

ECS171: Machine Learning

L15: Unsupervised Learning II

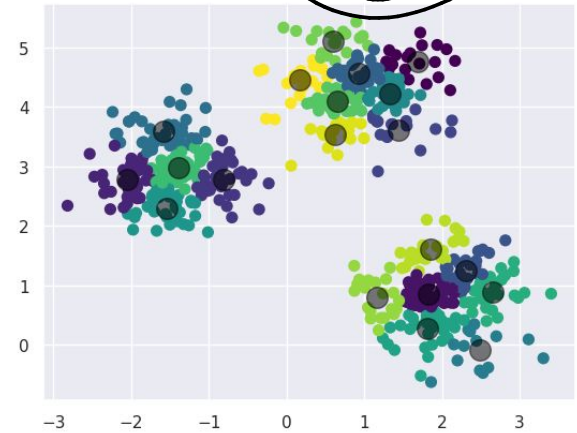
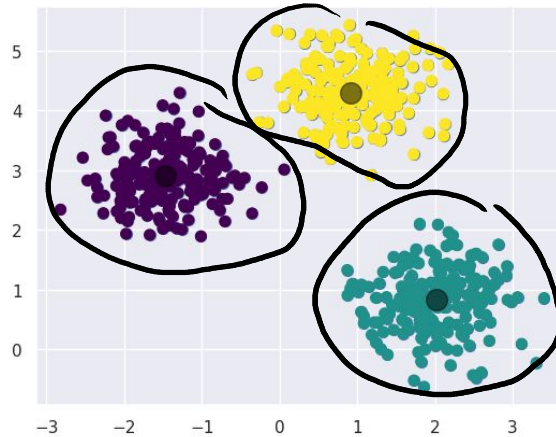
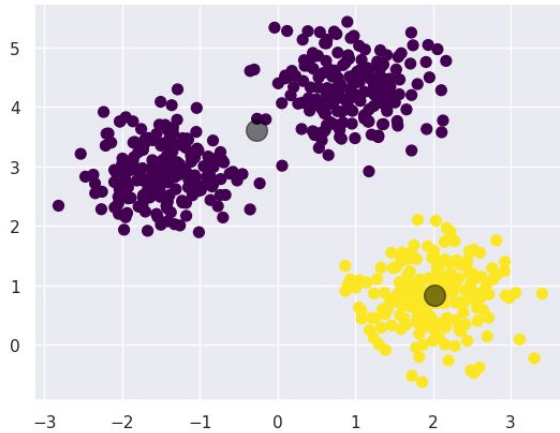
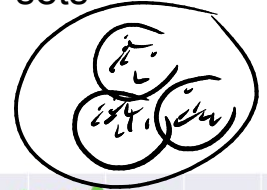
Instructor: Prof. Maïke Sonnewald
TAs: Pu Sun & Devashree Kataria

Intended Learning Outcomes

- Describe the difference between internal and external validation
- Describe the important components of internal validation and how various methods use different aspects of cluster properties
 - Describe and apply the Silhouette and 'elbow' methods
 - Critically evaluate the underlying assumptions of the methodologies
- Describe the approach of external validation, it's benefits and shortcomings.
 - Describe the Jaccard score
- Describe and apply information Theoretic aspects of model selection
- Describe the Association Rule type of Unsupervised Learning
 - Know different 'rules' and appreciate different algorithms

Validation and model selection

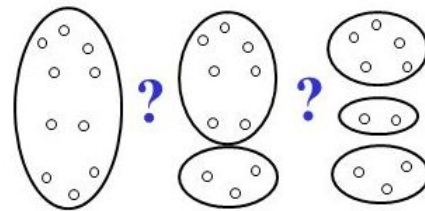
- Evaluate the 'goodness' of the results to compare:
 - Clustering methods (k-Means, Fuzzy c-means, DBSCAN etc..), cluster sets
 - Compare the results of analysis to externally known results
 - Parameter tuning e.g. determine the 'correct' number of clusters



Measuring Cluster Quality and Validity

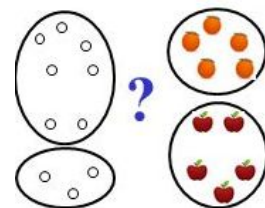
Internal Index

- Validate without external information
- With different numbers of clusters
- Solve the number of clusters



External Index

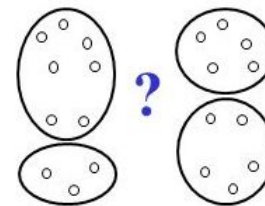
- Validate against 'ground truth'
- Compare two clusters (how similar?)



Information content and cross-validation

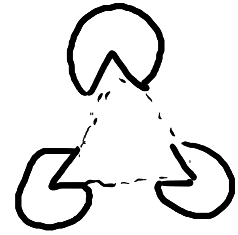
- Information criteria: Akaike and Bayesian (among many)

AIC BIC

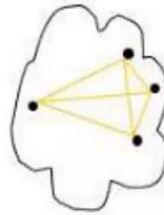


Internal quality indices

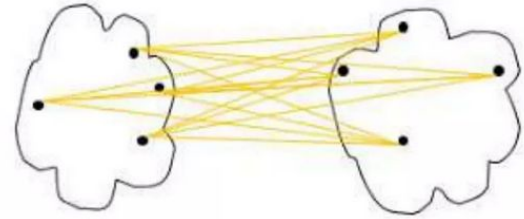
Gestalt psychology



- Compactness/Cohesion
 - How closely related are the objects (data) in a cluster?
- Separation
 - How distinct or well-separated is a cluster from the other clusters?
- Examples include:
 - 'Elbow' inertia and distortion
 - Silhouette score
 - Modified Hubert statistics
 - Calinski-Harabasz
 - I index
 - SD validity index
 - Many more....
- Most metrics make strong statistical assumptions about the data and cluster shapes



cohesion

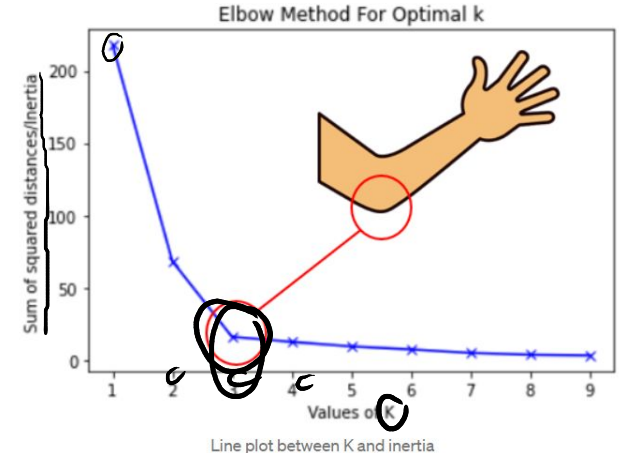
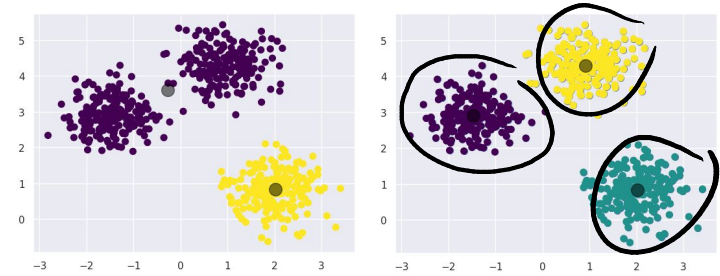


separation

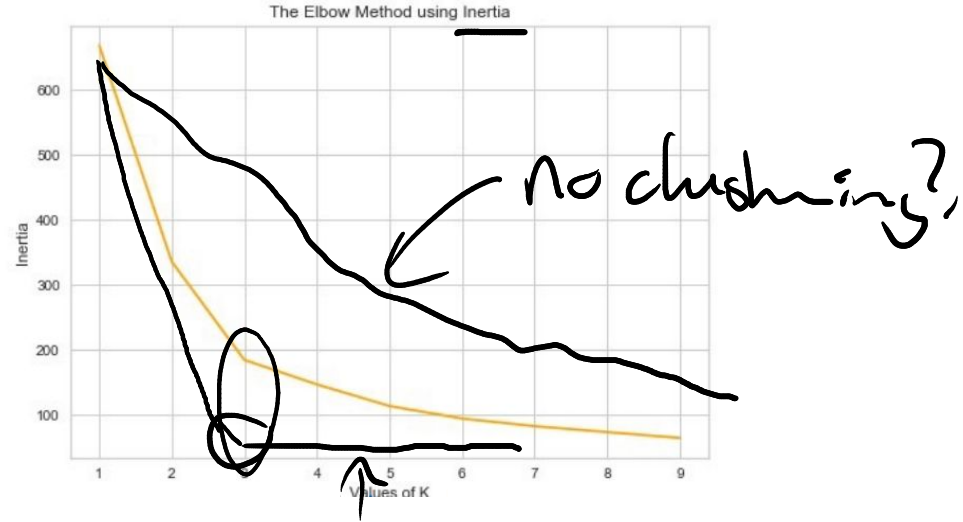
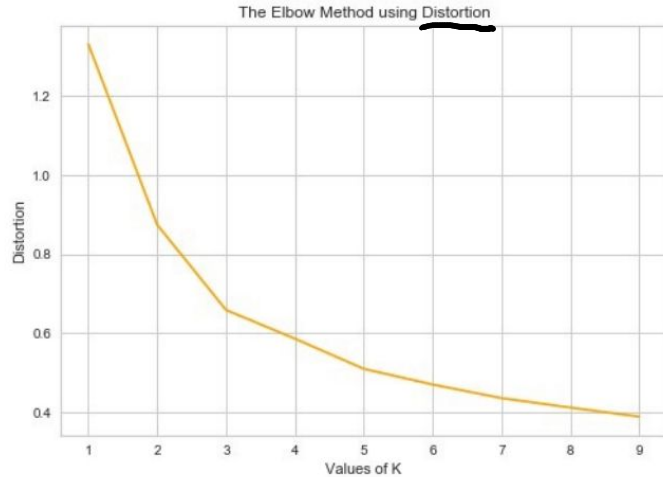
The 'elbow' method

- Line plot between cluster number (K) and the inertia/distortion metric from the data.
- There is a marked reduction in variation with K =3, but after that, the variation doesn't go down as quickly
- Inertia:
 - Inertia is the Sum of squared distances of samples to their closest cluster center
- Distortion:
 - The distortion score is computed as the average of squared errors/distances (SSE) of samples to their closest cluster center

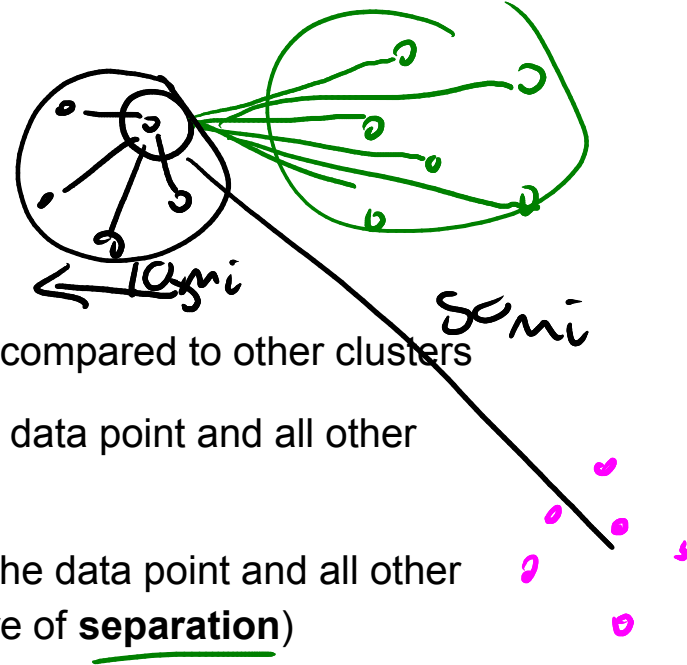
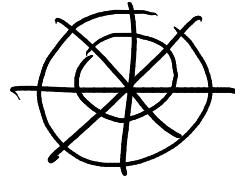
$$\frac{1}{N} \sum_{i=0}^n (Data X_i - Centroid X)^2 + (Data Y_i - Centroid Y)^2 \dots$$



Distortion and inertia example



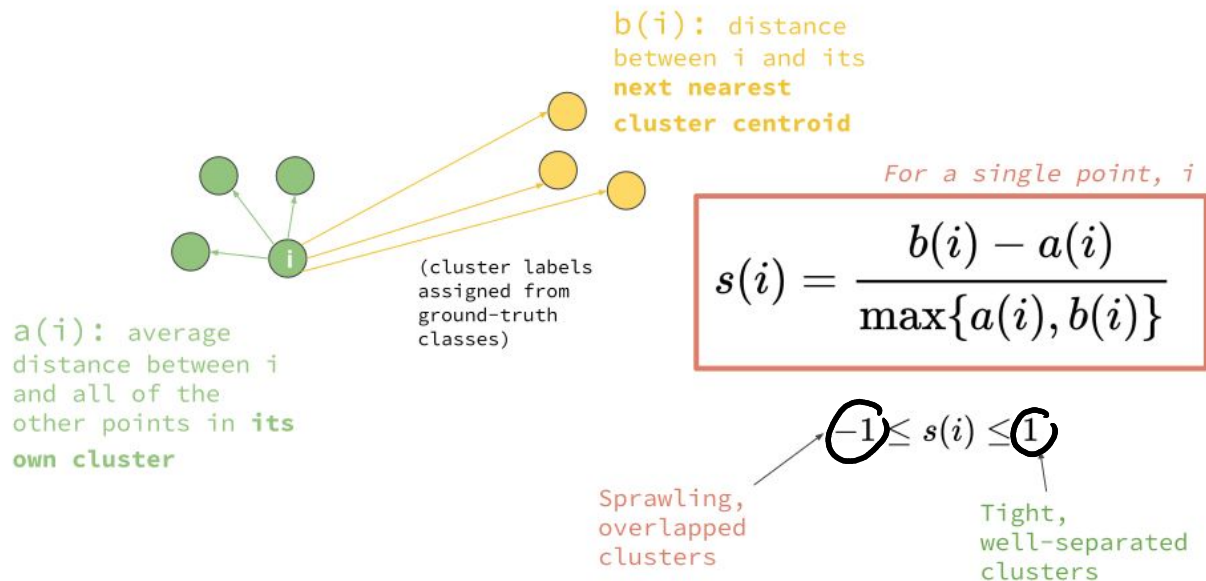
Silhouette score (S)



- **Calculated for each datapoint**, varies from -1 to 1
- A measure of how similar a data point is to its own cluster compared to other clusters
- Mean intra-cluster distance = Mean distance between the data point and all other data points in the same cluster. (measure of **cohesion**)
- Mean nearest-cluster distance = Mean distance between the data point and all other nearest cluster(s) that the sample is not a part of. (measure of **separation**)

$$S = \frac{\text{mean nearest clusters distance} - \text{mean intra cluster distance}}{\max(\text{mean nearest clusters distance}, \text{mean intra cluster distance})}$$

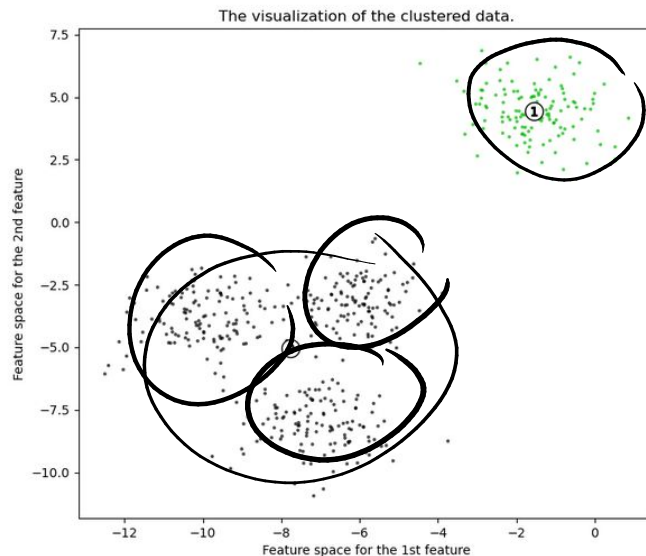
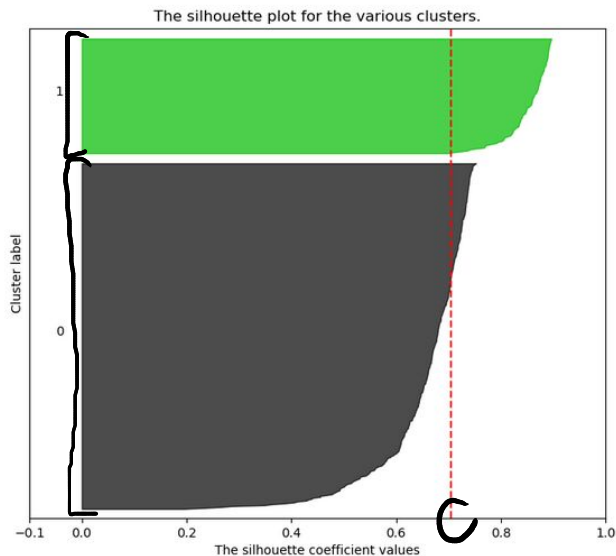
Silhouette Score



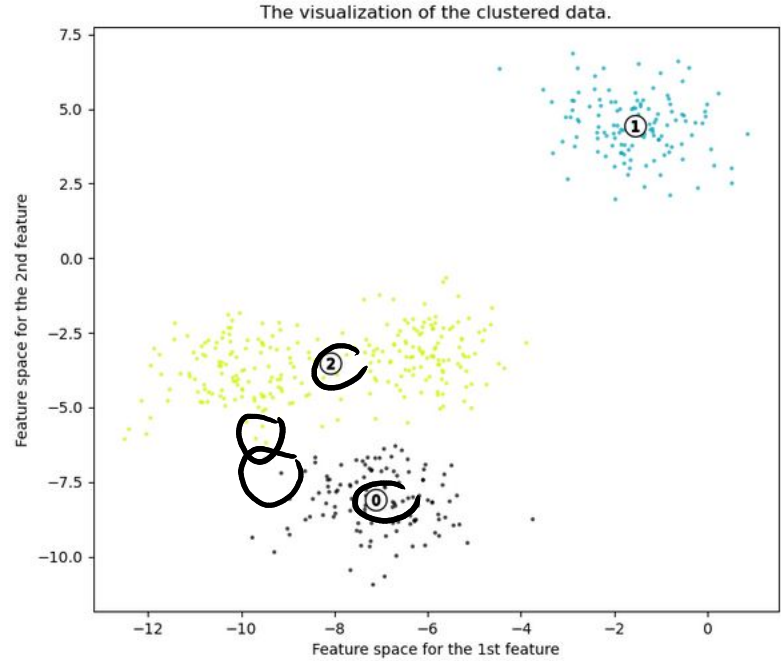
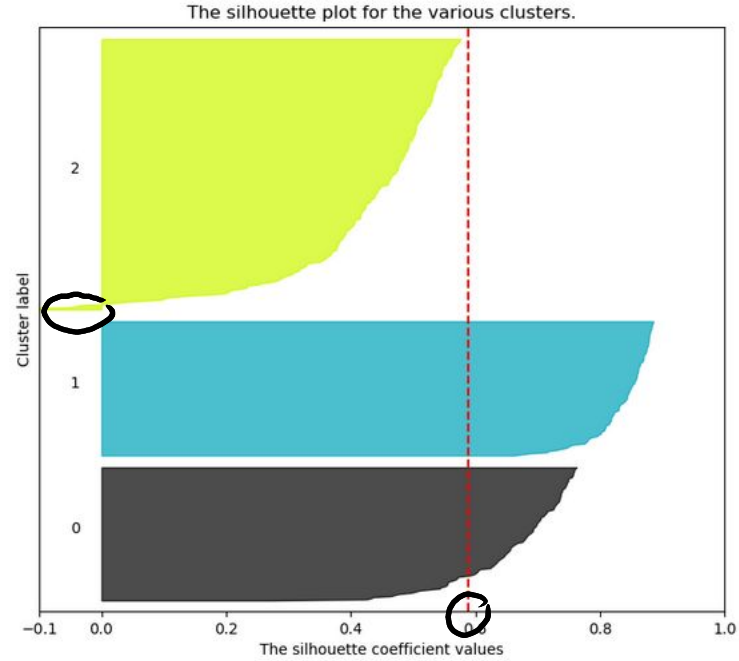
Silhouette score

$k=2$

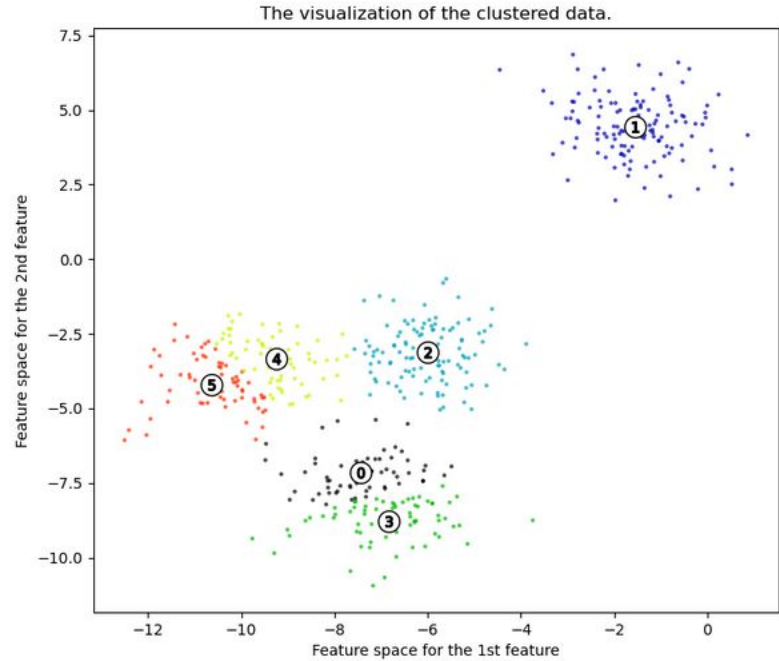
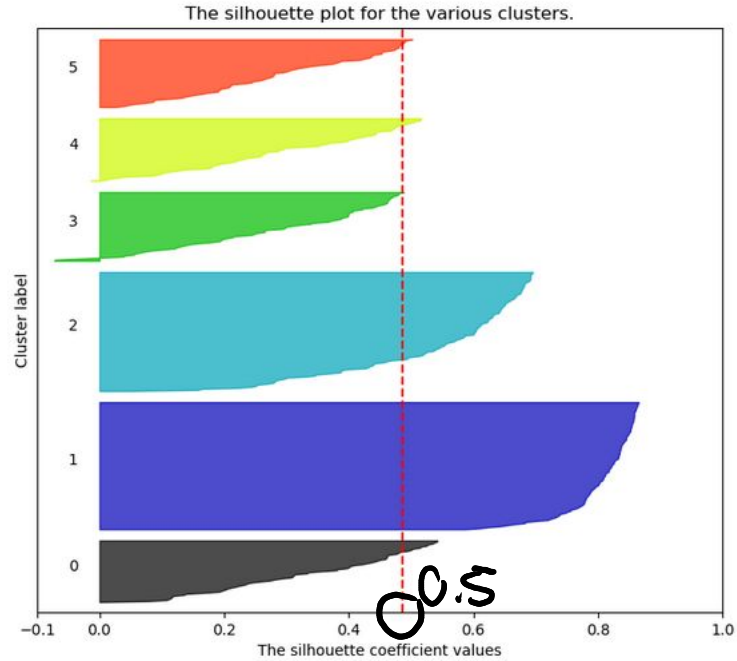
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 2$



Silhouette analysis for KMeans clustering on sample data with $n_clusters = 3$



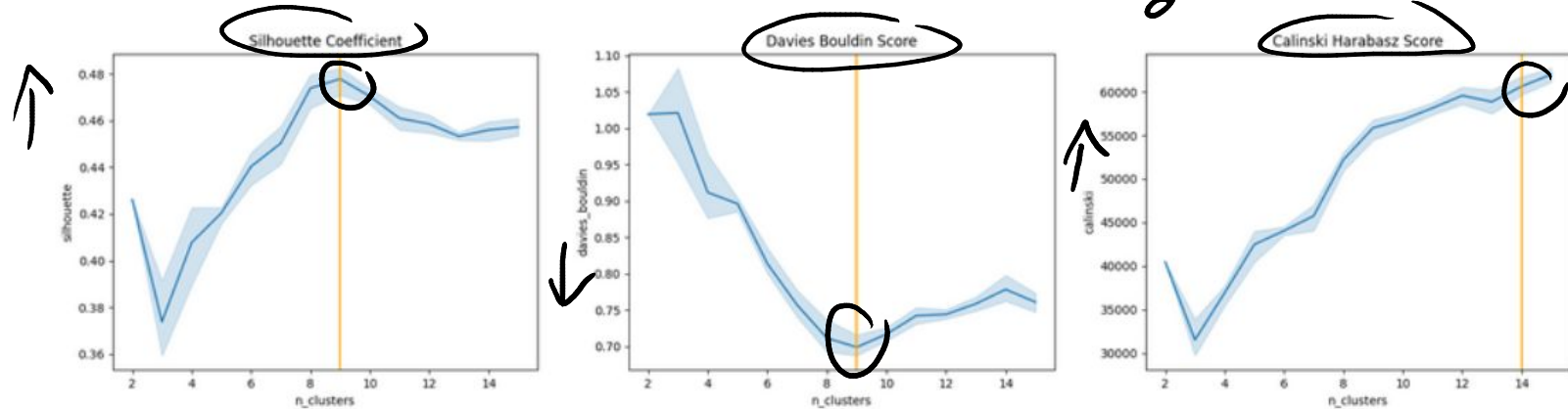
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 6$



Different metrics give different answers

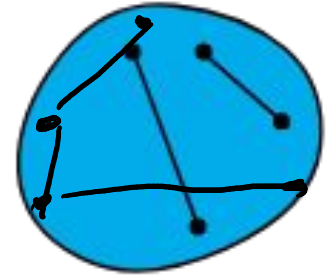
Often, different metrics give different suggestions for the same data and algorithm

This is a red flag that the method should be discarded

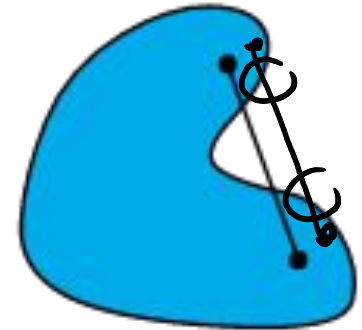


Problem with non-convexity

- For the internal validation methods to work perfectly one requires a convex distribution
- If the area is concave, meaning there is a 'gap' where the cluster is not present, but the distance between two points would be measured through the gap
- With a concave function, our metrics of compactness and separation do not work



convex

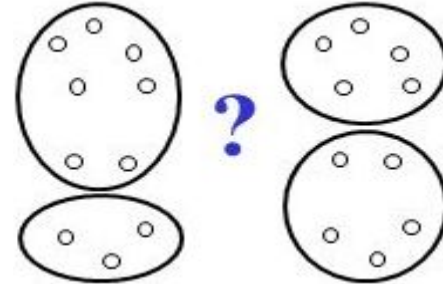
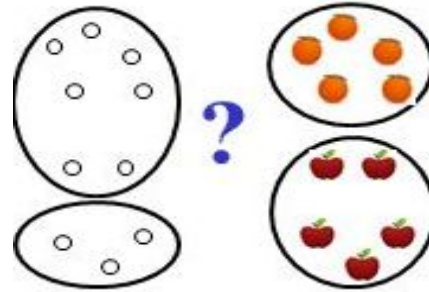


concave

External Validation

Compare against ground truth

- E.g. externally provided class labels
- Scores:
 - Homogeneity score
 - Completeness score
 - Rand Index
 - F-score
 - Jaccard
 - Many more...



Jaccard Score

Fuzzy c-Means

J = Jaccard distance

A = set 1 (e.g. cluster 1)

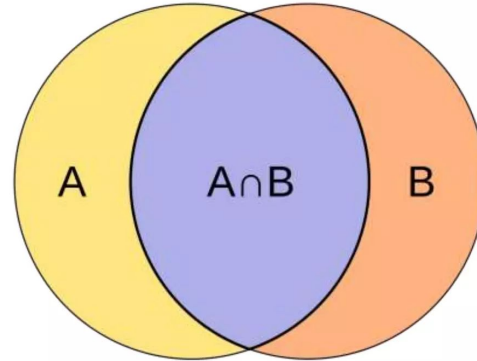
B = set 2 (e.g. cluster 2)

$A: \{x_1, x_2, x_3, x_4\}$

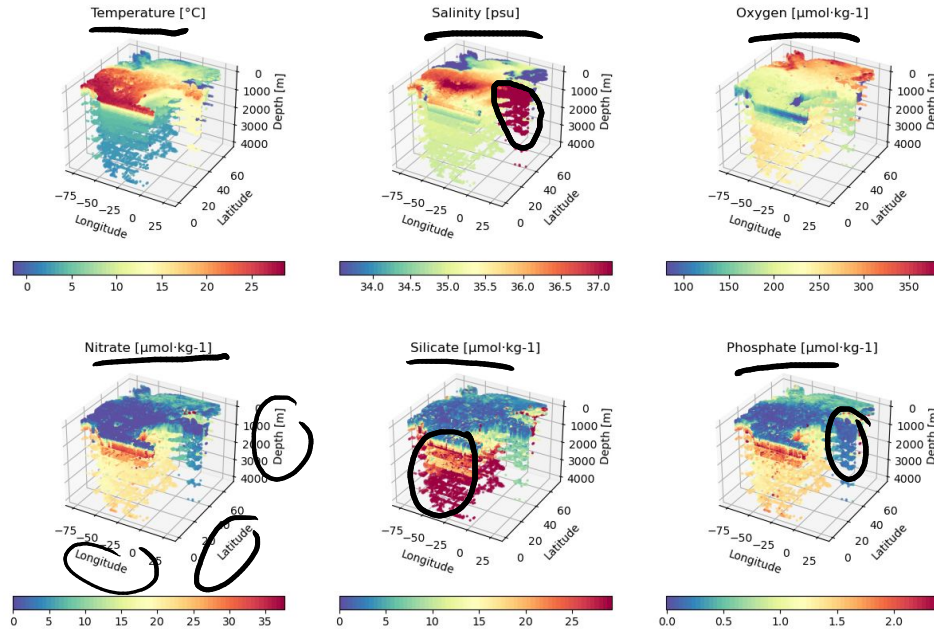
externally-known clustering

The Jaccard index, or the Jaccard similarity coefficient, used for gauging the similarity and diversity of sample sets.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

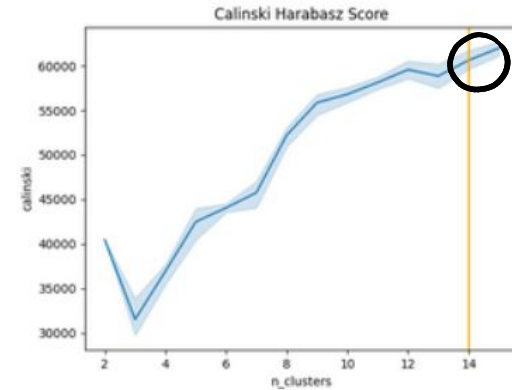
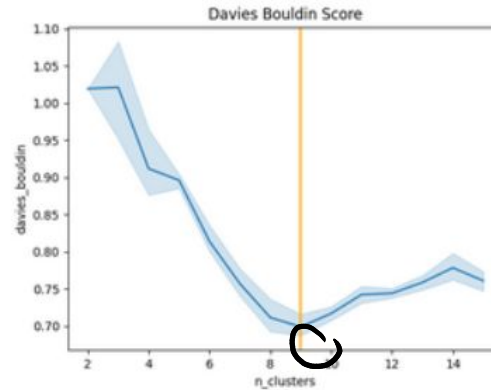
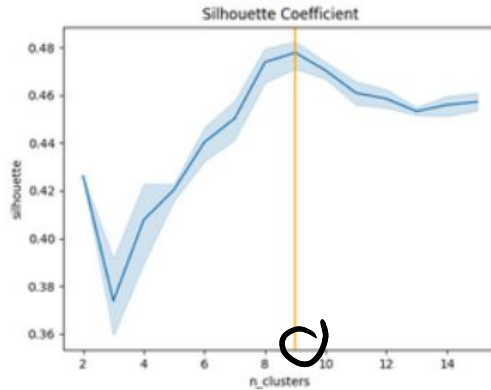


Case study: Nutrients in the ocean



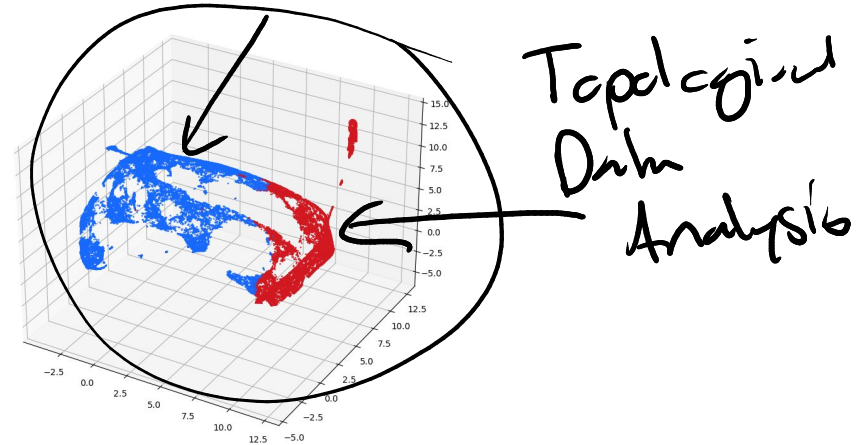
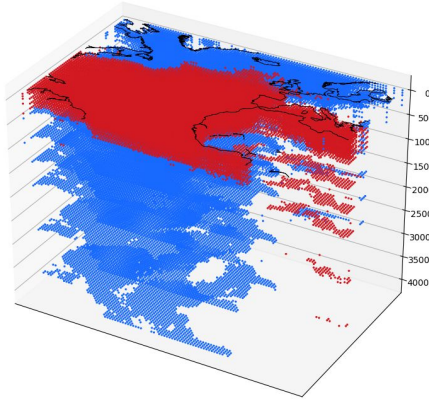
Case study: Nutrients in the ocean

- Clustering the data gives confusing results.
- What is happening?



External validation example

- The ML method was 'only' able to find hot and cold waters, missing out the 'interesting' structure
- Manifold approximation illustrates the highly non-convex data structures highlighting why the methods failed
- Here, visual inspection of the manifold and of the clusters in physical space highlights the failure



Information criteria: AIC/BIC

The ‘information’ content, how well we fit the data, is used to estimate how ‘good’ a model is

Akaike information criterion (AIC) (Akaike, 1974) is a technique based ‘fines’ for in-sample fit to estimate the likelihood that a model will predict future values

- A good model has minimum AIC among all the other models

Bayesian information criterion (BIC) (Stone, 1979) measures the trade-off between model fit and complexity of the model

The AIC and BIC use the likelihood

$$\text{AIC} = 2k - 2 \ln(\hat{L})$$

$$\text{BIC} = K \ln(n) - 2 \ln(\mathcal{L}),$$

where n is the number of datapoints and \mathcal{L} is the likelihood:

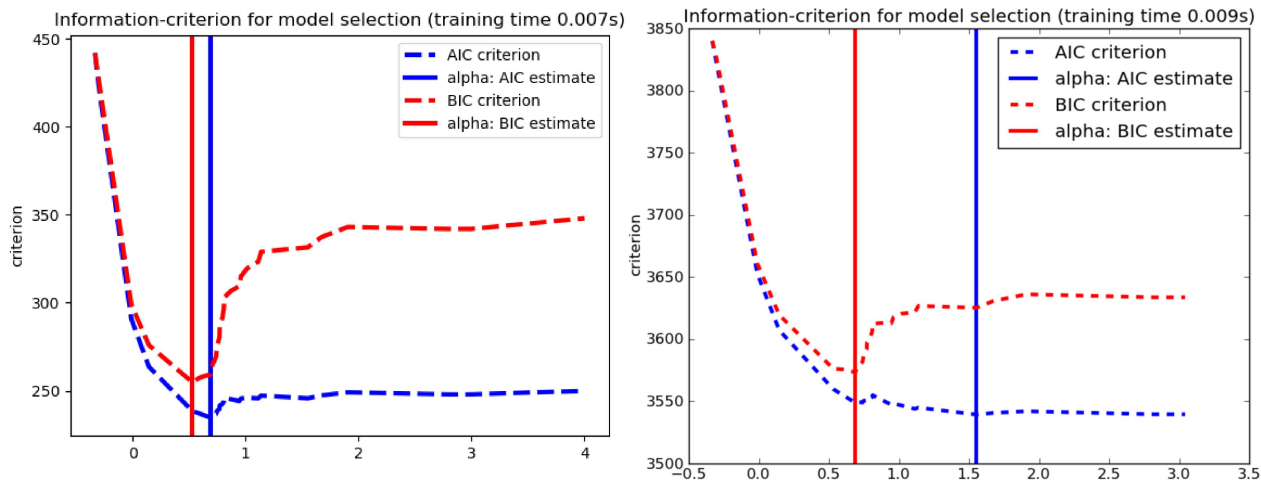
$$\mathcal{L} = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\zeta_i - \hat{\zeta}_i)^2}{2\sigma^2}\right).$$

Information criteria: AIC/BIC

A lower AIC or BIC value indicates a better fit

Ideally, the AIC should asymptote and the BIC go up as the model complexity increases beyond the ideal

There is not always a 'perfect fit'



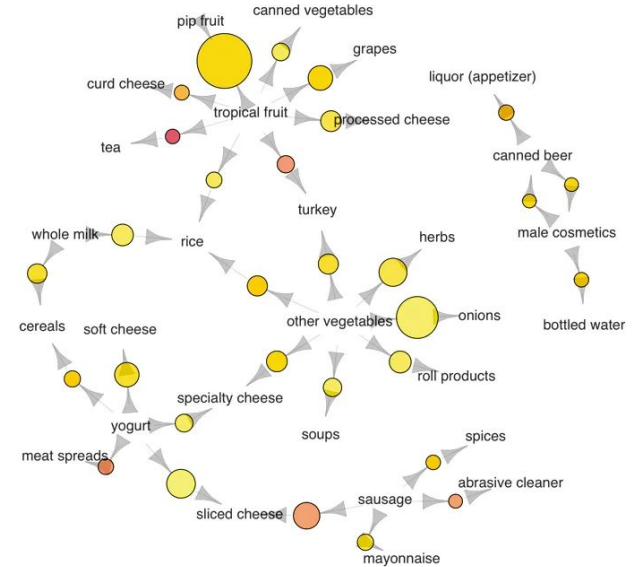
Association Learning

Dig into large amounts of data and discover interesting relations between attributes

- Market Basket Analysis, Intrusion Detection, Web Usage Mining, medical diagnosis etc.

For example, you find out that people who purchase milk and bread, also tend to purchase butter.

- Target advertisement
- Place products in store



Association Learning rules

- **Support:** How popular is an itemset. Used to find the frequency of a certain itemset appearing in the dataset.

$$Support(A) = Frequency(A)$$

- **Confidence:** How likely item B is purchased when item A is purchased, expressed as $(A \rightarrow B)$.

$$Confidence(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(A)}$$

- **Lift:** How likely an item A is purchased while controlling how popular item B is.

$$Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)}$$

Association Learning algorithms

- **Apriori:** This algorithm uses frequent datasets to generate association rules.
 - Apply an iterative approach/level-wise search where k-frequent itemsets are used to find k+1 itemsets
 - Uses a *Breadth-First Search* algorithm and *Hash-Tree* to calculate the itemset efficiently
 - Apriori algorithm works in a horizontal sense imitating the ***Breadth-First Search*** of a graph
- **Eclat:** Eclat algorithm stands for ***Equivalence Class Transformation***.
 - The ECLAT algorithm works in a vertical manner just like the ***Depth-First Search*** of a graph
 - Has faster execution than Apriori Algorithm
- **F-P Growth:** The F-P Growth algorithm stands for ***Frequent Pattern***
 - Improved version of the Apriori Algorithm
 - The FP-Growth Algorithm is an alternative way to find frequent item sets without using candidate generations, thus improving performance.
 - It uses a ***Divide-and-Conquer*** strategy
 - The core of this method is the usage of a special data structure named ***Frequent-Pattern Tree (FP-tree)***, which retains the item set association information. The purpose of this frequent tree is to extract the most frequent patterns.