UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

Unsupervised learning - algorithms for data partitioning (part 2)

Andrzej Janusz
Daniel Kałuża

THE PLAN

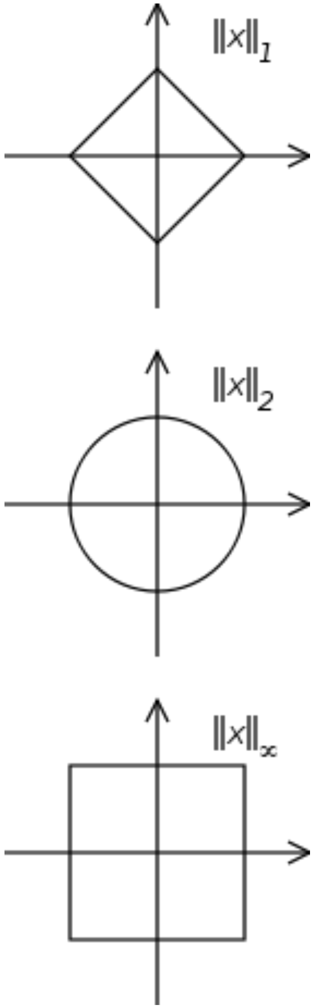
- A recap of the previous lectures.
- Hierarchical clustering.
- Density-based algorithms.
- Deep neural networks for clustering.
- Semi-(un)supervised learning.
- Examples of commonly used algorithms.
- Summary.

Unsupervised learning

- Given a cloud of data points, we want to understand its structure.
    - A common step in the exploratory data analysis.
    - Provides insights about the data.
    - Helps at identifying outliers.
    - <u>Can help in finding a reliable initial data batch for active learning!</u>
- Desirable properties of clustering:
    - Members of a cluster should be similar to each other.
    - Dissimilar samples should be placed in different clusters.
    - Clustering can be efficiently re-computed to allow interactive data analysis.

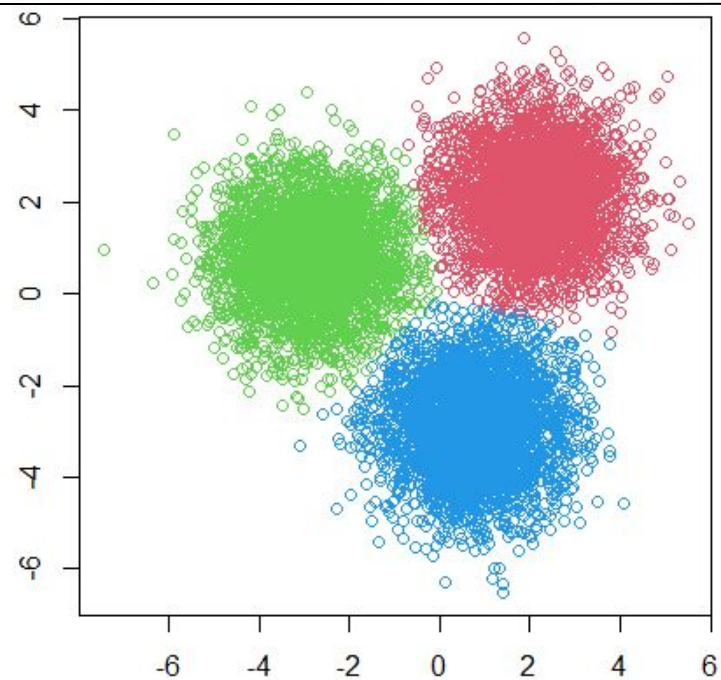
Distance and similarity metrics

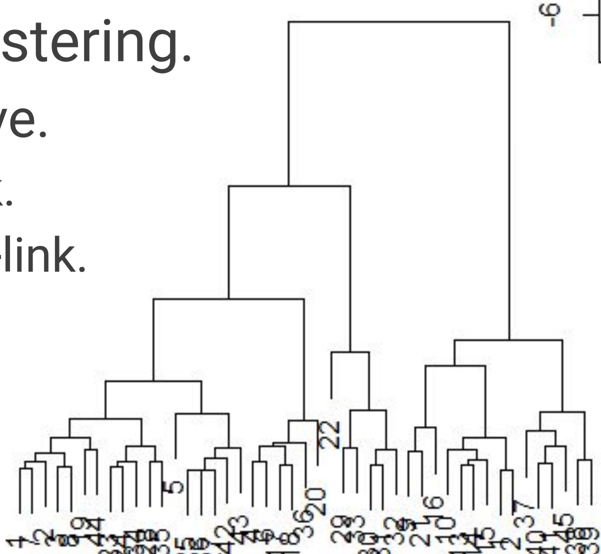
- Most of the clustering algorithms make use of some similarity or distance measure.
- Examples of popular distance measures:
  - Minkowski distances: $d(p,q) = \sqrt[m]{\sum_i^n (p_i - q_i)^m}$
    - Manhattan distance (p = 1)
    - Euclidean distance (p = 2)
  - The Canberra distance: $d(p,q) = \sum_i^n \frac{|p_i - q_i|}{|p_i| + |q_i|}$
  - We don't always need all properties of a distance metric.
- Examples of popular similarity measures:
  - Cosine similarity (of vectors): $sim(p,q) = \frac{p \cdot q}{\|p\| \, \|q\|}$
  - Jaccard similarity (of sets): $sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$



Taxonomy of clustering algorithms

- Flat-partition clustering:
  - Hard partitioning:
    - k-means, BFR.
    - EM, DBscan.
  - Soft partitioning:
    - fuzzy c-means.
    - rough c-means.
- Hierarchical clustering.
  - Agglomerative.
    - single-link.
    - complete-link.
  - Divisive.
    - Daina.
- Hybrids.

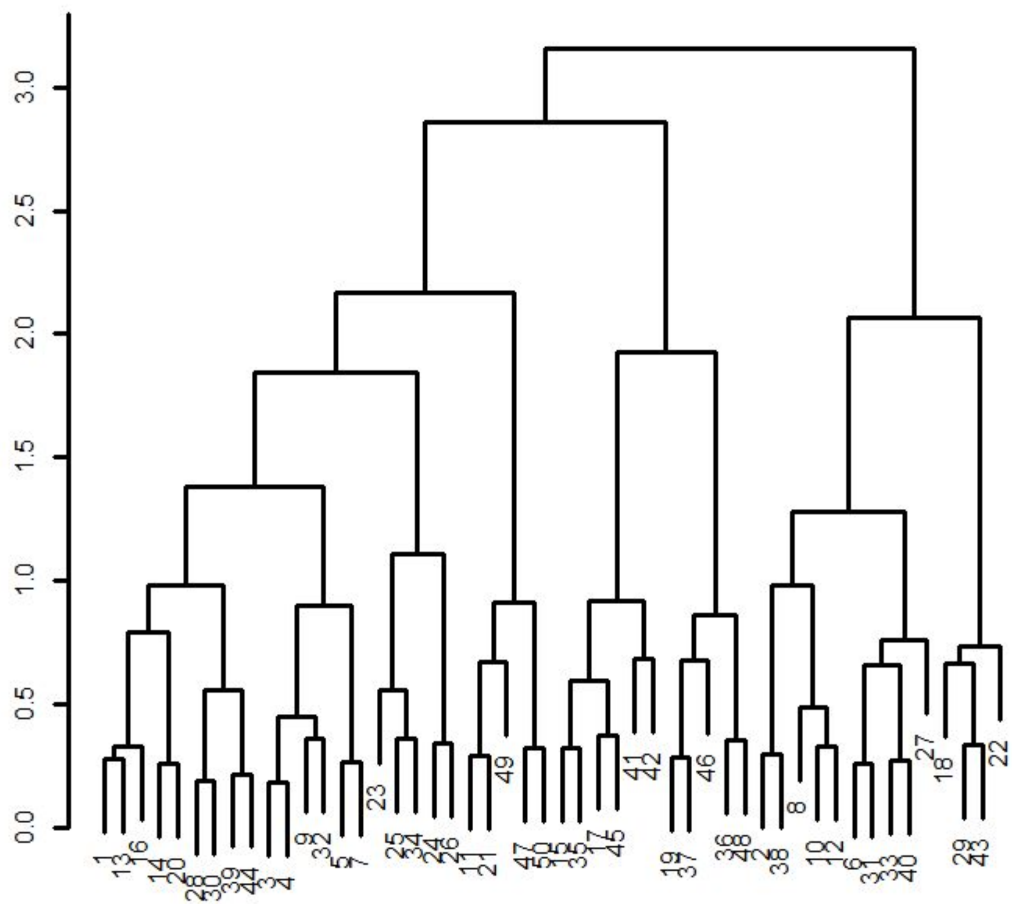




UNIVERSITY OF WARSAW

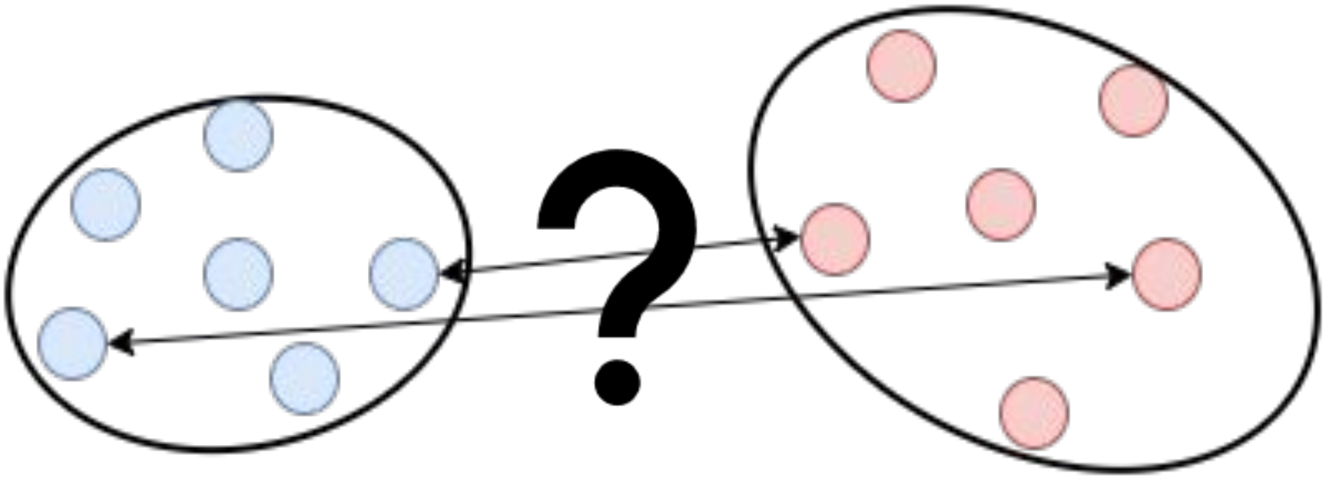
Agglomerative clustering

- The most common approach to the hierarchical clustering.
- Starts with as many clusters as there are data samples.
- In each iteration, join two most “closest” clusters until all samples are in a single group.
- How to define the cluster proximity?



Linkage functions

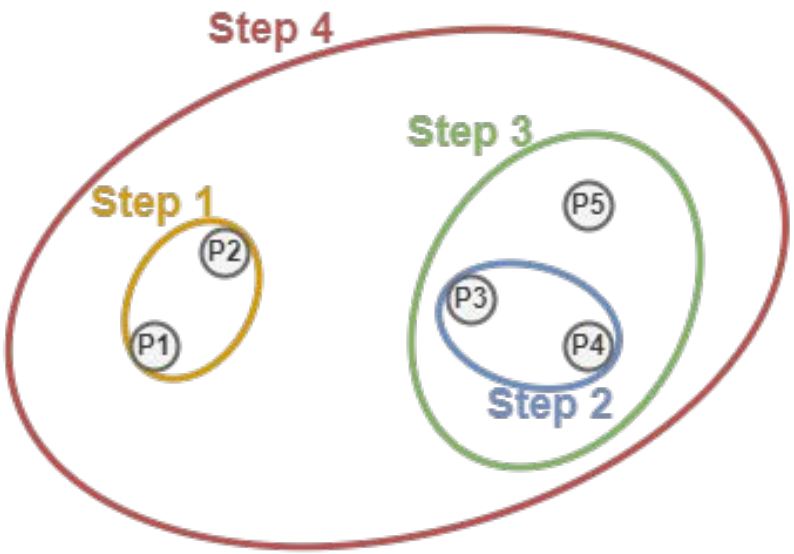
- Linkage functions are used to measure cluster proximity.
- The choice of a linkage function determines the shape of the resulting clustering tree.
- Typical implementations of hierarchical clustering use distance matrices to efficiently compute the linkage values.

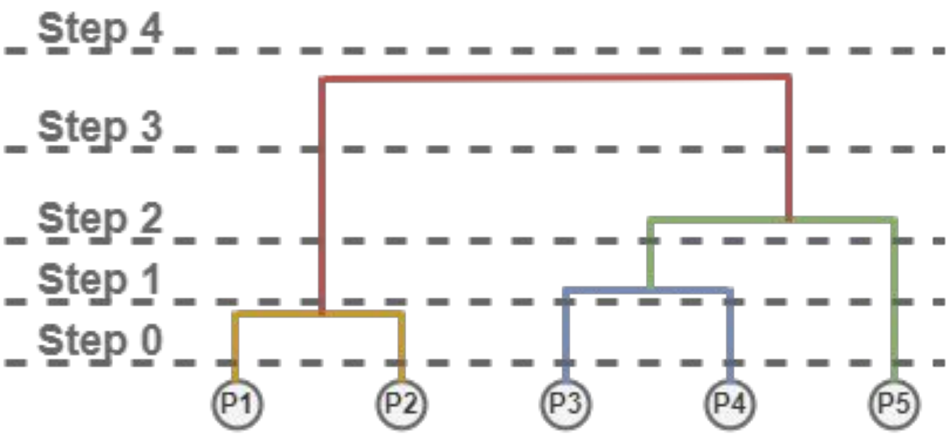


Exemplary linkage functions

- Single linkage: $$D(C_1, C_2) = \min_{c_1 \in C_1, c_2 \in C_2} d(c_1, c_2)$$
- Complete linkage: $$D(C_1, C_2) = \max_{c_1 \in C_1, c_2 \in C_2} d(c_1, c_2)$$
- Average linkage: $$D(C_1, C_2) = \frac{1}{|C_1| \times |C_2|} \sum_{c_1 \in C_1} \sum_{c_2 \in C_2} d(c_1, c_2)$$
- Centroid linkage: $$D(C_1, C_2) = d\left(\frac{1}{|C_1|} \sum_{c_1 \in C_1} c_1, \frac{1}{|C_2|} \sum_{c_2 \in C_2} c_2 \right)$$
- Ward's linkage: $$D(C_1, C_2) = \sum_{c \in C_1 \cup C_2} d\left(c, \frac{1}{|C_1| + |C_2|} \sum_{c' \in C_1 \cup C_2} c' \right)^2$$

An example

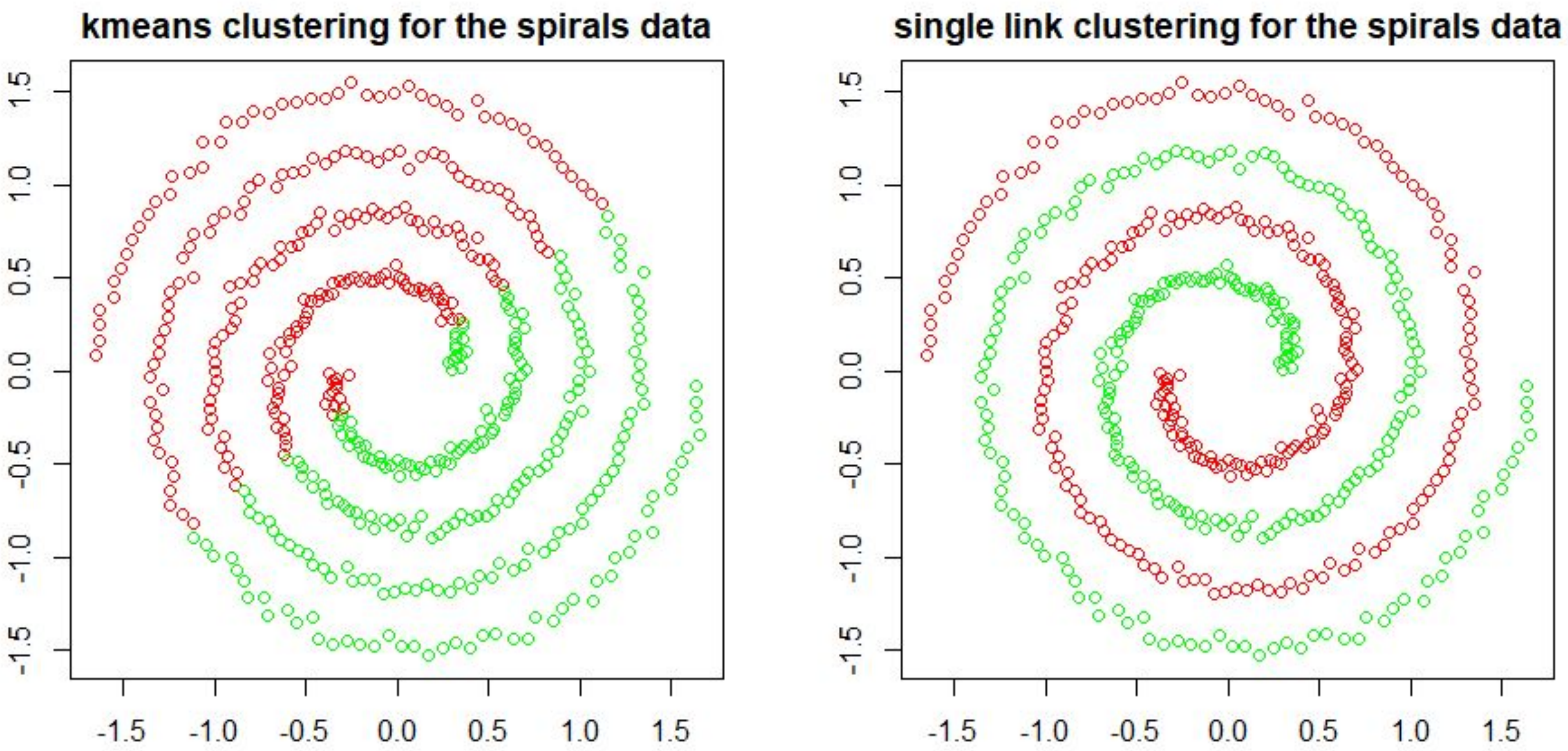

Step 4
Step 3
Step 1
P2
P5
P3
P1
P4
Step 2


Step 4
Step 3
Step 2
Step 1
Step 0
P1
P2
P3
P4
P5

Divisive clustering

- Unlike in the agglomerative approach, we start from a single cluster containing all data.
- We recursively divide "*the largest*" cluster into two smaller ones, in a way that maximizes a predefined gain function.
  - We measure the cluster "*size*" in terms of its *compactness*, not the actual cardinality.
  - Gain functions measure the difference between the cluster "*size*" and the sum of resulting cluster "*sizes*".
- An exemplary algorithm: DIANA.
  - The "*size*" is expressed as the cluster variance.

Discovering complex cluster shapes



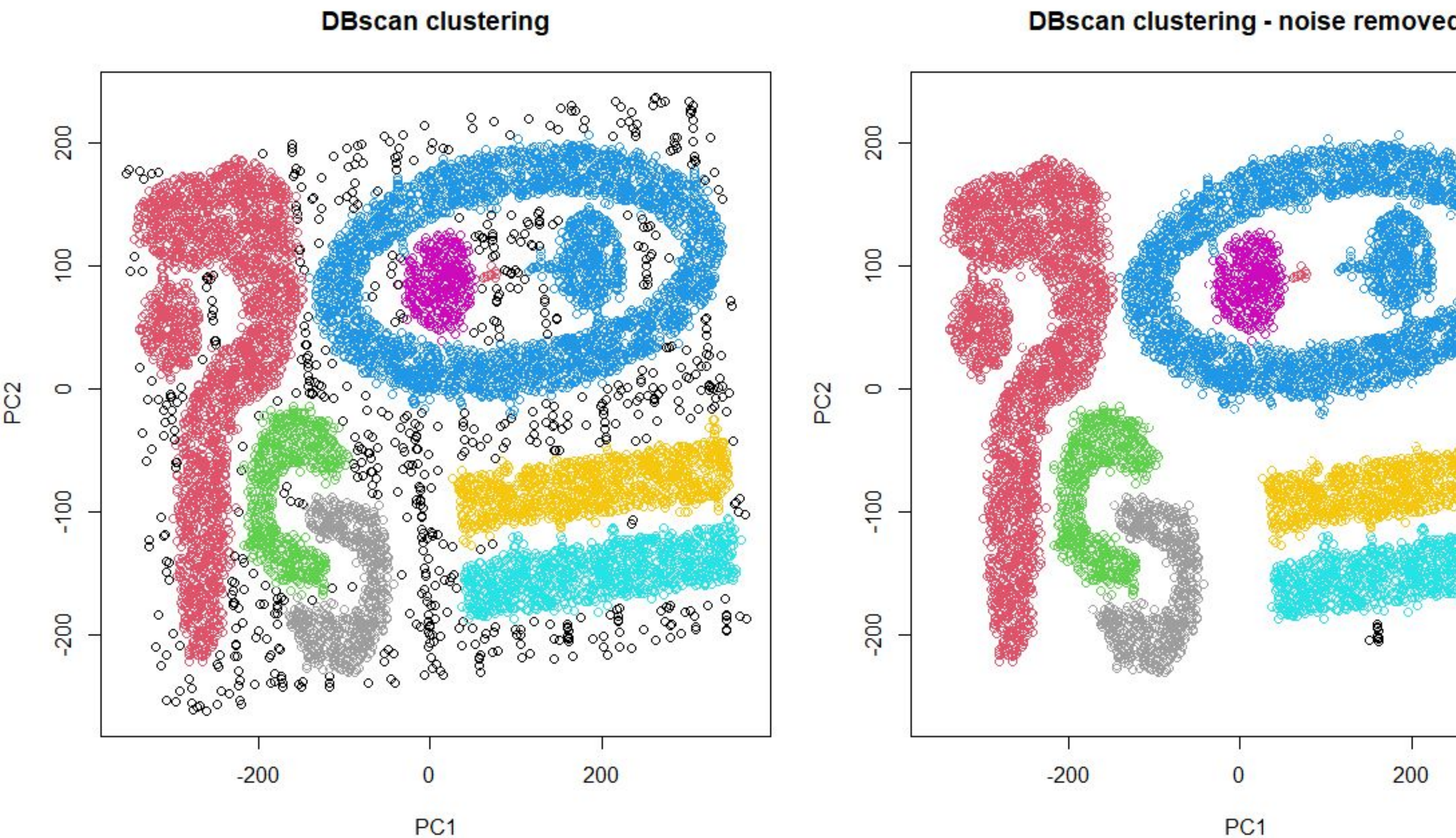
Problems with hierarchical clustering

- **<u>Computational complexity!</u>**
  - For most of the linkage functions, the agglomerative approach has at least $O(n^2 \log(n))$ time complexity.
  - Hierarchical clustering usually requires the computation of a distance matrix (at least $O(n^2)$ memory requirements).
- Quality of hierarchical clustering depends on the selection of a distance metric and the linkage function.

Density-based clustering

- Density-based clustering algorithms identify regions of high data density that are separated by regions of low data density.
  - Density is typically understood as a number of data samples inside a sphere of a fixed radius.
- Major features of the density-based clustering:
  - Can discover clusters of arbitrary shape.
  - Can handle noise and detect outliers.
  - Need density parameters as a clustering condition.
    - Often doesn’t need explicit number of clusters.

Example

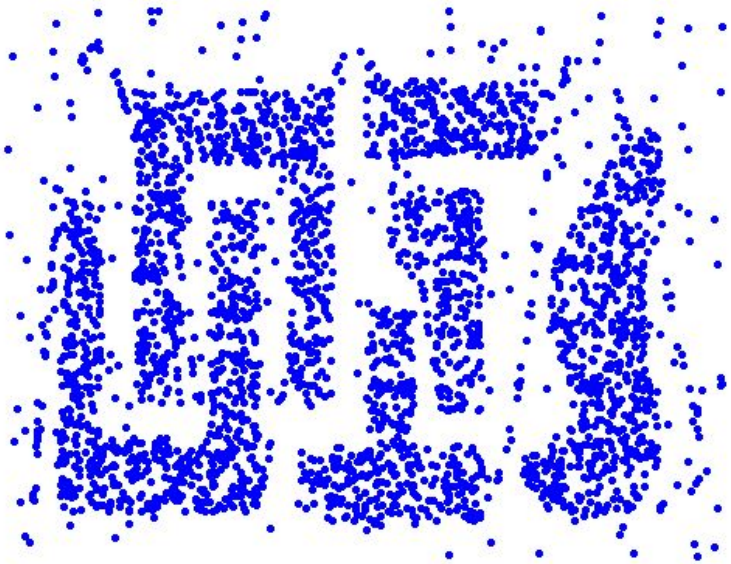

DBscan clustering
DBscan clustering - noise removed
PC1
PC2
-200
0
200
100
-100

The DBSCAN algorithm

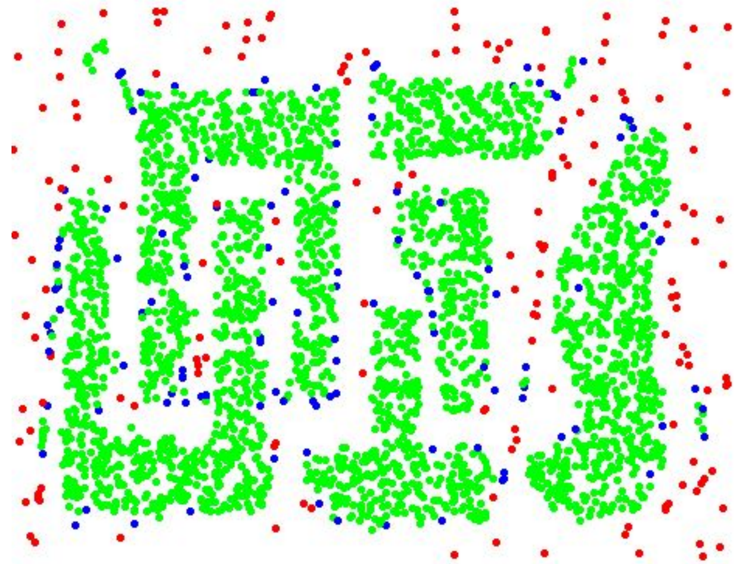
- **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise:
  - One of the most widely used clustering algorithms.
- Main input parameters:
  - ***Eps*** - the radius of local neighborhoods.
  - ***MinPts*** - the minimal number of points in a neighborhood.
- The algorithm divides data into three categories:
  - Core points (CP) - dense cluster interiors.
  - Border points (BP) - have fewer than *MinPts* neighbors within *Eps*, but are in the neighborhood of some core point.
  - Noise points (NP) - all remaining data points.

Core points, border points, and noise points

Original data:

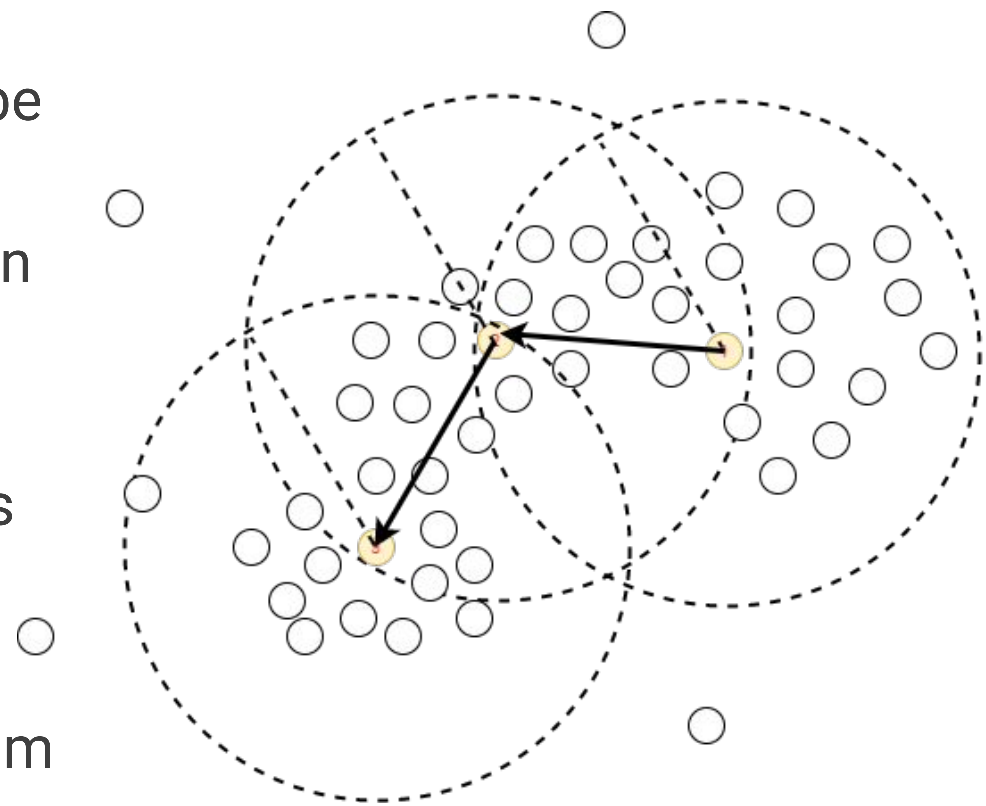


Core points, **border points**, and **noise points**:



How does it work?

- Two points are density-reachable is they can be connected by dense neighborhoods (there is a chain of core points with distances < Eps between them).
- Any two density-reachable CPs will be placed in the same cluster.
- If a BP is density-reachable from a CP, it will be placed in the same cluster.
- All NPs will be discarded.



MinPts = 20

The algorithm

Set clusterID := 0; Draw a single sample u from U:

1. If u <u>is not a CP</u>, assign a null label to u (e.g., zero label).
2. If u <u>is a CP</u>, create a new cluster with clusterID := clusterID+1;

   Assign the label clusterID to u;

   Find all points density-reachable from u and assign them to clusterID (you may re-assign cases with the null label);

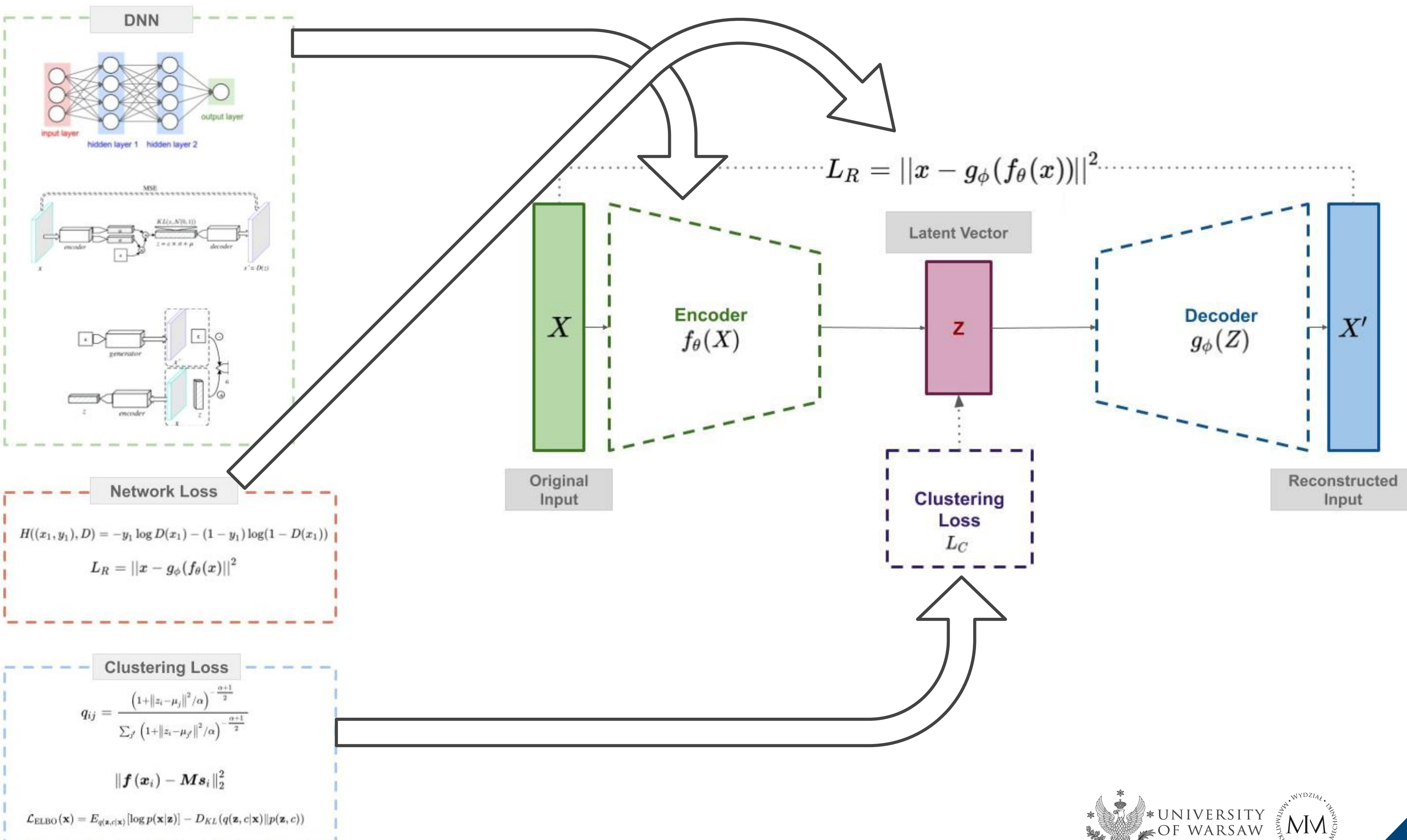
Repeat the above procedure until you visit all cases from U.

DBSCAN implemented using, e.g., KD-trees, has O(nlog(n)) time complexity.

Deep learning vs. the clustering task

- Typically, we have a lot of unlabeled data.
  - It seems to be a perfect circumstance for deep learning.
- Deep clustering models have two essential components:
  - **Deep neural network architecture** for the representation learning.
    - Auto-encoders - a very common choice.
    - Generative model-based architectures.
    - Direct cluster optimization approaches.
  - **Loss functions.**
    - Cluster assignment losses, e.g., *k-means loss*.
    - Cluster regularization losses, e.g., *locality preserving loss*, *group sparsity loss*.

Building blocks of deep clustering methods



Exemplary DNN clustering algorithm

- Deep Clustering Network (DCN).
  - Simple and intuitive model.
  - Uses an autoencoder to learn a data representation suitable for k-means clustering.
  - Uses the k-means loss function:

$$L_{k-means} = \sum_{u \in U} \left(\ell\left(g(f(u)), u\right) + \frac{\lambda}{2} ||f(u) - M s_u||_2^2 \right)$$

    where *f*, *g* are encoder and decoder functions, respectively.
  - Optimizes the representation and the centroids matrix *M*.

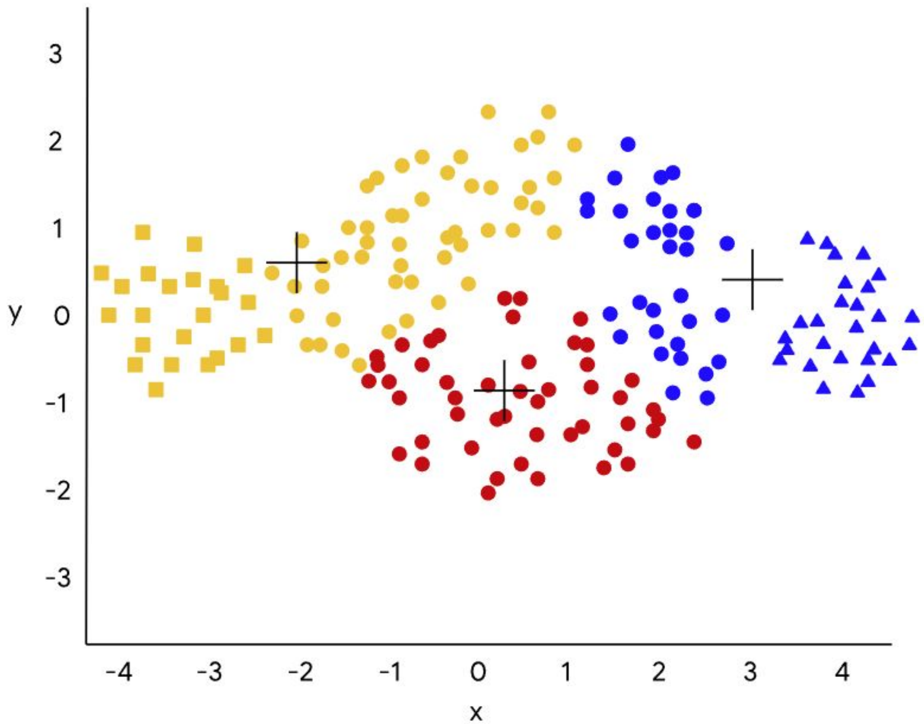
- Obtains good results on popular benchmarks:
  - MNIST: NMI = 0.63
  - 20Newsgroup: NMI = 0.48
  - RCV1(20): NMI ranging from 0.88 to 0.63 (depending on the number of considered clusters).

Semi-(un)supervised learning

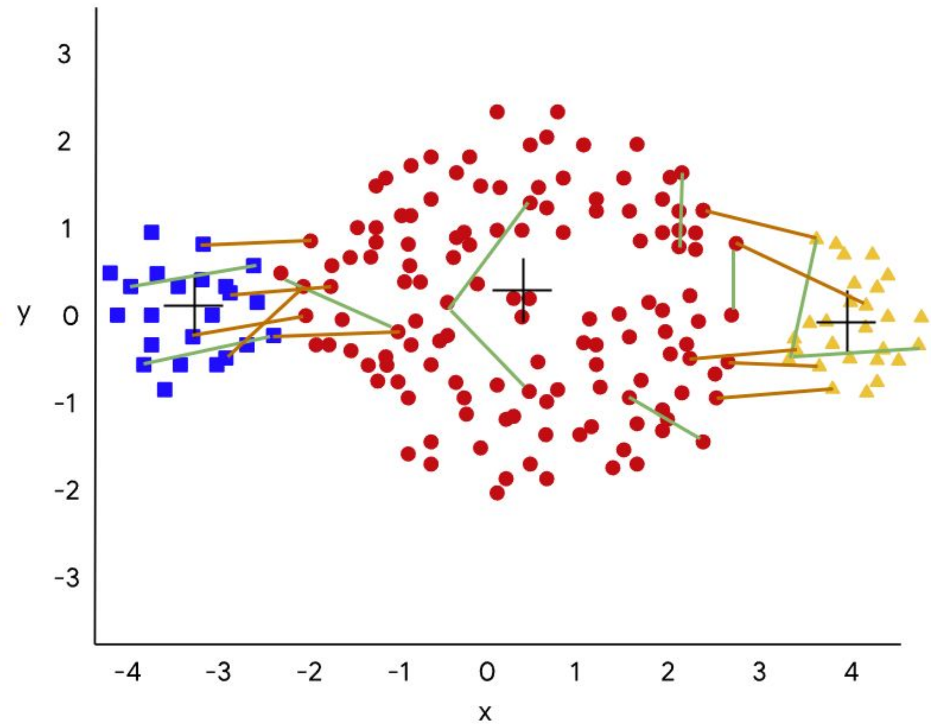
- Sometimes, we may not have class labels but we can get some similarity constraints:
  - Must-links - two cases have to be placed in the same cluster.
  - Cannot-links - two cases must not be in the same cluster.
- Exemplary algorithm: COP k-means.
  - Check constraint consistency. Extend and fix the constraint set if possible.
  - Initialize k cluster centers.
  - Repeat until convergence:
    - **Assign phase**: objects are assigned to the closest cluster center without violating constraints.
    - **Update phase**: cluster centers are updated to the mean of constituent objects.

Semi-(un)supervised clustering example

standard k-means



constrained k-means



Active learning can be used to acquire constraints!

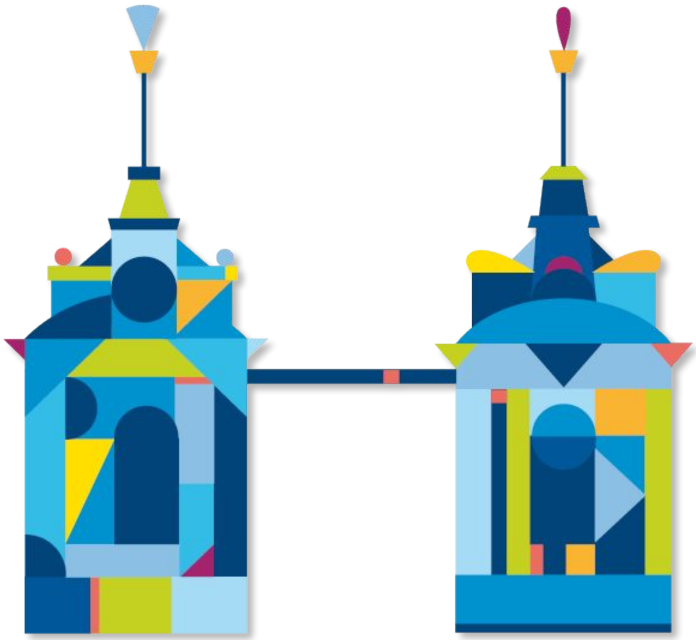
Summary

- We continued the discussion on the unsupervised learning task.
- We discussed agglomerative and divisive approaches to hierarchical clustering.
- We talked about density-based clustering methods using as an example the DBSCAN algorithm.
- We considered deep learning point of view on the clustering task and discussed the Deep Clustering Network model.
- We talked about clustering with constraints with an example of COP k-means.

UNIVERSITY OF WARSAW

Literature:

1. L. Kaufman, P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons. (2009).
2. E. Schubert, J. Sander, M. Ester, H.P. Kriegel, X. Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. ACM Trans. Database Syst. 42, 3, Article 19, 21 pages. https://doi.org/10.1145/3068335. (2017).
3. A. Ganesan, F. Ferraro, T. Oates. Locality Preserving Loss: Neighbors that Live together, Align together. In Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 122–139, Kyiv, Ukraine. ACL. (2021).
4. B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong. Towards K-means-friendly spaces: simultaneous deep learning and clustering. In Proceedings of ICML 2017. JMLR.org, 3861–3870. (2017).
5. Y. Hong, S. Kwong. Learning Assignment Order of Instances for the constrained k-means clustering algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Vol 39, No 2. April, (2009).
6. L. Saul, S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research, 4, 119-155. (2003).
7. M. Bilenko, S. Basu, R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In Proceedings of ICML 2004. ACM, New York, NY, USA, 11. https://doi.org/10.1145/1015330.1015360 (2004).



QUESTIONS OR COMMENTS?

a.janusz@mimuw.edu.pl

or

d.kaluza@mimuw.edu.pl