INFO 284 – Machine Learning

Clustering

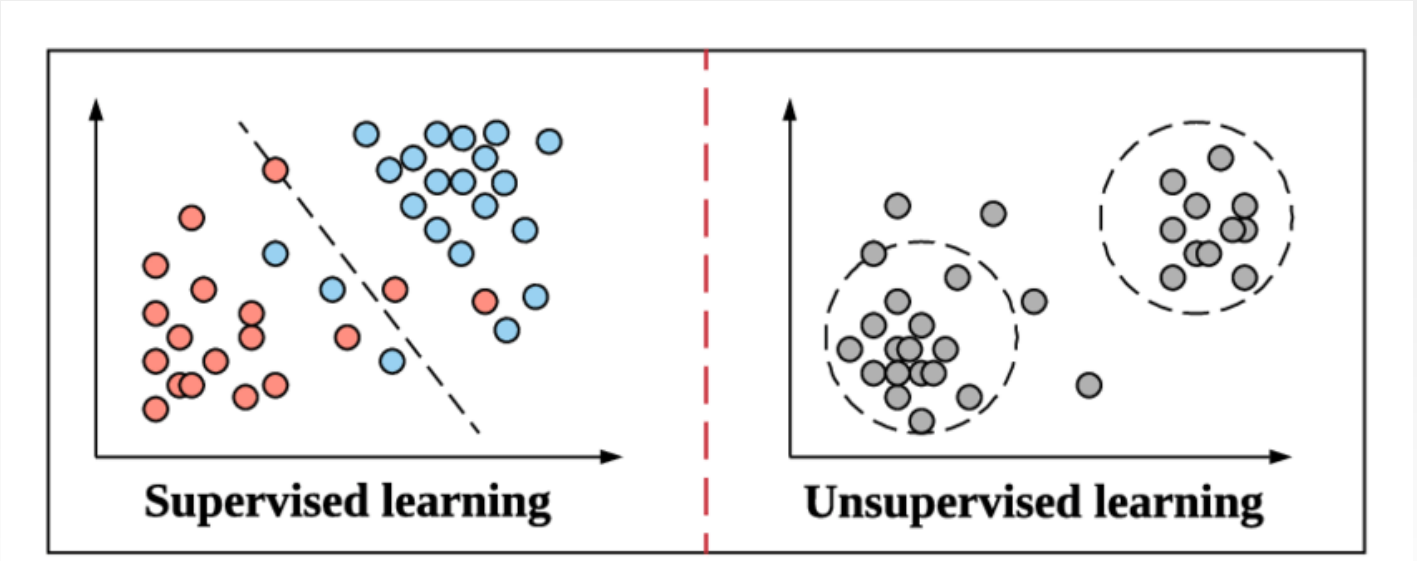
Bjørnar Tessem

UNIVERSITY OF BERGEN

Types of unsupervised learning

- Clustering
  - Find inherent groupings in the data
    - grouping customers by purchasing behaviour
    - news by topic
- Unsupervised transformations
  - Create new representations of the data
    - easier to visualise
    - easier to learn from
  - Dimensionality reduction

Clustering

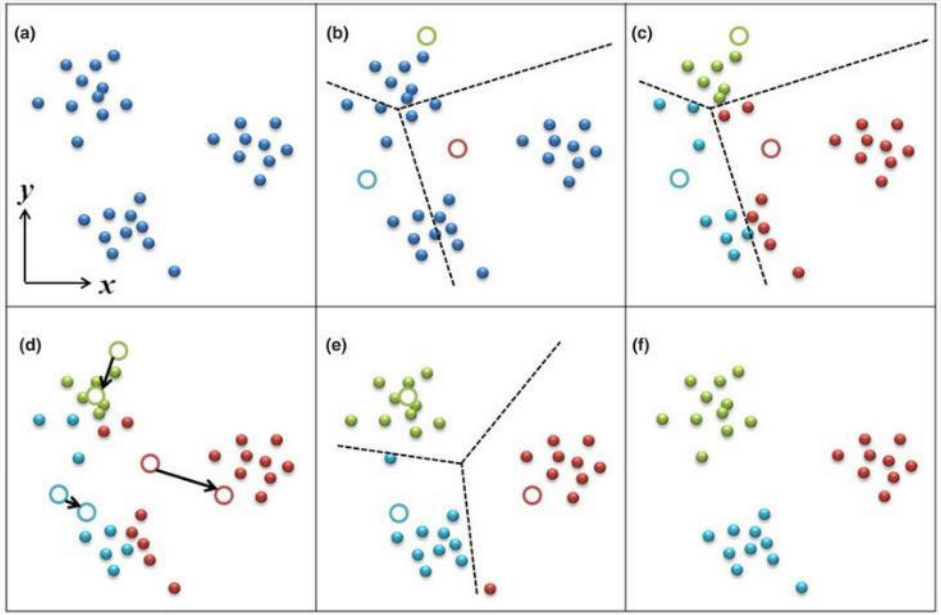


Clustering methods

- K-Means
  - Iteratively move cluster centers
- Agglomerate clustering
  - Merge close groups
- DBSCAN
  - Include close data points in clusters

- Important parameters
  - Number of clusters
  - Distance measures

K-means clustering



k-Means Clustering

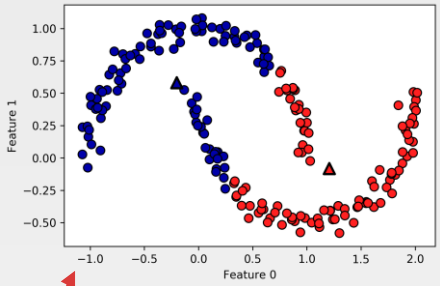
- Partition data points into *k* clusters
  - Each data point belongs to the cluster with the nearest average
  - Minimizes within-cluster variances.
- Algorithm
  1. Initialize with k random data points as center points.
  2. While cluster membership has changed for at least one data point:
     a. Assign all data points to the closest cluster center (centroid)
     b. Reset each centroid as the average of the data points assigned to it
  3. Return k current center points and clusters

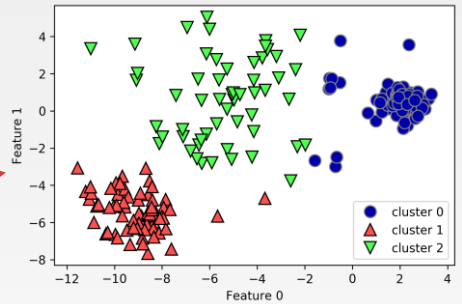
k-Means Clustering Properties

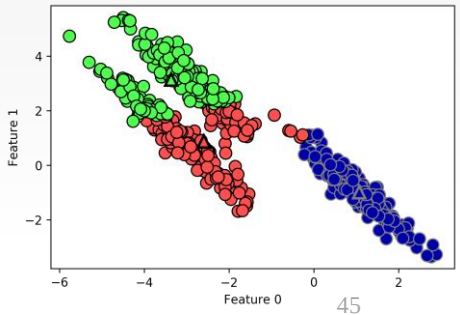
- Pros
  - easy to understand and implement
  - runs relatively quickly
  - scales easily to large data sets
- Cons
  - Relies on random initialization
  - Restrictive on the shape of the data
  - Requires you to specify the number of clusters you are looking for
- Scikit
  - Runs the algorithm 10 times with 10 different random initialisations and keeps the best
  - MiniBatchKMeans class for handling very large datasets

k-Means Clustering Failure Cases

- A cluster is defined by its centre and is a convex shape.
- => non-convex shapes will not be clustered correctly
- k-means clustering assumes all clusters have the same “diameter”
- k-means clustering assumes all the directions are equally important for each cluster



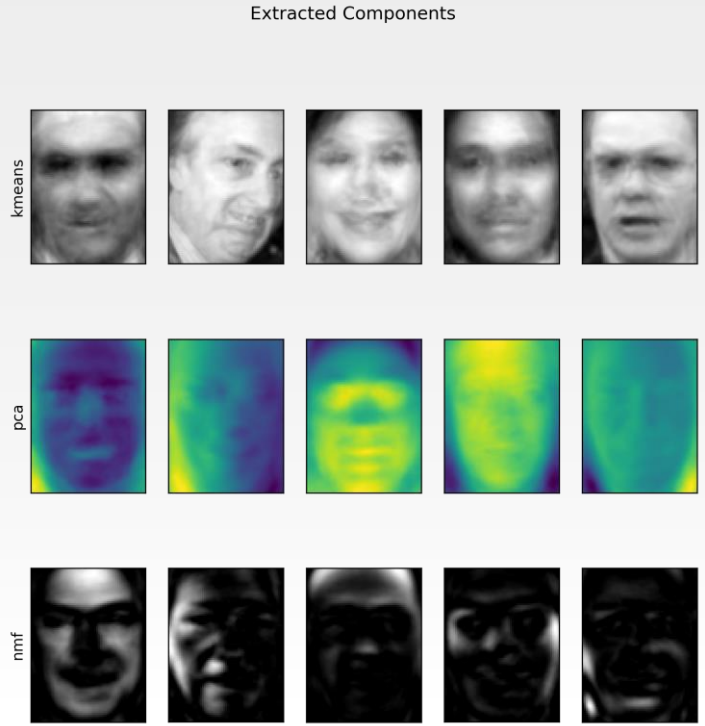




Vector quantization

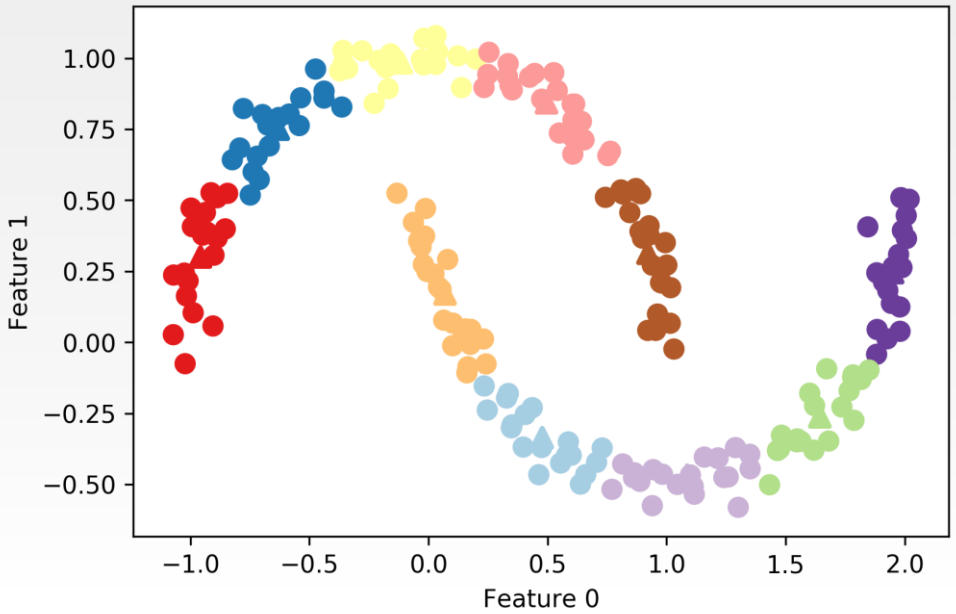
- k-means as a decomposition method
- PCA - find directions of maximum variance in the data
- NMF - find additive components, corresponding to “extremes” or “parts” of the data
- PCA, NMF - Express the data points as a sum of components
- k-means
  - represent each data point using its cluster centre
  - The number of clusterings user determined
  - cluster centres as a “single component” that represents all the data points in the cluster

Vector quantization vs PCA & NMF

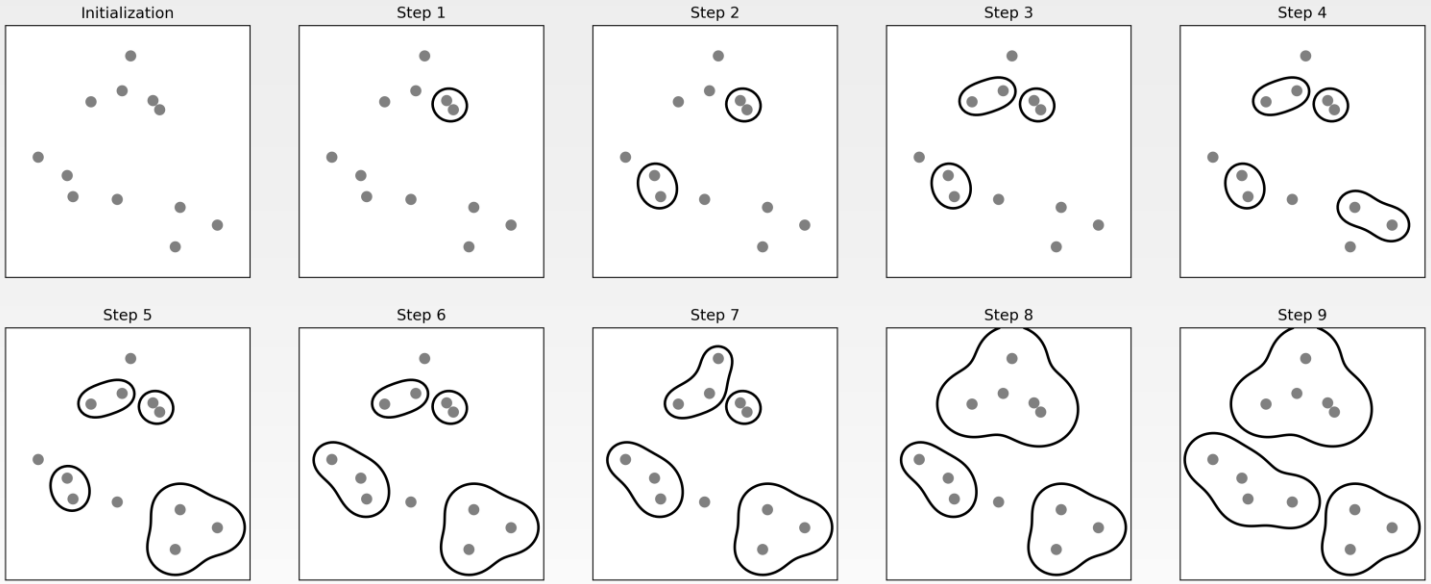


Vector quantization at low dimensionality

we can use more clusters than there are input dimensions and classes.



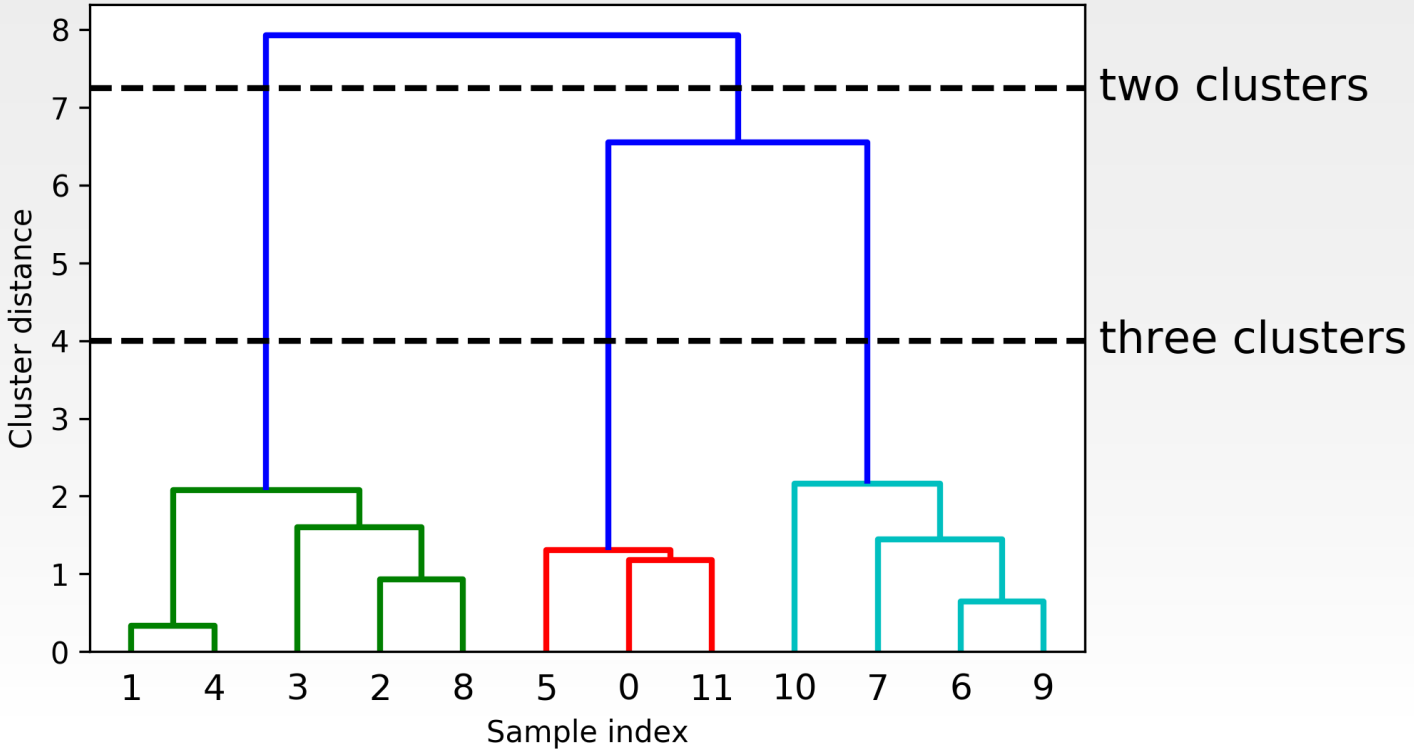
Agglomerative clustering



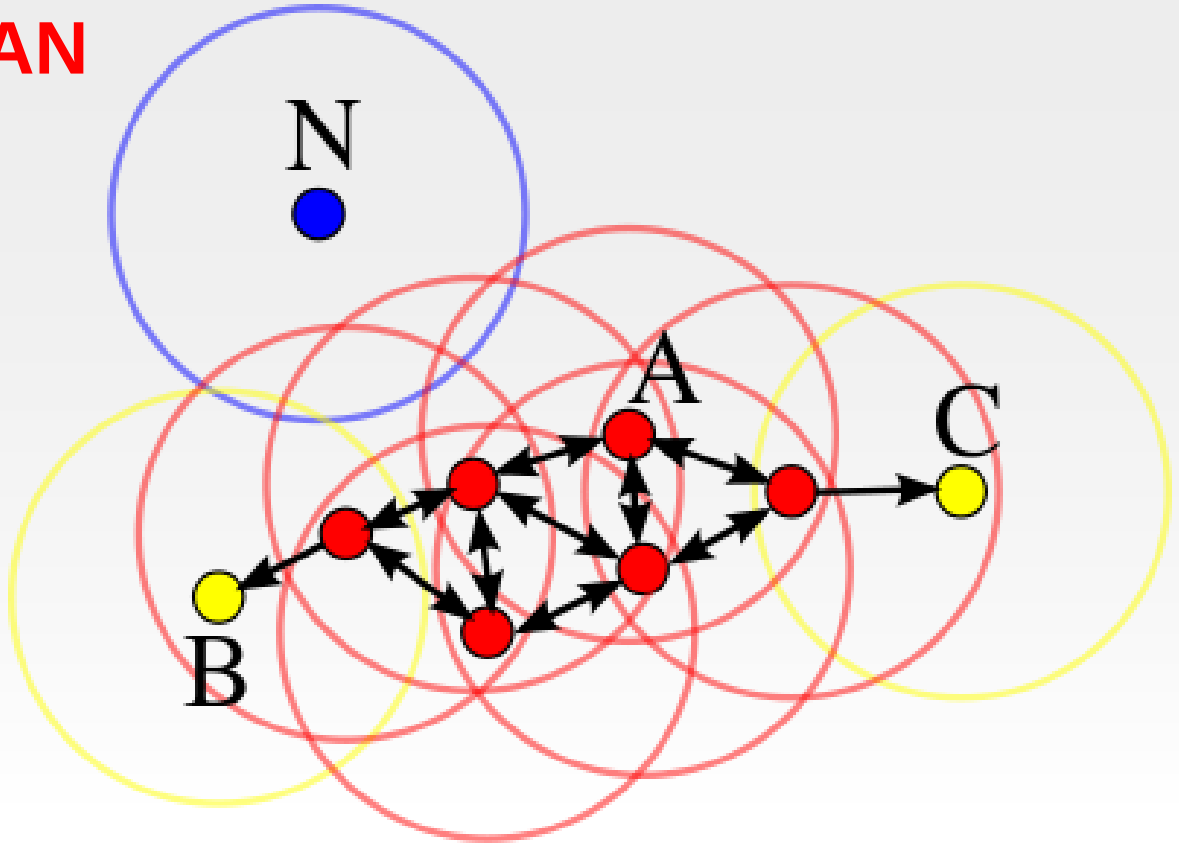
Agglomerative clustering

- Each point is declared as its own cluster
- Two most similar clusters are merged until some stopping criterium is satisfied
- Different measures for similarity yield different clustering algorithms
- In scikit-learn: ward, average, complete,single

Dendrograms



DBSCAN



DBSCAN algorithm

- Start by picking an arbitrary point
- Find all points within a distance of eps(ilon) or less from the start point
- If you find less than min_samples points, then
    - label the start point as “noise” (= does not belong to any cluster),
- else
    - label it as “core sample” and assign it a new cluster label
- Visit all the neighbours within eps distance of the new core samples.
    - If they do not belong to a cluster they are assigned to the new cluster label just created.
    - If they satisfy the core samples criterium mark as «core sample»
- The cluster grows until there are no more core samples left to grow from in the cluster
- Pick another unvisited point by random and repeat

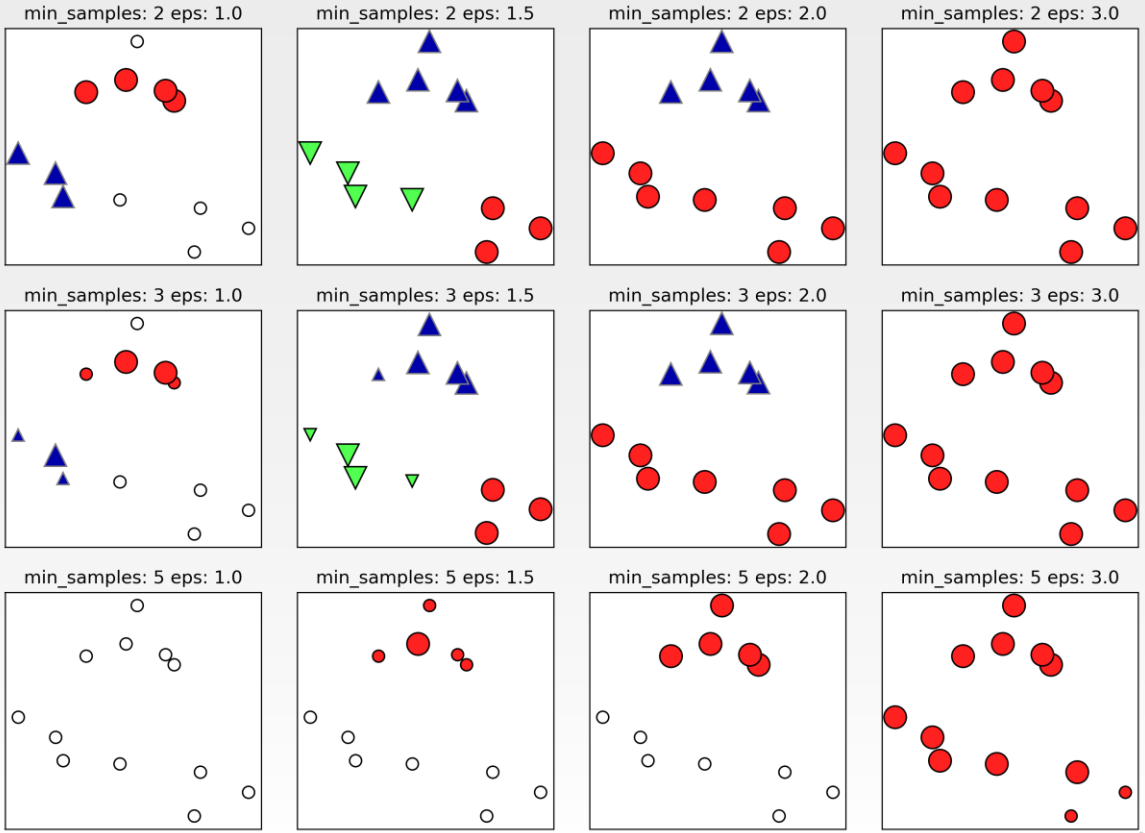
DBSCAN properties

- DBSCAN works by identifying points that are in “crowded” regions of the feature space - referred to as “dense regions” of the feature space
- At the end there are three kinds of points:
  - core samples,
  - points within eps of core samples (**boundary points**)
  - noise
- Multiple runs of the algorithm
  - the same clustering of core points and noise
  - boundary points cluster memberships depend on the order in which the core samples have been labeled

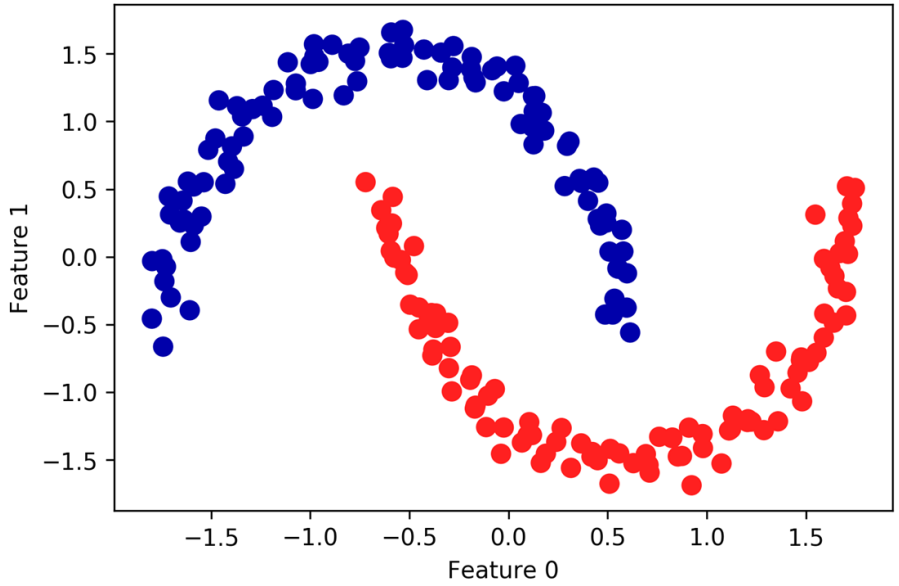
DBSCAN Pros and Cons

- Pros
  - It doesn’t require specification of number of clusters.
  - It can find arbitrarily shaped clusters.
  - Notion of noise making it robust vs outliers.
  - Algorithm is designed to use with databases that can accelerate regional queries.
- Cons:
  - The algorithm is not deterministic on border points.
  - It is sensitive to the value of eps. Setting eps requires understanding of scale of the data.
  - Distance measure used has a big impact of the performance
  - It cannot cluster datasets with large differences in densities.

DBSCAN – Variation in parameters

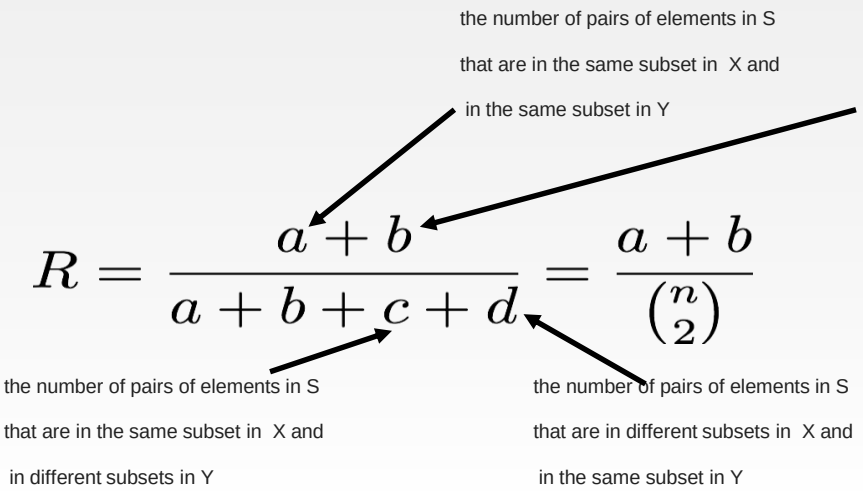


DBSCAN and half moon data



Evaluating with ground truth

- Done during clustering algorithm development phase to estimate clustering quality
- Adjusted rand index (ARI)



$$S = \{o_1, \ldots, o_n\}$$

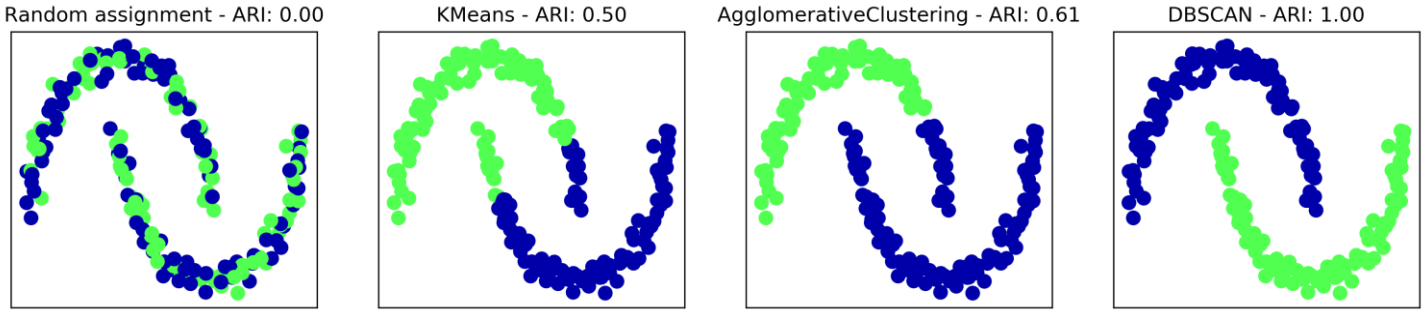
$$S = X_1 \cup \ldots \cup X_k$$

$$S = Y_1 \cup \ldots \cup Y_l$$

$$X = \{X_1, \ldots, X_k\}$$

$$Y = \{Y_1, \ldots, Y_l\}$$

ARI for half moon data



Evaluating without ground truth

- Silhouette coefficient is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).
- Ranges from −1 to +1. High value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.
- If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

Robustness-based clustering metrics

- Not available in scikit-learn
- Run an algorithm after adding some noise in the data or using different parameter settings and compare the outcomes
- If many algorithm parameters and many perturbations in the data return the same result, it is likely to be trustworthy
- Often necessary to inspect the clustering “manually”

uib.no