Cluster Analysis

Classification

Supervised learning = machine learning task of inferring a function from labeled training data.

Training data = a set of training examples.
Each example is a pair of

- an input object (typically a vector)
- a desired output value (also called the supervisory signal).

A supervised learning algorithm

- analyzes the training data
- produces an inferred function, which can be used for mapping new examples.

An optimal scenario:

- allows for the algorithm to correctly determine the class labels for unseen instances;
- requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way (see inductive bias).

Unsupervised learning = machine learning task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations).

The examples given to the learner are unlabelled.

So there is no evaluation of the accuracy of the structure that is output by the relevant algorithm.

This is one way of distinguishing unsupervised learning from supervised learning and reinforcement learning.

Soundness and completeness of algorithms

Definition:

An algorithm is **sound** if every result that is returned by it is indeed a solution. ■

Definition:

An algorithm is **complete** if every solution can be found by it. ■

Accuracy vs Precision vs Recall

Accuracy refers to the closeness of a measured value to a standard or known value.

For example, if in lab you obtain a weight measurement of 3.2 kg for a given substance, but the actual or known weight is 10 kg, then your measurement is not accurate. In this case, your measurement is not close to the known value.

Precision refers to the closeness of two or more measurements to each other.

Using the example above, if you weigh a given substance five times, and get 3.2 kg each time, then your measurement is very precise. Precision is independent of accuracy. You can be very precise but inaccurate, as described above. You can also be accurate but imprecise.

For example, if on average, your measurements for a given substance are close to the known value, but the measurements are far from each other, then you have accuracy without precision.

Another example: imagine a basketball player shooting baskets.

If the player shoots with accuracy, his aim will always take the ball close to or into the basket.

If the player shoots with precision, his aim will always take the ball to the same location which may or may not be close to the basket.

A good player will be both accurate and precise by shooting the ball the same way each time and each time making it in the basket.

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{True negative rate} = \frac{tn}{tn + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn} \qquad \text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

condition positive (P)	the number of real positive cases in the data
condition negatives (N)	the number of real negative cases in the data

true positive (TP)	case positive and predicted positive
true negative (TN)	case negative and predicted negative
false positive (FP)	case negative and predicted positive
false negative (FN)	case positive and predicted negative

What percent of predictions were correct?

The "accuracy"

What percent of positive cases did you catch?

The "recall"

What percent of positive predictions were correct?

The "precision"

Clustering vs classification

Classification = supervised learning, i.e. using labelled data

Clustering = unsupervised learning, i.e. using unlabelled data

Cluster analysis = Clustering

The task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).

Main task of exploratory data mining; common technique for statistical data analysis.

Used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved.

Various algorithms, that differ significantly in what constitutes a cluster and how to efficiently find them.

Popular notions of clusters include

- groups with small distances among the cluster members,
- dense areas of the data space,
- intervals of particular statistical distributions,
- geometrical configurations,
- groups set up on the shell, or contour of structures.

Clustering can be formulated as a multi-objective optimization problem.

The appropriate clustering algorithm and parameter settings (the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.

Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure.

It is often necessary to modify data pre-processing and model parameters until the result achieves the desired properties.

The notion of a "cluster" cannot be precisely defined, which is one of the reasons why there are so many clustering algorithms.

There is a common denominator: a group of data objects.

However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given.

The notion of a cluster, as found by different algorithms, varies significantly in its properties.

Understanding these "cluster models" is key to understanding the differences between the various algorithms.

A "clustering" (or cluster structure or substructure) is essentially a set of such clusters, usually containing all objects in the data set.

Additionally, it may specify the relationship of the clusters to each other, for example, a hierarchy of clusters embedded in each other.

Clustering methods can be roughly distinguished as:

- **Hard (crisp) clustering**: each object belongs to a cluster or not

- **Soft (fuzzy) clustering**: each object belongs to each cluster to a certain degree (for example, a likelihood of belonging to the cluster)

Finer distinctions possible:

- **Strict partitioning clustering**: each object belongs to exactly one cluster
- **Strict partitioning clustering with outliers**: objects can also belong to no cluster, and are considered outliers
- **Overlapping clustering (alternative clustering, multi-view clustering)**: objects may belong to more than one cluster; usually involving hard clusters
- **Hierarchical clustering (agglomerative vs. divisive)**: objects that belong to a child cluster also belong to the parent cluster
- **Subspace clustering**: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap

Typical cluster models include:

- **Connectivity models**: for example, hierarchical clustering builds models based on distance connectivity.

- **Centroid models**: for example, the k-means algorithm represents each cluster by a single mean vector.

- **Distribution models**: clusters are modeled using statistical distributions, such as multivariate normal distributions used by the expectation-maximization algorithm.

- **Density models**: for example, DBSCAN and OPTICS defines clusters as connected dense regions in the data space.

- **Subspace models**: in biclustering (also known as co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.

- **Group models**: some algorithms do not provide a refined model for their results and just provide the grouping information.

- **Graph-based models**: a clique (a subset of nodes in a graph such that every two nodes in the subset are connected by an edge) is a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques, as in the HCS clustering algorithm.

- **Neural models**: the most well known is the self-organizing map; can be characterized as similar to one or more of the above models, and including subspace models when neural networks implement a form of Principal Component Analysis or Independent Component Analysis.

There is no objectively "correct" clustering algorithm.

"Clustering is in the eye of the beholder."

The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally.

Possibly there is a mathematical reason to prefer one cluster model over another.

An algorithm designed for one kind of model will generally fail on a data set that contains a radically different kind of model.

For example, k-means cannot find non-convex clusters.

Connectivity-based clustering (hierarchical clustering)

Connectivity based clustering, also known as hierarchical clustering, is based on the core idea of objects being more related to nearby objects than to objects farther away.

These algorithms connect "objects" to form "clusters" based on their distance.

A cluster can be described largely by the maximum distance needed to connect parts of the cluster.

At different distances, different clusters will form, which can be represented using a dendrogram, which explains where the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances.

In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to) to use.

Popular choices are known as

- single-linkage clustering (the minimum of object distances),
- complete linkage clustering (the maximum of object distances) or
- UPGMA ("Unweighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering).

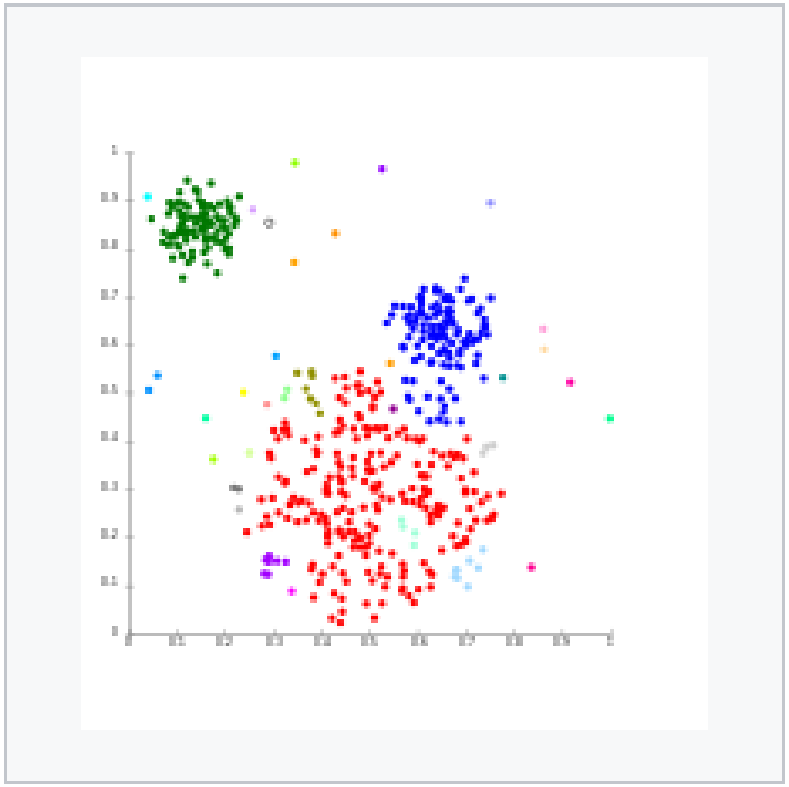
Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

These methods will not produce a unique partitioning of the data set, but a hierarchy from which the user still needs to choose appropriate clusters.

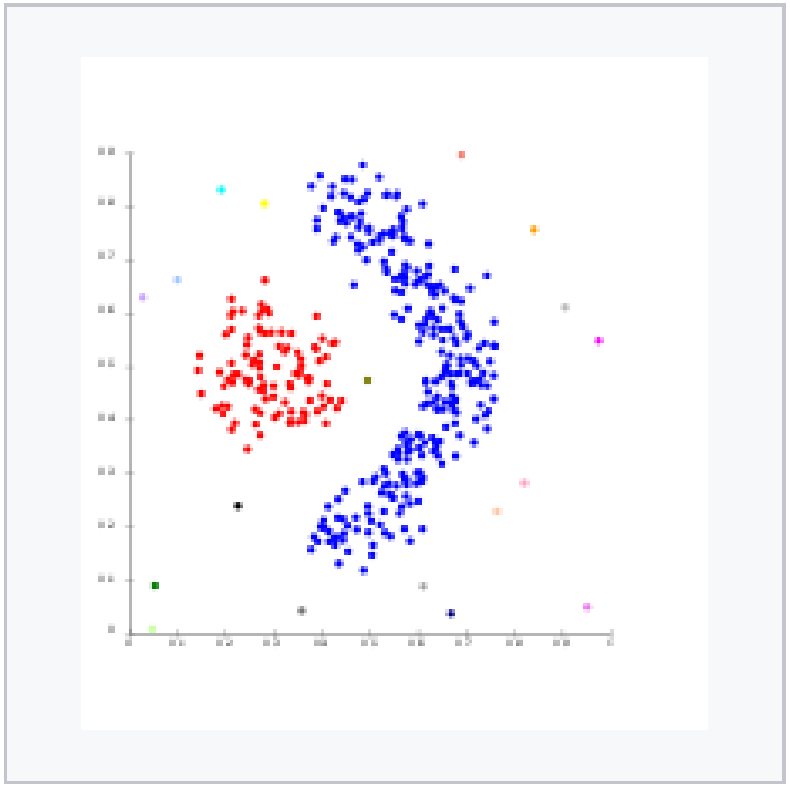
They are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering).

In the general case, the complexity is $O(n^3)$ for agglomerative clustering and $O(2^{n-1})$ for divisive clustering, which makes them too slow for large data sets.

In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering.



Single-linkage on Gaussian data. At 35 clusters, the biggest cluster starts fragmenting into smaller parts, while before it was still connected to the second largest due to the single-link effect.



Single-linkage on density-based clusters. 20 clusters extracted, most of which contain single elements, since linkage clustering does not have a notion of "noise".

Centroid-based clustering

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximative method is Lloyd's algorithm ("k-means algorithm"). It finds a local optimum, and is commonly run multiple times with different random initializations.

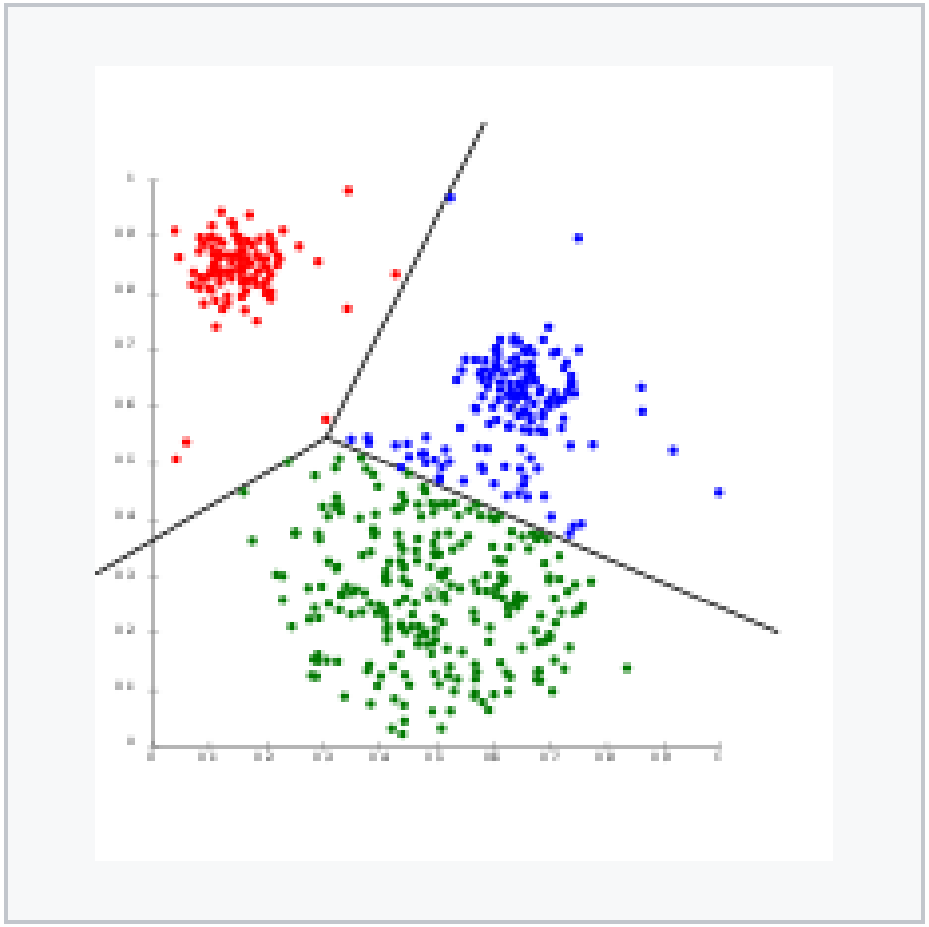
Variations of k-means often include such optimizations as

- choosing the best of multiple runs,
- restricting the centroids to members of the data set (k-medoids),
- choosing medians (k-medians clustering),
- choosing the initial centers less randomly (k-means++)
- allowing a fuzzy cluster assignment (fuzzy c-means).

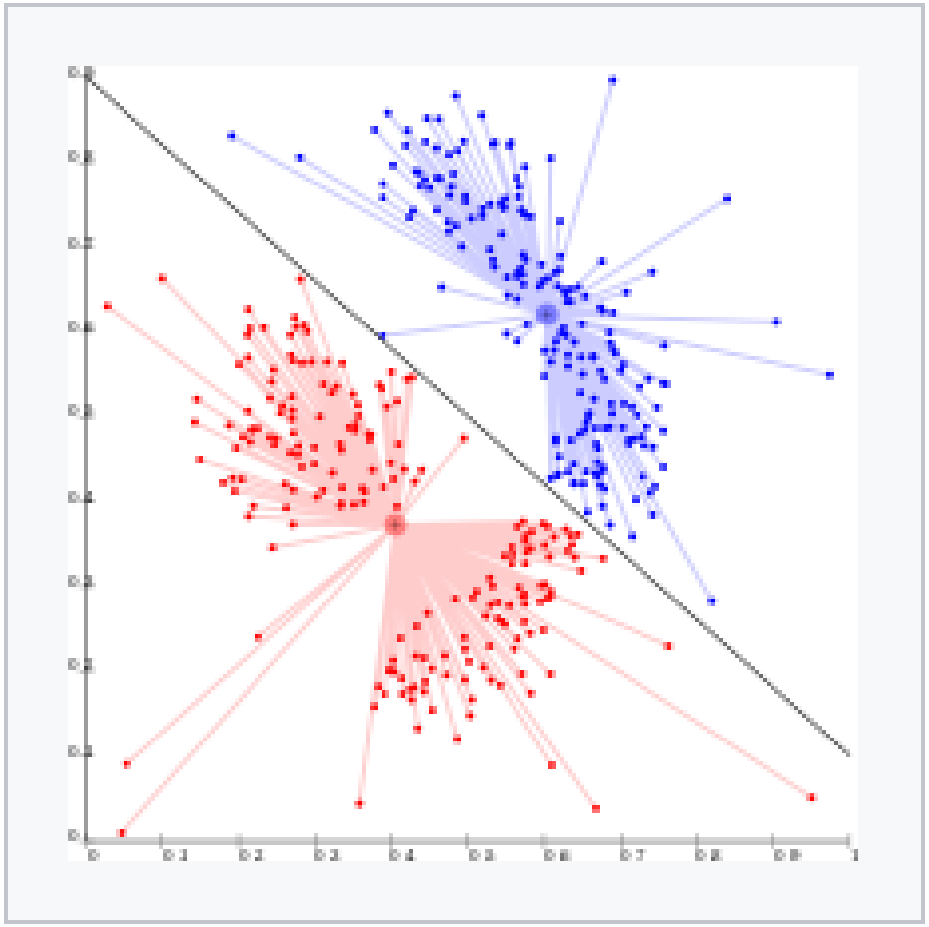
Most k-means-type algorithms require the number of clusters - k - to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders of clusters (which is not surprising since the algorithm optimizes cluster centers, not cluster borders).

K-means has a number of interesting theoretical properties.

- First, it partitions the data space into a structure known as a Voronoi diagram.
- Second, it is conceptually close to nearest neighbor classification, and as such is popular in machine learning.
- Third, it can be seen as a variation of model based clustering, and Lloyd's algorithm as a variation of the Expectation-maximization algorithm for this model.



K-means separates data into Voronoi-cells, which assumes equal-sized clusters (not adequate here)



K-means cannot represent density-based clusters

Distribution-based clustering

The clustering model most closely related to statistics is based on distribution models. Clusters can easily be defined as objects belonging most likely to the same distribution. This approach closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

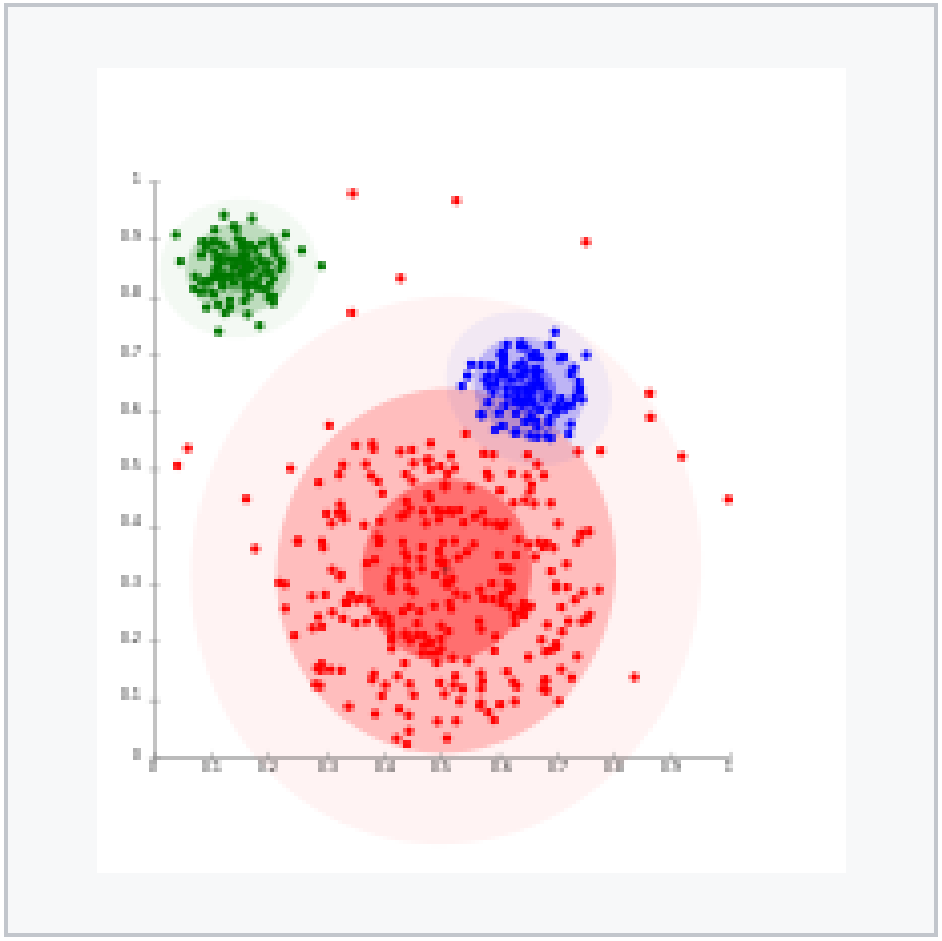
The theoretical foundation of these methods is excellent.

They suffer from overfitting, unless constraints are put on the model complexity: A more complex model will usually be able to explain the data better, which makes choosing the appropriate model complexity inherently difficult.

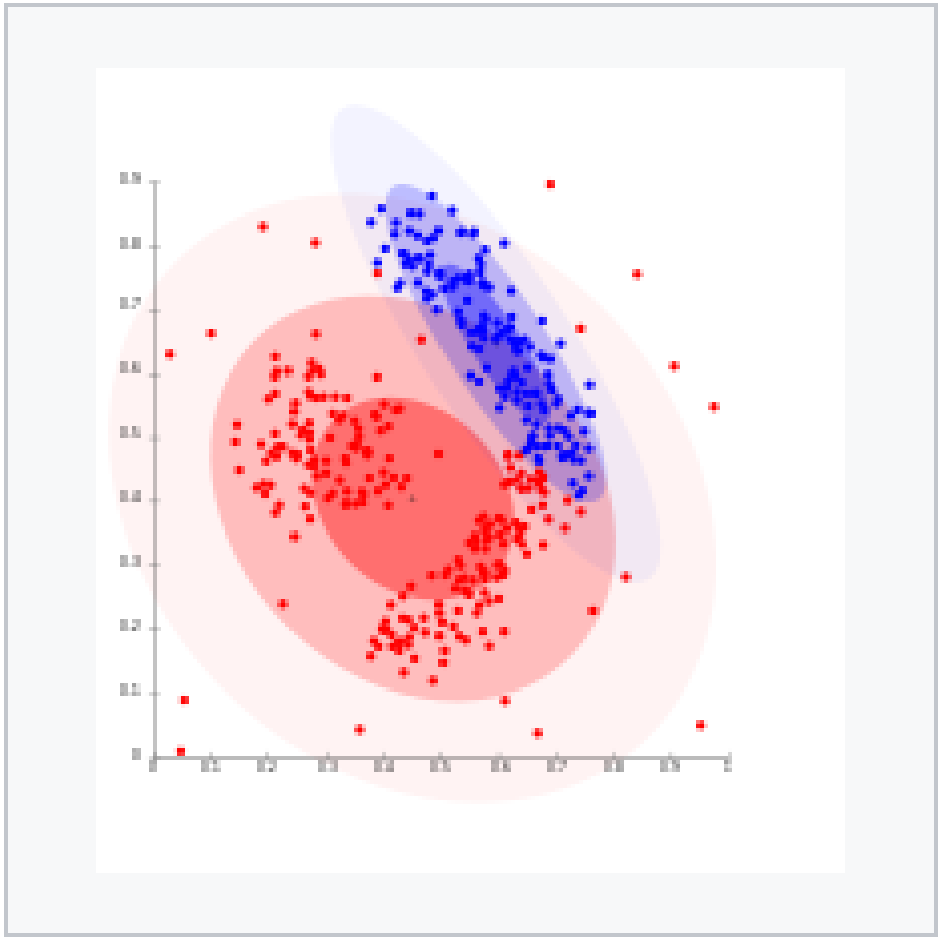
Distribution-based clustering produces complex models for clusters that can capture correlation and dependence between attributes. But for many real data sets, there may be no concisely defined mathematical model.

One prominent method is Gaussian mixture models.

- The data set is usually modelled with a fixed (to avoid overfitting) number of Gaussian (normal) distributions that are initialized randomly and whose parameters are iteratively optimized to better fit the data set.
- This will converge to a local optimum, so multiple runs may produce different results.
- In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to.
- For soft clustering, this is not necessary.



On Gaussian-distributed data, EM works well, since it uses Gaussians for modelling clusters



Density-based clusters cannot be modeled using Gaussian distributions

Density-based clustering

Clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

The most popular density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering, it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range.

Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times.

OPTICS is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ε, and produces a hierarchical result related to that of linkage clustering.

DeLi-Clu, Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter entirely and offering performance improvements over OPTICS by using an R-tree index.

The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders.

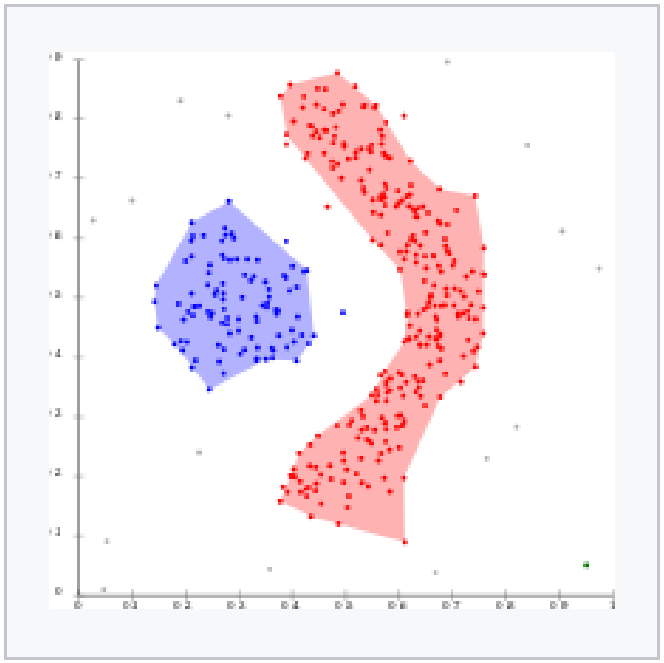
On data sets with overlapping Gaussian distributions - a common use case in artificial data - the cluster borders produced by these algorithms will often look arbitrary, because the cluster density decreases continuously.

On a data set consisting of mixtures of Gaussians, these algorithms are nearly always outperformed by methods such as EM clustering that are able to precisely model this kind of data.

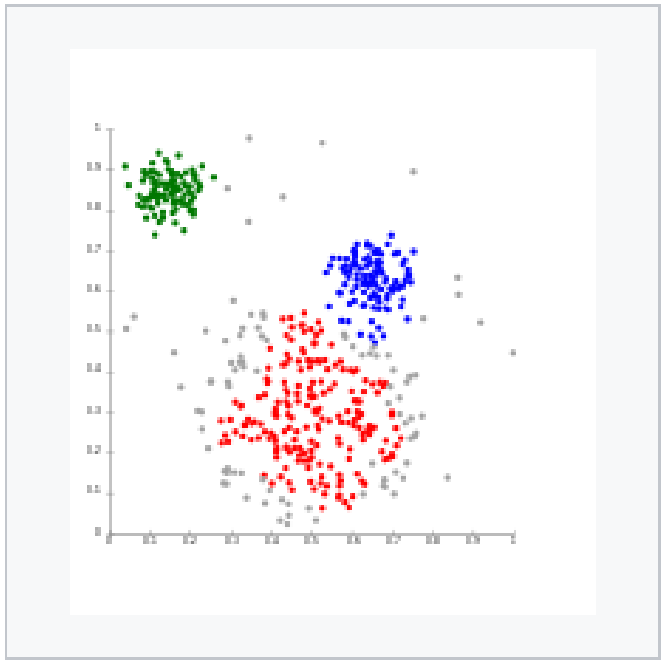
Mean-shift is a clustering approach where each object is moved to the densest area in its vicinity, based on kernel density estimation. Eventually, objects converge to local maxima of density.

Similar to k-means clustering, these "density attractors" can serve as representatives for the data set, but mean-shift can detect arbitrary-shaped clusters similar to DBSCAN.

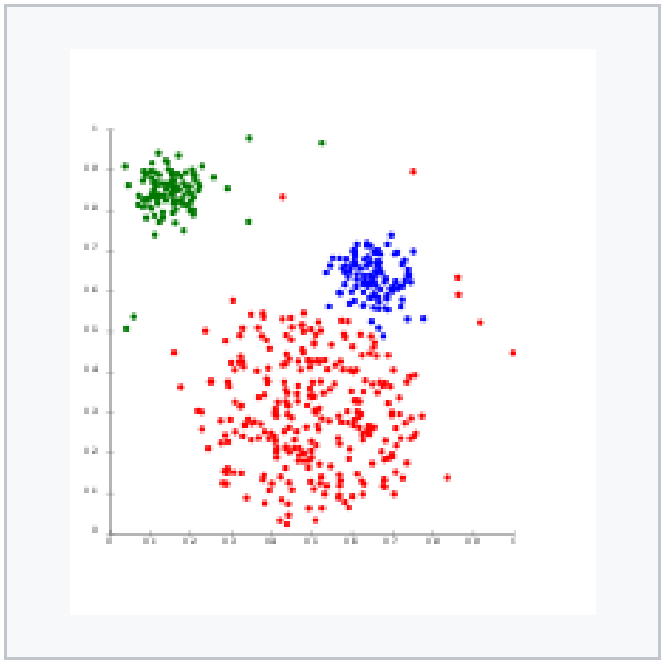
Due to the expensive iterative procedure and density estimation, mean-shift is usually slower than DBSCAN or k-Means. Besides that, the applicability of the mean-shift algorithm to multidimensional data is hindered by the unsmooth behaviour of the kernel density estimate, which results in over-fragmentation of cluster tails.



Density-based clustering with DBSCAN.



DBSCAN assumes clusters of similar density, and may have problems separating nearby clusters



OPTICS is a DBSCAN variant that handles different densities much better