UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

Unsupervised learning - algorithms for data partitioning (part 1)

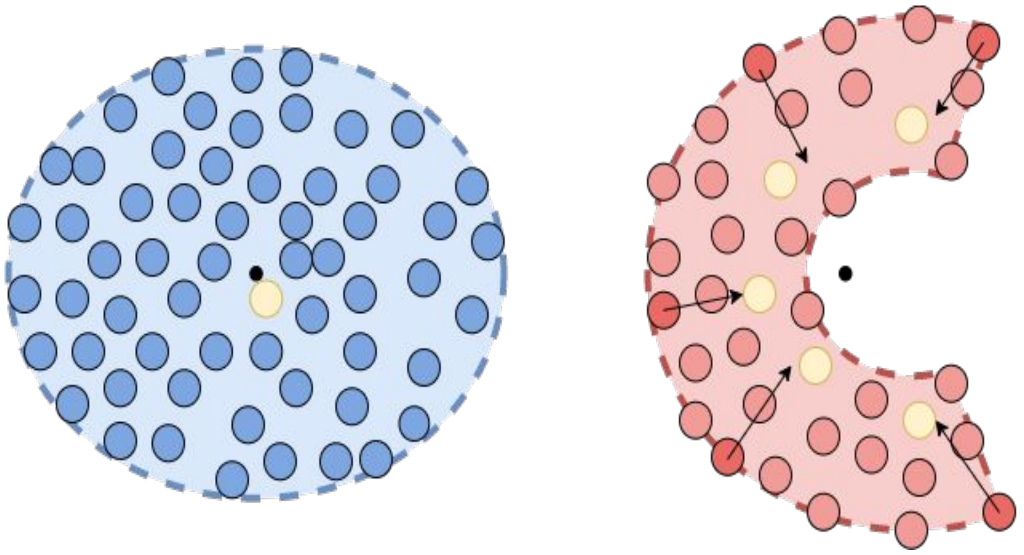
Andrzej Janusz
Daniel Kałuża

THE PLAN

- A recap of the previous lectures.
- What is the unsupervised learning?
- How can we evaluate the quality of clustering results?
- Various approaches to clustering.
- Examples of commonly used algorithms.
- Summary.

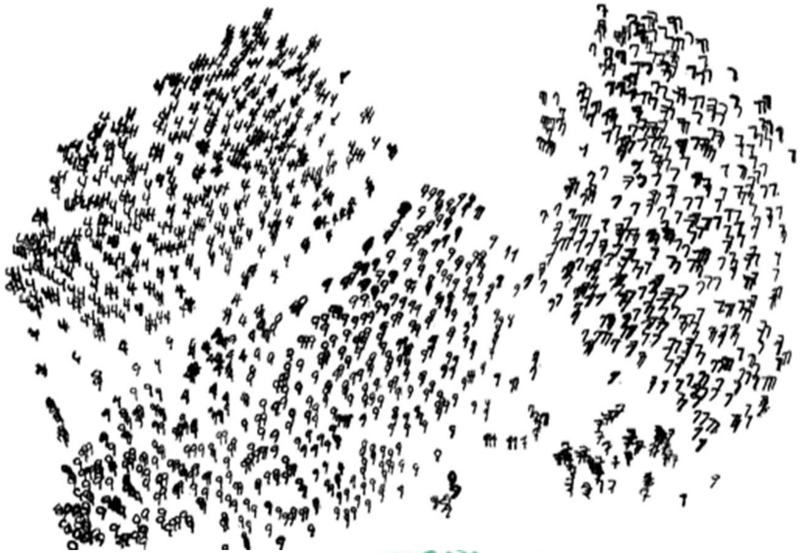
Choosing the initial data batch for AL

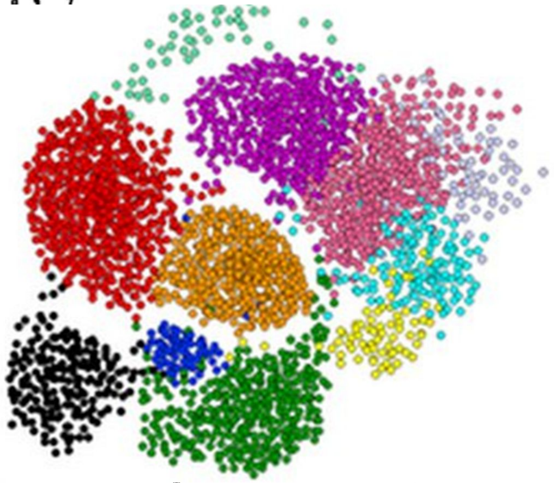
- We can use a clustering algorithm to divide the available data pool into a number of subsets.
- We independently sample the initial queries from each discovered data cluster.
  - We can select the samples nearest cluster centers.
  - Or we select cluster representatives.
- We aim to select representative, yet diverse cluster members.



Unsupervised learning

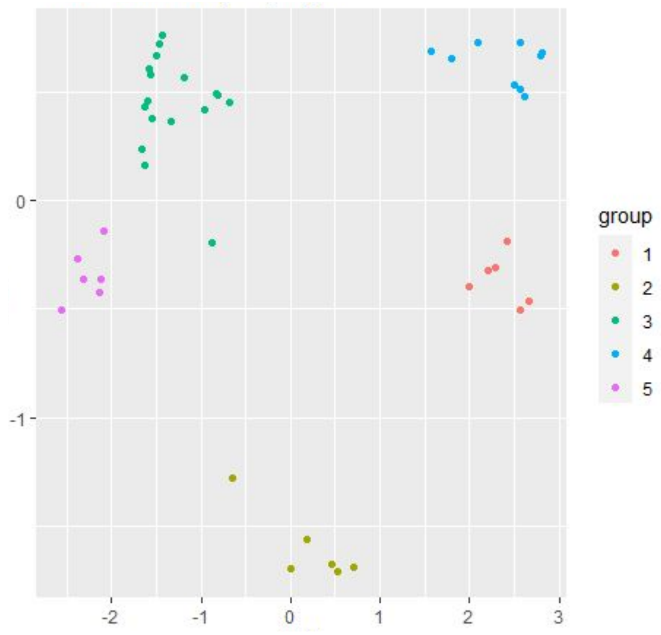
- Given a cloud of data points, we want understand its structure.
  - A common step in the exploratory data analysis.
  - Provides insights about the data.
  - Facilitates interaction with the data.
  - Helps at identifying outliers.
- Given a set of data points, group the points into some number of clusters, so that:
  - Members of a cluster are similar to each other.
  - Dissimilar points are in different clusters.





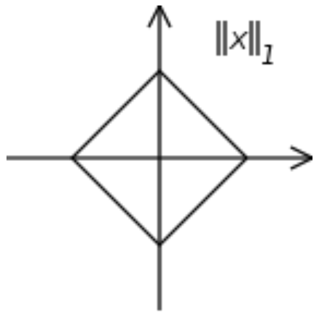
Why is it difficult?

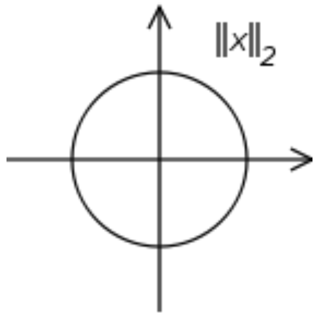
- When data have only two or three dimensions, the clustering looks easy.
- When the number of samples is low, the clustering looks easy.
  - In those cases, <u>the looks are not deceiving</u>.
- Typically, the dimensionality of real-world data is high.
  - In a highly dimensional space nearly all points are far away from each other.
  - Distance metrics may become unreliable.

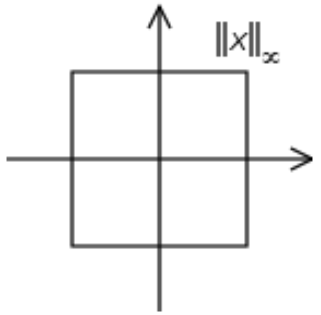


Distance and similarity metrics

- 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): $sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$







How can we evaluate clustering results?

- The quality of clustering is subjective.
  - The same objects can be clustered differently depending on the objective.
  - The right choice of the similarity measure is of paramount importance.
- External clustering quality measures.
  - The ground-truth clustering must be available.
  - Quality measures insensitive to permutations of cluster IDs, e.g., *clustering entropy, expected cluster purity*.
- Internal clustering quality measures.
  - Aim to reflect the cohesion and separation of clusters, e.g. *WSS, silhouette index, Dunn index*.

Image: Freepik.com

An example - (negative) clustering entropy

	Class 1	Class 2	Class 3	Total:
Cluster 1	45	10	0	55
Cluster 2	5	5	40	50
Cluster 3	0	35	10	45
Total:	50	50	50	150

- To compute the entropy of clustering results, we need to consider the class distribution in each cluster:

$$H_j = -\sum_{i}^{K} p_{ij} \log p_{ij} \qquad H_1 = -\left(\frac{45}{55} \cdot \log(\frac{45}{55}) + \frac{10}{55} \cdot \log(\frac{10}{55}) + 0\right)$$

- The clustering entropy is the expectation over all clusters:

$$H = -\sum_{j}^{M} \frac{n_j}{n} H_j \qquad H = -\left(\frac{55}{150} H_1 + \frac{50}{150} H_2 + \frac{45}{150} H_3\right)$$

An example - WSS and the silhouette index

- WSS measures the cluster compactness. If c_i is the center of the cluster C_j then WSS is

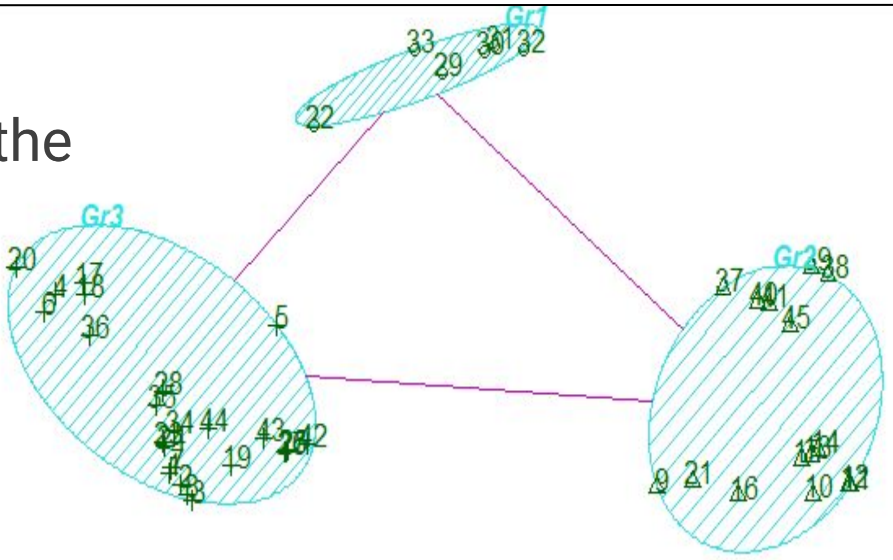
$$WSS(C) = \sum_{j}^{k} \sum_{u \in C_j} (u - c_j)^2$$

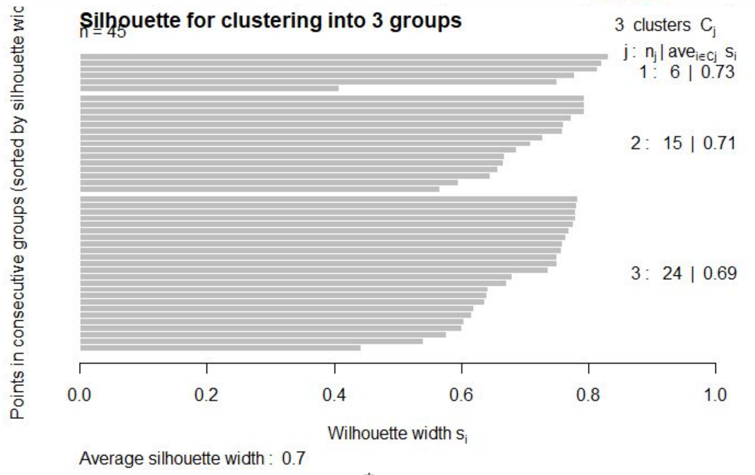
  - WSS is monotonic with k.

- The silhouette index also takes into account the distinctiveness of clusters.

$$Silh(C) = \frac{1}{N} \sum_{u} \frac{b(u) - a(u)}{max(a(u), b(u))}$$

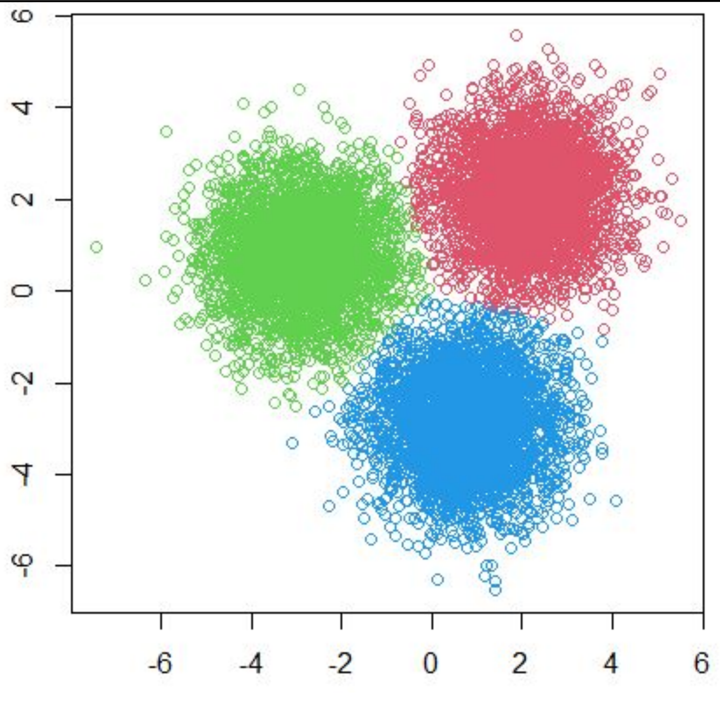
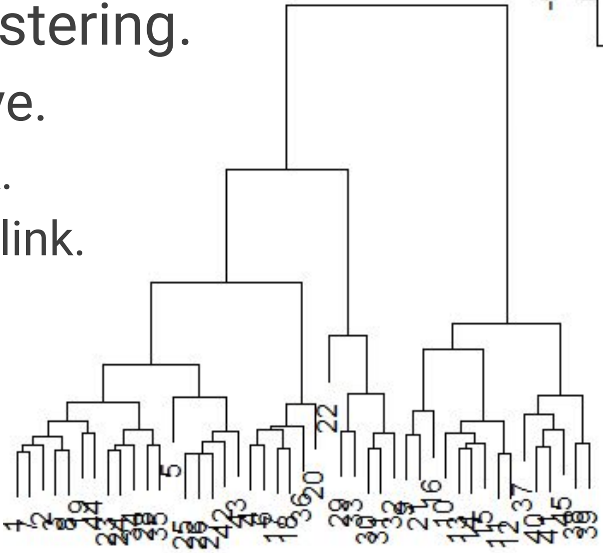
where *a(u)* and *b(u)* are avg. distances to samples from the same and the closest different cluster, respectively.





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.

The famous k-means algorithm

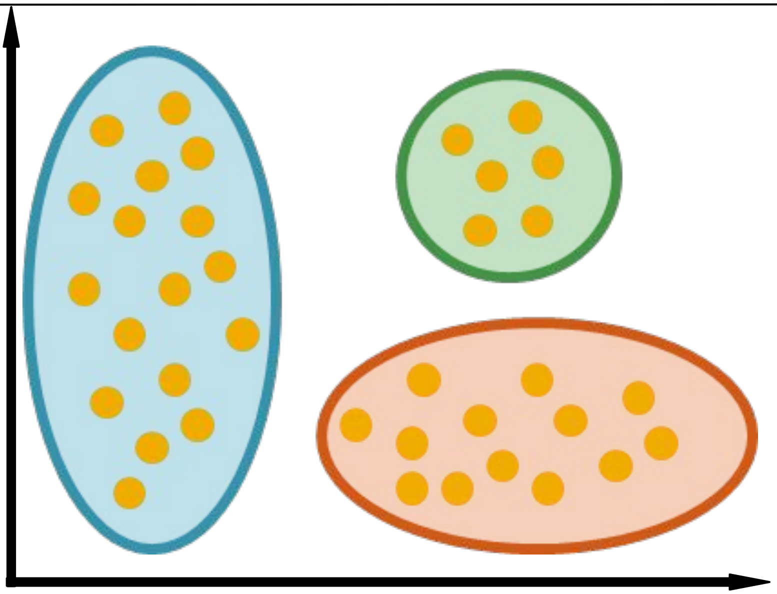
- One of the most commonly used DM algorithm:
  1. Initialize the algorithm by randomly selecting k points in the data space - they are the initial cluster centers.
  2. Repeat the following two steps until the algorithm converges:
     - Assign every data point to the nearest cluster center.
     - Compute new cluster centers by averaging the corresponding data points.
- <u>The most important is the initialization step!</u>
- The algorithm has linear time complexity with regard to the number of data points.

Problems with k-means

- We need to guess the right number of clusters.
  - We can use an intrinsic quality metric.
- Clusters need to be separable using hyper-spheres.
  - More complex shapes are problematic.
  - Outliers may distort the clustering results.
- We are constrained to the euclidean distance.
  - Finding cluster centers can be difficult for other metrics.
  - May not work too well for high-dimensional data.

The BFR algorithm - general information

- BFR is a modification/improvement of the k-means algorithm.
  - It can handle a broader variety of cluster shapes (i.e., ellipses).
  - It is designed to work with out-of-memory data.
- The main assumption is that clusters are Gaussian along each axis.
  - The cluster centers are means of points - just as in the k-means algorithm.
  - Variances in each dimension of the data can be different.
- BFR is memory-efficient.



The BFR algorithm - the idea

- In BFR data is processed in chunks - one chunk at a time.
  - The chunk size depends on available memory.
  - Data points from a chunk are summarized to make room for the next chunk.
- We assign each data point to one of three sets:
  - **Discard set (DS)** contains points that are close enough to a cluster center to be assigned to a cluster, summarized, and forgotten.
  - **Compression set (CS)** contains groups of points that are close to each other but not sufficiently close to any cluster center.
  - **Retained set (RS)** contains isolated data points that are still waiting to be assigned to some compression set.

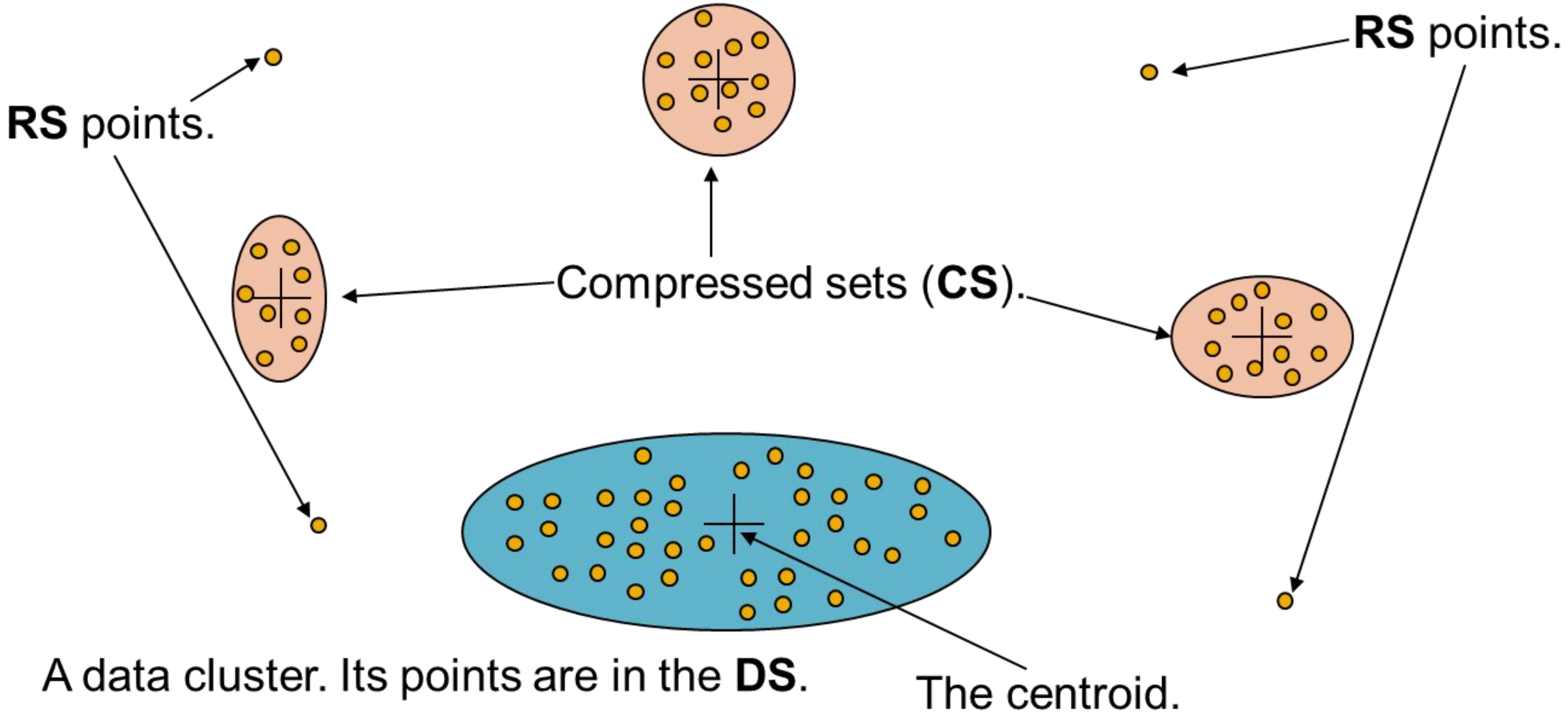
Data summaries in BFR

- Groups of points in BFR are summarize using simple statistics:
  - The number of points in the group: ***N***.
  - The vector ***SUM***, whose ith component is the sum of the attribute values of the points in the ith dimension.
  - The vector ***SSQ***, whose ith component is the sum of squares of the attribute values in the ith dimension.
- The summaries are very useful:
  - **The center** of a group can be computed as: ***SUM / N***.
  - **The variance** in the ith dimension: $(SSQ_i / N) - (SUM_i / N)^2$
- <u>We may summarize a whole cluster with just ***2d + 1*** numbers!</u>

The BFR algorithm (1)

- We load the first data chunk and we randomly choose ***k*** data points - they will be the initial cluster centers.
  - We can also do it by, e.g., running a clustering algorithm on a small sample of data.
- We identify points in the chunk that are sufficiently close to some cluster center, add them to the cluster, update the cluster summary, and discard them (we add them to DS).
  - A few passes through the data chunk might be needed.
  - Alternatively, we may do only one pass, and make all cluster updates at once.
- We cluster the remaining points using any in-memory algorithm.
  - Clusters are summarized in the compressed set (CS).
  - Outlying points go to the retained set (RS).

BFR “galaxies” of points



The BFR algorithm (2)

- Load remaining chunks of data (one at a time) and repeat the previous two steps until all chunks are processed.
  - Add RS points from previous iterations when clustering points that were not assigned to any cluster.
  - Consider merging groups in the CS after processing each chunk.
- After processing the last chunk, try to merge all data not in DS to the clusters.
  - Try merging all groups in the CS to the closest cluster.
  - Alternatively, create new clusters.
  - Leave points from RS as outliers or assign them to the closest cluster.

The two pivotal decisions

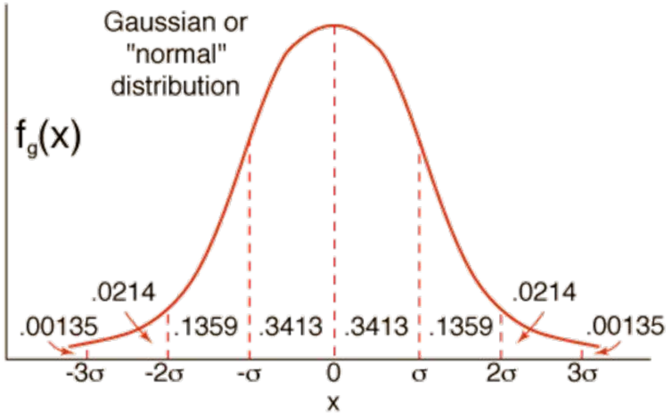
- When should we merge two groups from the CS?
  - Compute the variance of the resulting group and merge them if it is below a predefined threshold.
  - It can be easily done using the summaries.
- When can we add a new point to a summarized cluster?
  - We may estimate the likeliness that a point comes from the same distribution as the cluster.
    - Note that we assume that clusters are normally distributed.
  - We use the Mahalanobis distance metric.
  - All computations are done using the summaries.

Mahalanobis distance metric

- The Mahalanobis distance:

$$d(u,c) = \sqrt{\sum_{i=1}^{d} \left(\frac{u_i - c_i}{\sigma_i}\right)^2}$$

  - The computation of Mahalanobis distance between a point and a cluster summary is easy.
  - The standard deviations are just the square roots of the variances.

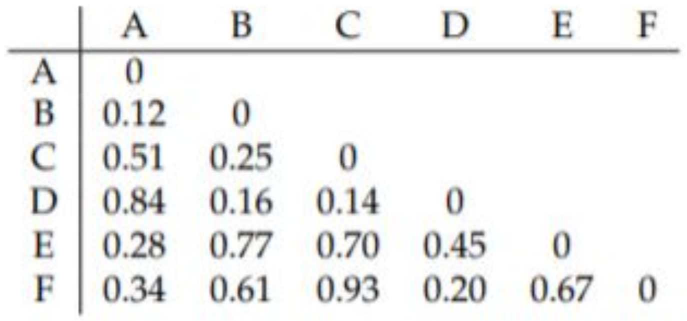


- If the cluster is normally distributed in d dimensions, then points within one standard deviation from the center have Mahalanobis distance $\leq \sqrt{d}$.
  - We assign a point to a cluster if its distance is smaller than a threshold, e.g. two standard deviations.

Finding medoids - the PAM algorithm

- A different extension of k-means.
  - Clusters are built around central objects, i.e., the medoids.
  - We make use of a distance matrix.
    - That comes with a performance cost…
  - We can use any distance metric or similarity function to define the medoid.

- The algorithm looks the same as the k-means but instead of a cluster center, we find the cluster medoid.
  - In each iteration for every group, we select the object for which the sum of distances to other objects in the cluster is minimal.



	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

Pros and cons of PAM

- Pros:
  - We get the most representative sample for free.
  - We can perform the clustering of any type of objects, not only points/vectors in a metric space.
  - PAM is more robust to outliers than k-means.
- Cons:
  - The similarity matrix requires a lot of computations and memory - $O(N^2)$ complexity.
  - Similar assumptions about the cluster shape and separability to k-means.

Approximating PAM - the CLARA algorithm

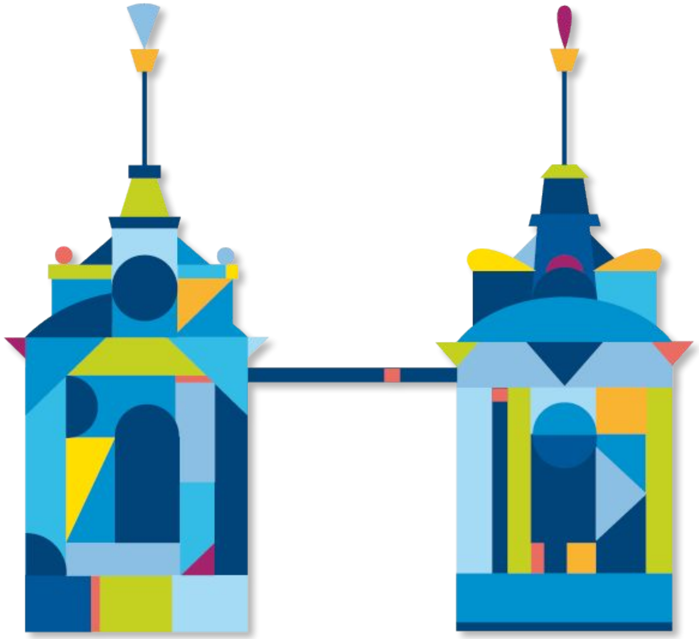
- Clustering LARge Applications:
  - Select L subsets of M samples.
  - For each subset:
    - Cluster the subset optimally using PAM.
    - Assign all other data samples to the nearest cluster medoid.
    - Compute some evaluation metric, e.g. silhouette index.
  - Select the clustering with the highest evaluation score.
- If we set values of L and M such that $LM^2 \approx N$, the complexity is linear.
- The idea behind the approximation in CLARA is generic.

Summary

- We discussed the problem of unsupervised learning.
- We considered a few examples of evaluation metrics for clustering results.
- We talked about several clustering algorithms that belong to the flat-partitioning category.
- In particular, we discussed two improvements over the k-means algorithm that are suitable for clustering large data pools, i.e., BFR and CLARA.

Literature:

1. R. C. Tryon. Cluster Analysis: Correlation Profile and Orthometric (factor) Analysis for the Isolation of Unities in Mind and Personality. Edwards Brothers. (1939).
2. L. Kaufman, P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons. (2009).
3. C. C. Aggarwal; C. K. Reddy, (eds.). Data Clustering : Algorithms and Applications. ISBN 978-1-315-37351-5. (2018).
4. P.S. Bradley, U.Fayyad, and C. Reina, Scaling Clustering Algorithms to Large Databases, KDD-98 Proceedings. (1998).
5. A. Rajaraman, J. Ullman, J. Leskovec. Mining of Massive Datasets. New York, NY, USA: Cambridge University Press. ISBN 1107015359. (2011).
6. M. Daoudi, S. Meshoul. Revisiting BFR Clustering Algorithm for Large Scale Gene Regulatory Network Reconstruction using MapReduce. In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications (BDCA'17). (2017).
7. W. Chih-Ping, L. Yen-Hsien, H. Che-Ming. Empirical Comparison of Fast Clustering Algorithms for Large Data Sets. DOI: 10.1109/HICSS.2000.926655. (2000).
8. R. T. Ng and J. Han. CLARANS: a method for clustering objects for spatial data mining, in IEEE Transactions on Knowledge and Data Engineering, vol. 14, no. 5, pp. 1003-1016, DOI: 10.1109/TKDE.2002.1033770. (2002).



QUESTIONS OR COMMENTS?

a.janusz@mimuw.edu.pl

or

d.kaluza@mimuw.edu.pl