

fruit	length	width	weight
fruit 1	165	38	172
fruit 2	218	39	230
fruit 3	76	80	145
fruit 4	145	35	150
fruit 5	90	88	160
…			
fruit n	…	…	…

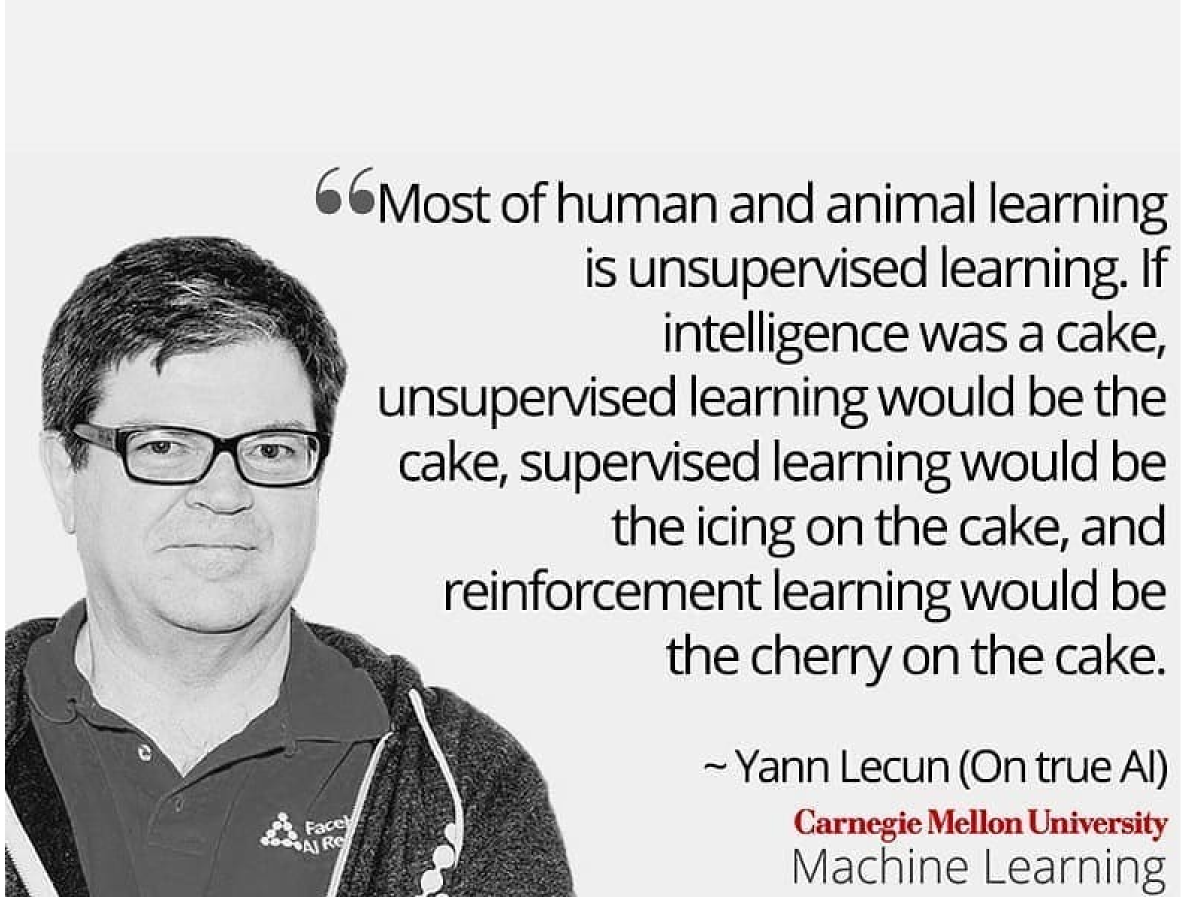
Unlabeled dataset



fruit	length	width	weight	label
fruit 1	165	38	172	Banana
fruit 2	218	39	230	Banana
fruit 3	76	80	145	Orange
fruit 4	145	35	150	Banana
fruit 5	90	88	160	Orange
…				
fruit n	…	…	…	…

Labeled dataset

Unsupervised learning



Source: Link

Yann Lecun's Cake Analogy

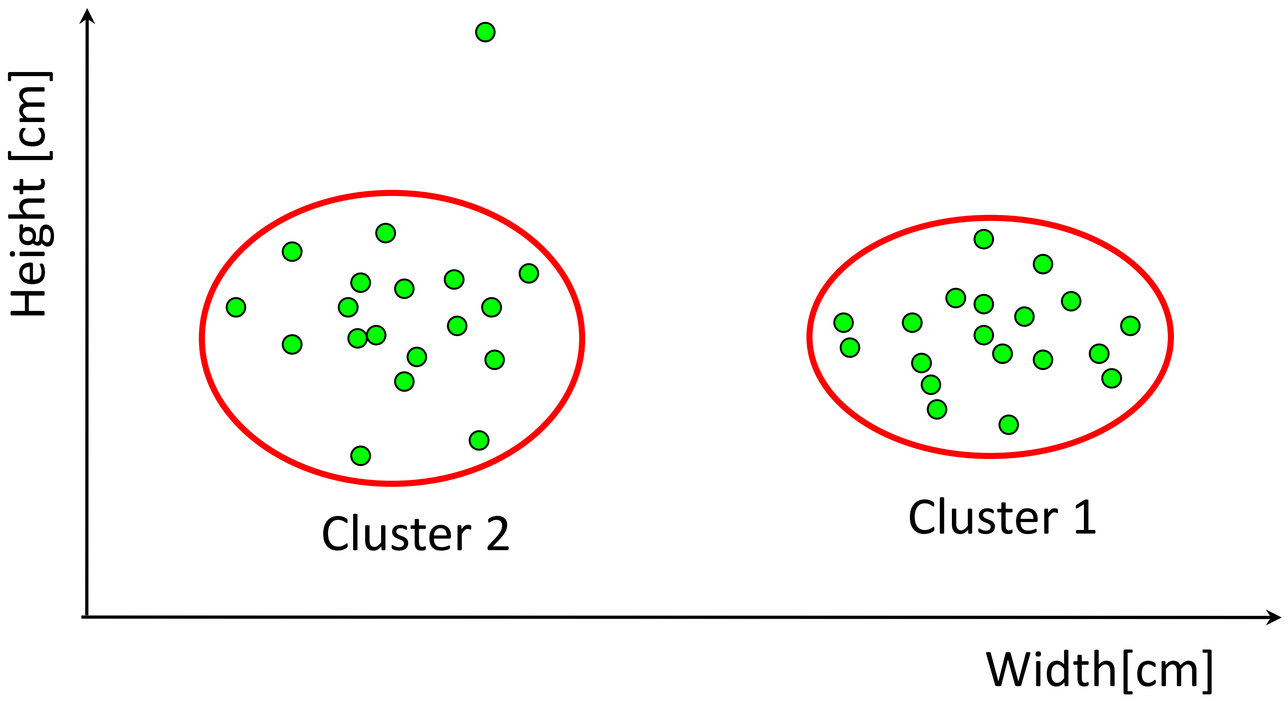
Y. LeCun

How Much Information is the Machine Given during Learning?

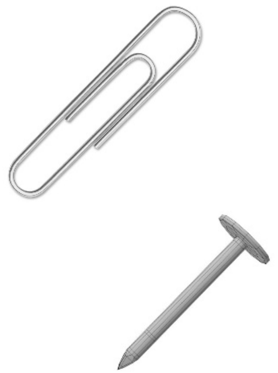
- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

 1.1: Deep Learning Hardware: Past, Present, & Future 59

Clustering: an example



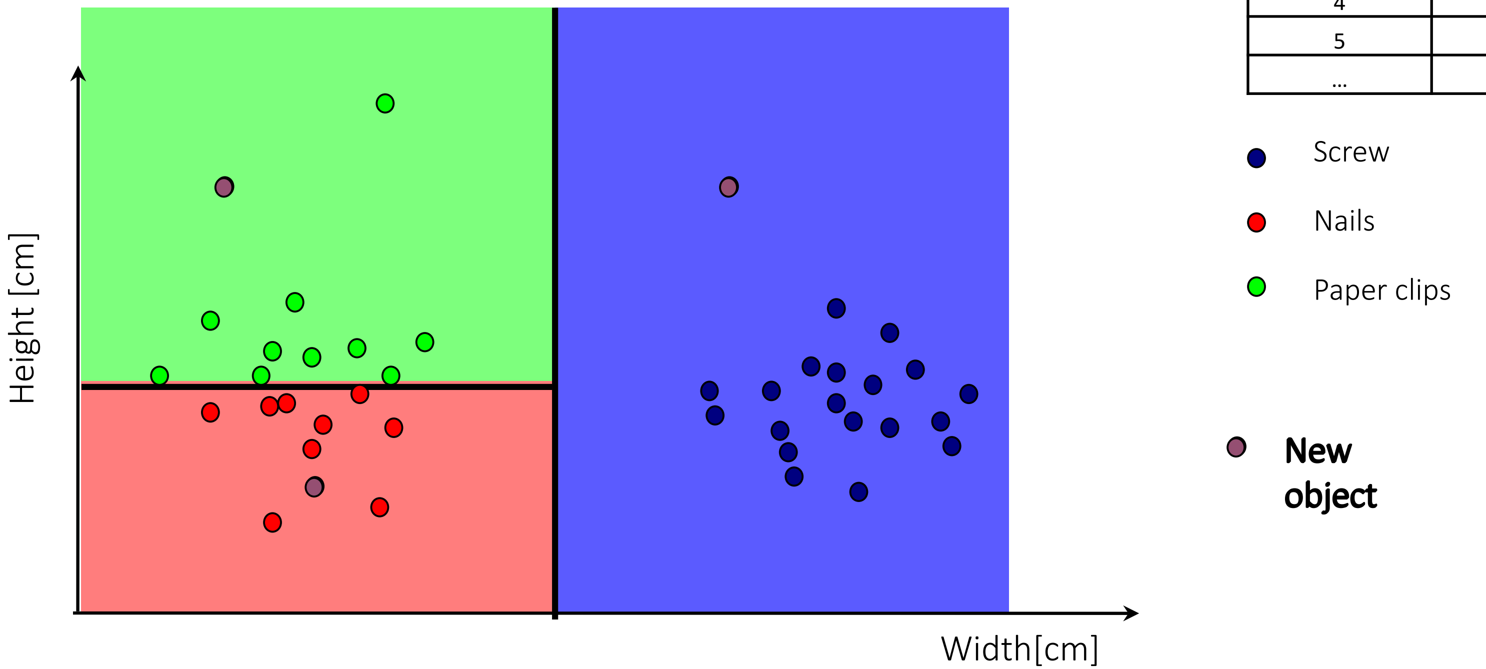
instance	width	height
1	2,6	4,5
2	3,7	7,3
3	4,1	6,5
4	8,5	8,1
5	9,5	5,5
…	…	…



Question:
Is there any structure in data (based on their characteristics, i.e., width, height)?

Classification: an example

instance	width	height	class
1	2,6	4,5	Screw
2	3,7	7,3	Nails
3	4,1	6,5	Paper Clips
4	8,5	8,1	Screw
5	9,5	5,5	Nails
…	…	…	…

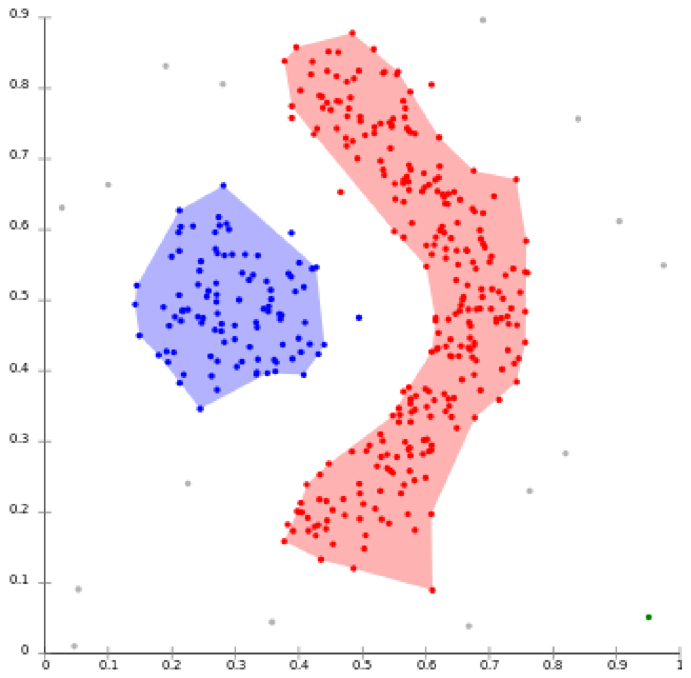


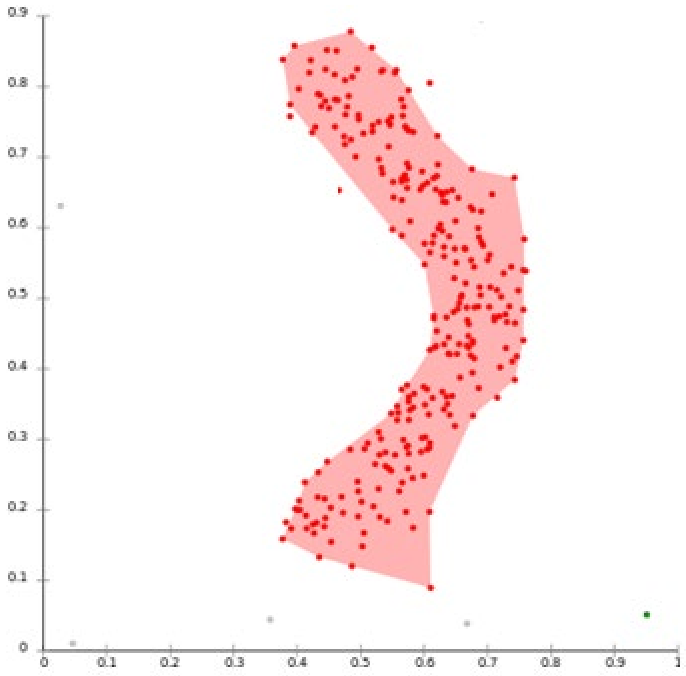
Question:

What is the class of a new object? Screw, nail or paper clip?

Why clustering?

- Clustering is widely used as:
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms





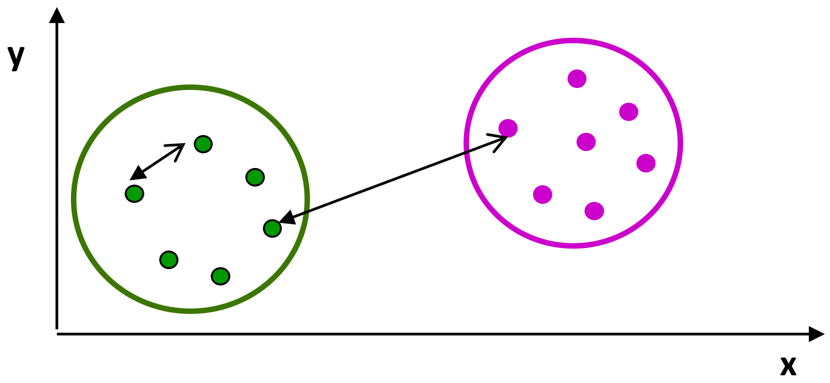
Source: http://en.wikipedia.org/wiki/Cluster_analysis

Example applications

- Marketing:
  - Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Telecommunications:
  - Build user profiles based on usage and demographics and define profile specific tariffs and offers
- Land use:
  - Identification of areas of similar land use in an earth observation database
- City-planning:
  - Identifying groups of houses according to their house type, value, and geographical location
- Bioinformatics:
  - Cluster similar proteins together (similarity wrt chemical structure and/or functionality etc)
- Web:
  - Cluster users based on their browsing behavior
  - Cluster pages based on their content (e.g. News aggregators)

The clustering task

- Goal: Group objects into groups so that the objects belonging in the same group are similar (high intra-cluster similarity), whereas objects in different groups are different (low inter-cluster similarity)
- A good clustering method will produce high quality clusters with
  - high intra-cluster similarity
  - low inter-cluster similarity



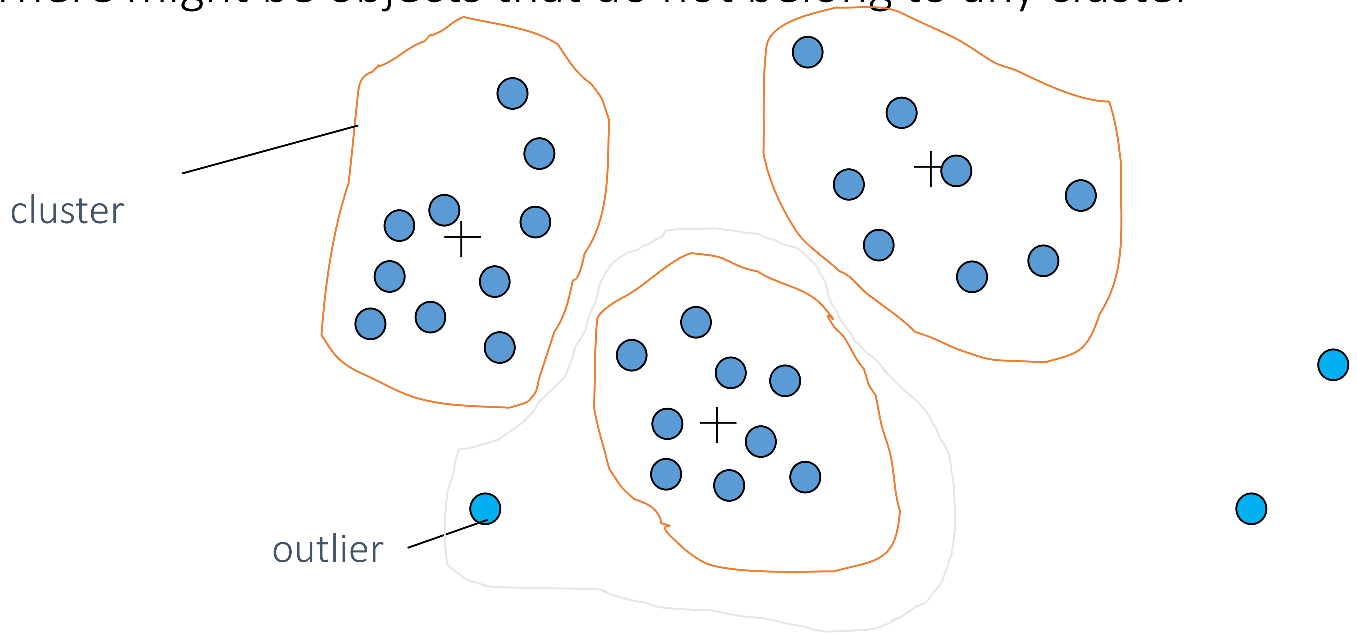
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Requirements for clustering

- Ideally, a clustering algorithm should fulfill the following requirements:
  - Discovery of clusters with arbitrary shape
  - Minimal requirements for domain knowledge to determine input parameters
  - Able to deal with noise and outliers
  - Incorporation of user-specified constraints
  - Interpretability and usability
  - Insensitive to the order of input records
  - Scalability
  - Ability to deal with different types of attributes
  - Ability to handle dynamic data
  - High dimensionality
  - …

Outliers vs Clusters

- There might be objects that do not belong to any cluster



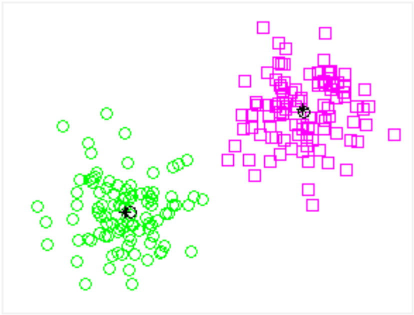
- Outliers can be removed at preprocessing. Some clustering algorithms (e.g., DBSCAN) also identify outliers
- There are cases where we are interested in detecting outliers, not clusters
  - More on outlier analysis in an upcoming lecture

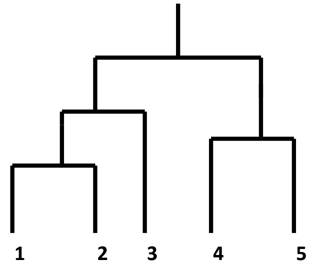
Outline

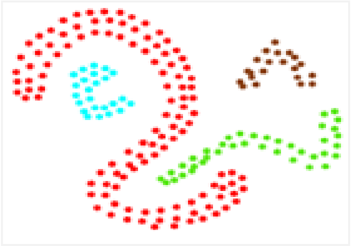
- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: kMeans
- Partitioning-based clustering: kMedoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

Major clustering methods

- Partitioning approaches:
  - Partition the data into several partitions/clusters based on some criterion, e.g., minimization of the sum of square errors
  - Typical methods: k-Means, k-Medoids, CLARANS
- Hierarchical approaches:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CHAMELEON
- Density-based approaches:
  - Based on connectivity and density functions
  - Typical methods: DBSCAN, OPTICS, DenClue

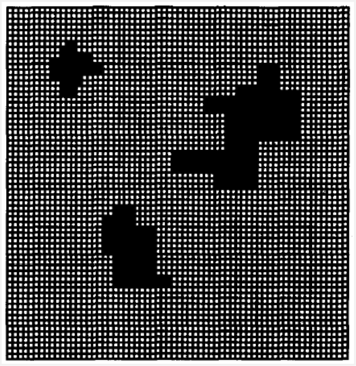


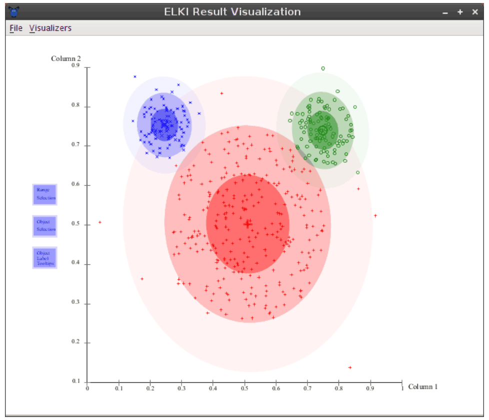


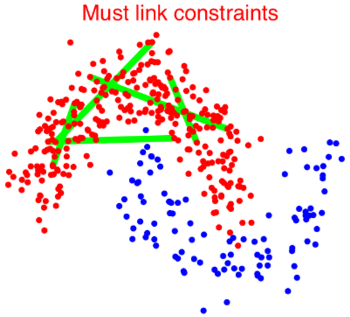


Major clustering methods

- Grid-based approaches:
  - partitioning the space via a grid
  - Typical methods: STING, WaveCluster, CLIQUE
- Model-based approaches:
  - A model is hypothesized for each of the clusters; the goal is to find the best models that explain the data
  - Typical methods: EM, SOM, COBWEB
- Constraint-based approaches:
  - Clustering by considering user-specified or application-specific constraints
  - Typical methods: COD (obstacles), constrained clustering
- ....

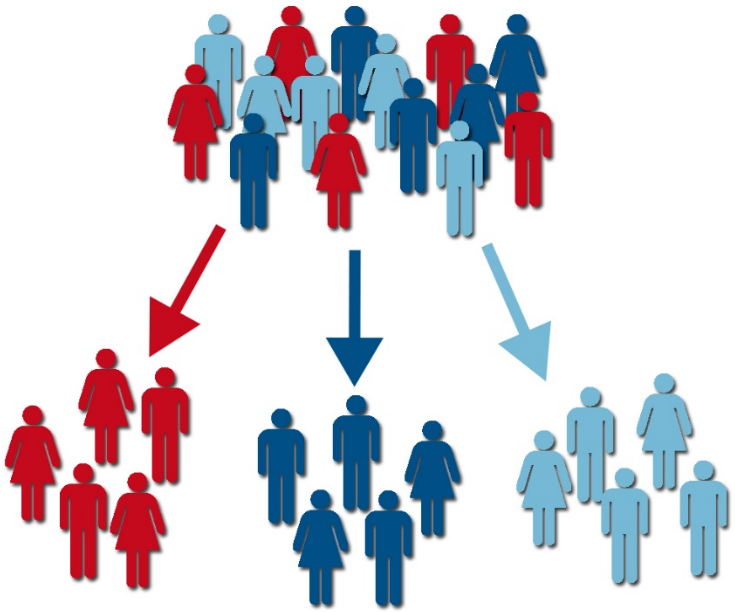






Cluster labeling

- After we extract the clusters, we typically want to describe them in a human interpretable way and not just by enumerating their members
  - Extensive description (enumerate cluster members)
  - Intensive description/cluster labeling (a more abstract description of the properties of the cluster members)

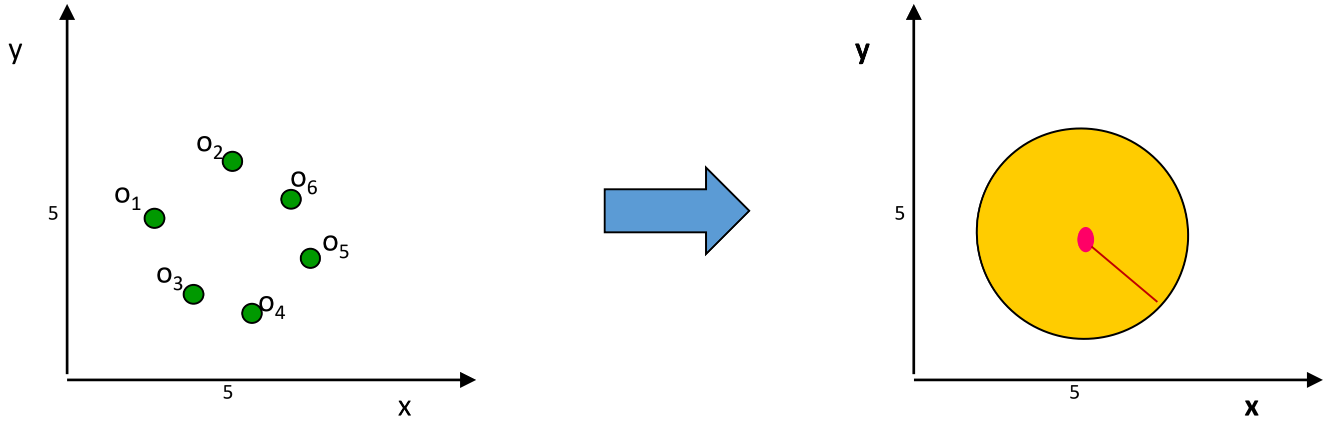


- Cluster labeling depends on
  - data types (e.g., numerical vs categorical)
  - extra information not used for clustering (like class labels)
  - …

Cluster labeling (numerical data, spherical clusters)

- In case of numerical data, spherical clusters are typically described via center and radius
- Centroid: the “center” of a cluster
- Radius: square root of average distance from any point of the cluster to its centroid

$$c_m = \frac{\sum_{i=1}^{n} p_i}{n}$$



$$r_m = \sqrt{\frac{\sum_{i=1}^{n} (p_i - c_m)^2}{n}}$$

Outline

- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: k-Means
- Partitioning-based clustering: k-Medoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

Partitioning methods idea

We will use the terms instances, objects, points interchangeably.

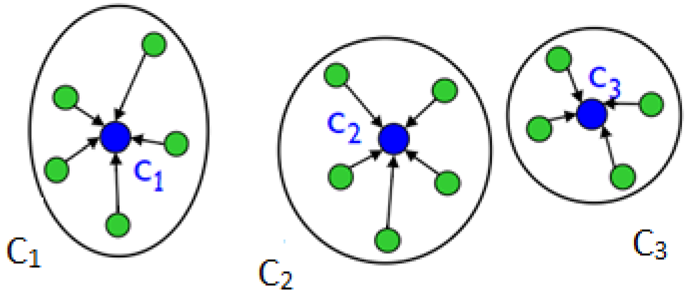
- Let a dataset *D* of instances described in a *d*-dimensional feature space
- Goal: Construct a partition of *D* into a set of *k* clusters
  - Each object belongs to exactly one cluster (hard or crisp clustering)
  - The number of clusters *k* is given in advance
- The partition should optimize the chosen partitioning criterion, i.e., minimize the intra-cluster distance
- Possible solutions:
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: k-Means and k-Medoids algorithms
    - k-Means: Each cluster is represented by the center of the cluster
    - k-Medoids: Each cluster is represented by one of the objects in the cluster

The k-Means problem

We use c_j to denote the cluster center of cluster C_j

- Given a dataset D of $|D|=n$ points in a d-dimensional space and an integer k
- <u>Task</u>: choose a set of k points $\{c_1, c_2,\ldots,c_k\}$ in the d-dimensional space to form clusters $\{C_1, C_2,\ldots,C_k\}$ such that the clustering cost is minimized:
  - The clustering cost is the aggregated intra—cluster distance, i.e., total square distance from point to the center of its cluster, the so-called centroid

$$Cost(C) = \underbrace{\sum_{i=1}^{k} \underbrace{\sum_{x \in C_i} d(x - c_i)^2}_{\textit{Cluster cost}}}_{\textit{Clustering cost}}$$



  - Same as variance if Euclidean distance (L_2 distance) is used
- This is an optimization problem, with the objective function to minimize the cost
- Enumerating all possible solutions and choosing the global optimum is infeasible (NP-hard).

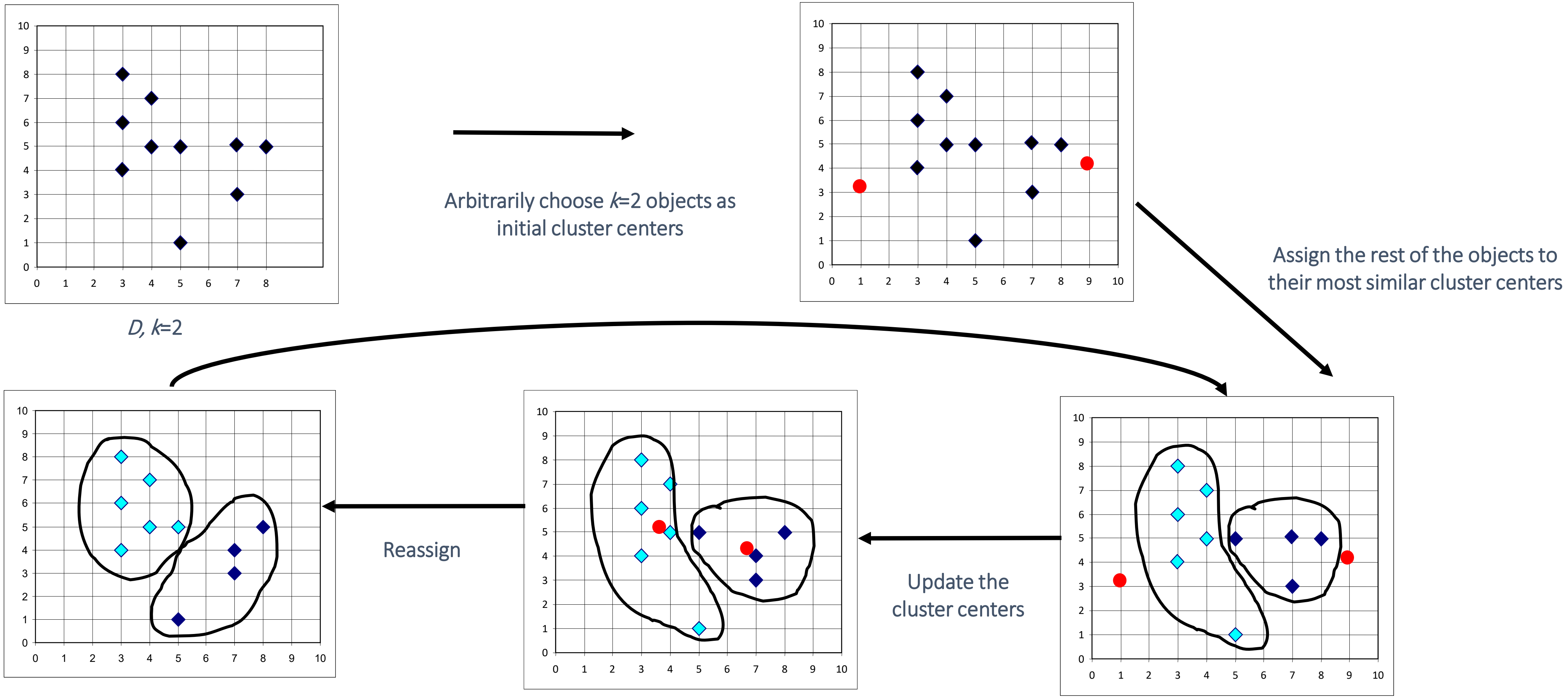
The k-Means algorithm (Lloyd's version)

- Given a dataset D of $|D|=n$ points in a d-dimensional space and the desired number of clusters k, the k-Means algorithm is implemented in four steps:
  - Randomly pick k objects as initial cluster centers $\{c_1, \ldots, c_k\}$.
  - Assign each point $x \in D$ to its closest cluster center c_j.
    - All points are assigned to some cluster $\{C_1, \ldots, C_k\}$
  - Update the center c_j of each cluster C_j based on the new point assignments.
  - Repeat until convergence.

- When to stop? Different approaches, e.g.:
  - cluster centers do not change
  - cost is not improved significantly
  - a max number of iterations t is reached
  - …

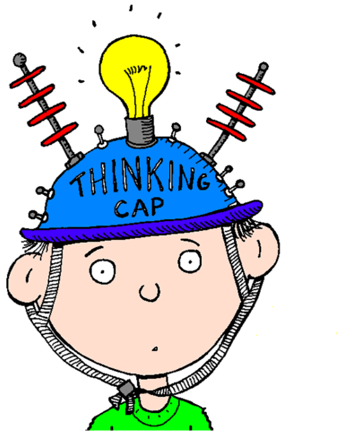
k-Means example

Note that the cluster centers are not real instances from *D*, also referred to as “virtual” centers



Short break (5')

- What is the complexity of k-Means?
  - Think for 1'
  - Discuss with your neighbours
  - Discuss in the class



k-Means pseudocode

Input: dataset D, $|D|=n$, # clusters k

- Randomly pick k objects as initial cluster centers $\{c_1, \ldots, c_k\}$.
- Assign the rest of the points to their closest cluster centers.
- Update the center of each cluster based on the new point assignments.
- Repeat until convergence.

k-Means complexity

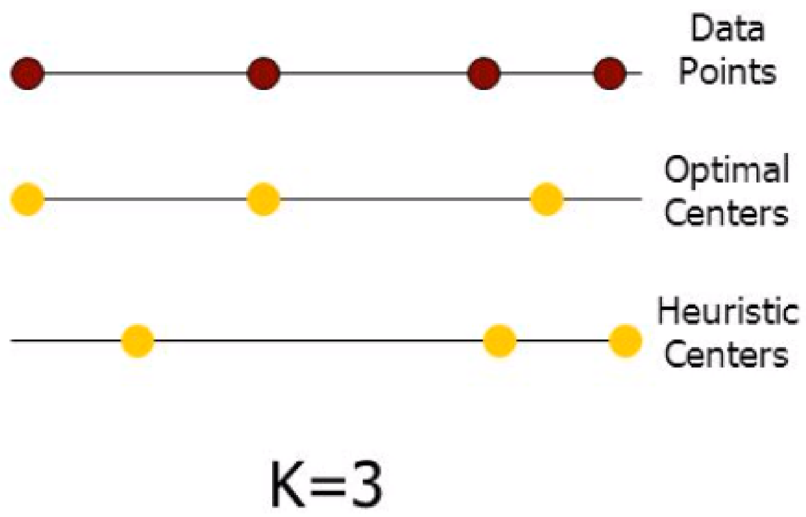
- Complexity
  - Relatively efficient
  - *O(tkn)*
    - *n*: is the number of objects
    - *k*: is the number of clusters
    - *t*: is the number of iterations.
  - Usually, *k, t* << *n*.

k-Means pseudocode

```
Input: dataset D, |D|=n, # clusters k

■ Randomly pick k objects as initial
  cluster centers {c_1, …, c_k}.

■ Assign the rest of the points to their
  closest cluster centers.

■ Update the center of each cluster based
  on the new point assignments.

■ Repeat until convergence.
```

k-Means convergence

- Has been shown to converge to a local optimum (locally optimal solution)
- But can converge to an arbitrarily bad solution compared to the optimal solution

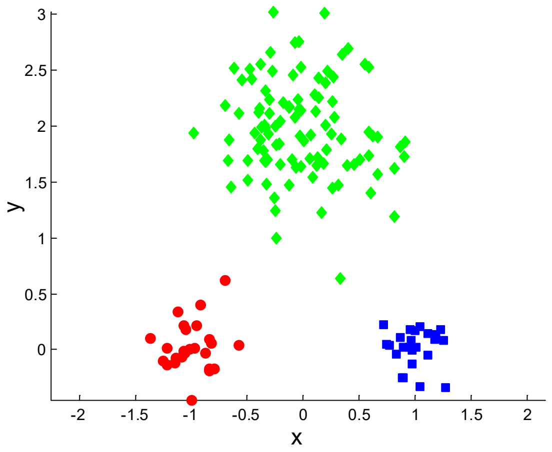


Selim, S. Z., & Ismail, M. A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1), 81-87.

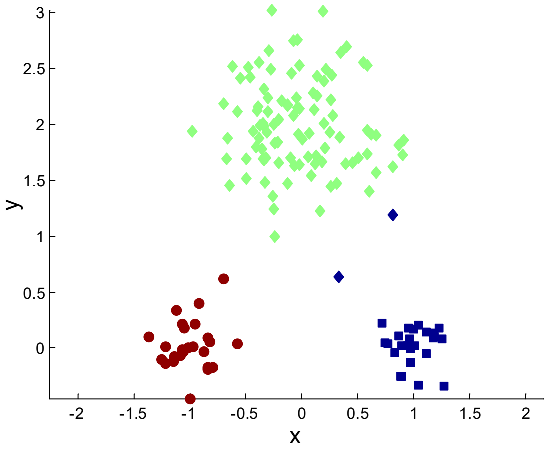
Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2004). A local search approximation algorithm for k-means clustering. *Computational Geometry*, *28*(2-3), 89-112.

k-Means convergence example

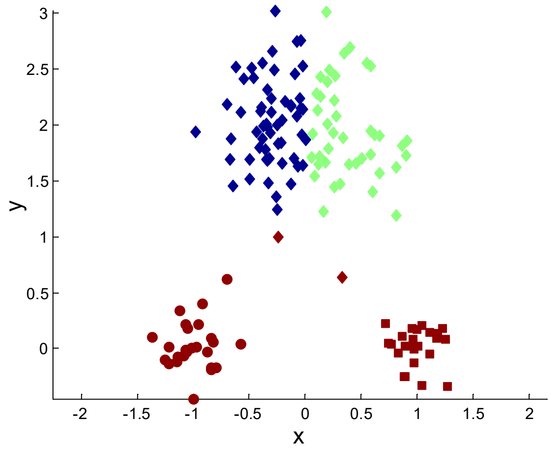
- k-Means converges to a local minimum



original points



optimal clustering

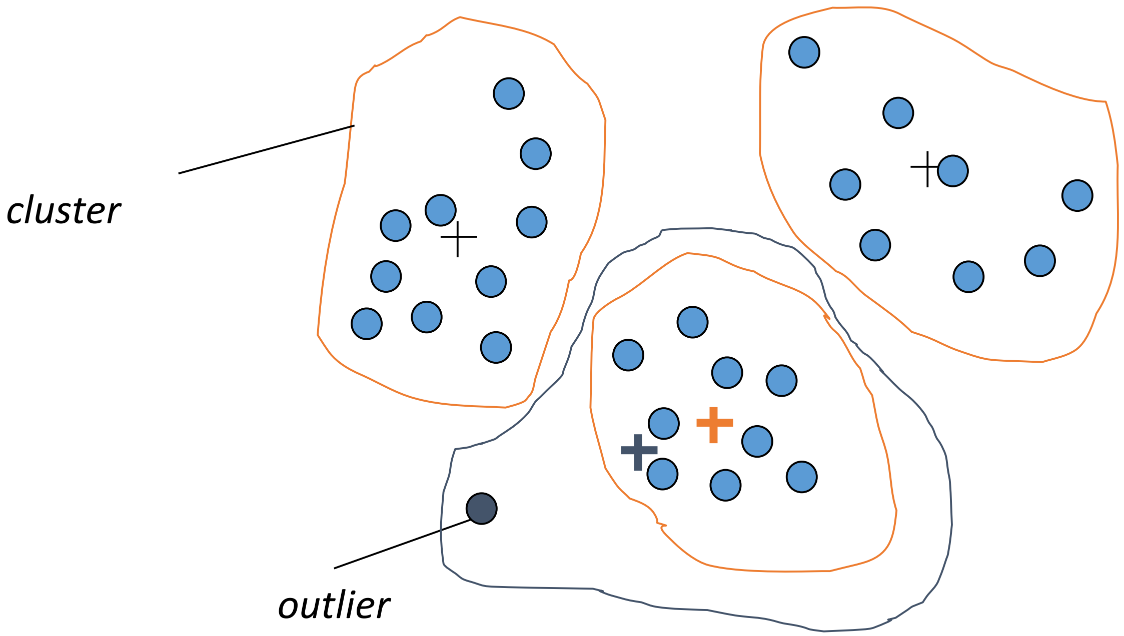


sub-optimal clustering

- Depends on the initialization: different starting points → different results (non-deterministic)
  - Idea: run several times with different initialization & pick the solution with the best clustering quality

k-Means and outliers

- k-Means is sensitive to outliers



- Outliers change the description of the clusters (e.g., centers are affected)
- One could remove outliers at a preprocessing step

k-Means example outliers

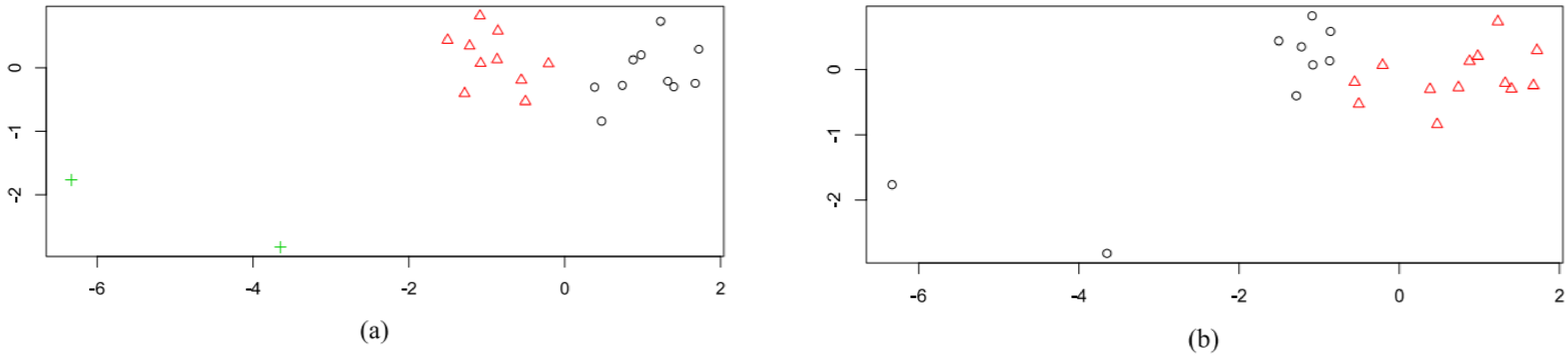
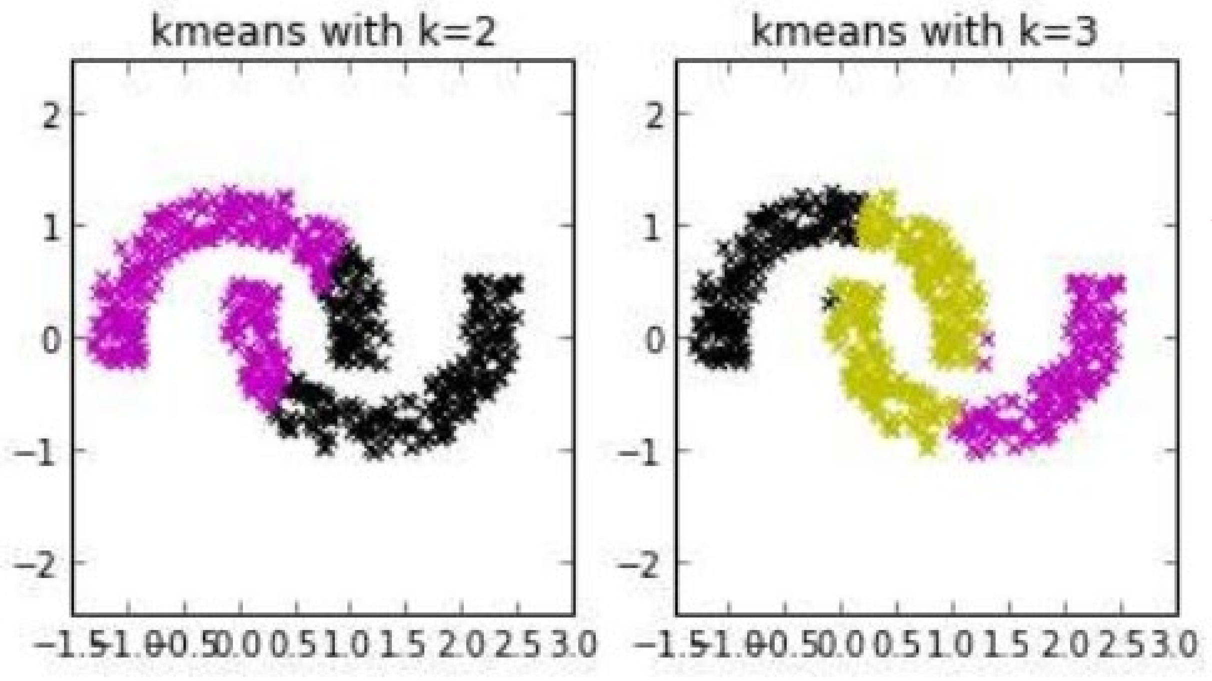


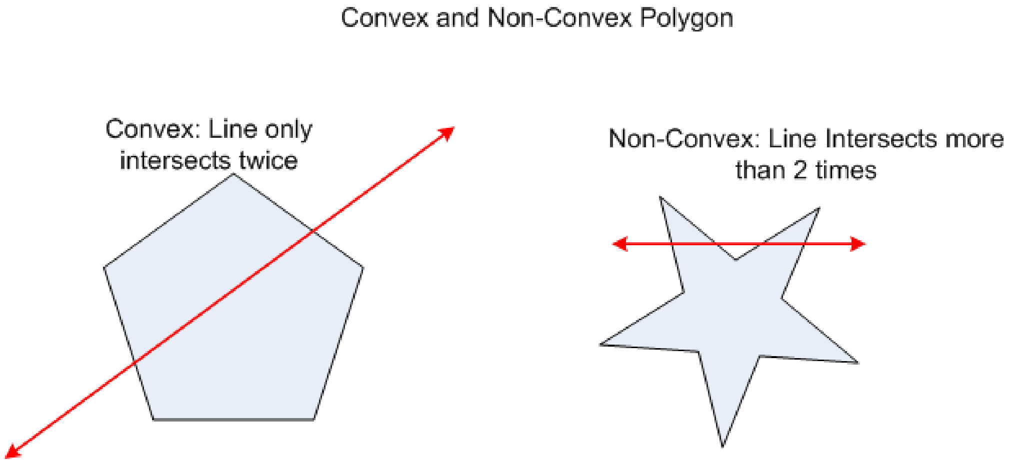
Fig. 1. An illustration showing that the *k*-means algorithm is sensitive to outliers. (a) A data set with two clusters and two outliers. The two clusters are plotted by triangles and circles, respectively. The two outliers are denoted by plus signs. (b) Two clusters found by the *k*-means algorithm. The two found clusters are plotted by triangles and circles, respectively.

Source: Guojun Gan and Michael Kwok-Po Ng. 2017. k-means clustering with outlier removal. Pattern Recogn. Lett. 90, C (April 2017), 8-14. DOI: https://doi.org/10.1016/j.patrec.2017.03.008 http://www.math.uconn.edu/~gan/ggpaper/gan2017kmor.pdf

k-Means properties

- Hard to find clusters with non-convex shapes





k-Means variants

- Many variants of the k-Means which differ in e.g.,
  - Different initialization of the *k* centers
    - Multiple runs
    - Not random selection of centers. e.g., pick the most distant (from each other) points as cluster centers (kMeans++ algorithm)
    - …
  - Different strategies to calculate cluster means
  - …
- Handling categorical data: *k*-modes (Huang'98)
  - Replacing means of clusters with modes (mode = value that occurs more often)
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters

k-Means: Lloyd's version vs MacQueen's version

- The k-Means algorithm we discussed thus far is the so called Lloyd's version
  - It fixes the centers, assigns the points to their closest centers, updates the centers, reassign the points …
- There is another version, the MacQueen's version that
  - Also called incremental/online algorithm (used also in streams)
  - The main difference to Lloyd's is that the *centroids are re-calculated every time a point is moved*.

k-Means MacQueen's version pseudocode

Input: dataset D, $|D|=n$, # clusters k

- Randomly pick k objects as initial cluster centers $\{c_1, …, c_k\}$.
- For each point $x \in D$
  - Assign x to its closest cluster center c_j.
  - Update the centers of the clusters (only those affected, so c_j in this case)
- Recalculate centroids
- Repeat until convergence.

k-Means Lloyd's version pseudocode

Input: dataset D, $|D|=n$, # clusters k

- Randomly pick k objects as initial cluster centers $\{c_1, …, c_k\}$.
- Assign the rest of the points to their closest cluster centers.
- Update the center of each cluster based on the new point assignments.
- Repeat until convergence.

k-Means overview

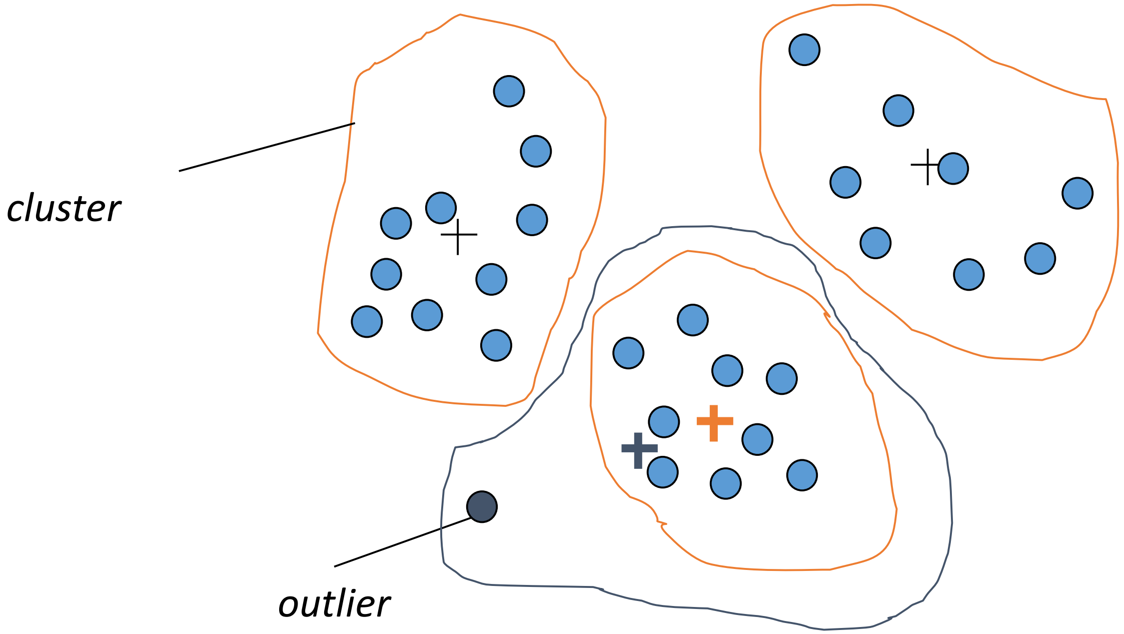
- Relatively efficient: $O(tkn)$, n: # objects, k: # clusters, t: # iterations. Normally, $k, t << n$.
- Finds a local optimum
- The choice of the initial centers can have a large influence in the results
- Weaknesses
  - Need to specify k, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes
  - Applicable only when mean is defined, then what about categorical data?

Outline

- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: k-Means
- Partitioning-based clustering: k-Medoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

From k-Means to k-Medoids

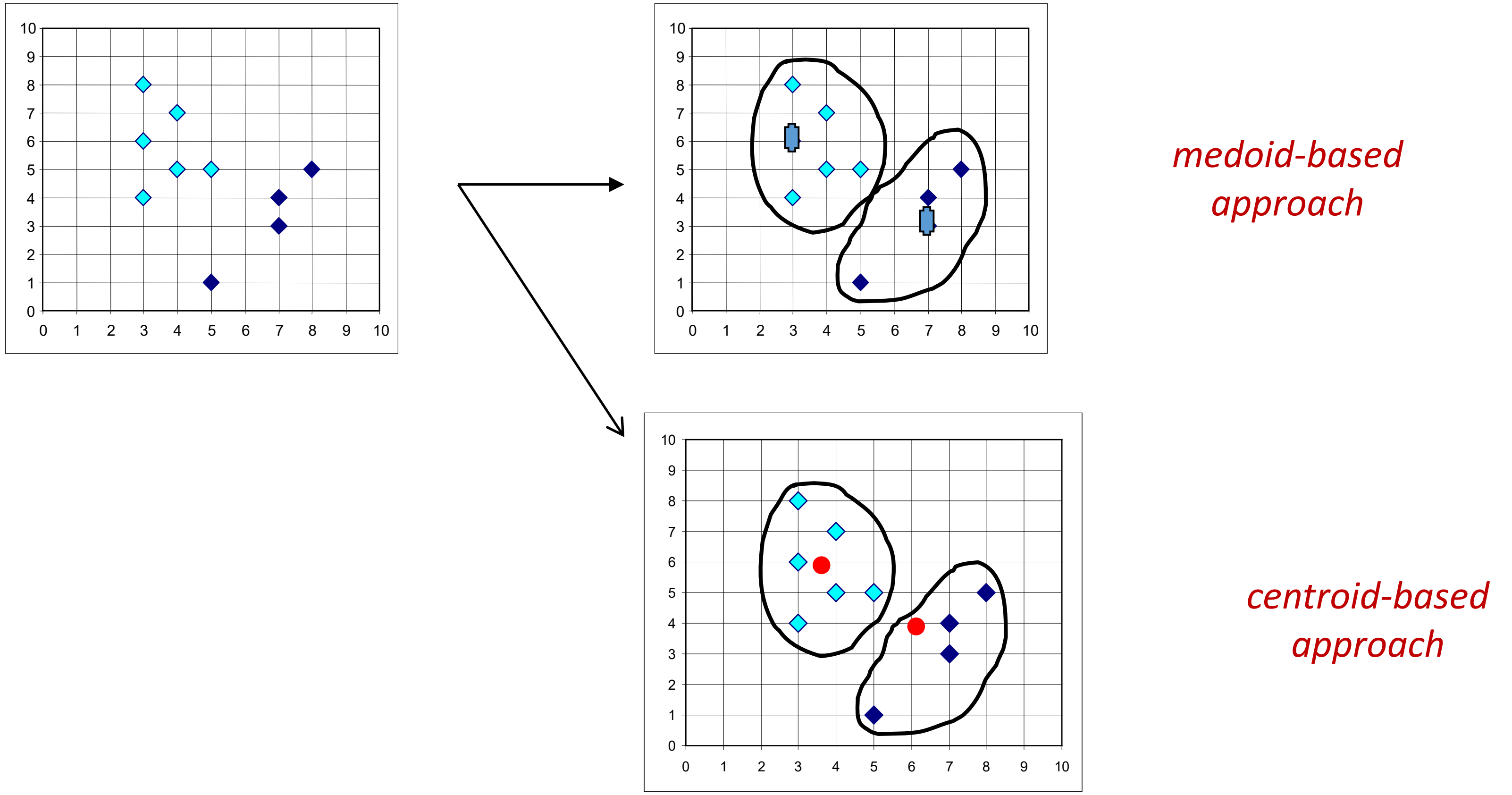
- The k-Means algorithm is sensitive to outliers!
  - an object with an extremely large value may substantially distort the distribution of the data.



- k-Medoids: Instead of taking the mean value of the objects in a cluster as a reference point, medoids can be used, which are the most centrally located objects in the clusters.

From k-Means to k-Medoids

- k-Means (clusters represented via ``virtual'' centers) vs k-Medoids (clusters represented via real instances)



The k-Medoids clustering algorithm

- Clusters are represented by real objects called medoids.
- PAM (Partitioning Around Medoids, Kaufman and Rousseeuw, 1987)
  - starts from an initial set of k medoids and iteratively replaces one of the medoids by one of the non-medoid points iff such a replacement improves the total clustering cost

K-Medoids pseudocode

Input: dataset D, $|D|=n$, # clusters k

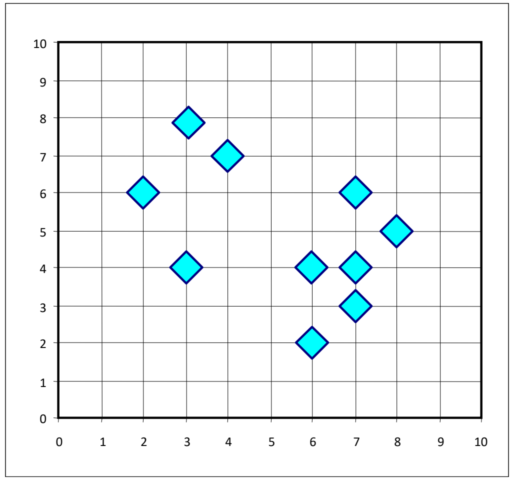
- Select k representative objects arbitrarily
- Repeat
  - Assign the rest of the objects to the k clusters
  - Representative replacement:
    - For each medoid m and each non-medoid object o, check whether o could replace m
    - Replacement is possible if the clustering cost is improved.
- Until no improvements can be achieved by any replacement

$$Cost(C) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x - c_i)^2$$

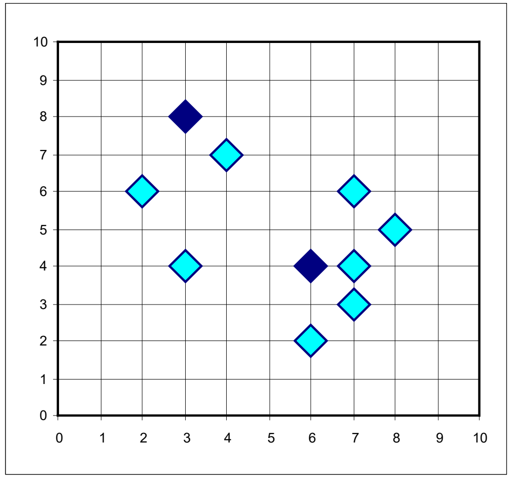
Clustering cost e.g., as in k-Means (c.f., slide 22)

PAM example: don’t swap case

k=2

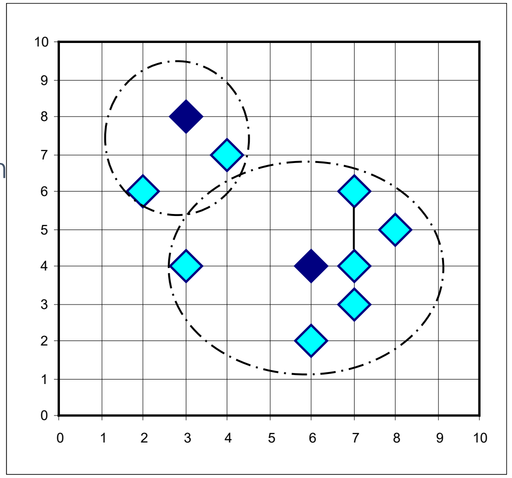


Arbitrary choose k object as initial medoids



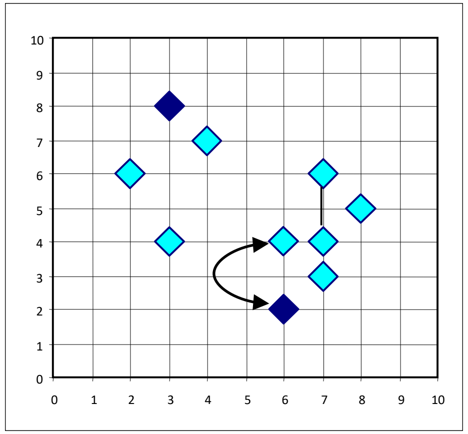
Assign each remaining object to nearest medoid

Total Cost = 19



Clustering cost computed using Manhattan distance (L_1)

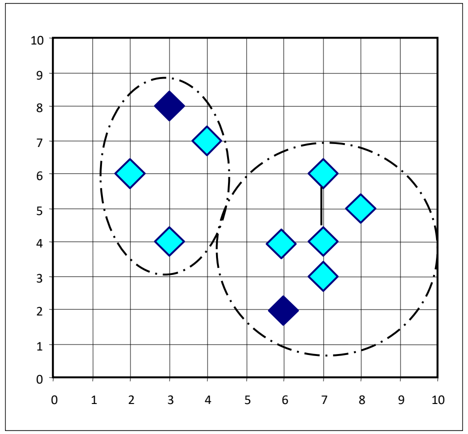
Randomly select a non-medoid object to replace a medoid



Reassign points

Compute new clustering cost

Total Cost = 26



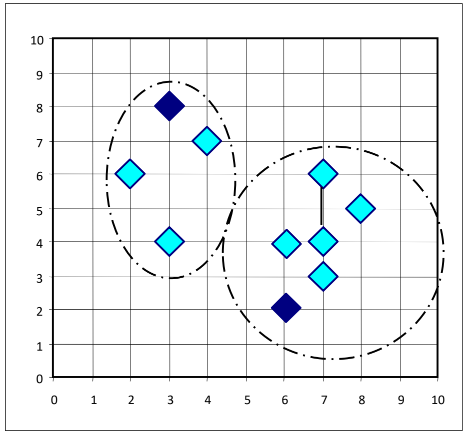
Swap if quality is improved.

26>19→ don’t swap

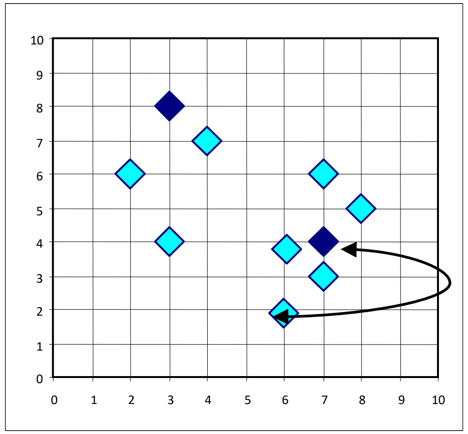
PAM example: swap case

k=2

Total Cost = 26

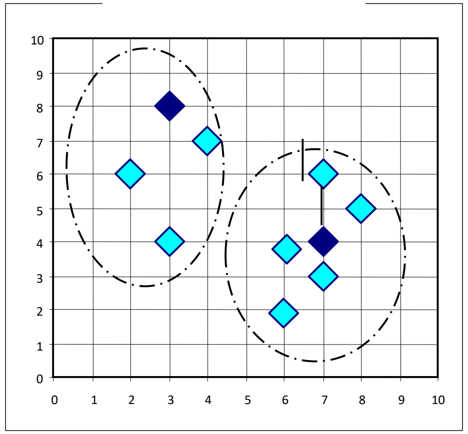


Randomly select a non-medoid object to replace a medoid



Assign each remaining object to nearest medoid

Total Cost = 18



Clustering cost computed using Manhattan distance (L_1)

Swap if quality is improved

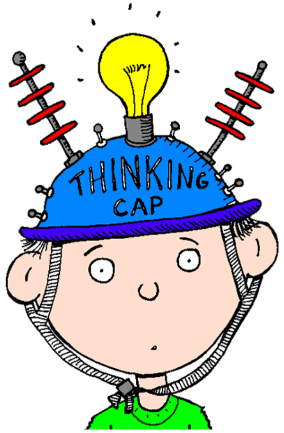
18<26 → swap

Do loop

Until no change

Short break (5')

- What is the complexity of kMedoids?
  - Think for 1'
  - Discuss with your neighbours
  - Discuss in the class



K-Medoids pseudocode

Input: dataset D, $|D|=n$, # clusters k

- Select k representative objects arbitrarily
- Repeat
  - Assign the rest of the objects to the k clusters
  - Representative replacement:
    - For each medoid m and each non-medoid object o, check whether o could replace m
    - Replacement is possible if the clustering cost is improved.
- Until no improvements can be achieved by any replacement

k-Medoids complexity

- PAM complexity
  - $O(k(n-k)^2)$ for each iteration

where *n* is # of data, *k* is # of clusters

- PAM works efficiently for small data sets but does not scale well for large data sets.

K-Medoids pseudocode

Input: dataset *D*, |*D*|=*n*, # clusters *k*

- Select *k* representative objects arbitrarily
- Repeat
  - Assign the rest of the objects to the *k* clusters
  - Representative replacement:
    - For each medoid *m* and each non-medoid object *o*, check whether *o* could replace *m*
    - Replacement is possible if the clustering cost is improved.
- Until no improvements can be achieved by any replacement

PAM overview

- Very similar to k-Means
- PAM is more robust to outliers comparing to k-Means because a medoid is less influenced by outliers or other extreme values than a centroid.
- PAM works efficiently for small data sets but does not scale well for large data sets.
  - $O(k(n-k)^2)$ for each iteration where *n* is # of data, *k* is # of clusters

- Sampling based method:
  - CLARA(Clustering LARge Applications)
  - CLARANS ("Randomized" CLARA)

Outline

- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: kMeans
- Partitioning-based clustering: kMedoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

What is the right number of clusters

- For partitioning-based clustering, *k* is required as input. Choosing the right *k* is challenging.
- Silhouette coefficient of an object *i* (Kaufman & Rousseeuw 1990)
  - Let *A* be the cluster to which *i* belongs
  - Let *a(i)* the distance of object *i* to *A* (the so-called best first cluster distance)

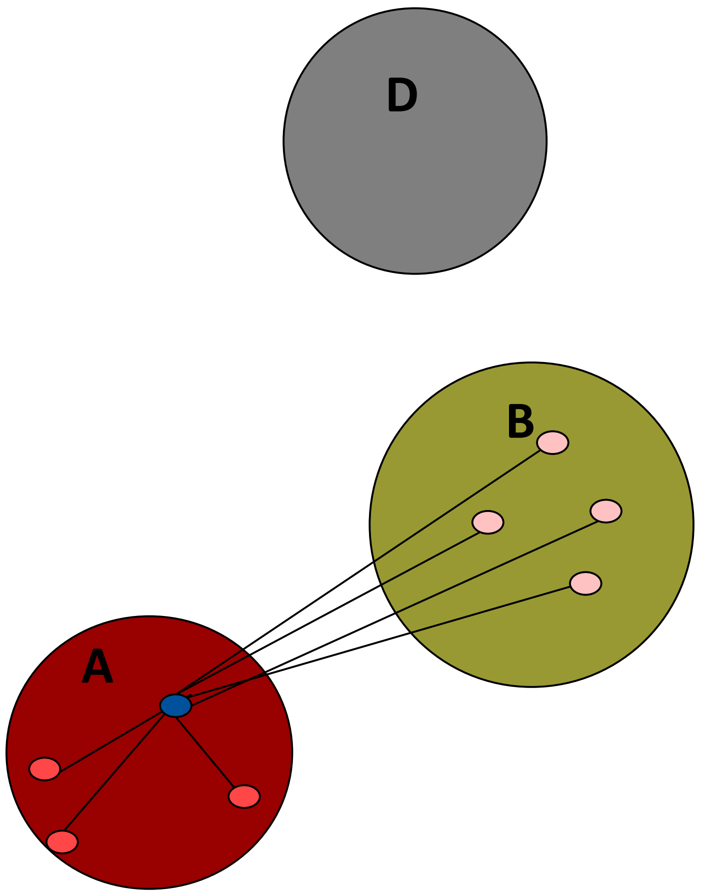
$$
\begin{aligned}
a(i) &:= \frac{1}{|A|-1} \sum_{j \in A, j \neq i} d(i,j) \\
&= \text{average dissimilarity of } i \text{ to all other objects of } A.
\end{aligned}
$$

  - Let *b(i)* be the distance of *i* to its second best cluster (we denote it by *B*)

$$
b(i) := \min_{C \neq A} d(i, C).
$$

where

$$
\begin{aligned}
d(i,C) &:= \frac{1}{|C|} \sum_{j \in C} d(i,j) \\
&= \text{average dissimilarity of } i \text{ to all objects of } C.
\end{aligned}
$$



What is the right number of clusters

- The Silhouette value *s(i)* of the object *i* is given by:

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

$-1 \leq s(i) \leq +1$

s(i) ~ -1 / 0 / +1 : bad / indifferent /good assignment

- *s(i)~1* → *a(i)* < < *b(i)*. Small *a(i)* means it is well matched to its own cluster *A*. Large *b(i)* means is badly matched to its neighboring cluster *B* → good assignment
- *s(i)~-1*→ the neighbor cluster *B* seems more appropriate → bad assignment
- s(i)~0→ in the border between the two natural clusters *A, B* → indifferent assignment

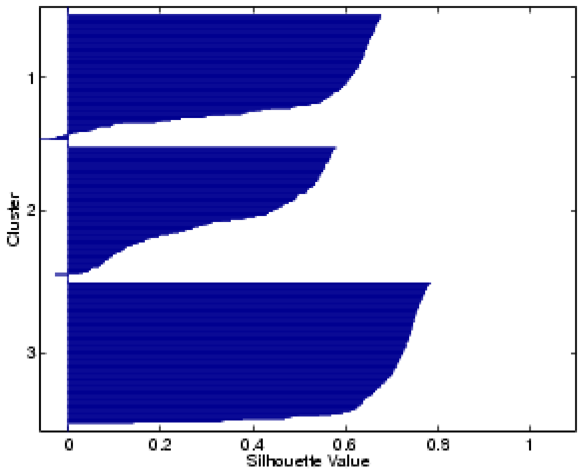
What is the right number of clusters

- The Silhouette coefficient of a cluster is the avg silhouette of all its objects
  - Is a measure of how tightly grouped all the data in the cluster are.
  - > 0,7: strong structure, > 0,5: usable structure ….

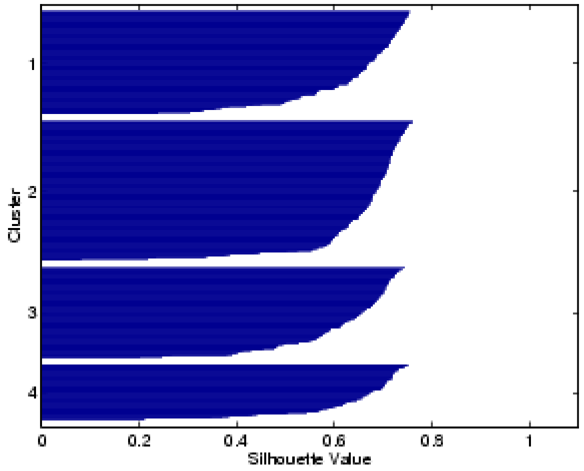
- The Silhouette coefficient of a clustering is the avg silhouette of all objects
  - is a measure of how appropriately the dataset has been clustered

What is the right number of clusters 2/2

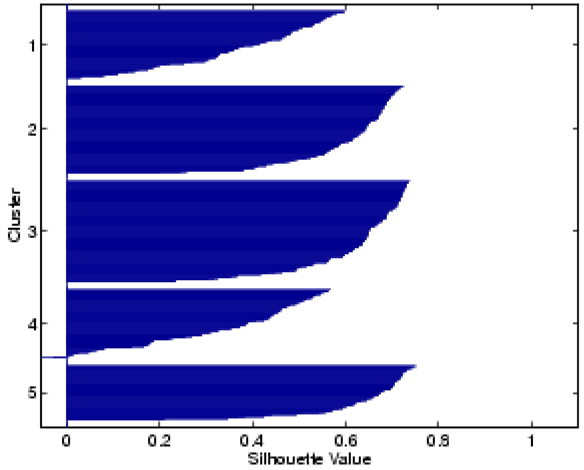
- The silhouette plot of a cluster A consists all its s(i) ranked in decreasing order.
- The entire silhouette plot of a clustering shows the silhouettes of all clusters below each other, so the quality of the clusters can be compared:



K=3

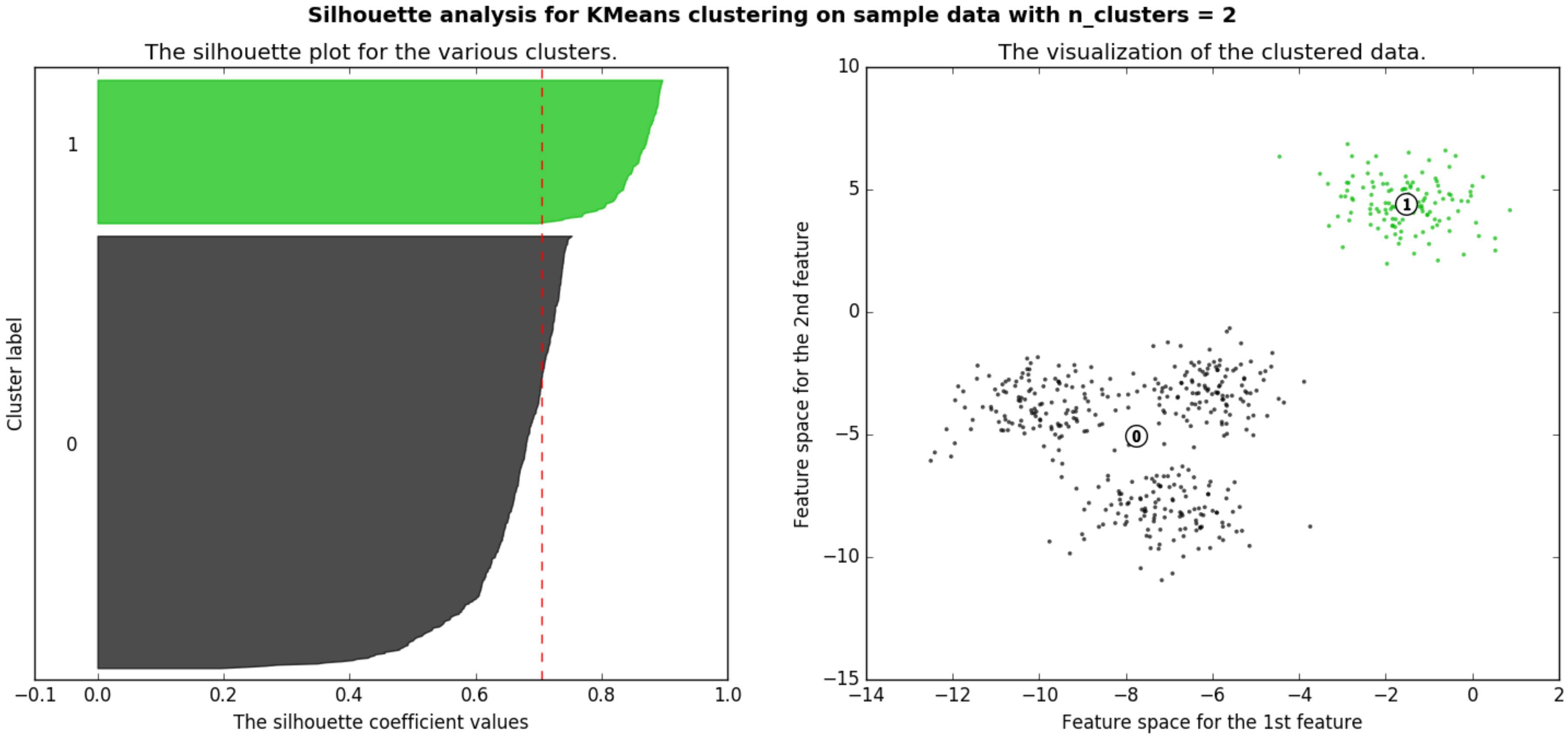


K=4



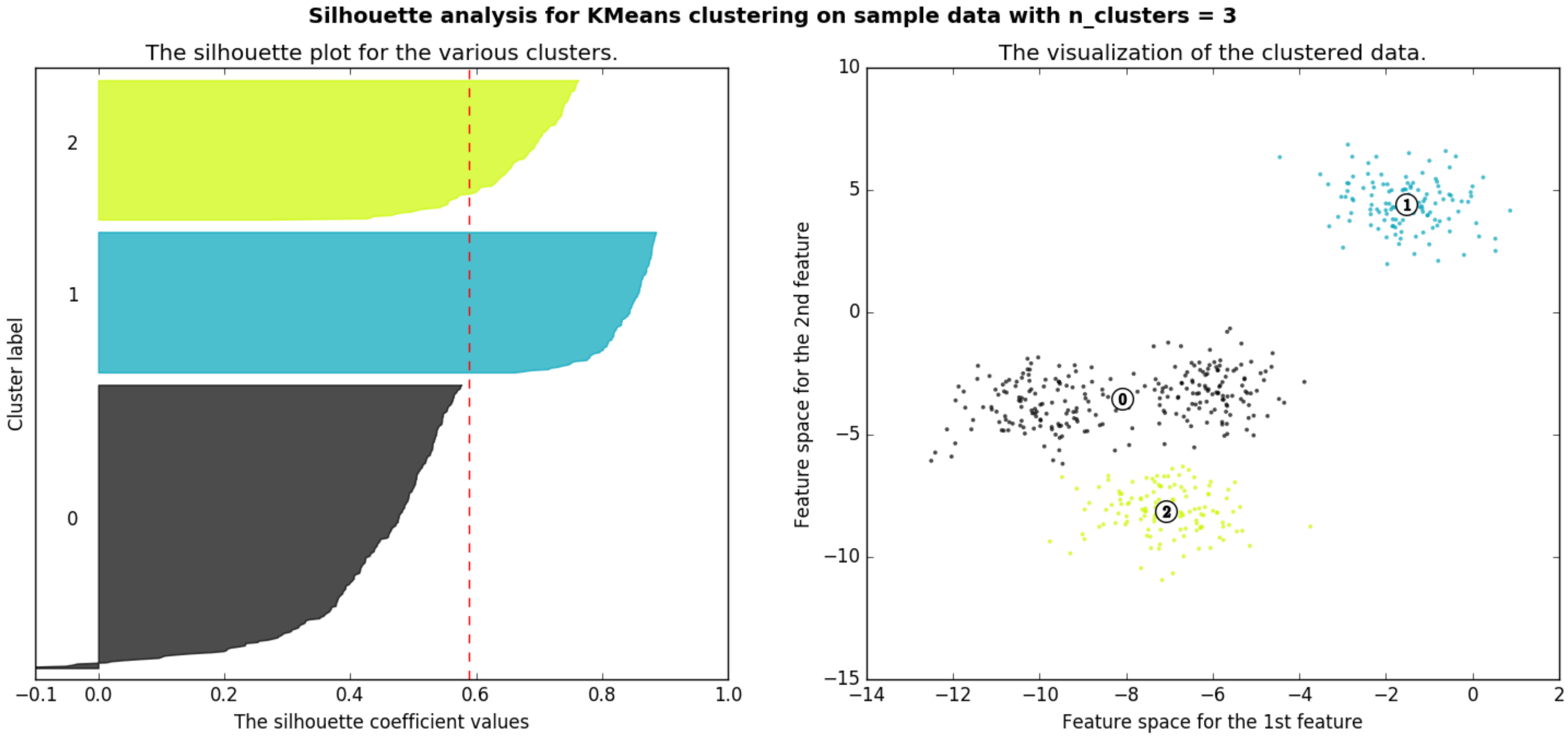
K=5

An example



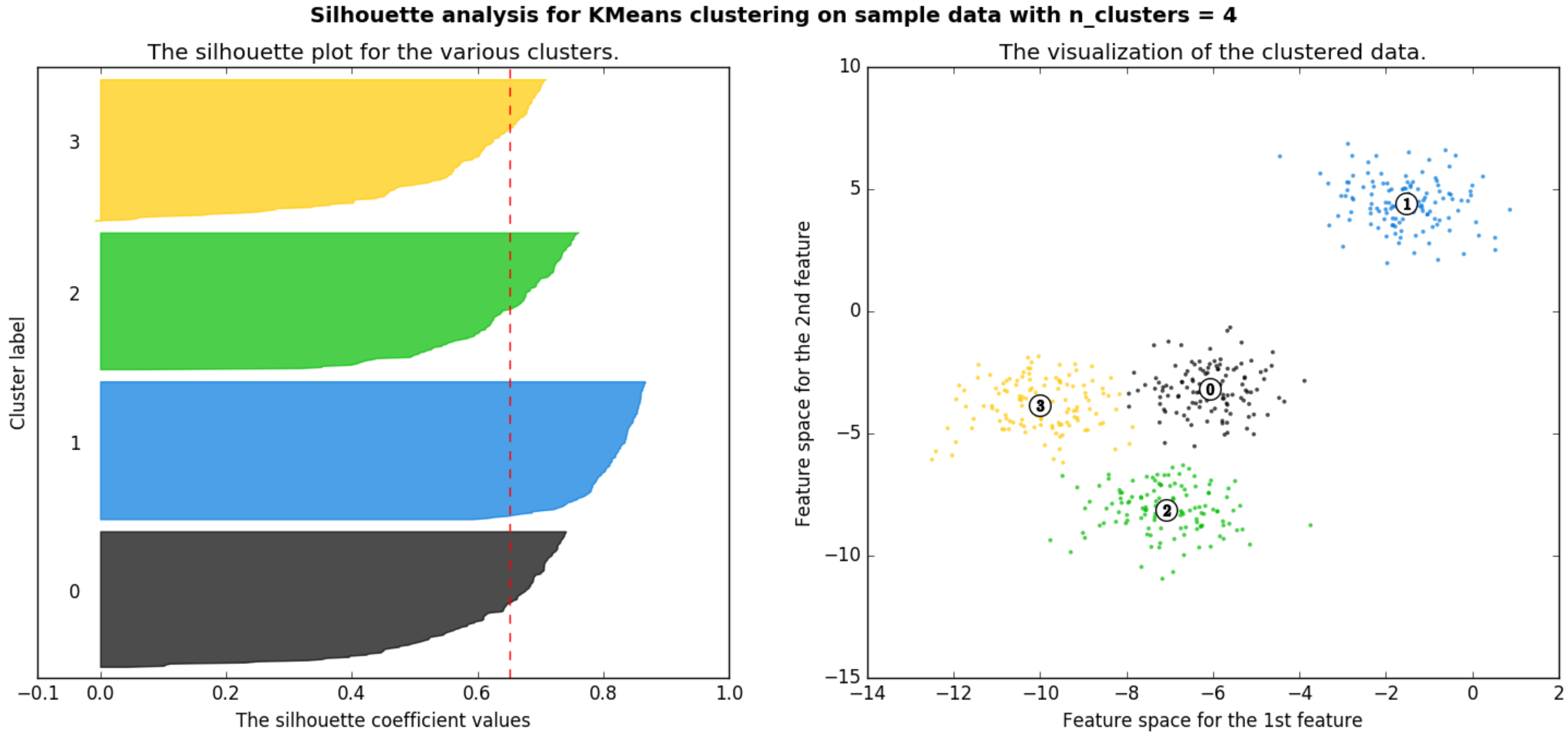
Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

An example



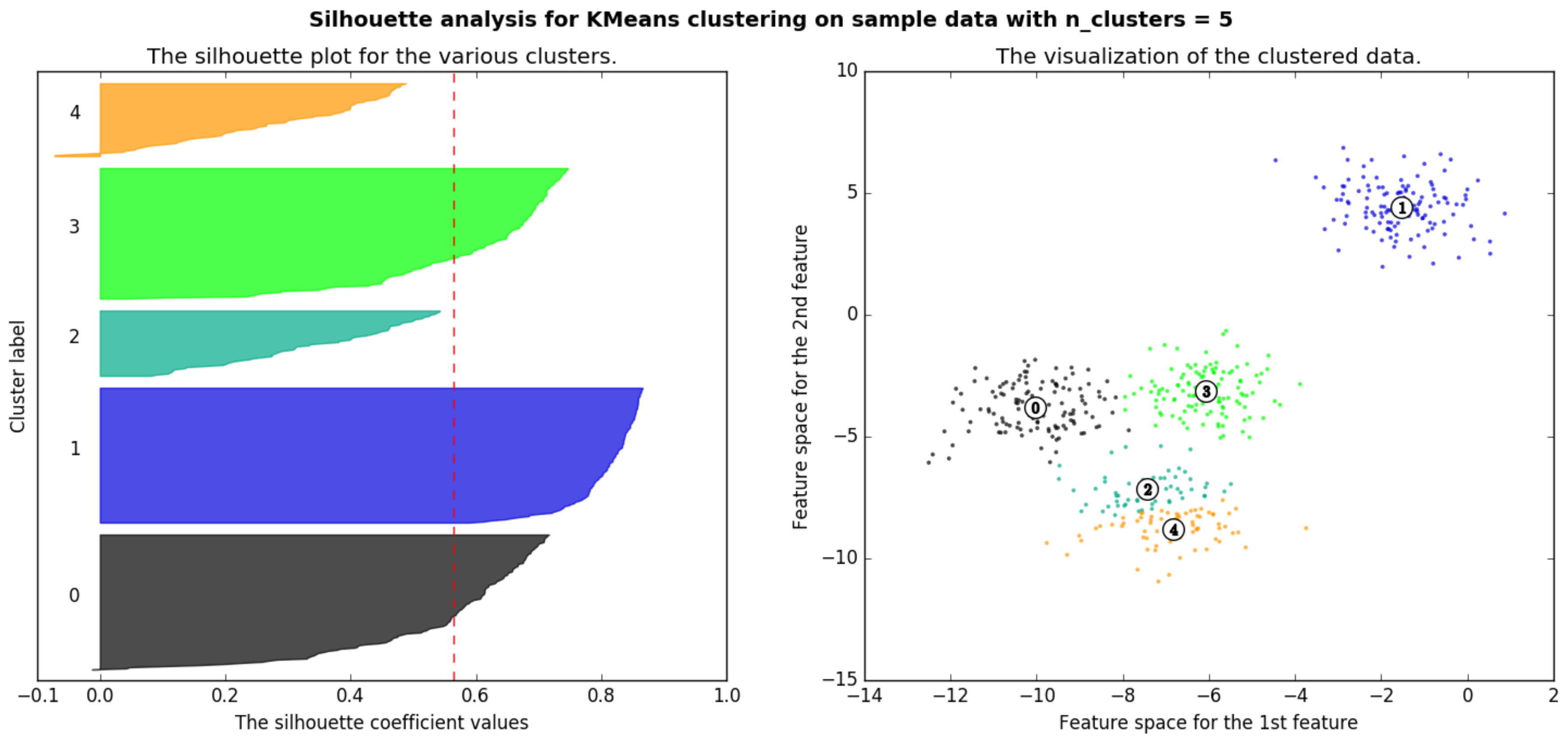
Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

An example



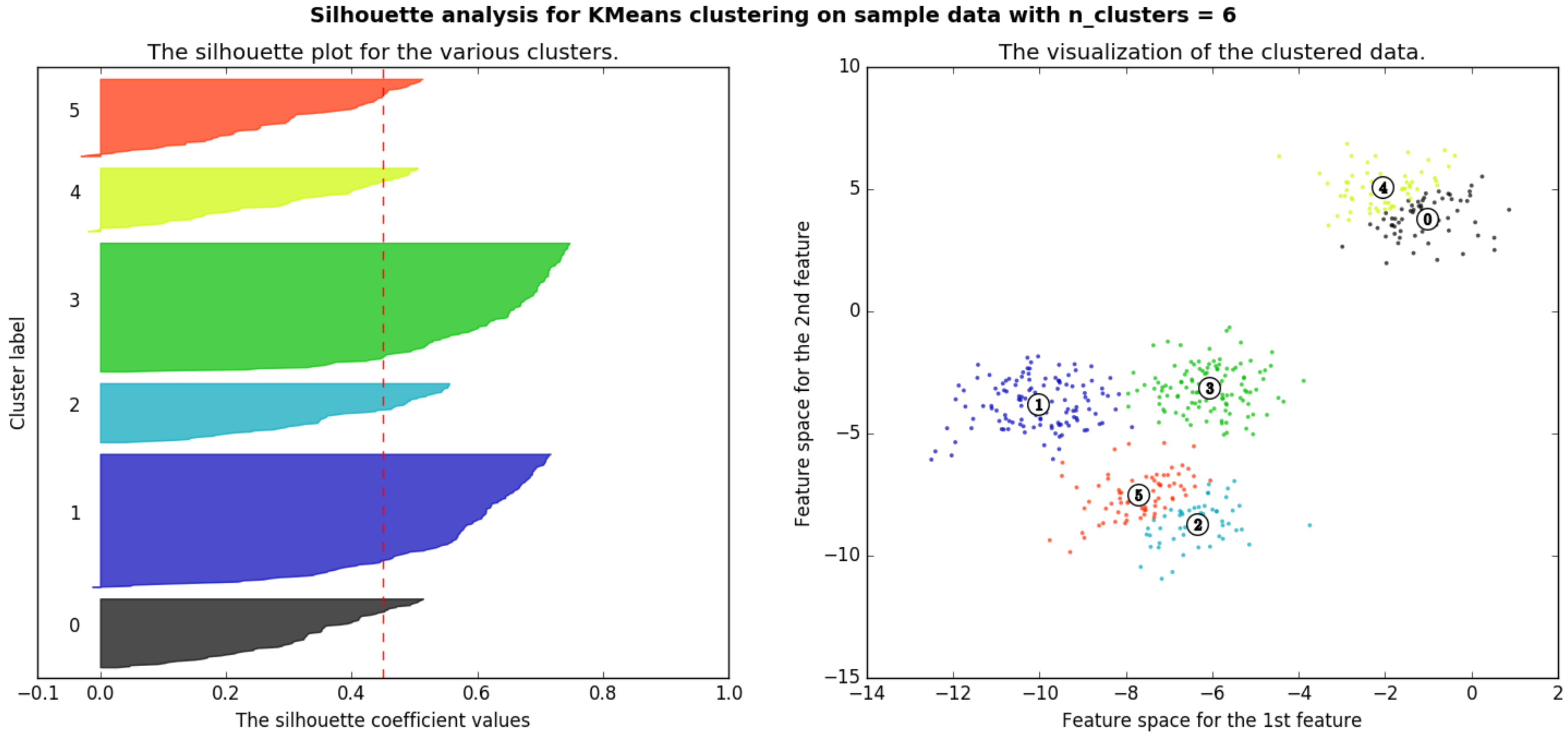
Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

An example



Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

An example



Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Outline

- Intro to unsupervised learning
- A categorization of major clustering methods
- Partitioning-based clustering: k-Means
- Partitioning-based clustering: k-Medoids
- Selecting k, the number of clusters
- Things you should know from this lecture & reading material

Overview and Reading

- Overview
  - Clustering basics
  - Partitioning based clustering
  - kMeans, kMedoids
  - Selecting number of clusters
- Reading
  - Tan P.-N., Steinbach M., Kumar V book, Chapter 8.
  - Data Clustering: A Review, https://www.cs.rutgers.edu/~mlittman/courses/lightai03/jain99data.pdf
  - *k*-means++: The Advantages of Careful Seeding, http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf
  - Extensions to the *k*-Means Algorithm for Clustering Large Data Sets with Categorical Values, Huang' 98.
  - The k-means clustering technique: General considerations and implementation in Mathematic, https://core.ac.uk/download/pdf/27210461.pdf

Hands on experience

- Try k-Means, k-Medoids on some dataset
  - E.g., use the seed dataset: https://archive.ics.uci.edu/ml/datasets/seeds
- Choose best *k* using Silhouette analysis: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Thank you

Questions/Feedback/Wishes?

Acknowledgements

- The slides are based on
  - KDD I lecture at LMU Munich (Johannes Aßfalg, Christian Böhm, Karsten Borgwardt, Martin Ester, Eshref Januzaj, Karin Kailing, Peer Kröger, Eirini Ntoutsi, Jörg Sander, Matthias Schubert, Arthur Zimek, Andreas Züfle)
  - Introduction to Data Mining book slides at http://www-users.cs.umn.edu/~kumar/dmbook/
  - Pedro Domingos Machine Lecture course slides at the University of Washington
  - Machine Learning book by T. Mitchel slides at http://www.cs.cmu.edu/~tom/mlbook-chapter-slides.html
  - (DTs) J. Fürnkranz slides from TU Darmstadt (https://www.ke.tu-darmstadt.de/lehre/archiv/ws0809/mldm/)
  - Thank you to all TAs contributing to their improvement, namely Vasileios Iosifidis, Damianos Melidis, Tai Le Quy, Han Tran.