University of Warsaw
Faculty of Economic Sciences

Unsupervised Learning

Winter Semester, 2025/2026

Unsupervised learning: clustering

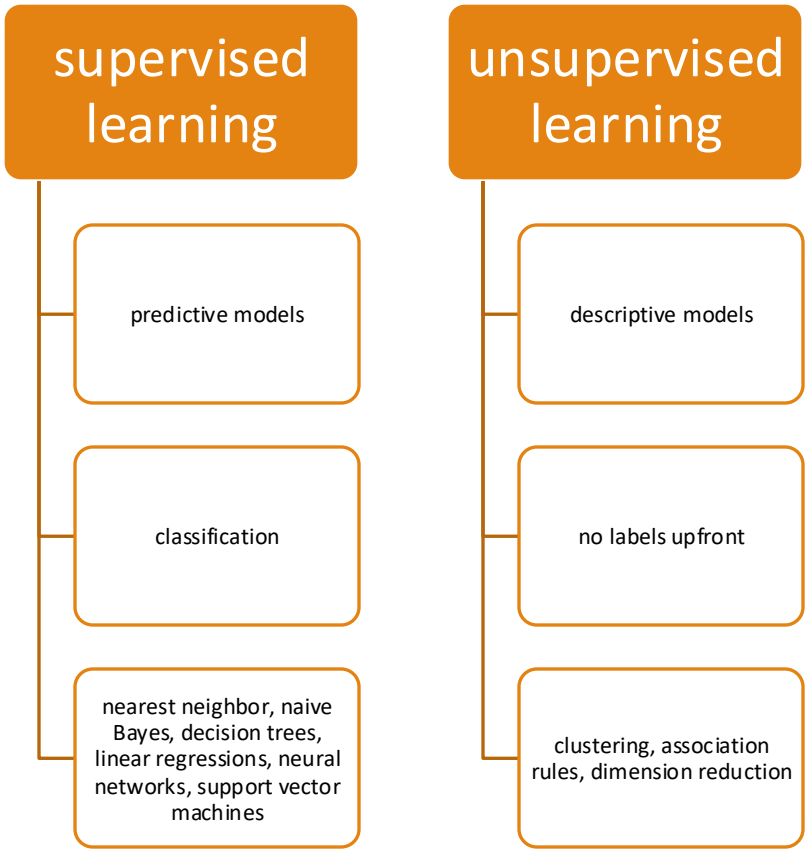
Literature

Clustering: literature

- Hennig, Ch.; Meila, M.; Murtagh, F.; Rocci, R. (2016). Handbook of Cluster Analysis. CRC Press.
- Bousquet, O.; von Luxburg, U.; Raetsch, G., eds. (2004). Advanced Lectures on Machine Learning. Springer-Verlag.
- Duda, Richard O.; Hart, Peter E.; Stork, David G. (2001). Unsupervised Learning and Clustering. Pattern classification (2nd ed.). Wiley.
- Hastie, Trevor; Tibshirani, Robert (2009). The Elements of Statistical Learning: Data mining, Inference, and Prediction. New York: Springer.

Clustering: introduction

Machine learning approaches



Introduction to clustering

unsupervised machine learning task

automatically divides data into clusters of similar items

used for knowledge discovery

insight into the natural groupings found within data

general idea: group the data such that related items are placed together

Introduction to clustering

Records inside a cluster should be very similar to each other, but very different from those outside!

Introduction to clustering

Applications of clustering

- Segmenting customers into groups with similar characteristics or buying patterns for targeted marketing campaigns or detailed analysis of purchasing behavior by subgroup
- Detecting anomalous behavior, such as unauthorized intrusions into computer networks, by identifying patterns of use falling outside known clusters
- Simplifying extremely large datasets by grouping a large number of features with similar values into a much smaller number of homogenous categories
- And many more!

Even more applications

Marketing	Land use	City-planning	Insurance
Disaster studies	Image procesisng	Web	Bioinformatics

Introduction to clustering

Clustering

- Creates new data
  - Unlabeled examples are given a cluster label and inferred entirely from the relationships within the data
- Is about classifying unlabeled examples (unsupervised learning)
- Clustering output will tell you which groups of examples are closely related

Introduction to clustering

- Our aim is to generate high quality clusters with high intra-class similarity and low inter-class similarity
- The quality of clustering may be measured also by its ability to discover hidden patterns
- There is a separate “quality” function that measures the “goodness” of a cluster
- Similarity is expressed in terms of a distance function: *d(x,y)*
- To some extent clustering is done on the basis of the measure of the distance between observations
- The definitions of distance functions are usually very different for interval-scaled, boolean, categorical, ordinal and ratio variables
- Weights should be associated with different variables based on applications and data semantics
- Sometimes we are interested in discovering outliers (outlier analysis)

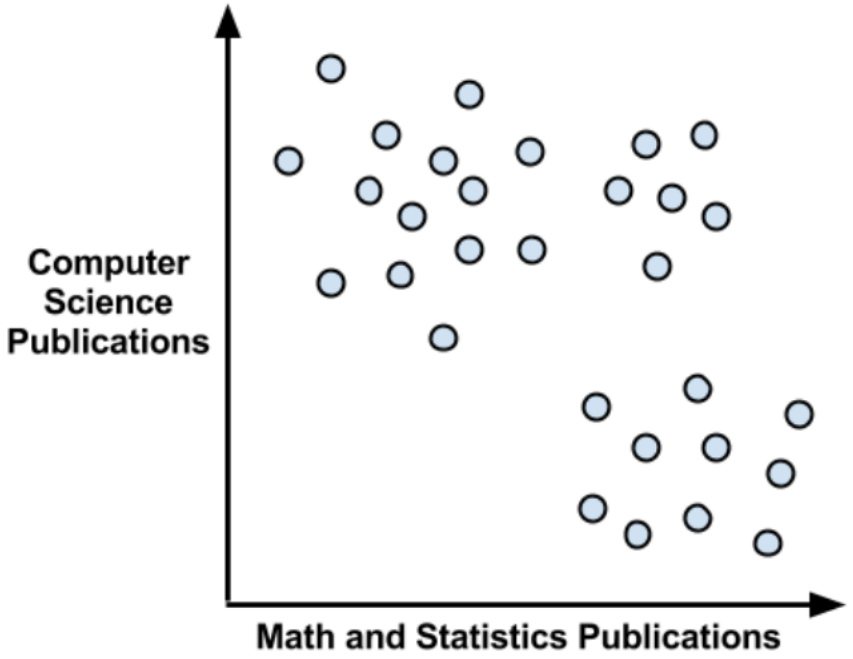
Introduction to clustering

- Centroids of clusters are points whose parameter values are means of the parameter values of all the points in the clusters
- Medoids are representative objects of a data set or a cluster with a data set whose average dissimilarity to all the objects in the cluster is minimal
  - medoids are always restricted to be members of the data set
  - most commonly used on data when a mean or centroid cannot be defined/is not representative (e.g. graphs)
- Generally, the distance between two points is taken as a common metric to asess the similarity among the components of a population
- The commonly used distance measure is the Euclidean metric

Introduction to clustering

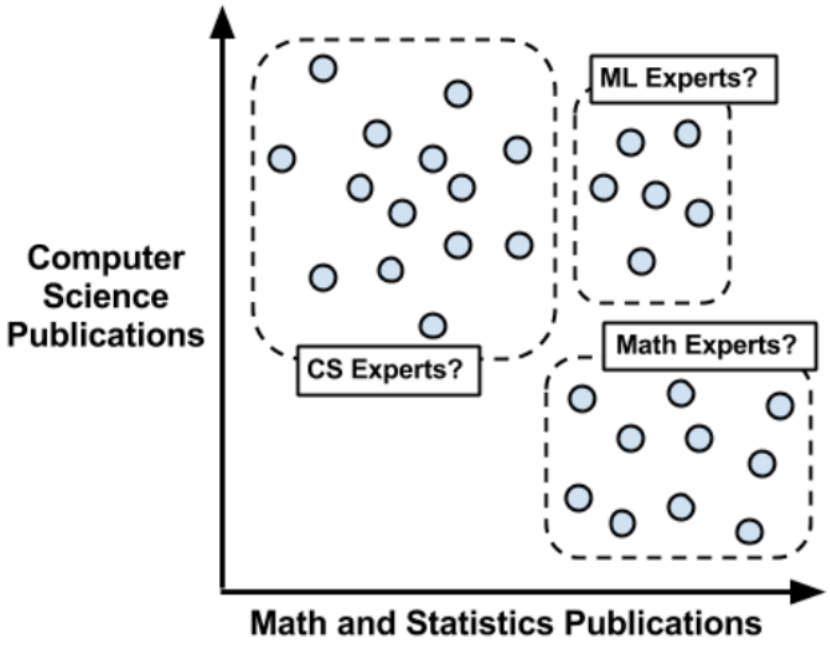
- The most prominent examples of clustering algorithms are:
  - Centroid-based clustering (k-means, k-medoids,…)
  - Connectivity-based clustering (hierarchical clustering)
  - Distribution-based clustering (Gaussian mixture models)
  - Density-based clustering (DBSCAN, OPTICS,…)

Example (Lantz, 2013)



- Consider publication track of scholars
- Mark the publications on the plot
- We see some kind of pattern there
- We cannot use the supervised learning approach as we do not know the true class value for each point (real area of expertise of a scholar)

Example (Lantz, 2013)



- Our groupings were formed visually
- We identified approx clusters as closely grouped data points
- Using a measure of how closely the examples are related, they can be assigned to homogenous groups

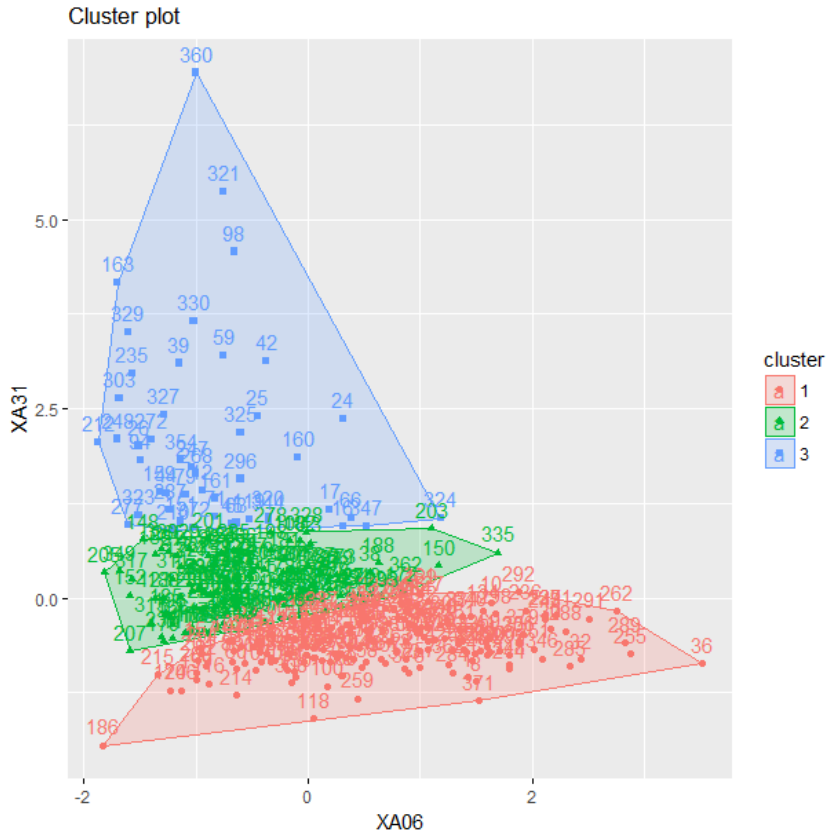
Example

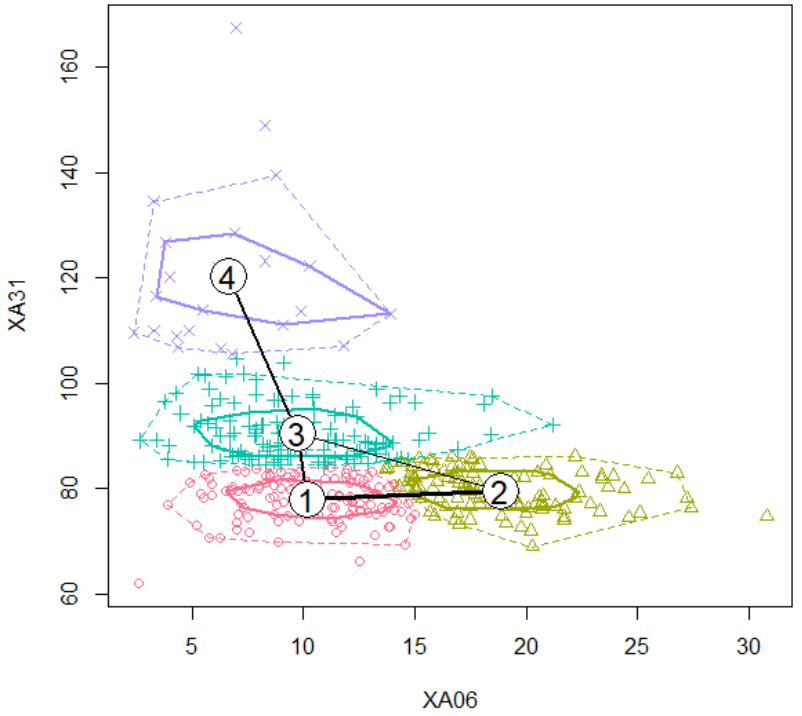
ID	feature1	feature2	feature3	feature4	feature5	feature6
object1	…	…	…	…	…	…
object2	…	…	…	…	…	…
object3	…	…	…	…	…	…
object4	…	…	…	…	…	…
object5	…	…	…	…	…	…
object6	…	…	…	…	…	…
object7	…	…	…	…	…	…
object8	…	…	…	…	…	…
object9	…	…	…	…	…	…
object10	…	…	…	…	…	…

Example

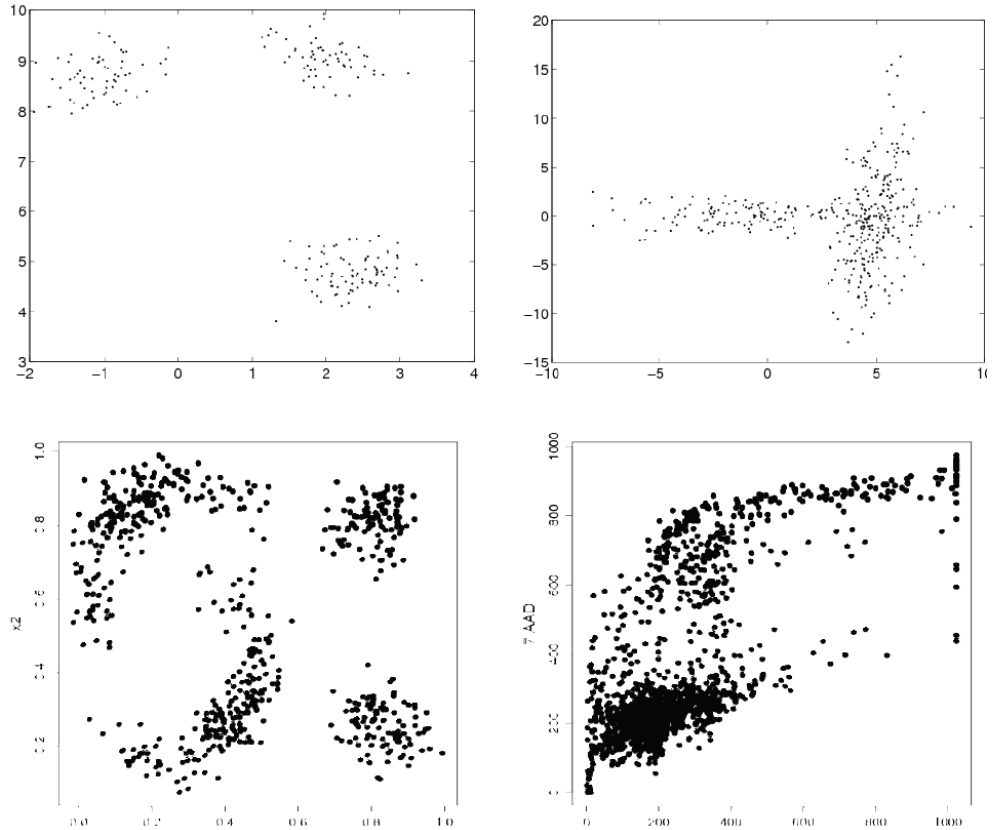
ID	feature1	feature2	feature3	feature4	feature5	feature6	**cluster**
object1	…	…	…	…	…	…	**1**
object2	…	…	…	…	…	…	**1**
object3	…	…	…	…	…	…	**3**
object4	…	…	…	…	…	…	**2**
object5	…	…	…	…	…	…	**3**
object6	…	…	…	…	…	…	**4**
object7	…	…	…	…	…	…	**4**
object8	…	…	…	…	…	…	**1**
object9	…	…	…	…	…	…	**1**
object10	…	…	…	…	…	…	**2**

Example





Example



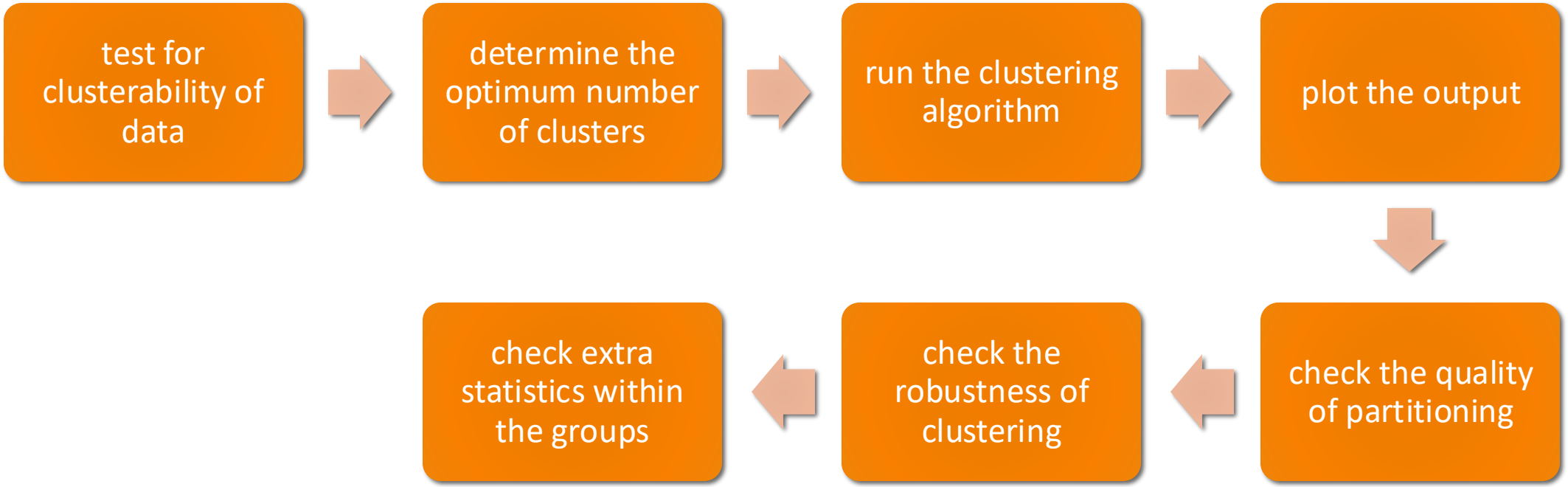
Introduction to clustering: short summary

- Cluster is a collection of data objects
- Data objects within the same cluster are similar but dissimilar to the objects in other clusters
- Cluster analysis is about grouping a set of data objects into clusters
- We have no predefined classes and clustering is a sort of unsupervised classification
- Clustering is usually used as a preprocessing step for other algorithms or as a stand-alone tool in order to get insight into data
- Clustering is hard to evaluate, but very useful
- Its application is subjective

Introduction to clustering: short summary

- Our goal is to group *n* objects into *k* clusters of similar objects
- Observations in the same group are similar, whereas observations in different groups are different
- As an input we provide a table regarding objects and features
- As an output we would like to get an extra variable by objects, to which cluster a given object belongs

Introduction to clustering: typical procedure



Distance-based approach

Distance-based approach

- The choice of distance metric should be made based on theoretical concerns from the domain of study
- A distance metric needs to define similarity in a way that is sensible for the field of study
- Where there is no theoretical justification for an alternative, the Euclidean should generally be preferred, as it is usually the appropriate measure of distance in the physical world

Distance-based approach

- First, we have to assign a proper distance measure between data
- Then, we look for a partition, where the distance between objects within parition is minimized and distance between objects from different clusters is maximised
- We have to define a distance measure, but it may be unclear how to assign it
- We do not know what relative weighting we should give to one attribute versus another
- This approach is applied to partitioning methods
- Example of distance measure – Euclidean distance

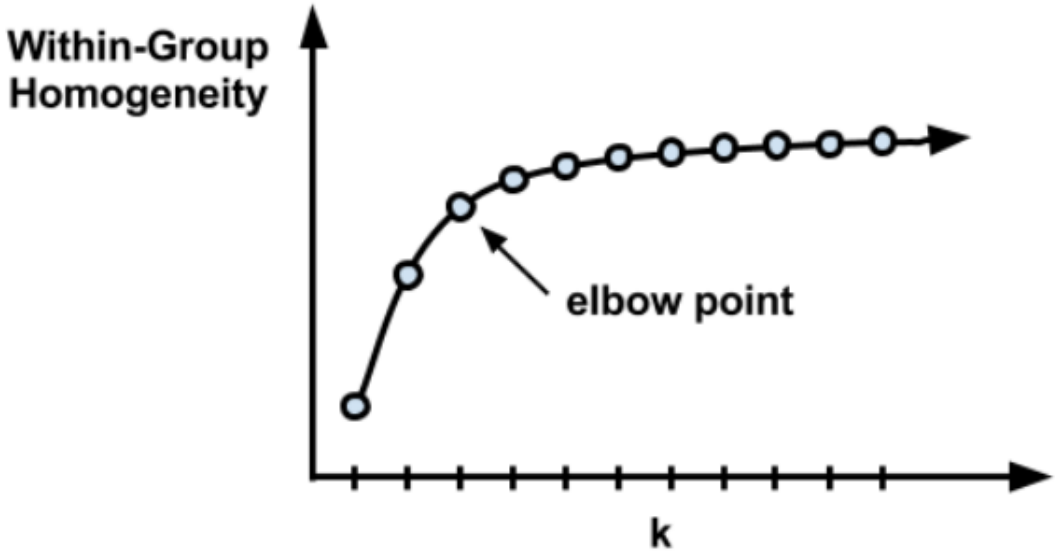
$$\mathrm{dist}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

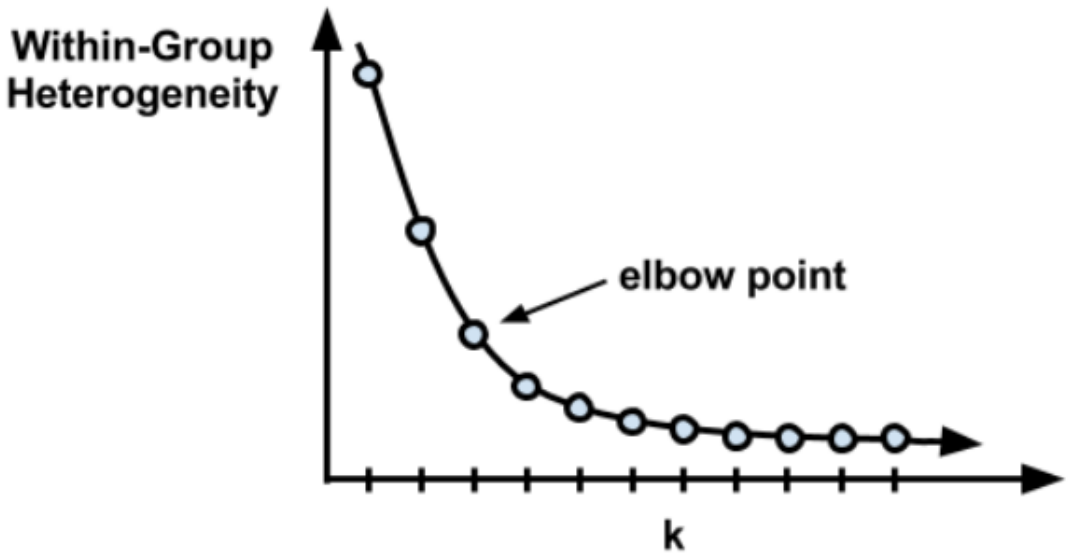
Appropriate number of clusters

Appropriate number of clusters

- Setting the *k* to a large number will improve the homogeneity of the clusters, but it risks overfitting the data
- Good to have a priori knowledge about the true groupings (with respect to features of data objects)
- Rule of thumb (too simplistic):
  - *k* equal to the square root of (*n*/2) where *n* is the number of data objects in the dataset
  - be aware that this will suggest you very high number of clusters for large datasets
- Elbow method (too simplistic as well):
  - the homogeneity within clusters is expected to increase as additional clusters are added (heterogeneity will continue to decrease with more clusters)
  - find *k* that there are diminishing returns beyond that point (elbow point)
  - within group homogeneity
    - similarity of values for the individual units that comprise the groups

Appropriate number of clusters (Lantz, 2013)





Appropriate number of clusters

- Visualize your dataset, as sometime it suggests you a lot
- Check out *k* set to various numbers
- Compute relevant statistics, e.g.:
  - silhouette
  - total within sum of squares
  - dissimilarity
  - AIC
  - BIC

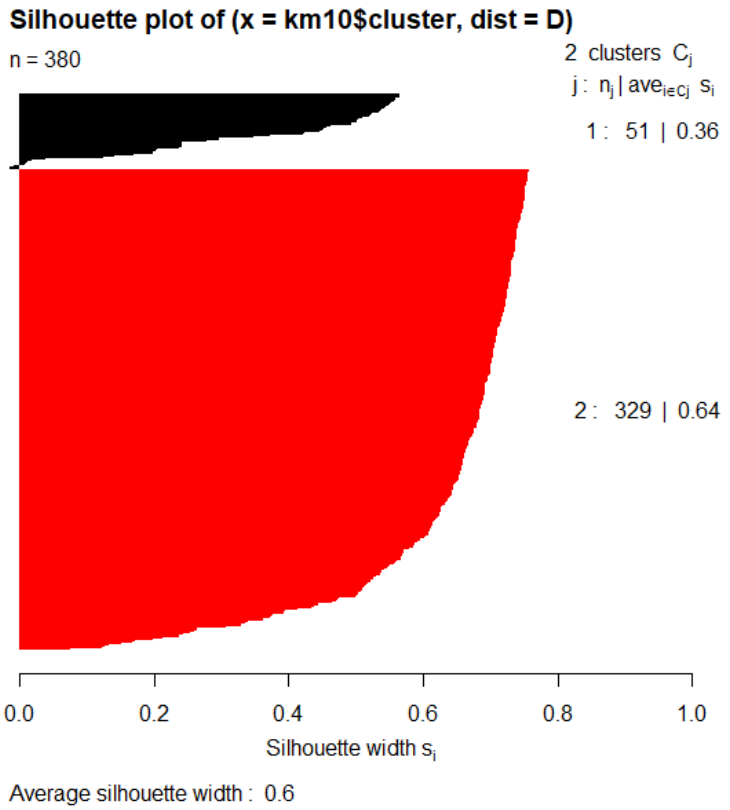
Appropriate number of clusters

- Silhouette analysis may be applied to study the separation distance between the resulting clusters
- Silhouette width

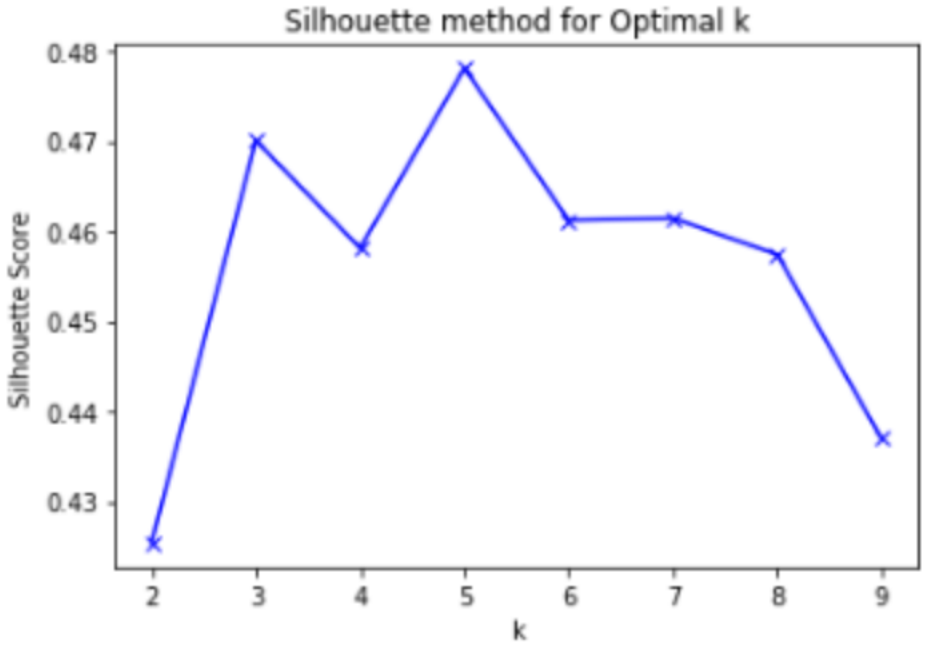
$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

- Where a is the mean intra cluster distance: mean distance to the other instances in the same cluster. b depicts mean nearest cluster distance i.e. mean distance to the instances of the next closest cluster
- A value close to 1 implies that the instance is close to its cluster is a part of the right cluster
- A value close to -1 means that the value is assigned to the wrong cluster
- A value close to 0 implies that the sample is on or very close to the decision boundary between two neighboring clusters

Appropriate number of clusters



Appropriate number of clusters



Silhouette Method

Clustering: k-means

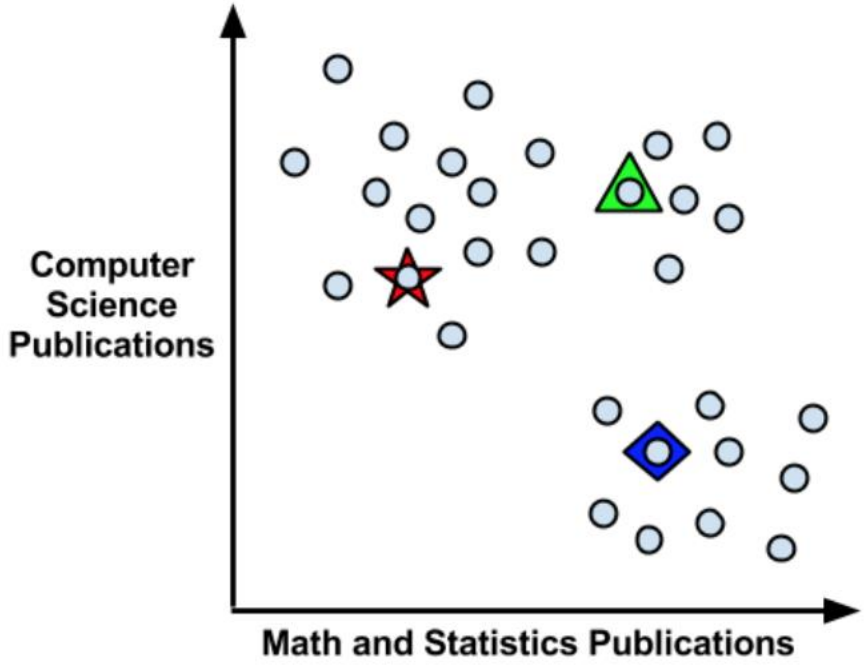
K-means

- One of the most popular approaches
- Assign each of the *n* examples to one of the *k* clusters, where *k* is a number that has been defined
- Goal: minimize the differences within cluster and maximize the differences between clusters
- The algorithm finds locally optimal solutions (inspecting the whole range would not be feasible)
- Then, it updates the assignments by adjusting the cluster boundaries according to the examples that currently fall into the cluster
- The process of updating and assigning occurs several times until making changes no longer improves the cluster fit
- Then, the process stops and the clusters are finalized

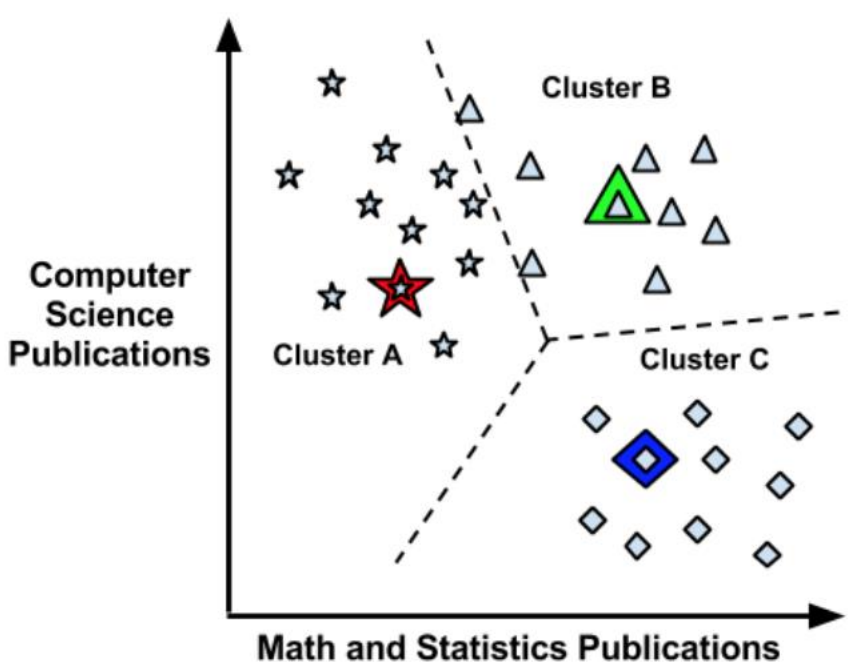
K-means

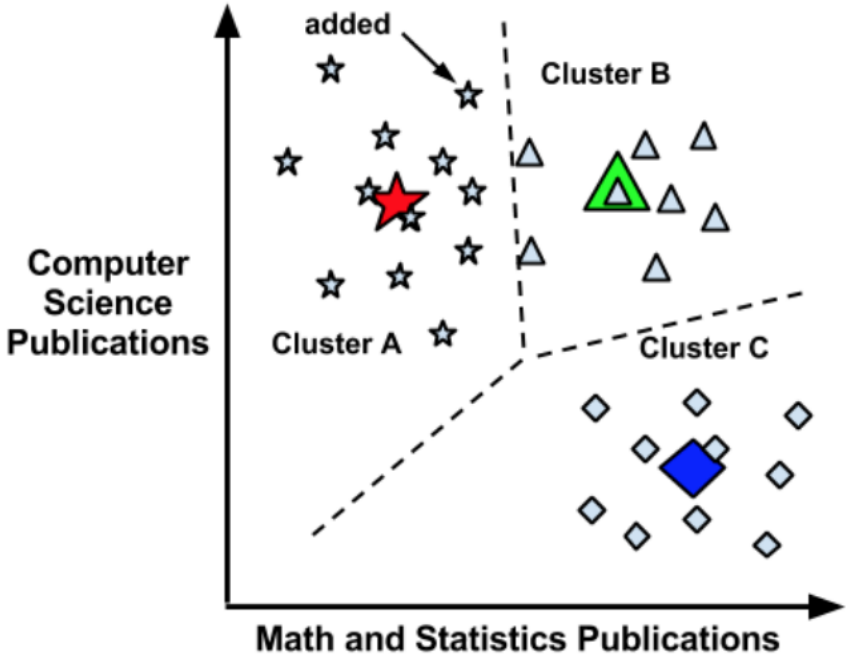
- The algorithm starts with choosing *k* points in the feature space to serve as the cluster centers
- These centers are the catalyst that spurs the remaining examples to fall into place
- These points may be chosen i.a. by selecting *k* random data objects
- Initial cluster centers may be selected in other ways
  - Choose random values occurring anywhere in the feature space
  - Randomly assign each example to a cluster and then proceed with updating the assignment

Example (Lantz, 2013) with k=3

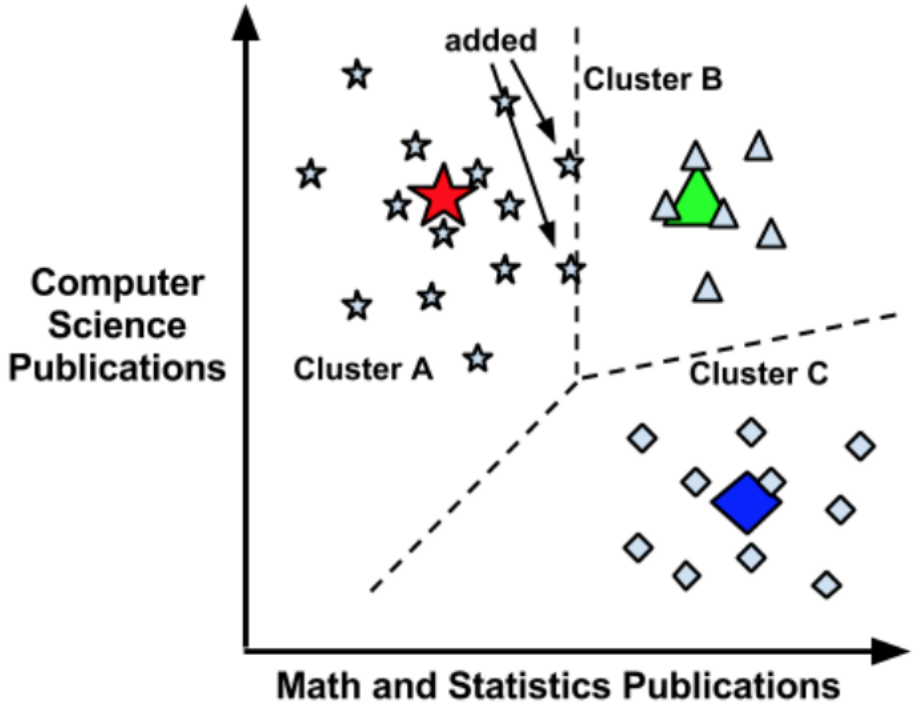


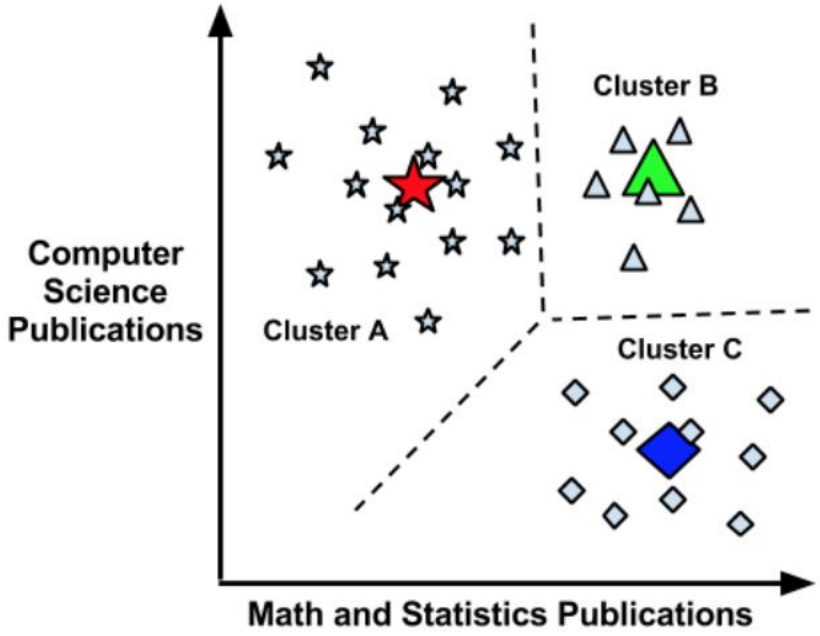
Example (Lantz, 2013) with k=3 voronoi diagrams





More updates





K-means: short summary

- This approach employs cluster center (means) to represent cluster
- We assign data elements to the closest cluster (center)
- The centroid's position is recalculated everytime a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters
- Our aim (in general) is to minimize the square error of the intra-class dissimilarity
- After we assign each object to the cluster with the closest center, we compute the new centers of the clusters
- We may also use the initialization procedure (initial partitions and number of clusters), update centers or try to move an object to another cluster
- The number of component equals to the final required number of clusters

K-means: short summary

- Our objective

$$\text{minimize} \sum_{i=1}^{K} \sum_{\mathbf{x}_i \in D_j} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$$

- The algorithm converges to a local minimum of its objective function
- We have to specify the number of clusters in advance

K-means

STRENGTHS

- Use simple principles for identifying clusters which can be explained in non-statistical terms
- It is highly flexible and can be adapted to address nearly all of its shortcomings with simple adjustments
- It is fairly efficient and performs well at dividing the data into useful clusters

WEAKNESSES

- Is less sophisticated than more recent clustering alhorithms
- As it uses an element of random chance, it is not guaranteed to find the optimal set of clusters
- It is unable to handle noisy data and outliers
- It is not suitable to discover clusters with non-convex shapes
- Requires a reasonable guess as to how many clusters naturally exist in the data

Quantitative example

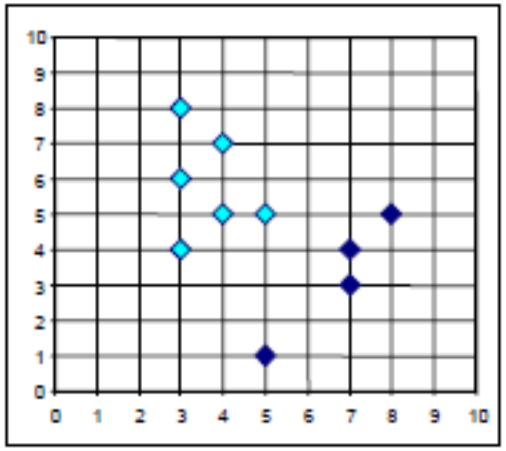
K-MEANS

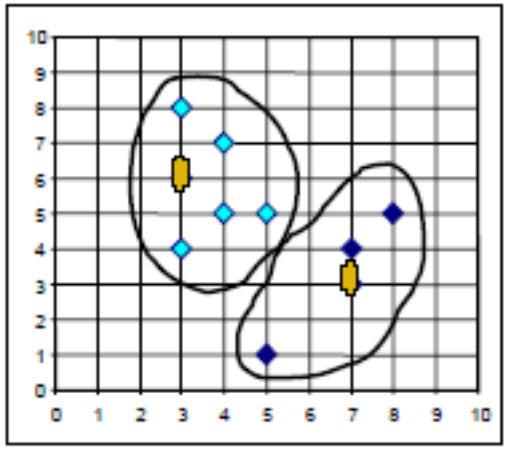
Clustering: partitioning around medoids

PAM

- Medoid is the most centrally located object in a cluster
- We start from some initial set of medoids and iteratively replace one of the medoids with one of the non-medoids and check if it improves the total distance of the resulting clustering
- Repeat until there is no change
- PAM is suitable for small datasets
- It is more robust than k-means in the presence of noise and outliers because a medoids less influenced by outliers or other extreme values than a mean

PAM





PAM: short summary

- The algorithm searches for k representative objects in a data set (k medoids) and then assigns each object to the closest medoid in order to create clusters
- Its aim is to minimize the sum of dissimilarities between the objects in a cluster and the center of the same cluster (medoid)
- It is known to be more than k-means as it is considered to be less sensitive to outliers

Extra: CLARA (clustering large applications)

- CLARA draws multiple samples of the dataset, then applies PAM on each sample, and gives the best clustering as the output
- The advantage of CLARA is that it deals with larger datasets than PAM
- Its weaknesses:
  - its efficiency depends on the sample size
  - when the sample is biased, a good clustering for samples may not be also good for the whole dataset

Quantitative example

PAM

Hierarchical clustering

Hierarchical clustering

- In this method, each object is assigned to its own cluster; then the algorithm proceeds iteratively, at each stage joining the two most similar clusters, continuing until there is just a single cluster
- At each stage, distances between clusters are recomputed by a dissimilarity formula according to the particular clustering method that is in use
- It produces a set of nested clusters organized as a hierarchical tree and may be visualized as a dendrogram
- We do not need to state the numer of clusters k as an input, but we have to propose the terminal condition

Hierarchical clustering

- There are two main types of hierarchical clustering: agglomerative and divisive
- In the agglomerative approach we start with the points as individual clusters and at each step we merge the closest pair of clusters until only one cluster is left [bottop up]
- In the divisive approach, we start with one, all-inclusive cluster and at each step, we split a cluster until each cluster contains a points [top down]
- Traditional hierarchical algorithms are based on a similarity or distance martix
- Dendrogram is a tree data structure, which presents hierarchical clustering techiques; each level of a dendrogram shows clusters for that level

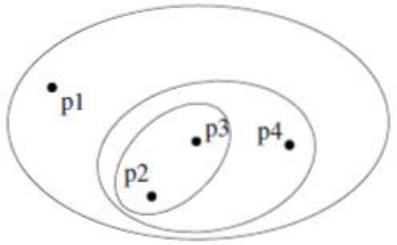
Linkage methods

- After selecting a distance metric, it is necessary to determine from where distance is computed just to be able to decide whether to split or merge clusters
- It can be computed between
  - the two most similar parts of a cluster (single-linkage)
  - the two least similar bits of a cluster (complete-linkage)
  - the center of the clusters (mean or average-linkage), or some other criterion
- Various linkage criteria have been developed
- The choice of linkage criteria should be made based on theoretical considerations from the domain of application
  - a key theoretical issue is what causes variation
- Where there are no clear theoretical justifications for the choice of linkage criteria, Ward's method is usually the fine default
  - this method works out which observations to group based on reducing the sum of squared distances of each observation from the average observation in a cluster

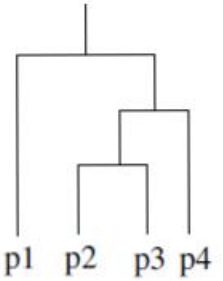
Hierarchical clustering

- The advantage of hierarchical clustering is that we do not have to assume any particular number of clusters (we can cut the dendrogram at the proper level in order to obtain the desired numer of clusters) and it could correspond to some meaningful taxonomies (i.a. in economics or biology)
- However, computation is usually complex in time and space (breaking large clusters, sensitivity to noise and outliers, difficulty in handling different sized clusters and convex shapes)
- Once a decision is made to combine two clusters, it cannot be undone
- We cannot correct erroneous merges or splits
- It may incorporate other techiques (i.a. microclustering)

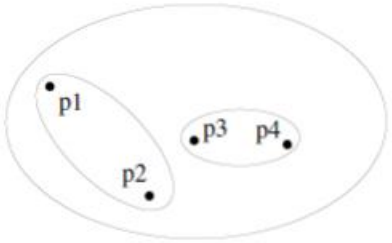
Hierarchical clustering



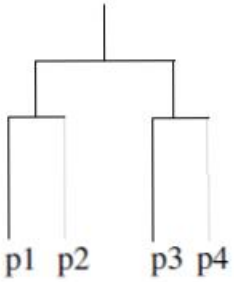
Traditional Hierarchical Clustering



Traditional Dendrogram

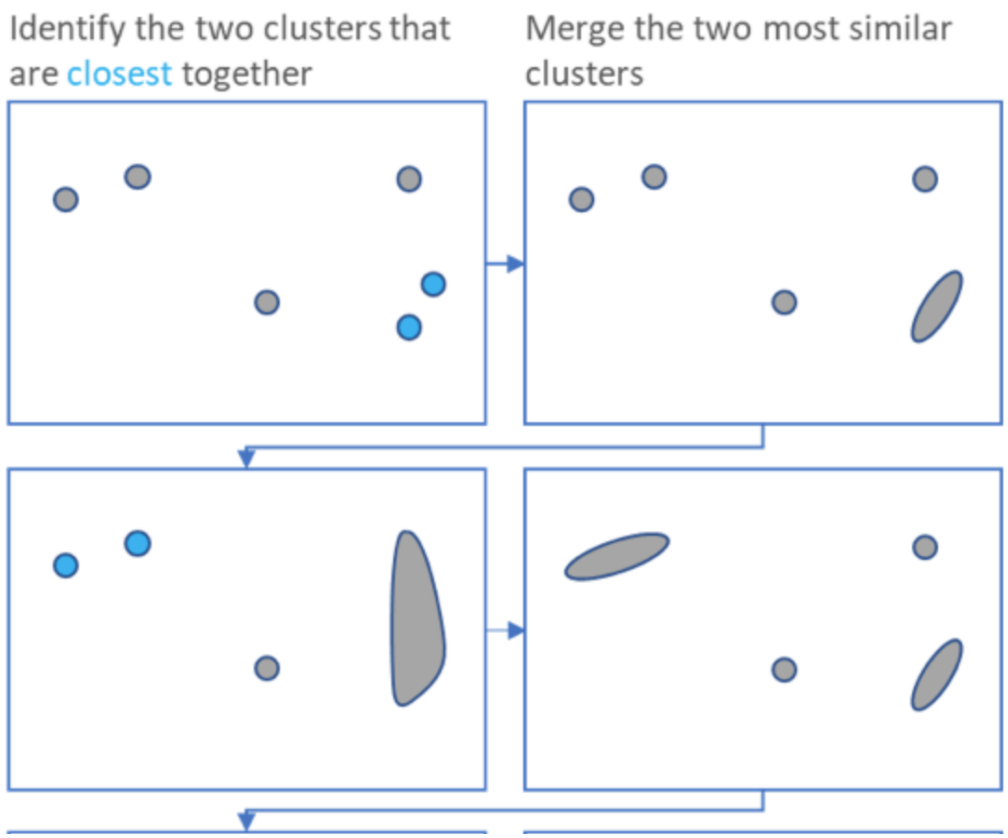


Non-traditional Hierarchical Clustering

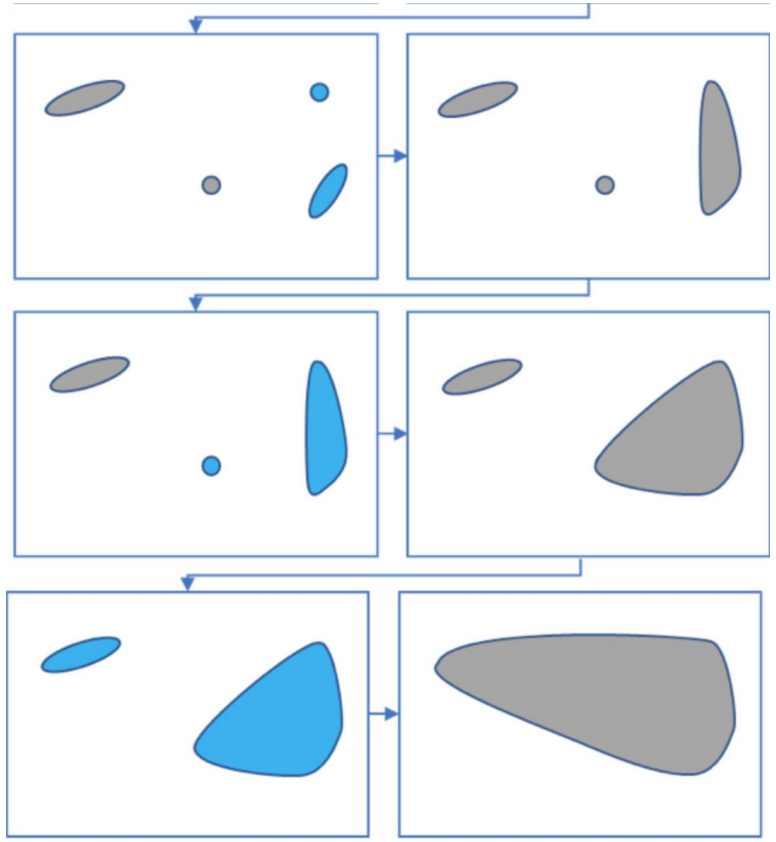


Non-traditional Dendrogram

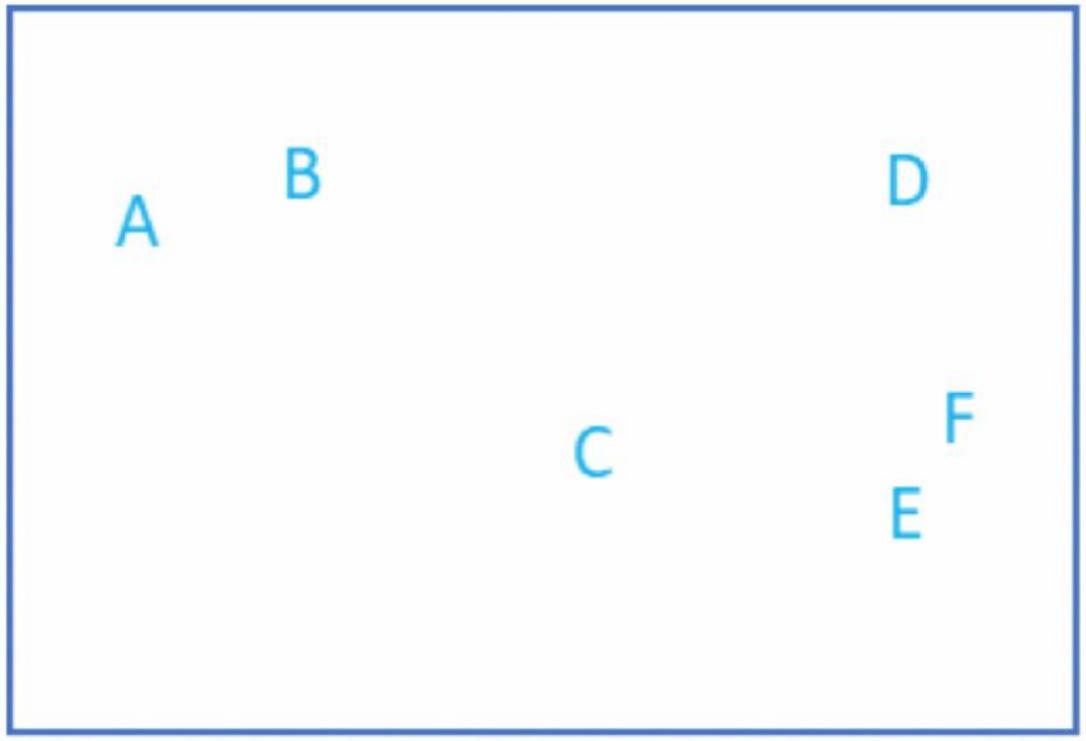
Hierarchical clustering (Bock)

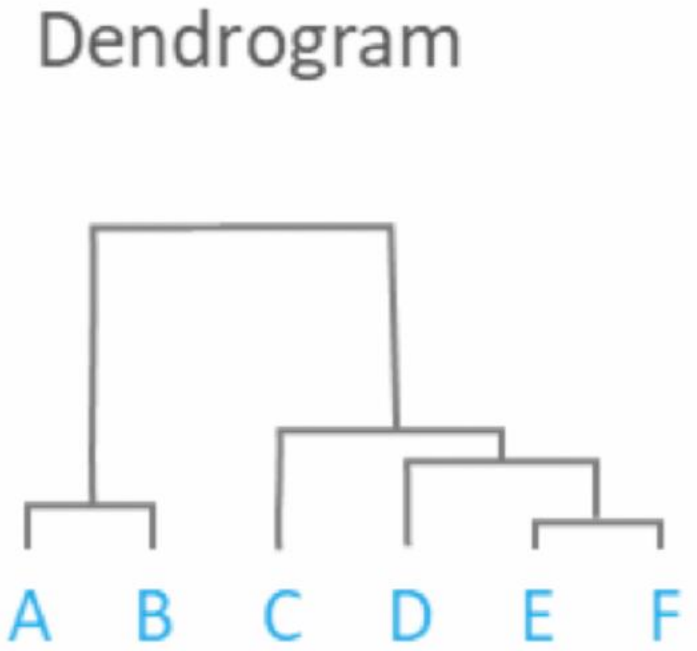


Hierarchical clustering (Bock)

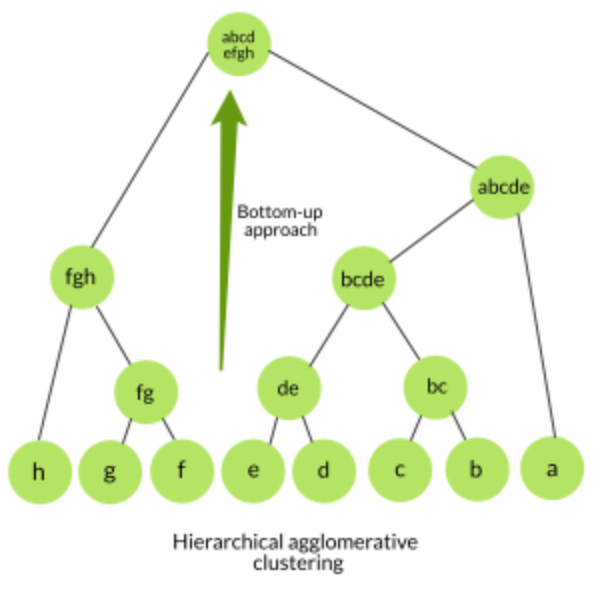


Hierarchical clustering (Bock)

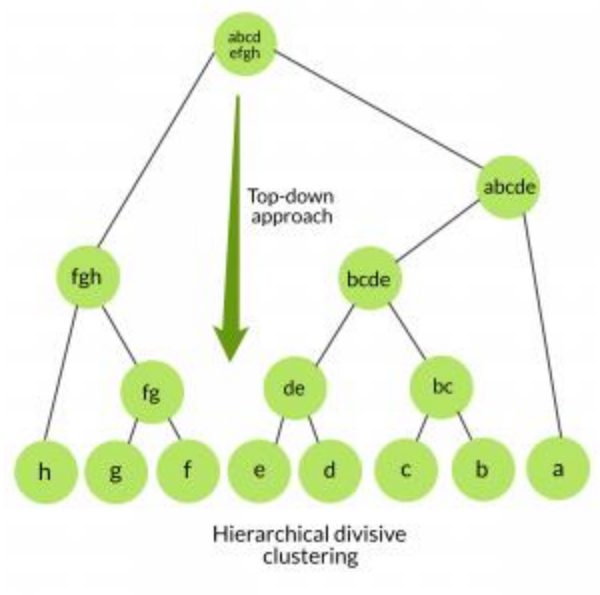




Hierarchical clustering



Hierarchical agglomerative clustering



Hierarchical divisive clustering

Quantitative example

HIERARCHICAL CLUSTERING

Thank you!