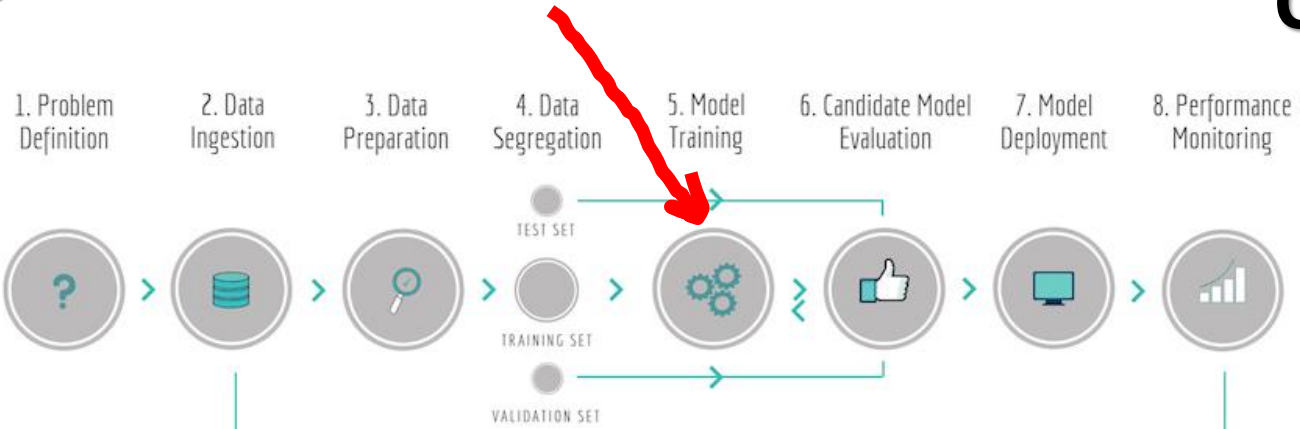
Universidade do Minho
Departamento de Informática
ISLab
Synthetic Intelligence Lab

DADOS e APRENDIZAGEM AUTOMÁTICA

Unsupervised Learning

MESTRADO (integrado) EM ENGENHARIA INFORMÁTICA

Contents



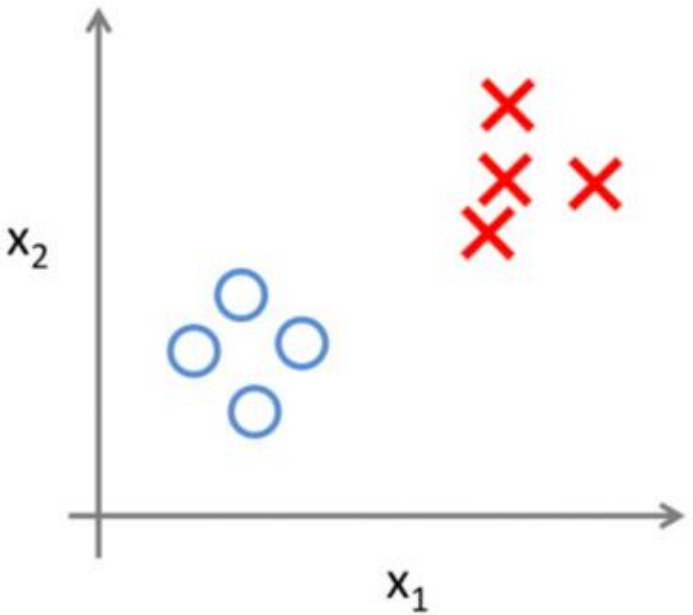
Unsupervised Learning:

- **Dimensionality Reduction**
- **Anomaly Detection**
- **Clustering**

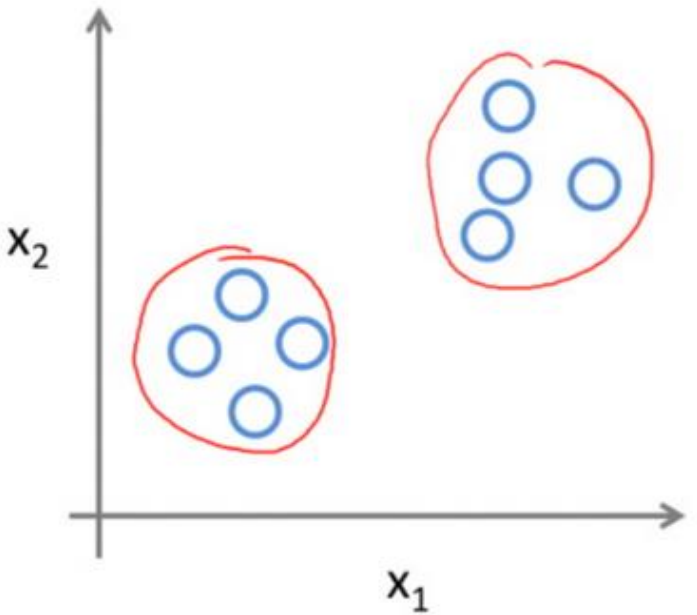
! Unsupervised learning is hard, because there is no error metric to evaluate how well the algorithm performed.

Unsupervised Learning

Supervised Learning
Labeled data

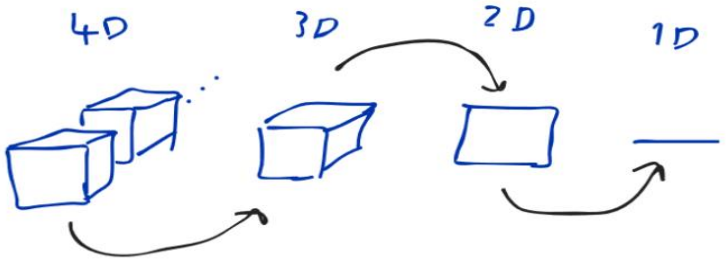


Unsupervised Learning
unlabeled data



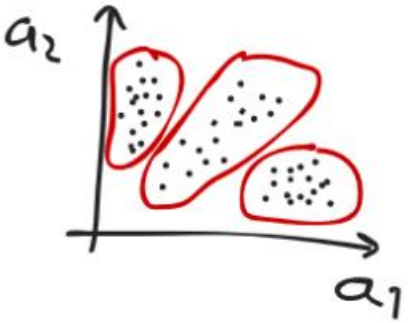
Unsupervised Learning - tasks

Dimensionality Reduction - task of reducing the number of input features in a dataset (not samples).



Anomaly Detection - task of detecting instances that are very different from the norm.

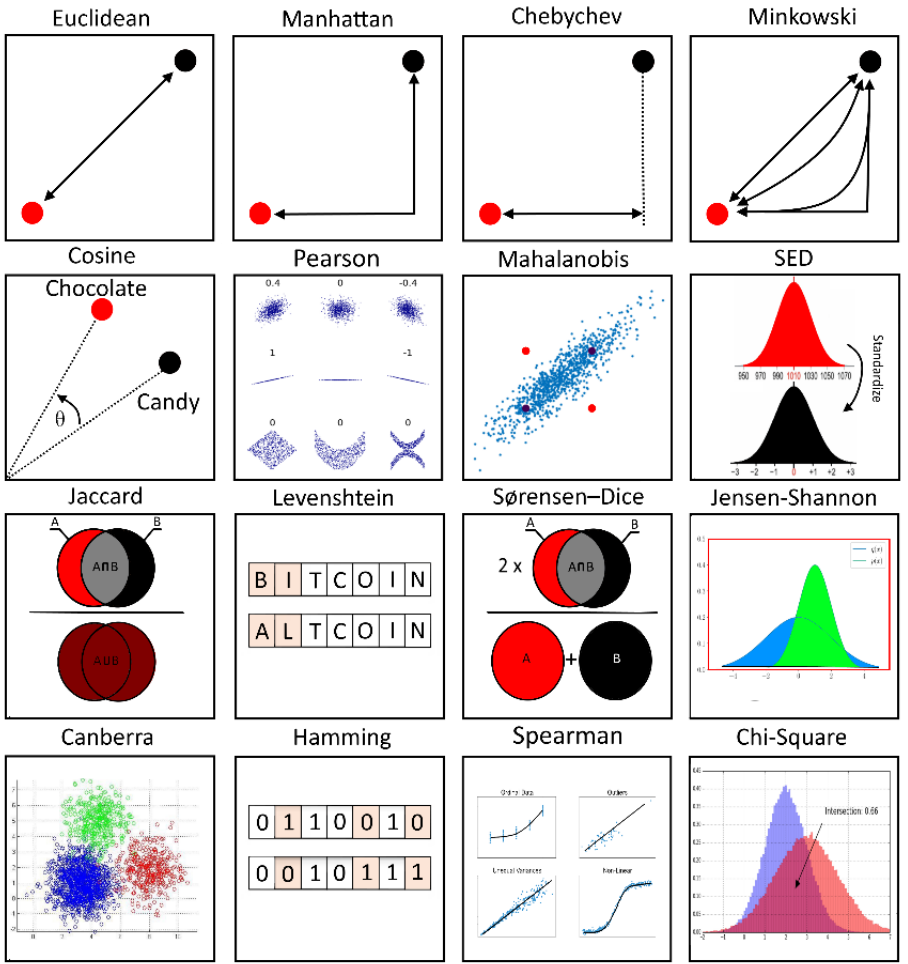
Clustering - task of grouping similar instances into clusters.



Distance measures

Similarity measures:

- Euclidean or Manhattan distance, for continuous attributes;
- Jacqard coefficient, for discrete/binary attributes;
- etc.



Unsupervised Learning - Algorithms

Dimensionality Reduction:

Principal Component Analysis (**PCA**);
Manifold Learning - LLE, Isomap, **t-SNE**;
Autoencoders and others.

Anomaly Detection:

Isolation Forest;
Local Outlier Factor;
Minimum Covariance Determinant;
other algorithms initially designed for dimensionality reduction or supervised learning.

Clustering:

K-Means;
Hierarchical Clustering and Spectral Clustering;
DBSCAN and OPTICS;
Affinity Propagation;
Mean Shift and BIRCH;
Gaussian Mixture Models;

Dimensionality reduction

Applications of dimensionality reduction algorithms:

Data Visualization & Data Analysis - reduce the number of input features to three or two and use data visualization techniques to get insights about the data.

Preparatory tool for other machine learning algorithms. More input features often make a prediction task more challenging to model. Since many algorithms (both from supervised and unsupervised learning (e.g. regression/classification, clustering)) do **not work well with sparse or high-dimensional data**, dimensionality reduction algorithms can greatly increase the quality of models.

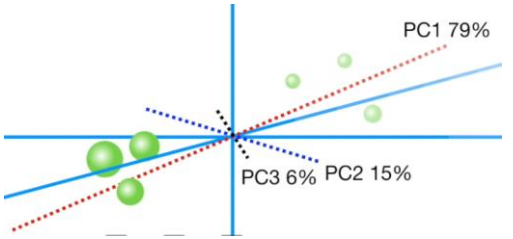
Methods are commonly divided into:

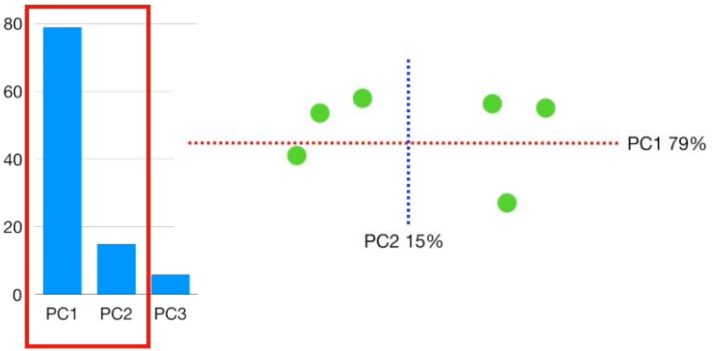
Feature Selection - find a subset of the input features (drops …).
Feature Projection (or Feature Extraction) - find the optimal projection of the original data into some low-dimensional space.

Dimensionality reduction - PCA

Principal Component Analysis (PCA) uses the projection of the original data into the principal components. The principal components are orthogonal vectors that describe the maximum amount of residual variation.

1 - **Standardize** the data
2 - Construct a **correlation matrix**
3 - Calculate the **eigenvectors/unit vectors and eigenvalues**. Eigenvalues are scalars by which we multiply the eigenvector of the covariance matrix.
4 - Sort the eigenvectors in highest to lowest order and **calculate the percentage of variation** that each PC accounts for.
5 - **Select the number of components**, (the so-called elbow method can be used). Plot a graph of the cumulative sum of the explained variance and then select the number of components that explains the desired ratio of information (usually 80% or 95%).





PCA versions:

Incremental PCA - for online learning or when data doesn't fit in memory
Randomized PCA - stochastic algorithm that allows to quickly estimate the first N components
Kernel PCA - kernel trick allows performing complex nonlinear projections

Universidade do Minho
Departamento de Informática
ISLab
Synthetic Intelligence Lab

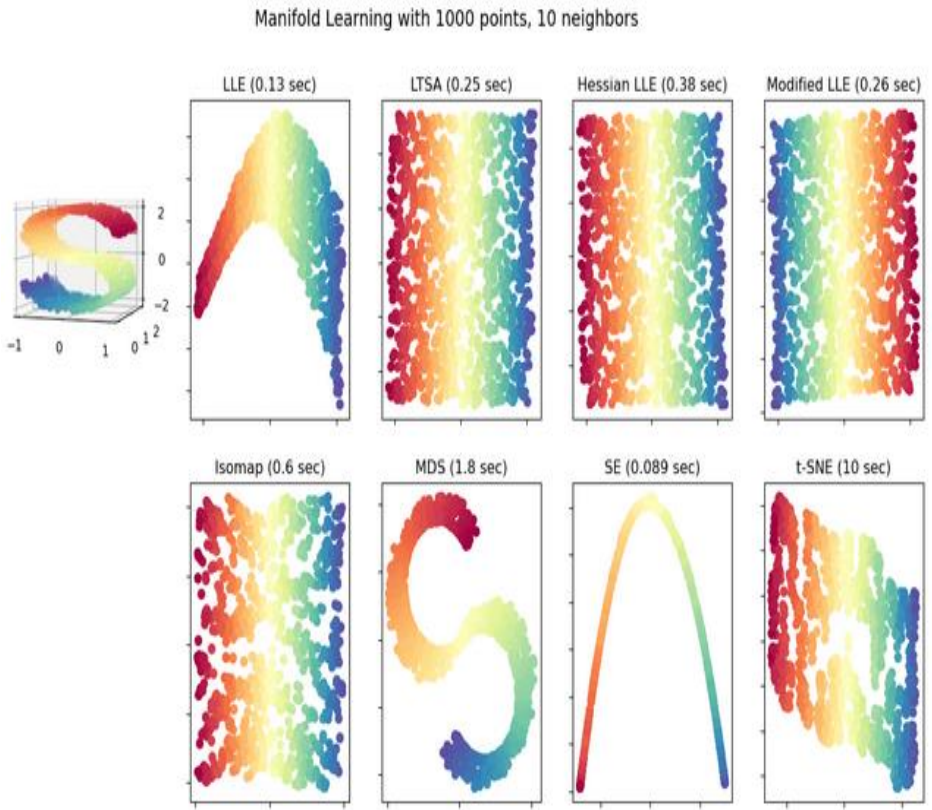
Dimensionality reduction – Manifold learning

Manifold Learning algorithms are based on some **distance measure conservation**. These algorithms are reducing the dimensionality while saving distances between objects.

LLE (Locally Linear Embedding) studies the linear connections between data points in the original space, and then tries to move to a smaller dimensional space, while preserving within local neighborhoods. There are a lot of modifications of this algorithm, like Modified Locally Linear Embedding (MLLE), Hessian-based LLE (HLLE), and others.

Isomap (short for Isometric Mapping) creates a graph by connecting each instance to its nearest neighbors, and then reduces dimensionality while trying to preserve the geodesic distances (distance between two vertices in a graph) between the instances.

t-SNE (**t-distributed Stochastic Neighbor Embedding**) reduces dimensionality by saving the relative distance between points in space - so it keeps **similar instances close** to each other and **dissimilar instances apart**. Most often used for data visualization.

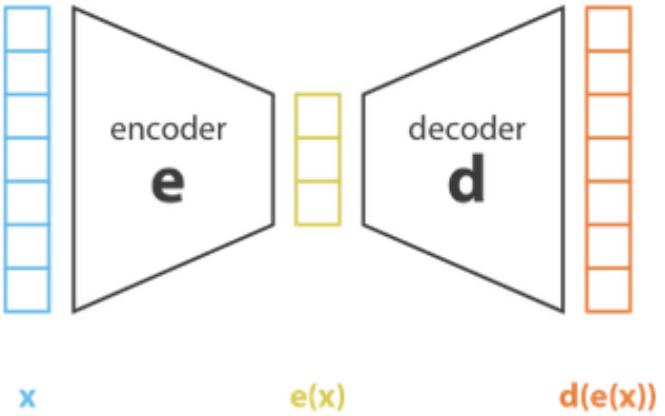


Dimensionality reduction – Autoencoders

Autoencoder is a artificial neural network that tries to output values that are as similar as possible to the inputs when the network structure implies *a bottleneck* - a layer where the number of neurons is much fewer than in the input layer.

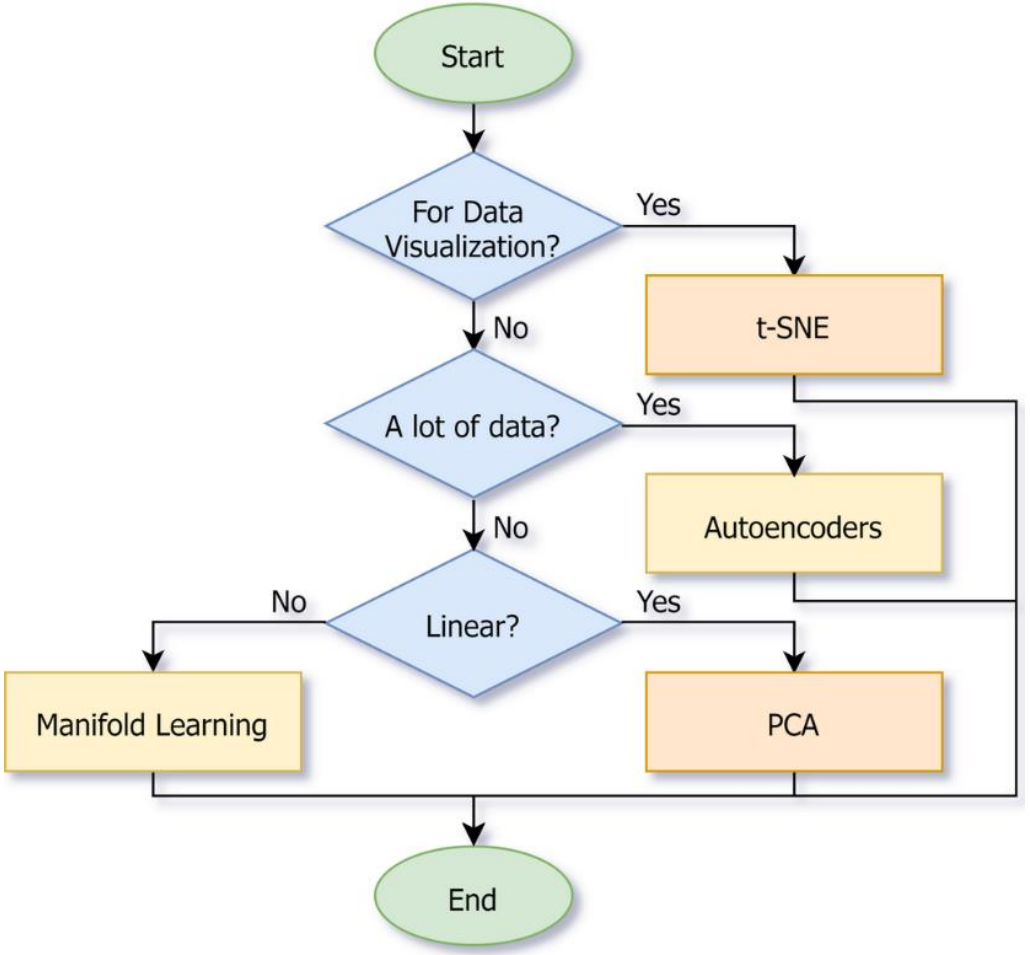
Autoencoder variations:

- **Denoising Autoencoders** that can help clean up the images or sound
- **Variational Autoencoders** that deal with distributions instead of specific values
- **Convolutional Autoencoders** for images
- **Recurrent Autoencoders** for time series or text



Dimensionality reduction – Algorithm selection

!!! Make sure you scaled the data.



Anomaly Detection

Anomaly detection (also outlier detection) is the task of detecting abnormal instances (outliers) for:

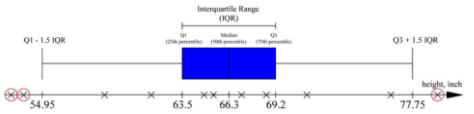
data cleaning - removing outliers from a dataset before training another model

anomaly detection tasks - fraud detection, detecting defective products in manufacturing etc.

ISLab
Synthetic Intelligence Lab
Universidade do Minho
Departamento de Informática

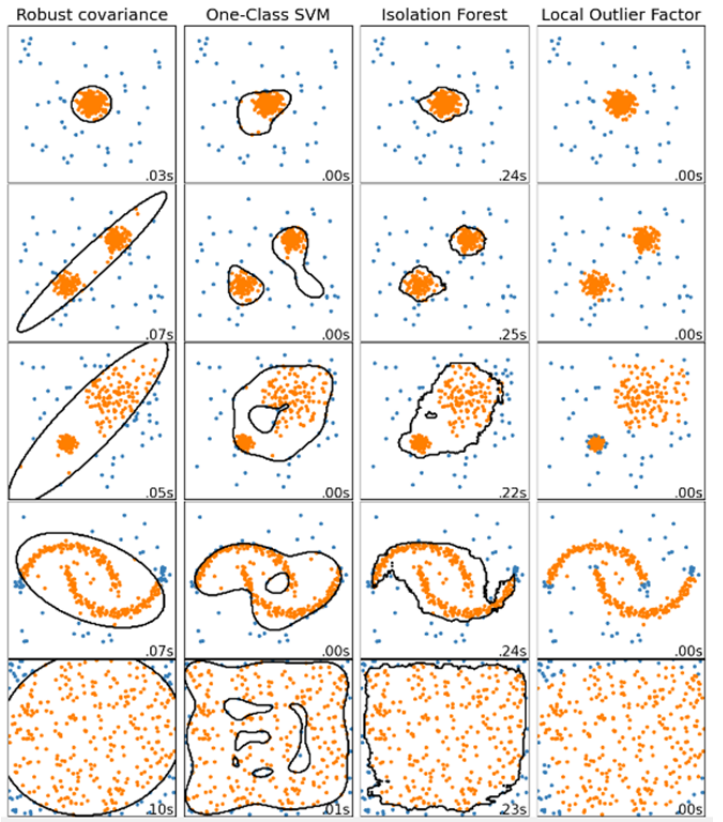
Anomaly Detection

Statistical Approaches (Interquartile Range (IQR)or Tukey Method for Outlier Detection).

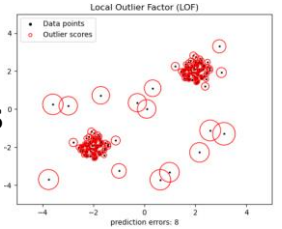


Clustering and dimensionality reduction algorithms

Isolation Forest and SVM (one-class SVM)

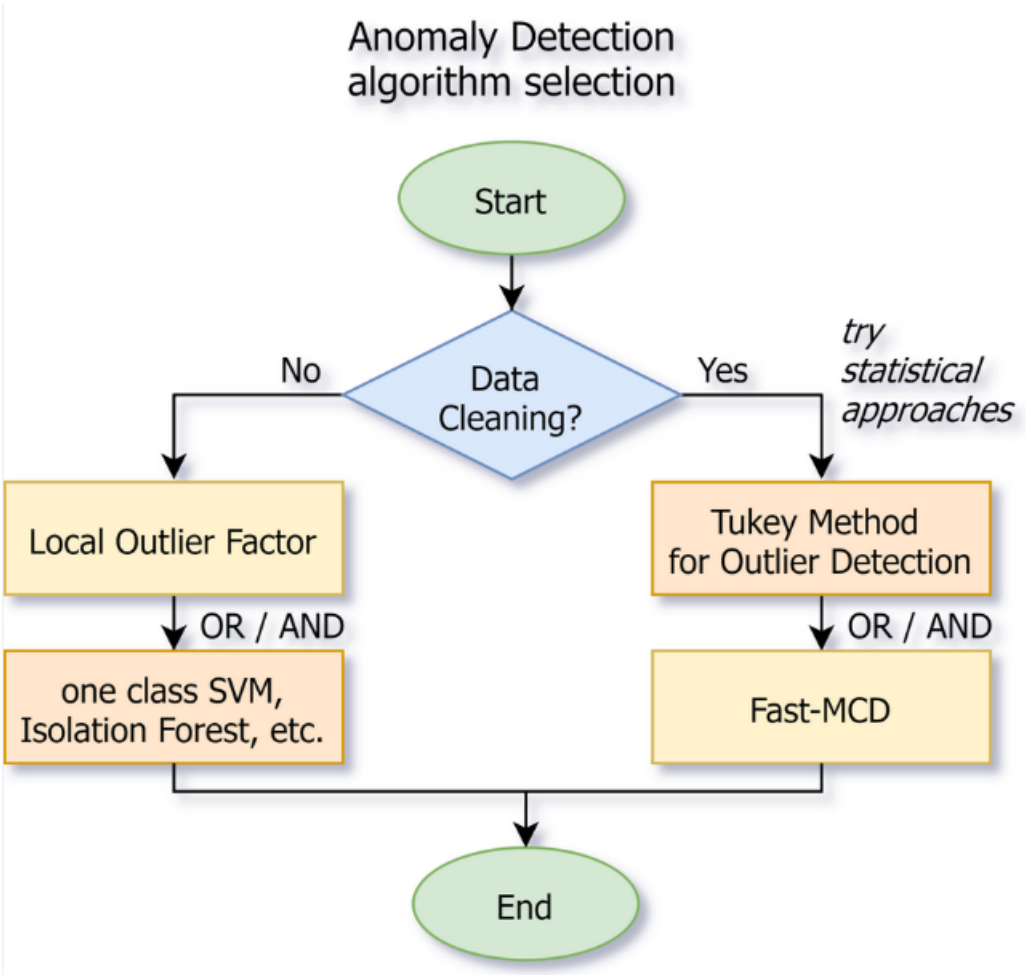


Local Outlier Factor (LOF) - based on the assumption that the anomalies are located in **lower-density regions**



Minimum Covariance Determinant (**MCD**) useful for data cleaning. It assumes that inliers are generated from a single Gaussian distribution, and outliers were not generated from this distribution.

Anomaly Detection – Algorithm selection



Clustering

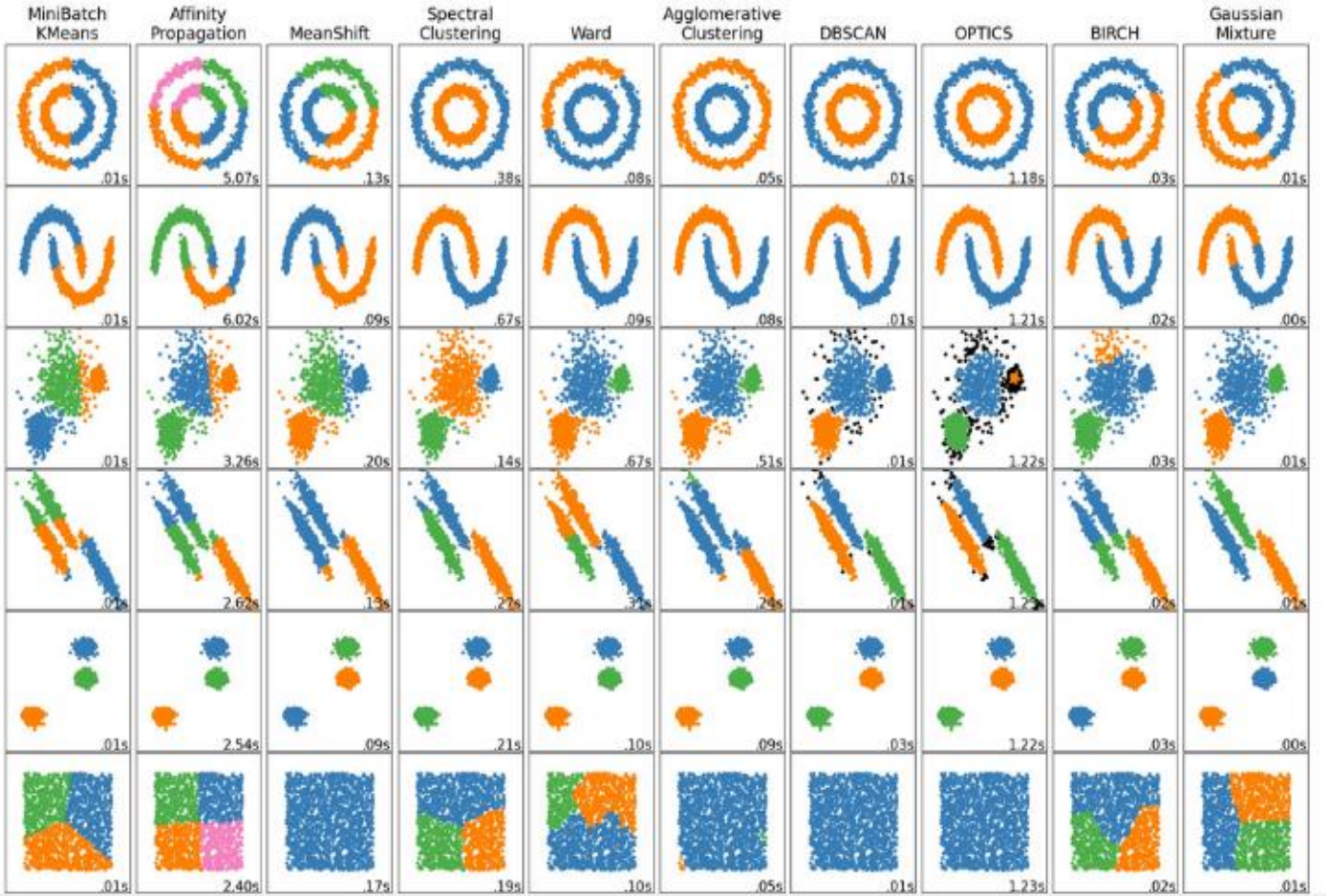
Clustering is the task of grouping the population of unlabeled data points into clusters in a way that data points in the same cluster are more similar to each other than to data points in other clusters.

Applications:

- Recommendation Engines
- Clustering similar news articles
- Medical Imaging
- Image Segmentation
- Anomaly detection
- Pattern Recognition

Clustering

All clustering algorithms require data preprocessing (e.g. dimensionality reduction) and standardization.



Clustering – K-Means

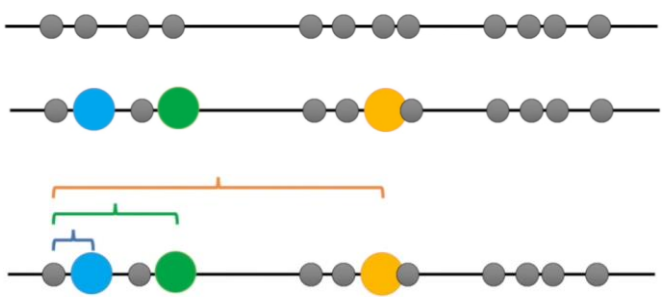
K-Means - Algorithm based on the centroid concept. Centroid is a geometric center of a cluster (mean of coordinates of all cluster points).

1st - Centroids are initialized randomly (this is the basic option, but there are other initialization techniques).

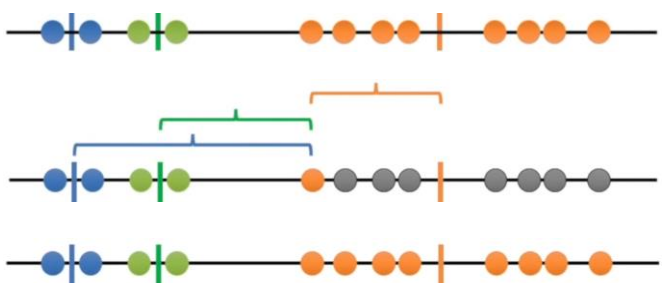
2nd -iteratively do the two following steps, while centroids are moving:

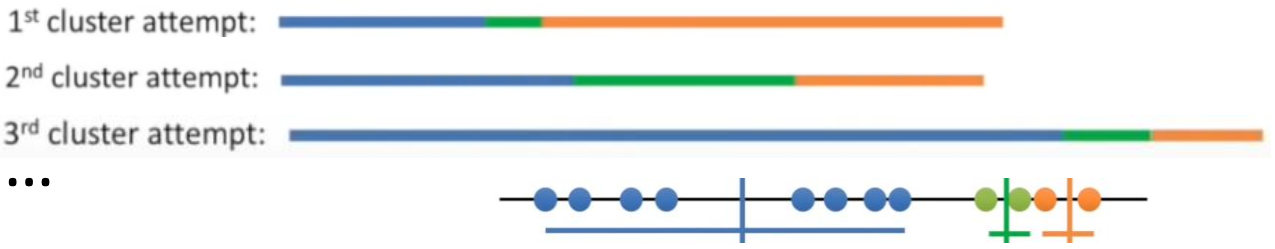
- Update the clusters - for each data point assign it a cluster number with the nearest centroid;
- Update the clusters' centroids - calculate the new mean value of the cluster elements to move centroids.

3rd – Calculate total variance



Calculate means:

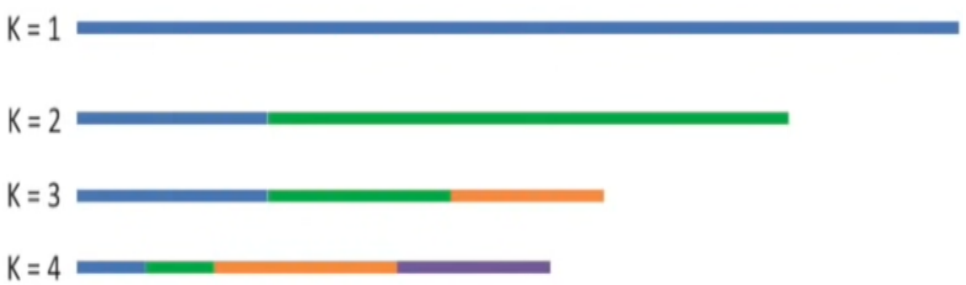


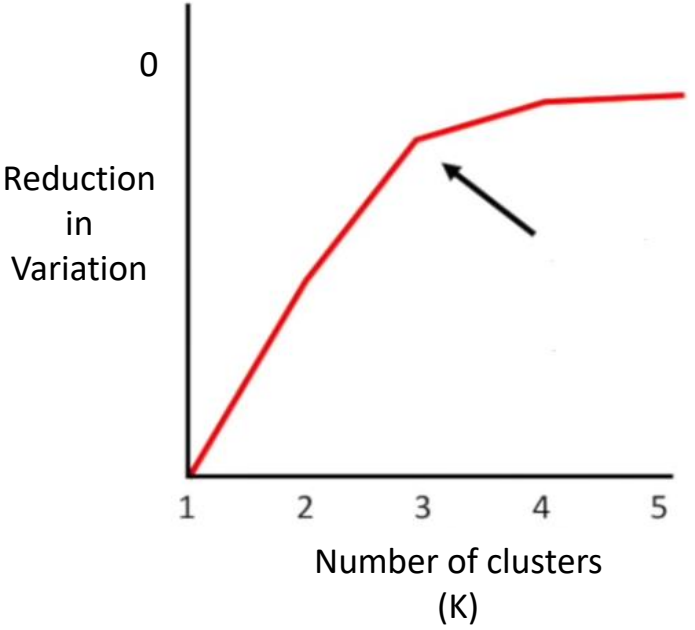


K = ?????

Elbow method

In cluster analysis, the **elbow method** is a heuristic used in determining the **number of clusters** in a data set. The method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use. The same method can be used to choose the number of parameters in other data-driven models, such as the **number of principal components** to describe a data set.





Clustering – K-Means

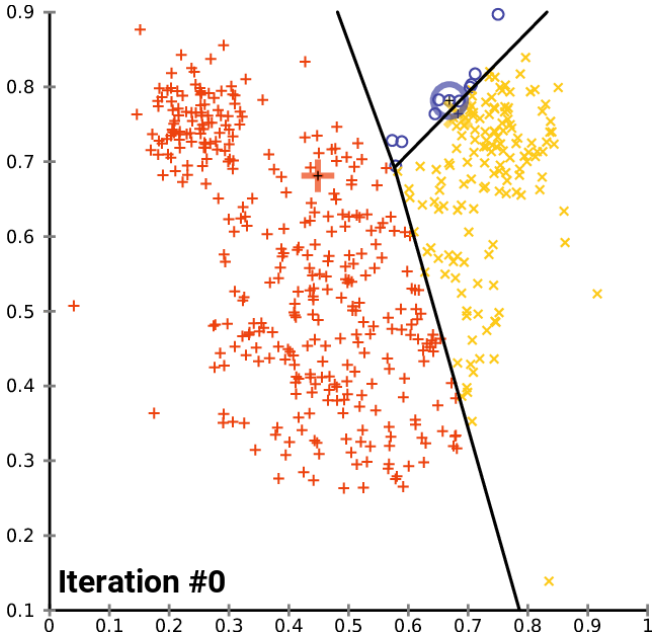
K-Means - Algorithm based on the **centroid concept**. Centroid is a geometric center of a cluster (mean of coordinates of all cluster points).

Strengths:

- **Simple** and intuitive;
- Scales to **large datasets**;
- As a result, we also have **centroids** that can be used as standard cluster representatives

Weaknesses:

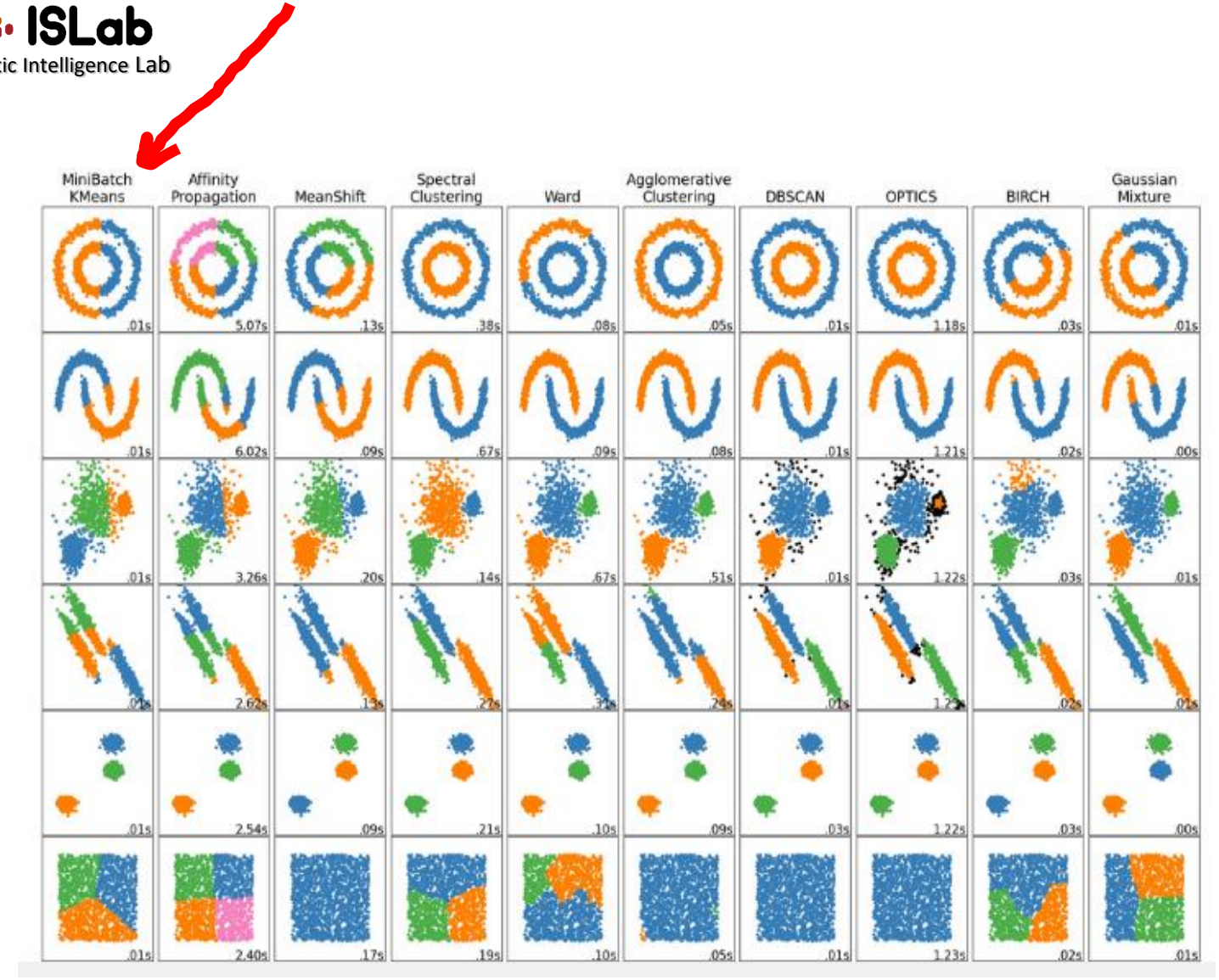
- Knowledge about the **number of clusters** is necessary and must be specified as a parameter;
- Does not cope well with a very large **number of features**;
- Separates only **convex and homogeneous** clusters well;
- Can result in poor local solutions, so it needs to be **run several times**.



Mini Batch K-Means – uses a random subsample instead of the whole dataset for calculations

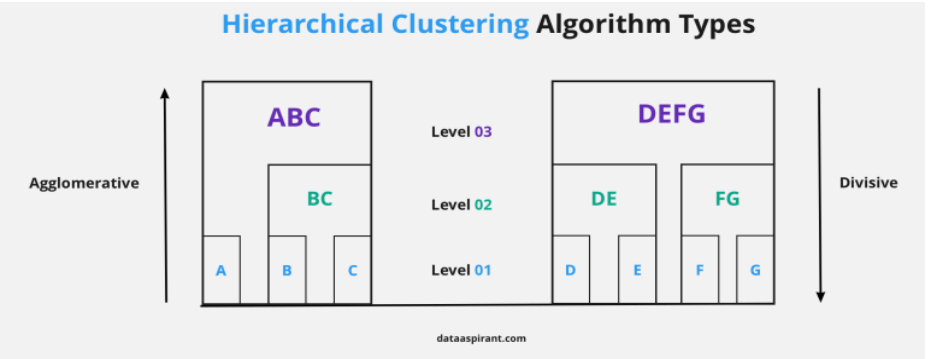
K-Medoids - It uses the dissimilarities(total mismatches) between the data points. The lesser the dissimilarities the more similar our data points are. It uses data points (Modoids) instead of means. **K-Modes** uses modes (most frequent value) as center.

Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

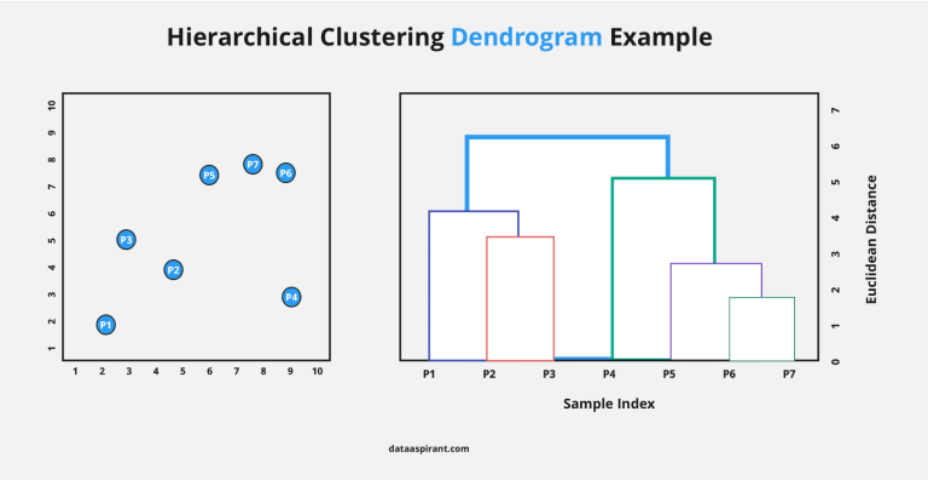
Clustering – Hierarchical Clustering

Hierarchical clustering (also Hierarchical Cluster Analysis (HCA) or **Agglomerative Clustering**) is a family of clustering algorithms that build a hierarchy of clusters during the analysis:
- Start with points as individual clusters.
- At each step, merge the closest pair of clusters until only one cluster (or K clusters left).



Strengths :
- **Simple** and intuitive;
- Works well when data has a hierarchical structure;
- Knowledge about the **number of clusters is not necessary**.

Weaknesses :
- Requires **additional analysis** to choose the resulting number of clusters;
- Can never undo any previous step;
- A greedy algorithm can result in poor local solutions.



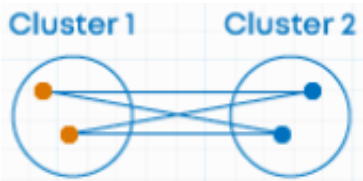
Clustering – distance

Measure of the distance between two clusters (linkage methods):

- **Single Linkage** – distance between closest elements in clusters

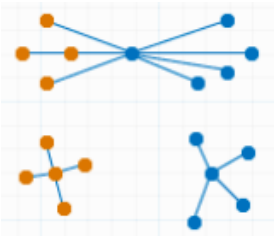
- **Complete Linkage** – maximum distance between elements in clusters

- **Average Linkage** – Average of the distance of all pairs

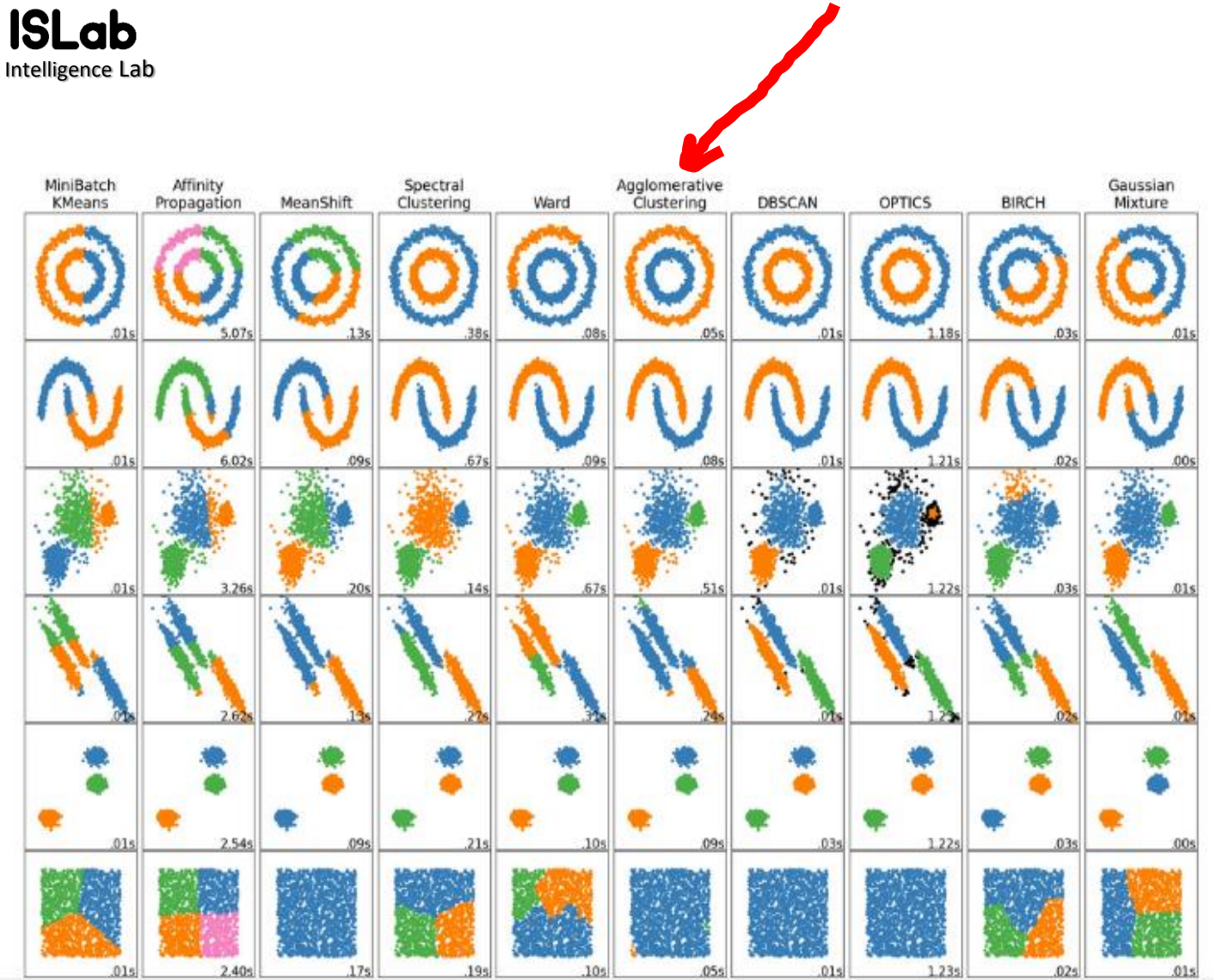


- **Centroid Linkage** – distance between the centroids of two clusters

- **Ward's Linkage** – Combining clusters where increase in within cluster variance is to the smallest degree (similarity of two clusters).



Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

Clustering – Spectral Clustering

Spectral clustering approach is based on **graph theory** and **linear algebra**. This algorithm uses the spectrum (set of eigenvalues) of the **similarity matrix** (that contains the similarity of each pair of data points) to perform dimensionality reduction. Then it uses some of the clustering algorithms in this low-dimensional space (sklearn.cluster.SpectralClustering class uses K-Means). If we have P data points each with N features, input matrix to K-means would be N by P, while input matrix to spectral clustering would be P by P

$$\begin{vmatrix} X_{11} & \dots & X_{1j} & \dots & X_{1p} \\ \dots & & \dots & & \dots \\ X_{i1} & \dots & X_{ij} & \dots & X_{ip} \\ \dots & & \dots & & \dots \\ X_{n1} & \dots & X_{nj} & \dots & X_{np} \end{vmatrix}$$

Strengths :

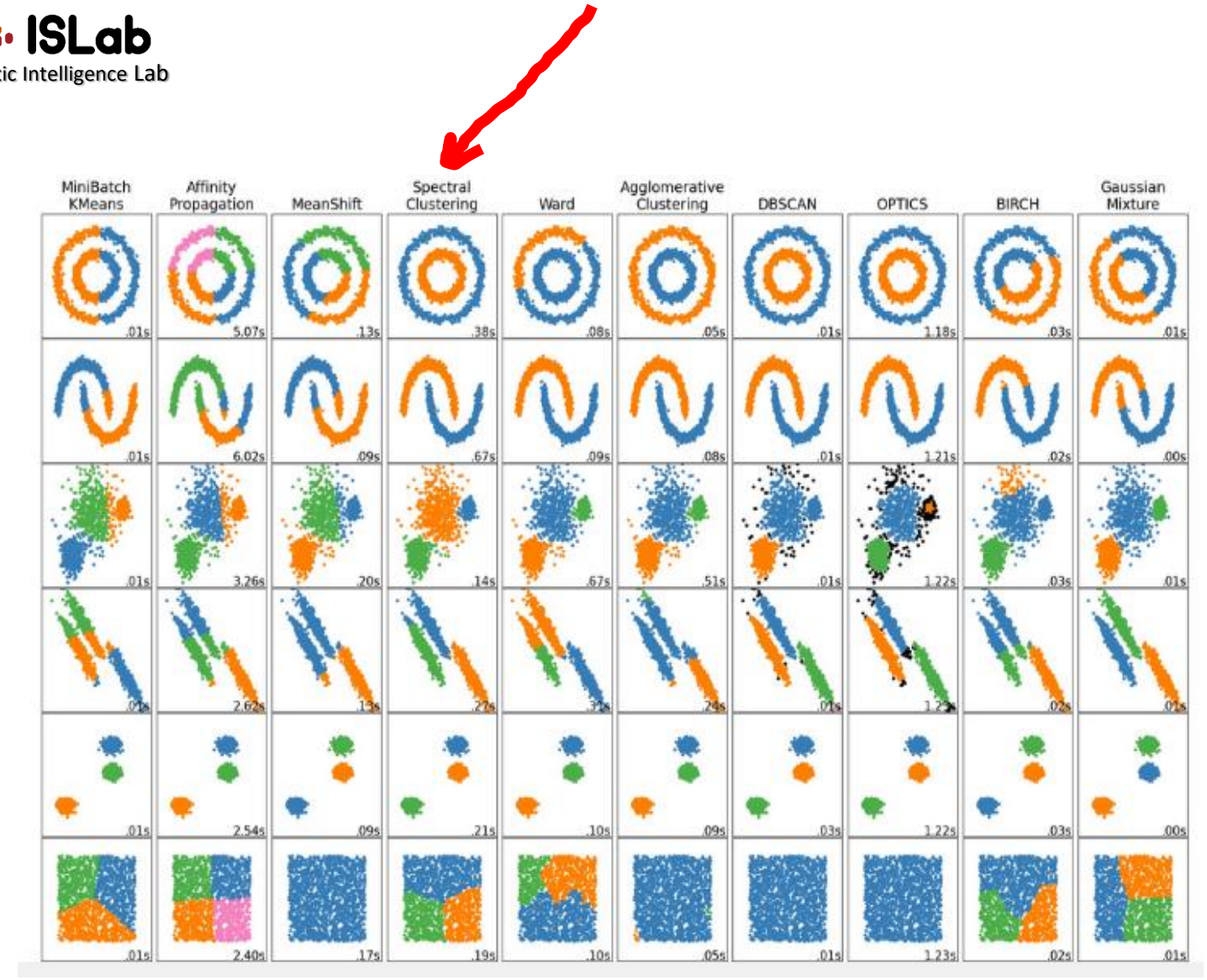
- Can detect **complex cluster** structures and shapes;
- Can be used to search for **clusters in graphs**.

Weaknesses :

- Knowledge about the **number of clusters** is necessary and must be specified as a parameter;
- Does not cope well with a very large number of instances;
- Does not cope well when the clusters have very different sizes.

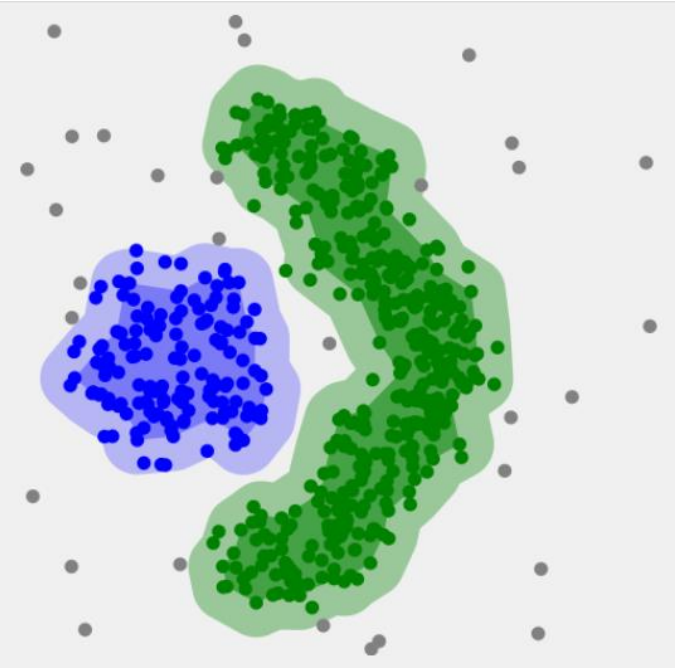
$$\begin{vmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & \dots & 0 & & \\ \dots & \dots & \dots & 0 & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{vmatrix}$$

Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

Clustering – DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise). In this algorithm, clusters are **high-density regions** (where the data points are located close to each other) **separated by low-density regions** (where the data points are located far from each other).
The central concept of the DBSCAN algorithm is the idea of a core sample, which means a sample located in an area of high density. Data point A is considered a core sample if at least min_samples other instances (usually including A) are located within **eps** distance from A.



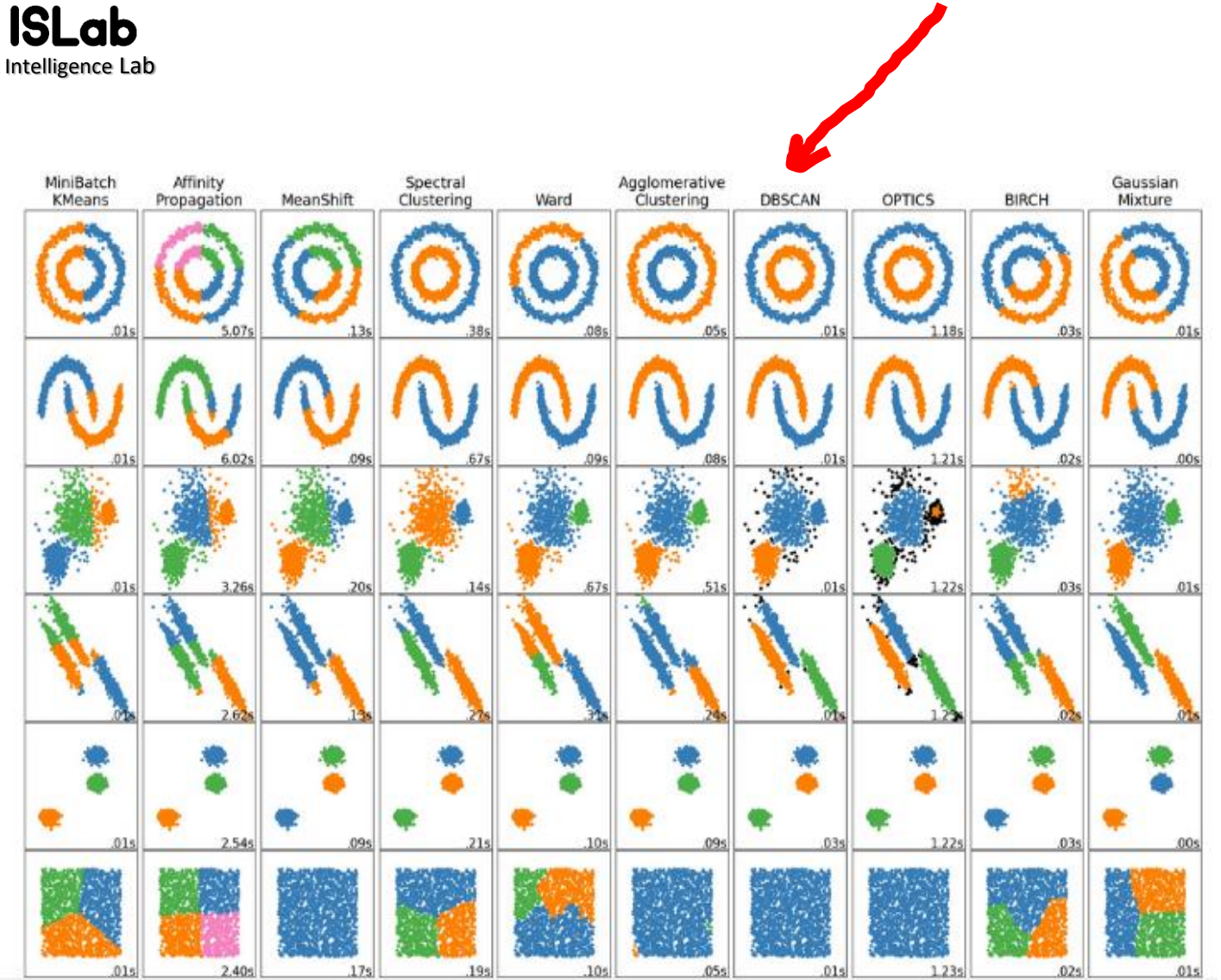
Strengths :

- Knowledge about the **number of clusters is not necessary**;
- Also solves the anomaly detection task.

Weaknesses :

- Need to select and tune the density parameter (**eps**);
- Does not cope well with **sparse data**.

Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

Clustering – Affinity Propagation

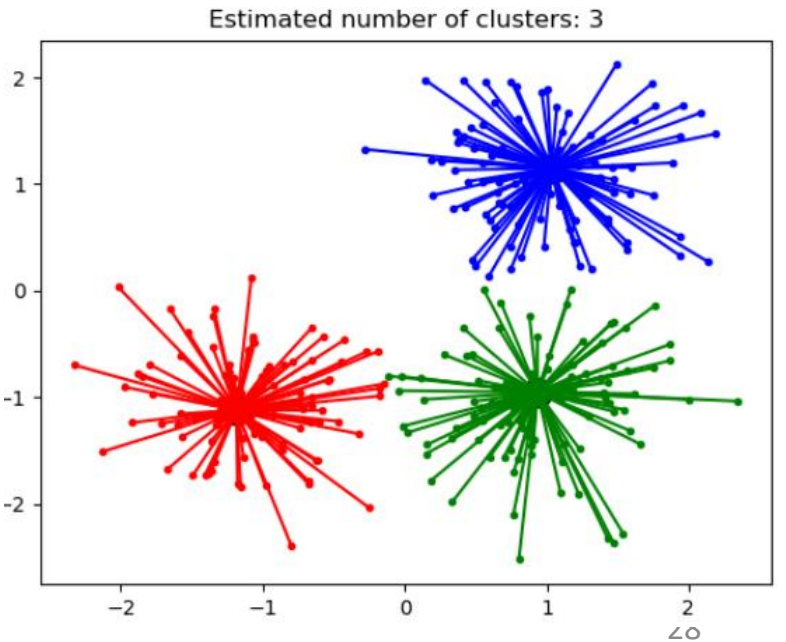
Affinity Propagation algorithm also does not require knowledge about the number of clusters. But unlike DBSCAN, which is a density-based clustering algorithm, affinity propagation is based on the idea of passing messages between data points. Calculating pairwise similarity based on some distance function (i.e. Euclidean distance) this algorithm then converges in some number of standard representatives. A dataset is then described using this small number of standard representatives, which are identified as the most representative instances on the particular cluster.

Strengths :

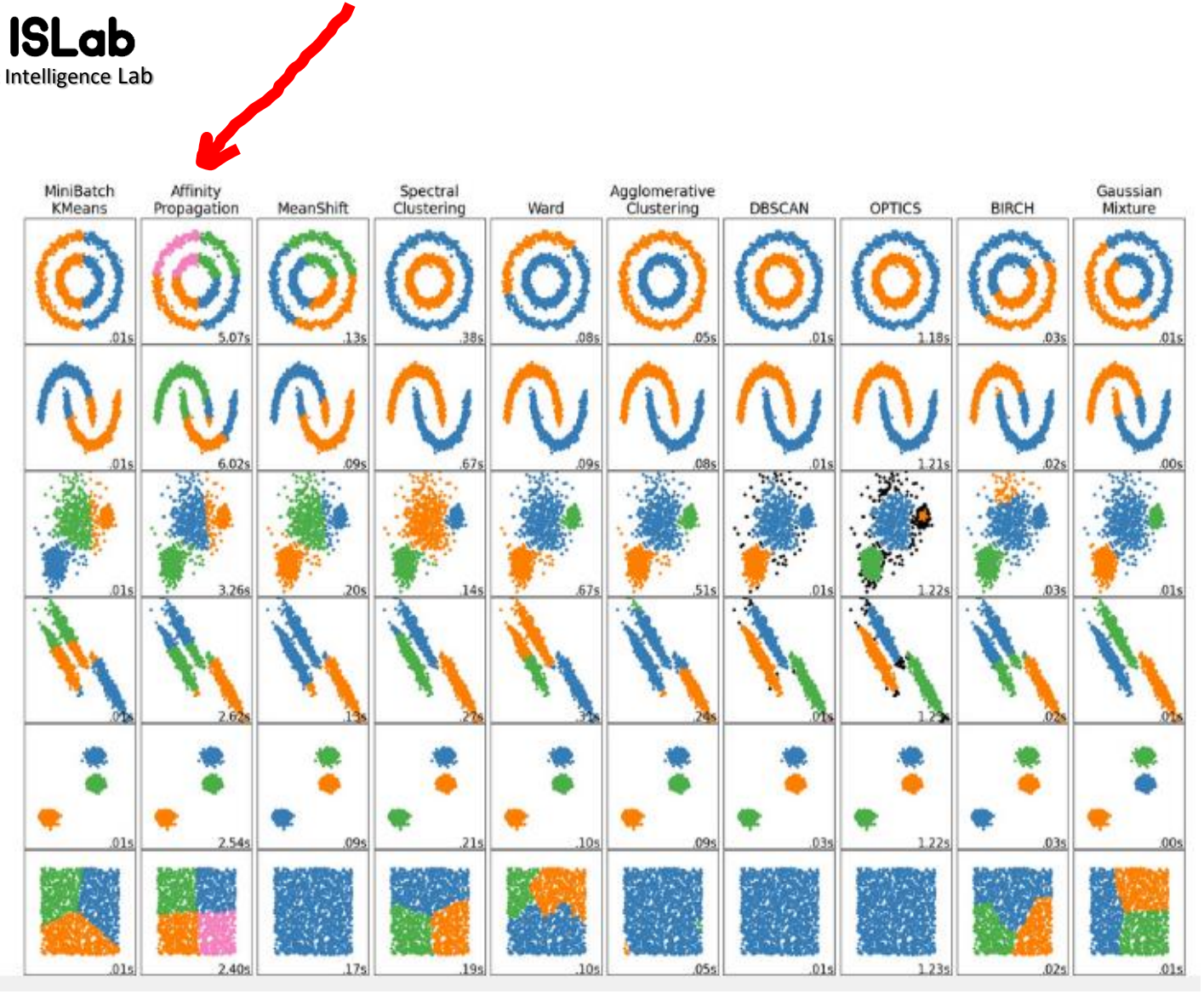
- Knowledge about the **number of clusters is not necessary**;
- As a result, we also have standard representatives of a cluster. Unlike K-Means centroids, these instances are not just average values, but real objects from the dataset.

Weaknesses :

- Works **much slower** than other algorithms due to computational complexity;
- Does not cope well with a **large number of instances**;
- Separates **only convex and homogeneous clusters well**.



Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture
.01s
5.07s
.13s
.38s
.08s
.05s
.01s
1.18s
.03s
.01s
.01s
6.02s
.09s
.67s
.09s
.08s
.01s
1.21s
.02s
.00s
.01s
3.26s
.20s
.14s
.67s
.51s
.01s
1.22s
.03s
.01s
.01s
2.62s
.13s
.27s
.31s
.29s
.01s
1.23s
.02s
.01s
.01s
2.54s
.09s
.21s
.10s
.09s
.03s
1.22s
.03s
.00s
.01s
2.40s
.17s
.19s
.10s
.05s
.01s
1.23s
.02s
.01s

Clustering – Mean Shift

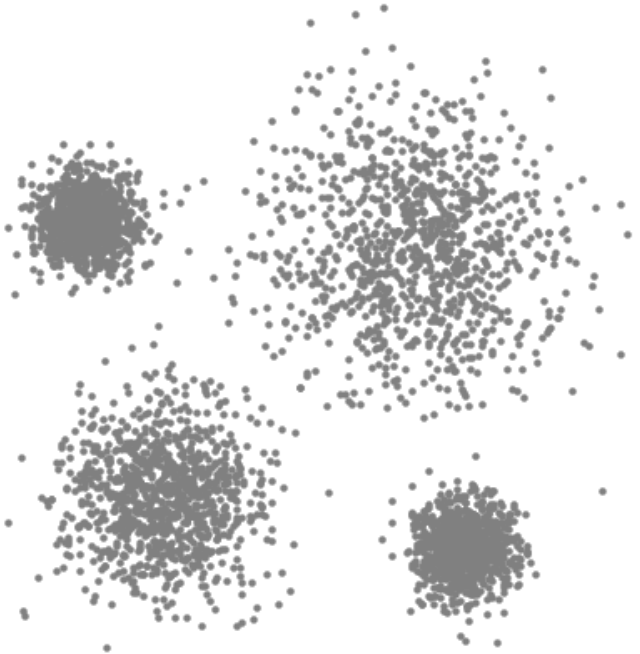
Mean Shift algorithm first places a circle of a certain size (radius of the circle is a parameter called **bandwidth**) in the center of each data point. After that, it iteratively calculates the mean for each circle (the average coordinates among the points inside the circle) and shifts it. These mean-shift steps are performed until the algorithm converges and the circles stop moving.

Strengths :

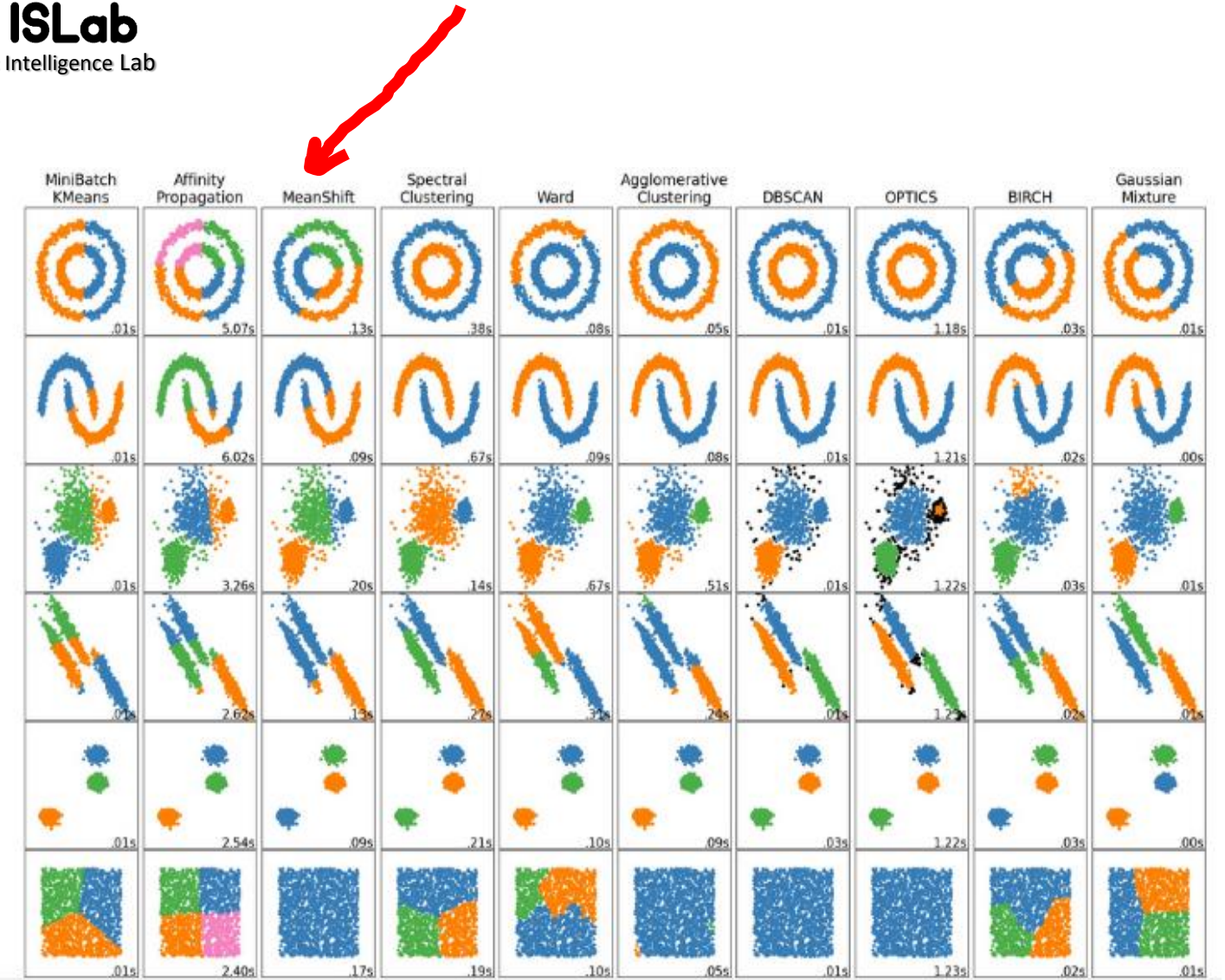
- Knowledge about the **number of clusters is not necessary**;
- Have just one hyperparameter: the **radius of the circles**;
- Solves density estimation task and calculate cluster centroids;
- Does not find a cluster structure where it is not actually present.

Weaknesses :

- Does not cope well with **sparse data** and with a large number of features;
- Does not cope well with a **large number of instances**;
- Does not cope well with **clusters of complex shapes**: tends to chop these clusters into pieces.



Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

Clustering – BIRCH

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies). This **hierarchical clustering algorithm** was designed **specifically for large datasets**. It requires only one scan of the dataset. During training, it creates a dendrogram containing enough information to quickly assign each new data instance to some cluster without having to store information about all instances in memory.

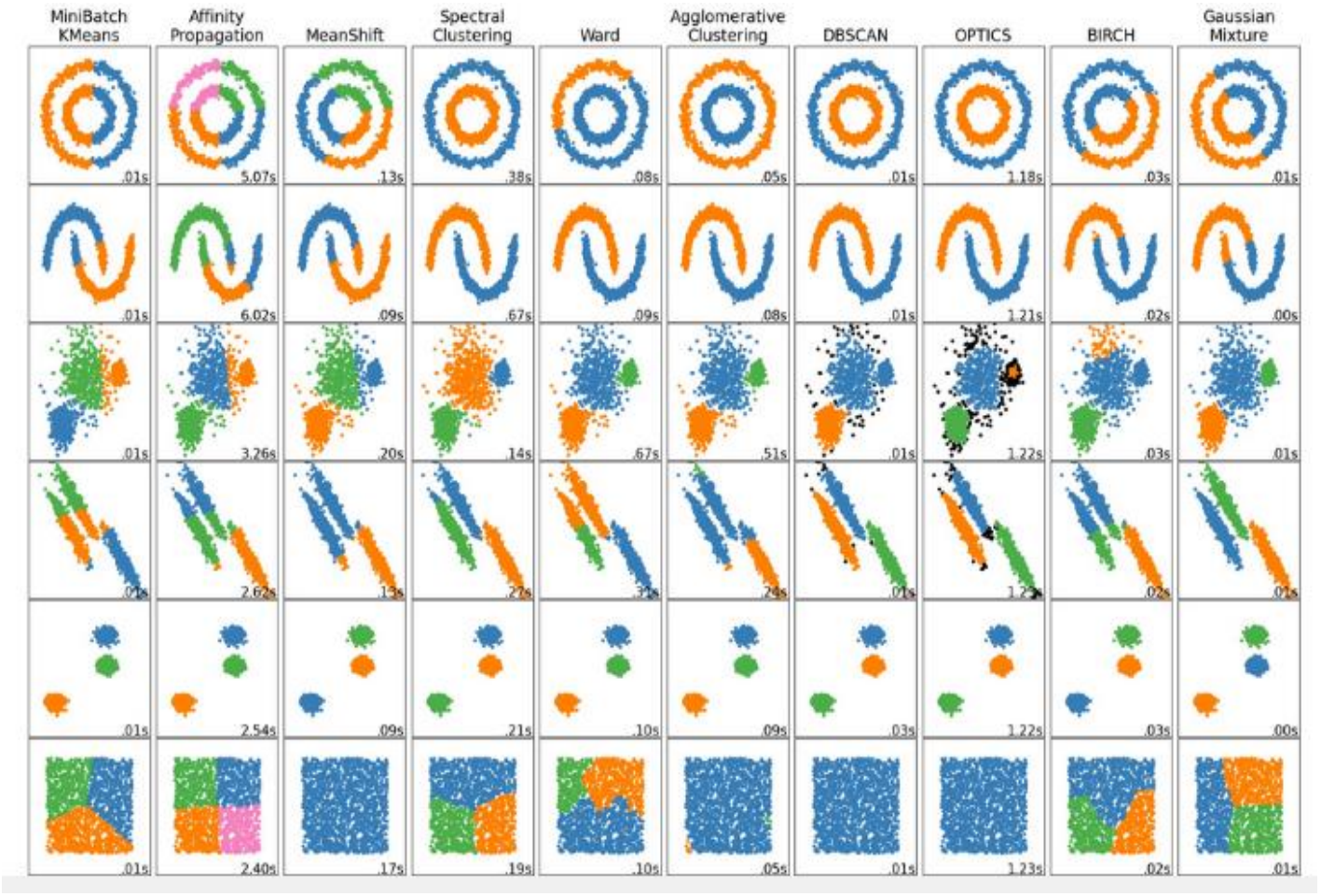
Strengths :

- Was designed **specifically for very large datasets**;
- Show the best quality for a given set of memory and time resources;
- Allows implementing online clustering.

Weaknesses :

- Does not cope well with a **large number of features**.

Clustering


MiniBatch KMeans
Affinity Propagation
MeanShift
Spectral Clustering
Ward
Agglomerative Clustering
DBSCAN
OPTICS
BIRCH
Gaussian Mixture

Clustering – Algorithm selection

The **clustering task is quite difficult** and have a wide variety of applications, so it's almost impossible to build some universal set of rules to select a clustering algorithm - all of them have advantages and disadvantages.

Things become better when you have **some assumptions about your data**, so data analysis can help you with that. What is the approximate number of clusters? Are they located far from each other or do they intersect? Are they similar in shape and density? All that information can help you to solve your task better.

If the number of clusters is unknown, a good initial approximation is the **square root of the number of objects**. You can also first run an algorithm that does not require a number of clusters as a parameter (**DBSCAN or Affinity Propagation**) and use the resulting value as a starting point.

References

- Data Mining: Concepts and Techniques
  Jiawei Han, Micheline Kamber
- Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations
  Ian Witten, Eibe Frank
- Introduction to Data Mining

  Pang Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar, ISBN 9780133128901, 2018
- https://scikit-learn.org/0.15/documentation.html