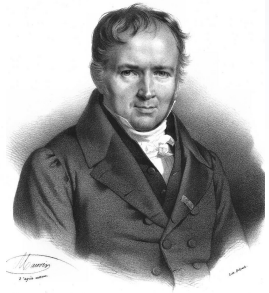
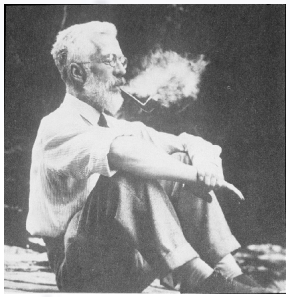
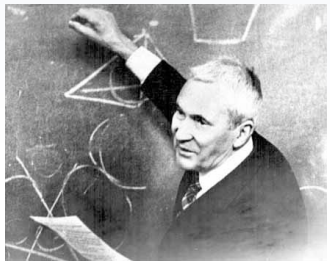
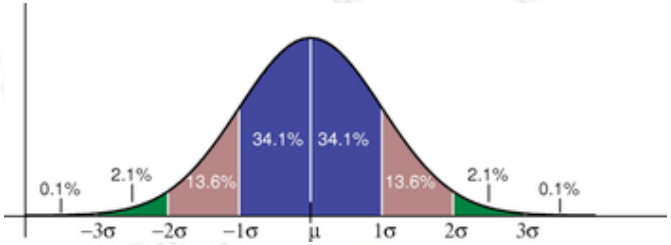
Applied ML
Clustering Algorithms



Troels C. Petersen (NBI)

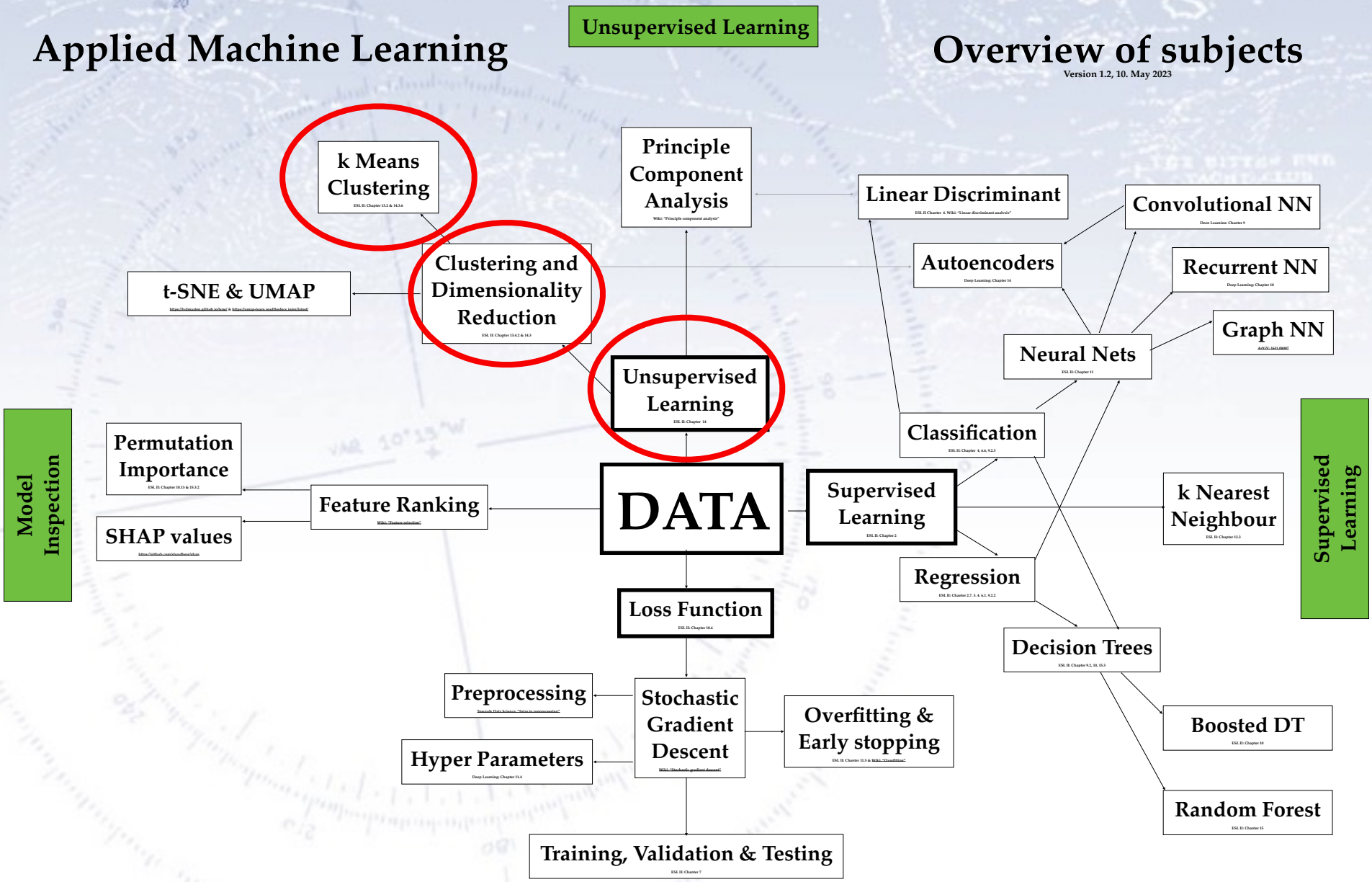


"Statistics is merely a quantisation of common sense - Machine Learning is a sharpening of it!"

Applied Machine Learning

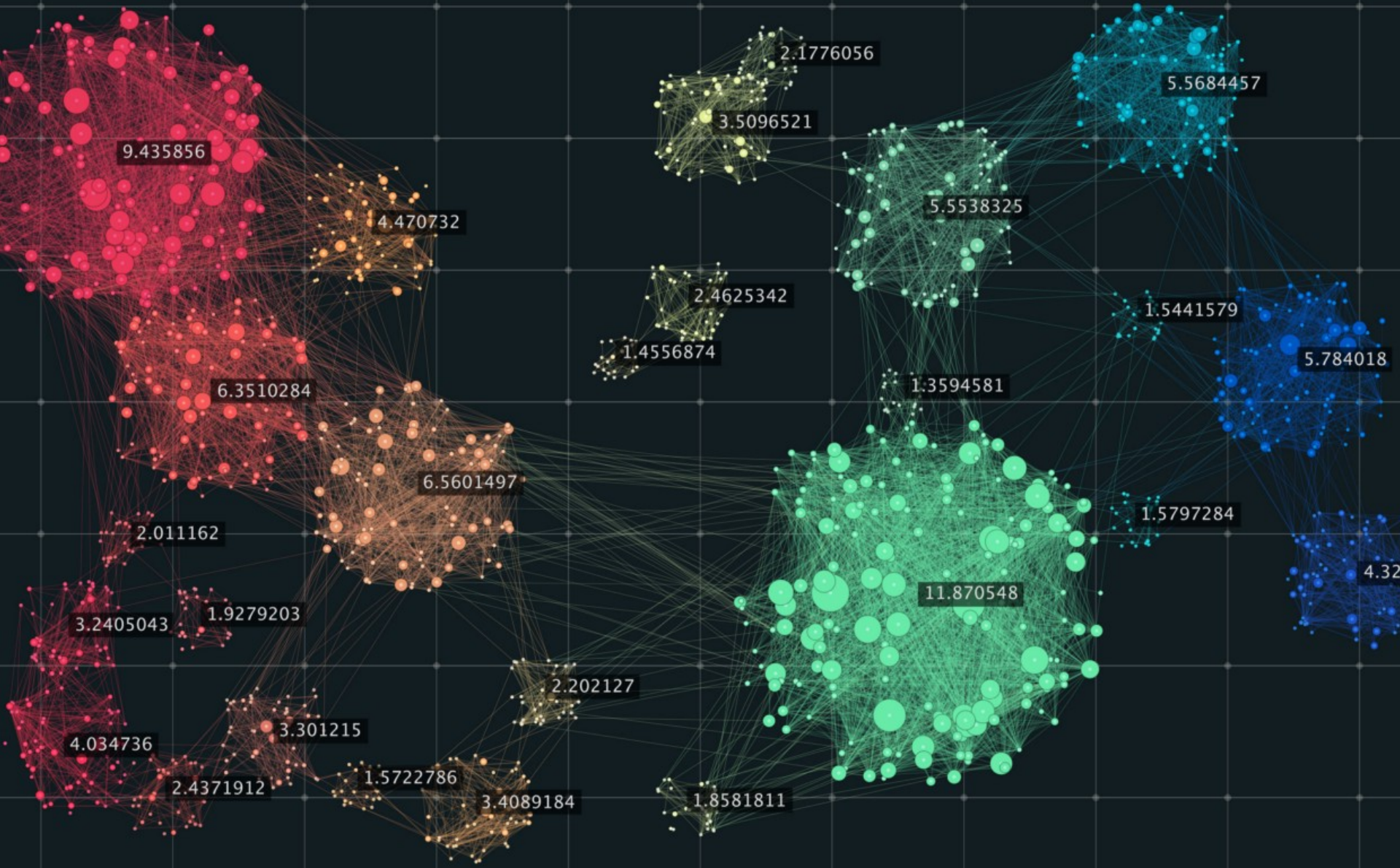
Overview of subjects

Version 1.2, 10. May 2023

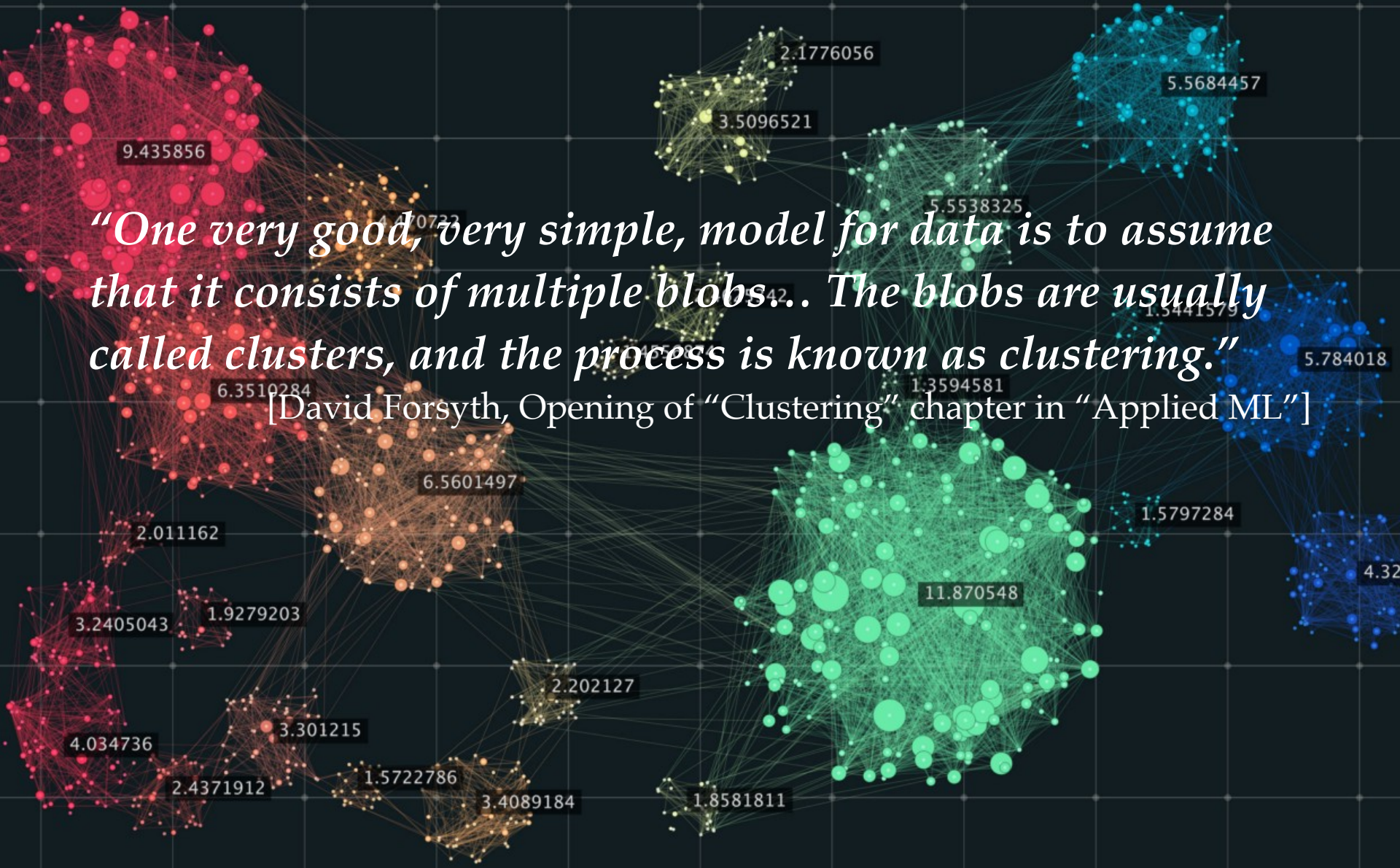


References:
Trevor Hastie et al.: "Elements of Statistics Learning II" (ELS II)
Ian Goodfellow et al.: "Deep Learning"
Wiki: Good reference for most subjects (only specified when essential)
Various blogs/githubs/papers for specific subjects.

Clustering… is an art!



Clustering… is an art!



"One very good, very simple, model for data is to assume that it consists of multiple blobs… The blobs are usually called clusters, and the process is known as clustering."

[David Forsyth, Opening of "Clustering" chapter in "Applied ML"]

The concept of clustering

Clustering is one of the two main "unsupervised" ML methods (the other being dimensionality reduction). It is to some extend related to classification much like dimensionality reduction is related to regression.

> *"Clusters presumably reflect some mechanism at work in the domain from which instances are drawn, a mechanism that causes some instances to bear a stronger resemblance to each other than they do to the remaining instances."*
>
> [Pages 141-142, Data Mining: Practical Machine Learning Tools and Techniques, 2016.]

The concept of clustering

Clustering is one of the two main "unsupervised" ML methods (the other being dimensionality reduction). It is to some extend related to classification much like dimensionality reduction is related to regression.

> "*Clusters presumably reflect some mechanism at work in the domain from which instances are drawn, a mechanism that causes some instances to bear a stronger resemblance to each other than they do to the remaining instances.*"
>
> [Pages 141-142, Data Mining: Practical Machine Learning Tools and Techniques, 2016.]

Clustering can be helpful in order to learn more about the data structure and problem domain, and requires no/little input to begin with.

A former student used to say:

"When I get new data, **I always run a clustering and a dimension reduction algorithm** on a random sample from it - that only costs computing time, and might reveal things, that it would take me a long time to discover otherwise."

Notice that "dimensionality reduction" (e.g. PCA) does not cluster data points, but possibly makes it easier to see patterns visually.

Evaluating clustering

Evaluation of identified clusters is subjective and may require a domain expert, although many clustering-specific quantitative measures do exist.

Typically, clustering algorithms are compared on synthetic datasets with pre-defined clusters, which an algorithm is expected to discover.

> "Clustering is an unsupervised learning technique, so it is hard to evaluate the quality of the output of any given method."
>
> [Page 534, Machine Learning: A Probabilistic Perspective, 2012.]

Evaluating clustering

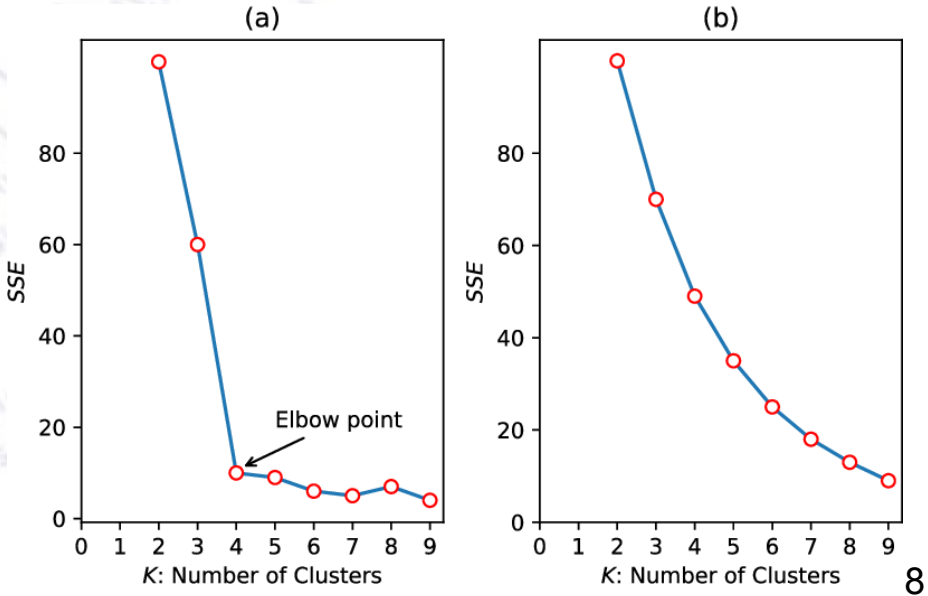
Evaluation of identified clusters is subjective and may require a domain expert, although many clustering-specific quantitative measures do exist.

Typically, clustering algorithms are compared on synthetic datasets with pre-defined clusters, which an algorithm is expected to discover.

> "Clustering is an unsupervised learning technique, so it is hard to evaluate the quality of the output of any given method."
>
> [Page 534, Machine Learning: A Probabilistic Perspective, 2012.]

One of the simple principles is that of the "Elbow Method".

If the loss function shows an "elbow" (sudden stop in rate of improvement), then that probably reflects some structure in the data.



Evaluating clustering

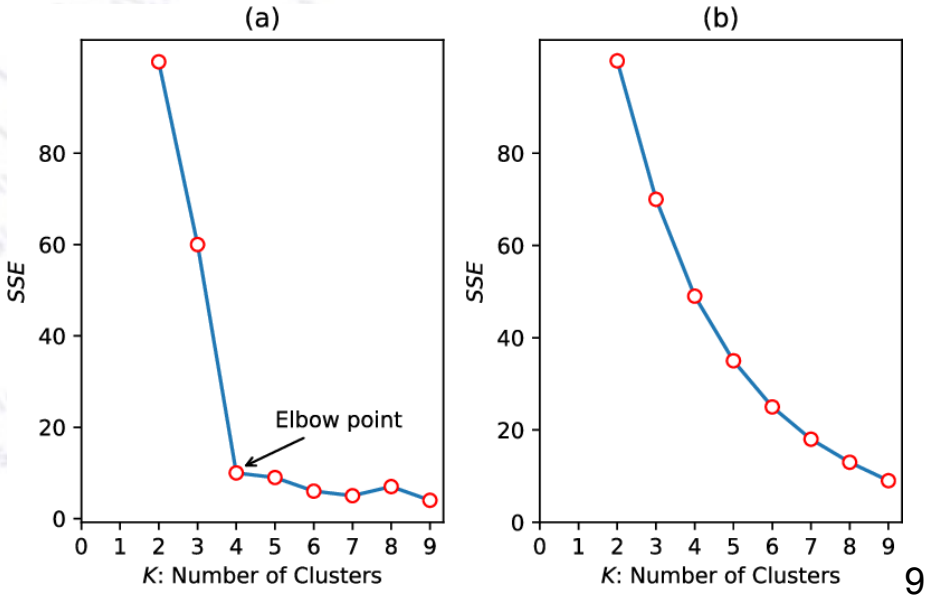
Evaluation of identified clusters is subjective and may require a domain expert, although many clustering-specific quantitative measures do exist.

Typically, clustering algorithms are compared on synthetic datasets with pre-defined clusters, which an algorithm is expected to discover.

One way of visually evaluating a clustering algorithm
is to combine it with a dimensionality reduction,
though one then observes the combined performance of the two.

One of the simple principles is that of the "Elbow Method".

If the loss function shows an "elbow" (sudden stop in rate of improvement), then that probably reflects some structure in the data.

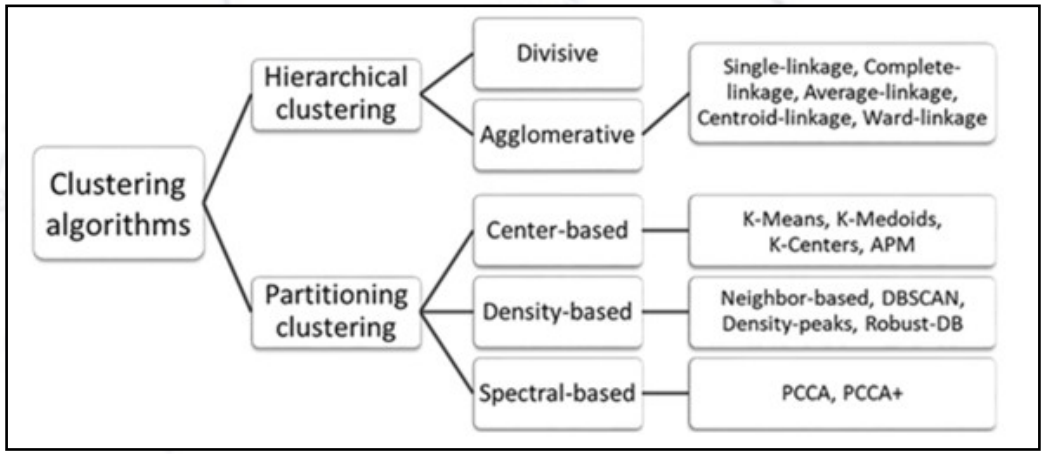


Clustering algorithms

Clustering is an "old field" and many philosophies (and algorithms) have been developed. They can roughly be reduced to two approaches:

- **Hierarchical clustering** algorithms are based on recursively either merging smaller clusters in to larger ones or dividing larger clusters to smaller ones.
- **Partitioning clustering** algorithms generate various partitions and then iteratively place each instance best in one of k mutually exclusive clusters.

Hierarchical clustering does not require any input parameters, while partitioning clustering algorithms require the number of clusters to start running. Hierarchical clustering returns a much more meaningful and subjective division of clusters but partitioning clustering results in exactly k clusters.

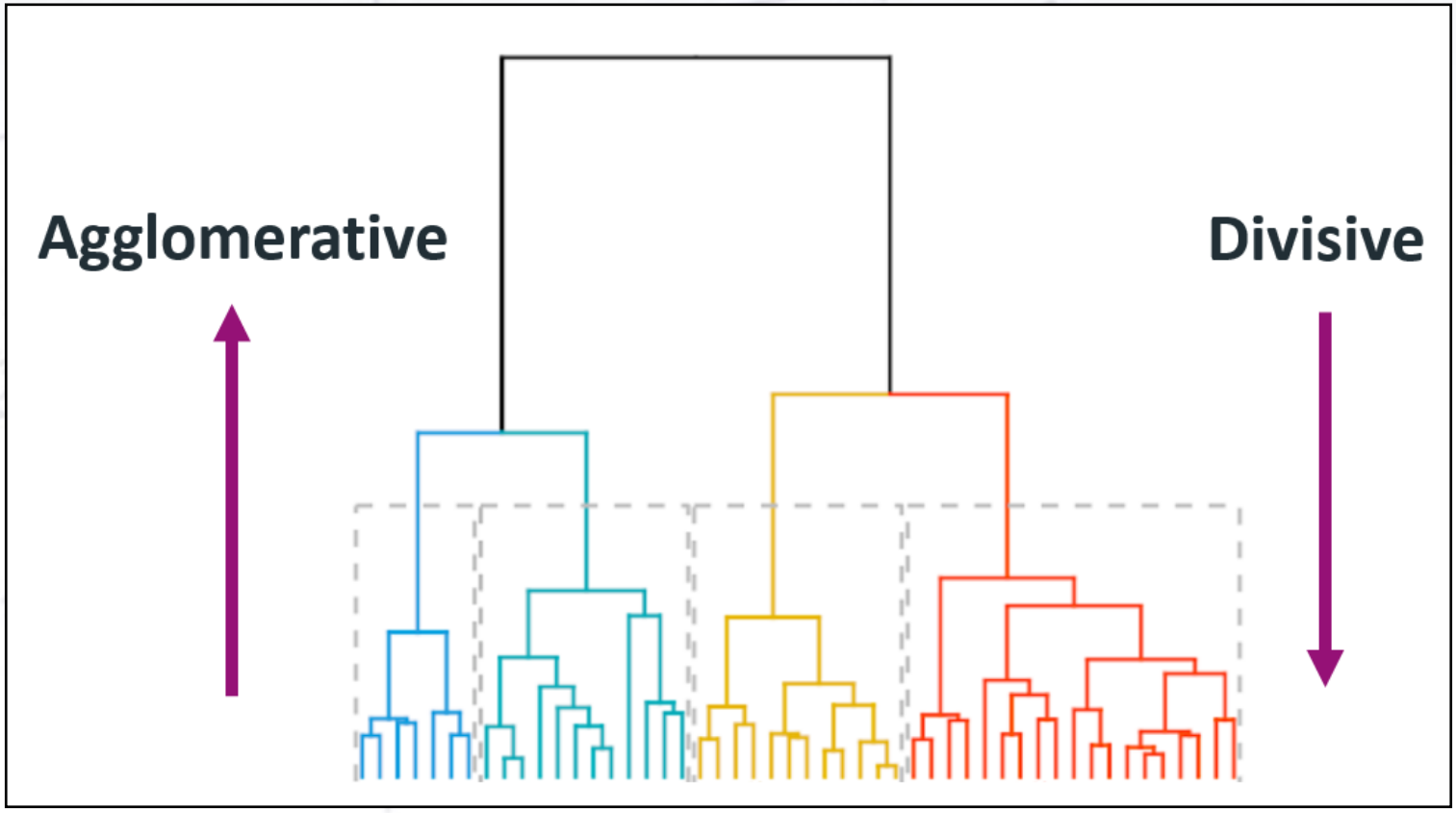


Hierarchical clustering algorithms

Hierarchical clustering algorithms can be further divided:

- **Agglomerative**: Merge smaller clusters into larger ones
- **Divisive**: Divide larger clusters into smaller ones.

The only requirement is a **similarity measure** to decide distance between cases.



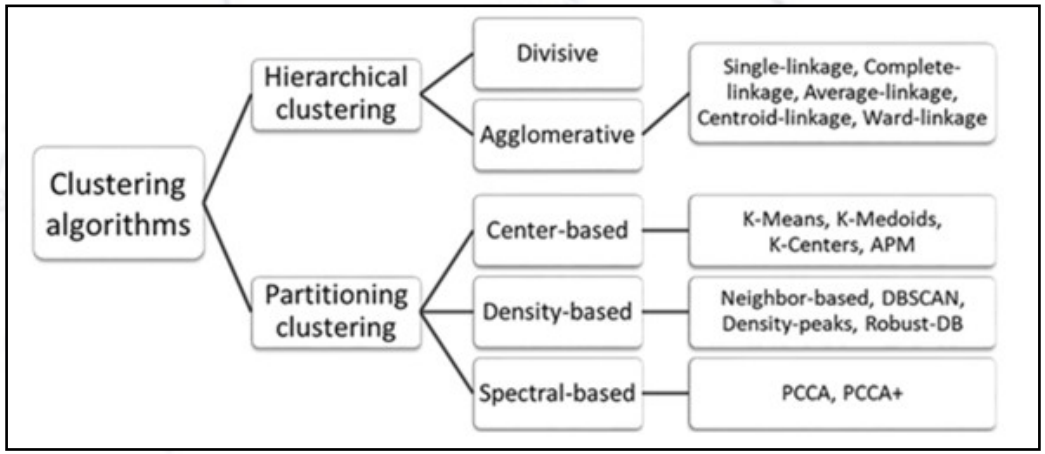
Partitioning clustering algorithms

Partitioning clustering algorithms can (also) be further divided:

- **Center-based**: Build clusters around (random?) centers (**k-Means**).
- **Density-based**: Build clusters around (high) densities (**DBSCAN**).
- **Spectral-based**: Uses eigenvalues of the similarity matrix to perform dimensionality reduction before clustering (**PCCA+**).

"k-Means clustering is the "go-to" clustering algorithm. You should see it as a basic recipe from which many algorithms can be concocted."

[David Forsyth, "Applied ML" chapter 8.2.6]

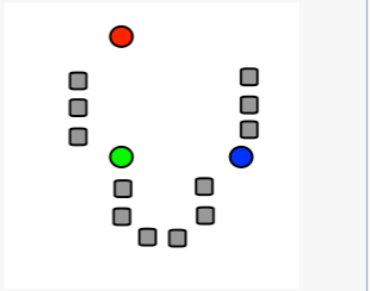


k-Means clustering

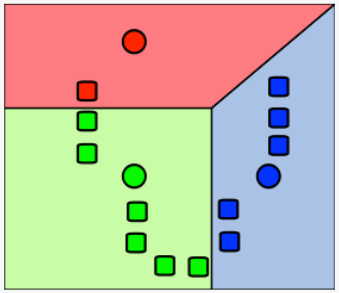
The recipe is to iterate the below points, until movements are “small”:

- Allocate each data point to the closest cluster center
- Re-estimate cluster centers from their data points.

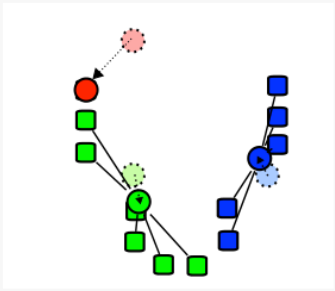
Demonstration of the standard algorithm



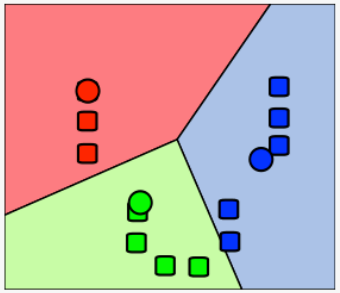
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.



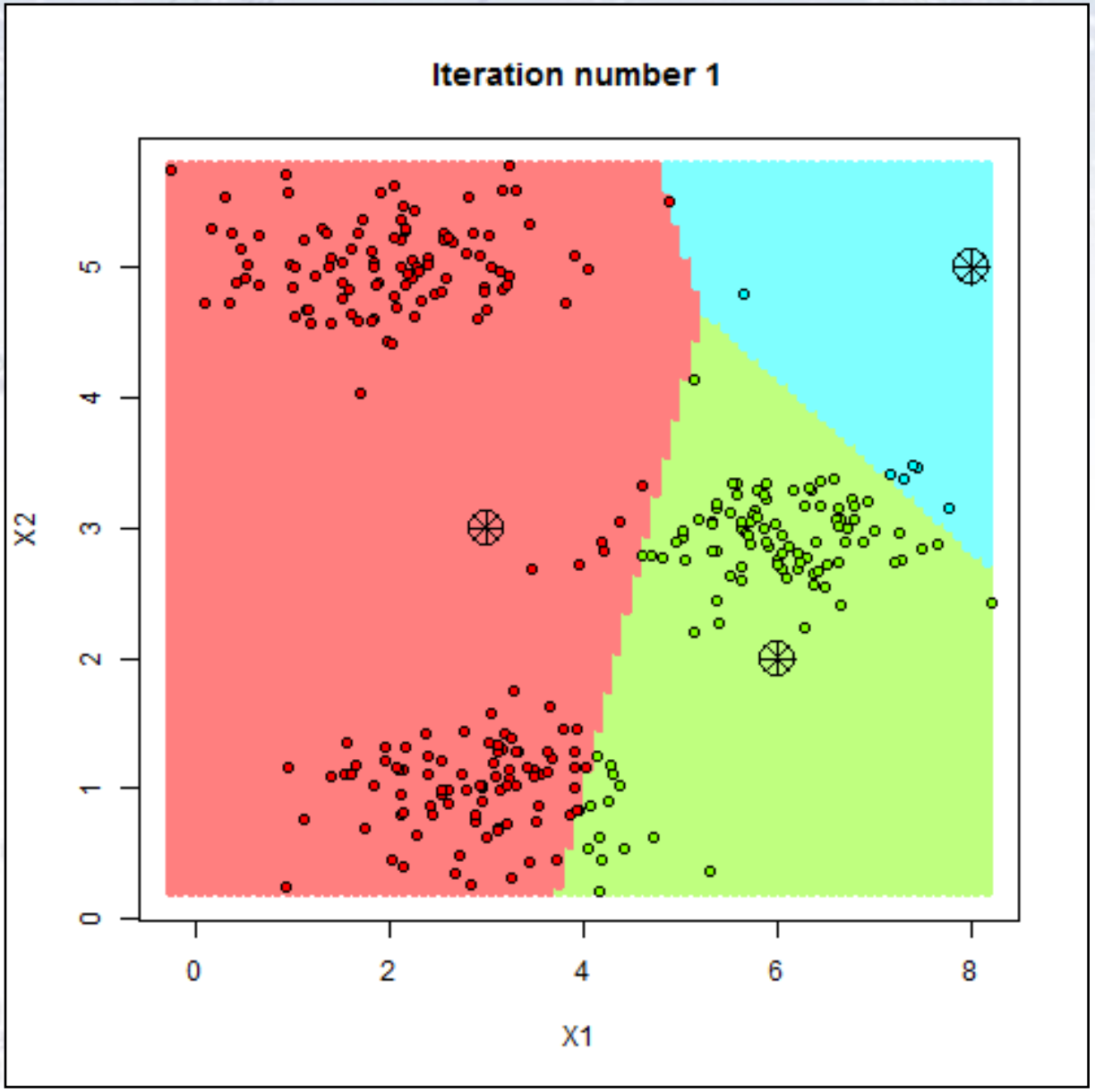
4. Steps 2 and 3 are repeated until convergence has been reached.

There are many variations, improvements, etc. that refines this algorithm.
Most notably are the k-means++ (better initial points) and k-mediods methods.

k-Means clustering

The recipe:

- Allocate each data point to the closest cluster center
- Re-estimate cluster centers from their data points.



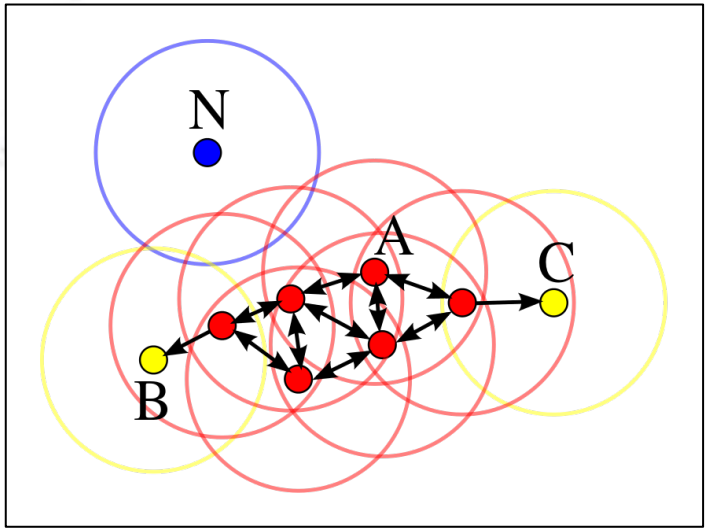
DBSCAN algorithm

DBSCAN classifies points as core points, reachable points and outliers:

- A point p is a **core point** if at least minPts points are within distance ε of it.
- A point q is **directly reachable** from p if point q is within distance ε from core point p. Points are only said to be directly reachable from core points.
- A point q is **reachable** from p if there is a path p_1, ..., p_n with $p_1 = p$ and $p_n = q$, where each p_{i+1} is directly reachable from p_i. Note that this implies that the initial point and all points on the path must be core points, with the possible exception of q.
- All points not reachable from any other point are outliers or noise points.

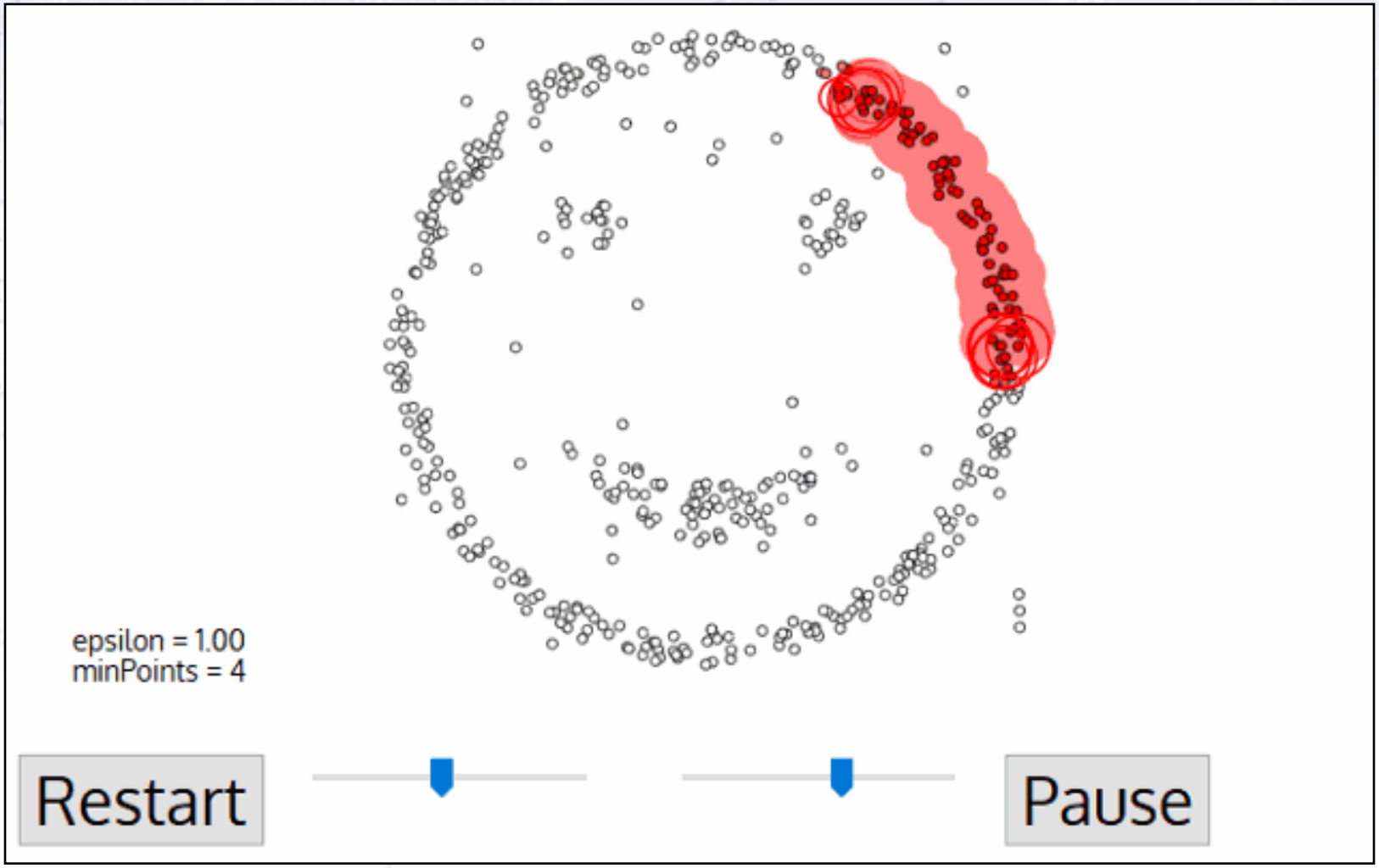
DBSCAN has two parameters: minPts and ε.

If p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it. Each cluster contains at least one core point; non-core points can be part of a cluster, but they form its “edge", since they cannot be used to reach more points.



DBSCAN algorithm

As can be seen, DBSCAN is a rather generic algorithm, capable of handling a large variety of data.



Expectation-Maximisation algorithm

The Expectation–Maximisation (EM) algorithm is an iterative method to find maximum likelihood estimates of parameters, where the model depends on unobserved latent variables.

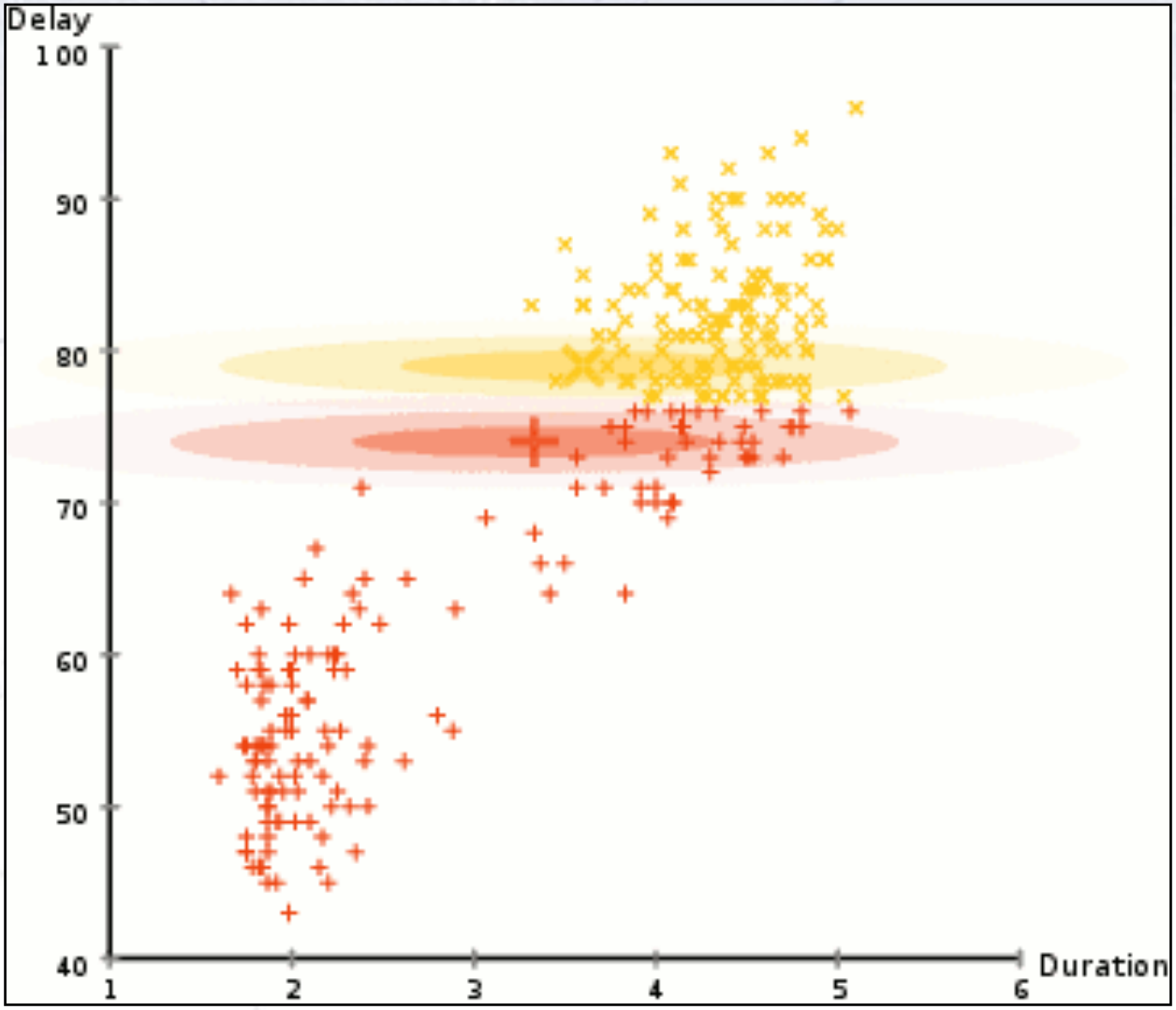
Based on the notion that data falls in "blobs", the model used in clustering is typically a Gaussian Mixture Model.

The EM algorithm alternates between performing an expectation (E) step, calculating the expected likelihood given current parameters, and a maximisation (M) step, computing new parameters maximising the expected likelihood found on the E step:

1. First, initialise the parameters θ to some random values.
2. Compute the probability of each possible likelihood value, given θ.
3. Use the just-computed likelihood values to compute a better estimate for the parameters θ. Iterate steps 2 and 3 until convergence.

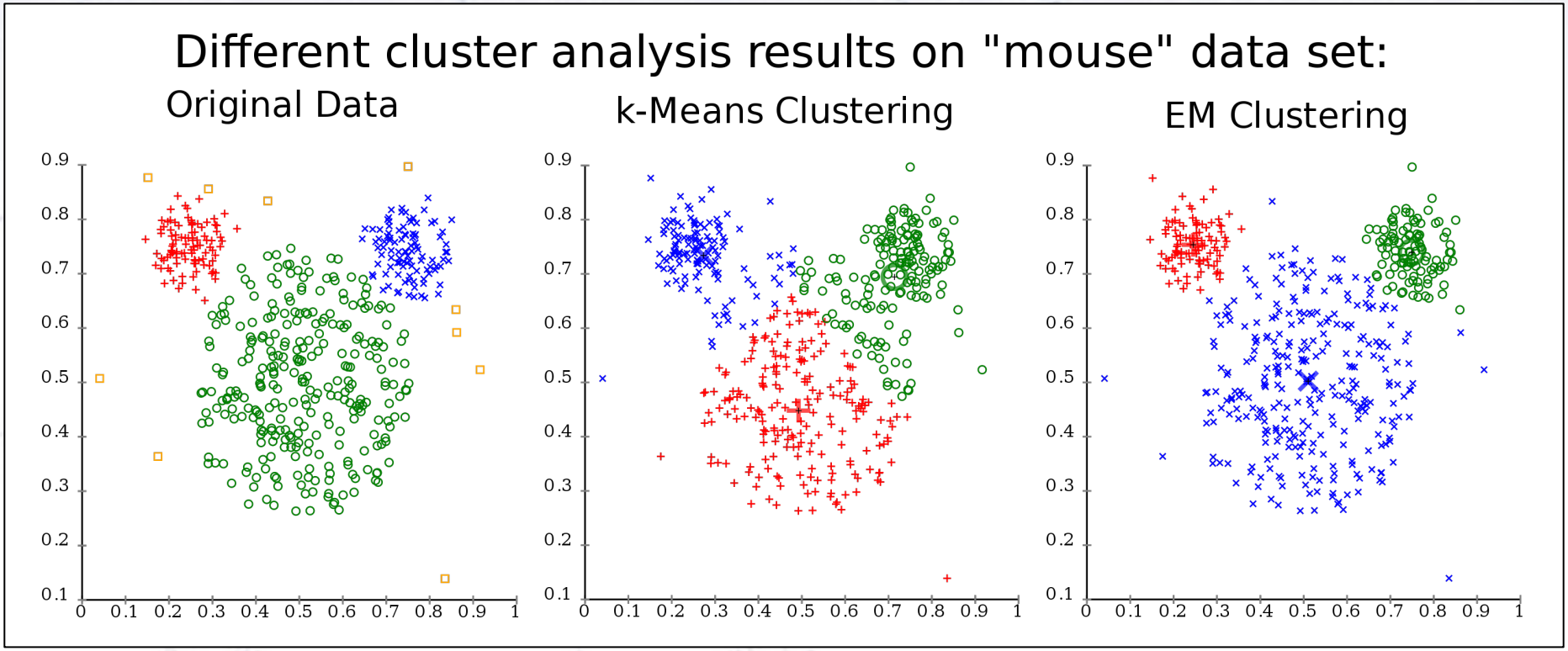
Expectation Maximisation algorithm

An example is shown below, applied to the eruption pattern of “Old Faithful”.



The "Mickey Mouse" test

The recipe is: iterate: allocate each data point to the closest cluster center; re-estimate cluster centers from their data points.

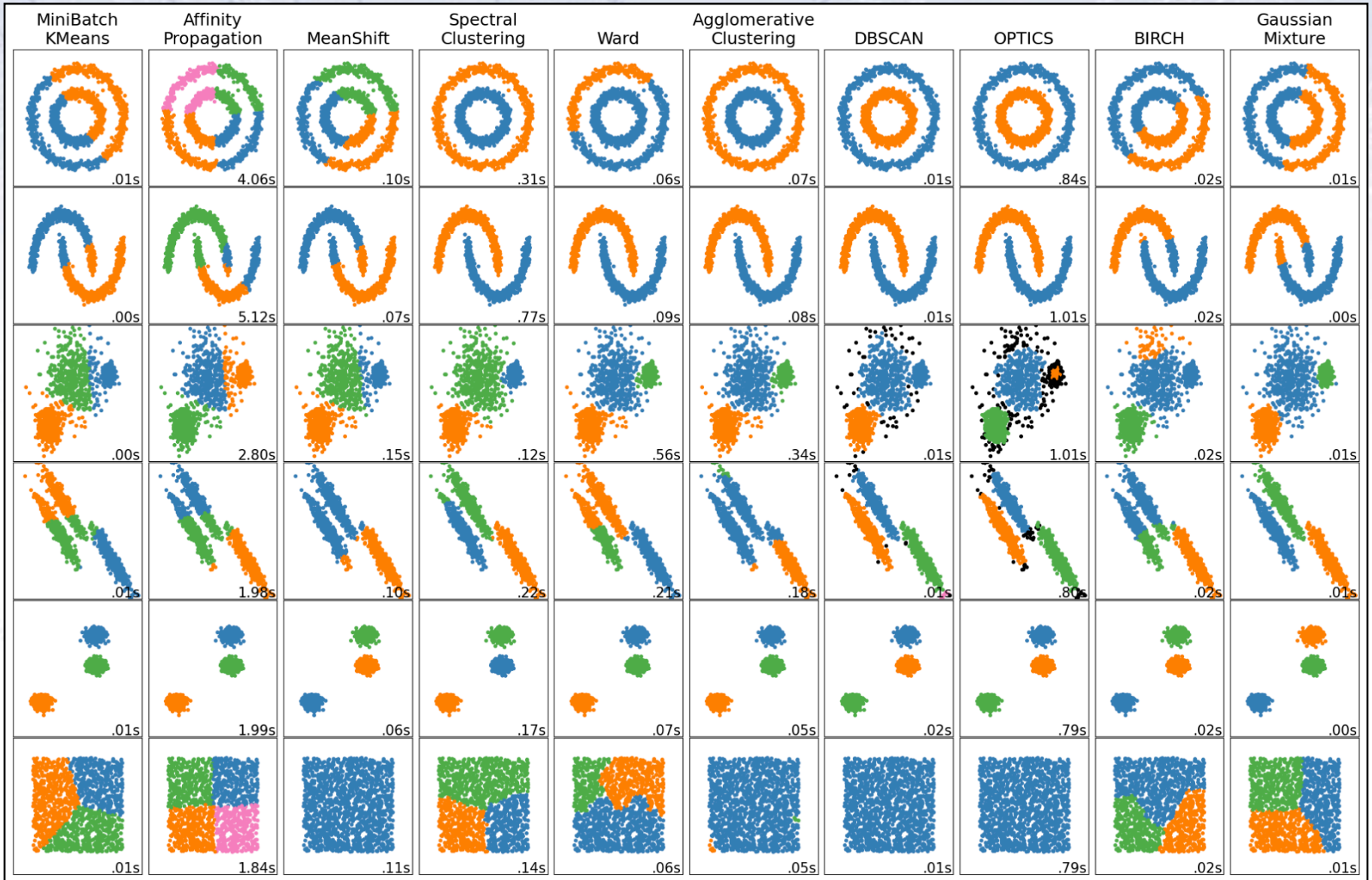


Clustering algorithms in scikit-learn

Scikit-Learn has a rather good selection of clustering algorithms:

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
OPTICS	minimum cluster membership	Very large `n_samples`, large `n_clusters`	Non-flat geometry, uneven cluster sizes, variable cluster density, outlier removal, transductive	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points

Clustering algorithms in scikit-learn



A comparison of the clustering algorithms in scikit-learn

Conclusions

Clustering is an “old” art form, for which there is a vast ocean of methods.

The K-means (and further developments) is the standard algorithm, if there is one such. DBSCAN is also an old (and awarded!) classic.

Note that like in dimensionality reduction, it is important to transform the input variables first, so that mean and variances are of order zero and unity.

It is HARD to evaluate the performance, and visual inspection and testing on similar (typically simulated) cases are some of few methods.

Bonus Slides

k-Means clustering