Perception for Autonomous Systems

Lecture 4 - 3D Point Cloud Processing (Clustering & Regression) (01/03/2021)

Outline/Content:

- A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?
- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

> Reading Material:
>
> - Book A, Sections 6.1 and 6.2
> - [Paper 1 - ICP] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992.
> - [Paper 2 - Spin Images] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 433-449, May 1999.
> - [Paper 3 - PFH] R. B. Rusu, N. Blodow and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," 2009 IEEE International Conference on Robotics and Automation, Kobe, 2009, pp. 3212-3217.
> - [Paper 4 - FPFH] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning Point Cloud Views using Persistent Feature Histograms,” in Proceedings of the 21st IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, September 22-26, 2008.

A taxonomy of ML algorithms

Supervised Learning: The target values are known:

- Regression: The target value is numeric
- Classification: The target value is nominal

Unsupervised Learning: The target values are unknown

- Clustering: Group together similar instances

Reinforcement learning: Interacting with a dynamic environment the system must perform a certain goal.

Why is ML important for Perception of Autonomous Systems? Article

Machine learning can be employed as a replacement for traditional computer-vision algorithms, making it useful in autonomous vehicles for **object detection, classification, segmentation, tracking, and prediction.**

Doing this will impact the system's level of determinism, safety, and security.

**The Benefits of Using ML for Object Detection and Classification*

ML algorithms

- achieve greater degrees of accuracy than vision-based systems (when trained sufficiently)
- are more adaptable and scalable than vision systems
- can easily handle large volumes of data
- can evolve without human input
- reliance on determinist behavior

> For greater elaboration on those bullet points, see the linked article.

The Limitations of Machine Learning for Object Detection and Classification

ML algorithms

- rely heavily on the data quality on which the ML model is trained on. High data quality crucial for good performance in the real world. Otherwise, it could lead to undesirable and possibly dangerous outcomes.
- take long to train/validate and require huge amounts of computing processing resources.
- are limited when it comes to teaching a system how to respond to something that humans have innately. For instance, the "sixth sense" that a car may be about to pull in front of you, or a truck may suddenly slam on the brakes.

> For greater elaboration on those bullet points, see the linked article.

What are Clustering and Regression?

Regression

Regression tries to assign correct significance (weights) to input variables and formulate a weighted linear combination of them that can predict the output variables.

- The number of input and output variables can vary depending on the specific problem
- The final output of the algorithm is a regression line
- The goal is to find an equation f() that models the relationship between input x and output y such as

Regression Models:

- Linear Regression: Assume that function f is linear.
- Locally Weighted Regression: Creates multiple linear models on small neighbor data.

> Other types of Regression models exist, but are not relevant for the continuing part of the course Link:
>
> - Logistic Regression
> - Lasso Regression
> - Ridge Regression
> - Elastic Net Regression
> - Stepwise Regression

The performance for the linear regression method is evaluated by an error function. The most used one is the Mean Squared Error (MSE):

Equation:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(t-y)^2$$

where t is the true value of the learned output and y is the predicted value.

Linear Regression Medium

Linear Regression assumes that f(x) is a linear combination between the inputs x and a set of regression coefficients b:

Equation:

$$y = f(b,x) = b_1x_1 + b_2x_2 + b_3x_3 + \ldots + b_nx_n + c = b^T x + c$$

The goal of linear regression is to find regression coefficients b that minimize the squared error between the true values of the output and the predicted values.

How do we minimize the squared error?

This can be done by using the Gradient Descent algorithm. The main goal of Gradient descent is to minimize the cost value of a function.

Gradient descent has an analogy in which we have to imagine ourselves at the top of a mountain valley and left stranded and blindfolded, our objective is to reach the bottom of the hill. Feeling the slope of the terrain around you is what everyone would do. Well, this action is analogous to calculating the gradient descent, and taking a step is analogous to one iteration of the update to the parameters.

Polynomial Regression Medium Article

Linear regression only works with linear data, while polynomial regression, which is a special case of linear regression, is used to deal with non-linear data points. In this case, a curvilinear relationship is established between input x and output y.

This is done by adding dimensions to the input data, where n is the degree of polynomial.

> Polynomial regression is seen as a special case of linear regression since it is still linear in the regression coefficients.

Equation:

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_n x^n$$

Where:

- Y is the target
- x is the predictor
- Omega_0 is the bias
- Omega_1 - Omega_n are the weights in the equation of the polynomial regression
- n is the degree of the polynomial

> The degree of polynomial can be found by cross-validation

Pros	Cons
Provides the best approximation of the relationship between the dependent and independent variables	The presence of one or two outliers in the data can seriously affect the result of the nonlinear analysis
A broad range of functions can be fit under it	Too sensitive to outliers
Polynomial basically fits a wide range of curvature	Presence of fewer model validation tools for the detection of outliers in nonlinear regression

Use cases for Regression on 3D point clouds:

- Line fitting
- Plane fitting
- Grid point fitting Paper
- Fitting other (non-linear) surfaces

> If you were wondering about whether 3D point clouds and 3D meshes are the same:
>
> - A point cloud is a collection of points to represent an object in the given co ordinate system. For example, in a 3 dimensional (X, Y, Z) co ordinate system, a point cloud represent a 3D object. Point clouds are used to create 3D meshes and other models used in 3D modeling
> - A 3D mesh is the structural build of a 3D model consisting of polygons. 3D meshes use reference points in X, Y and Z axes to define shapes with height, width and depth.

Clustering (or "Cluster Analysis")

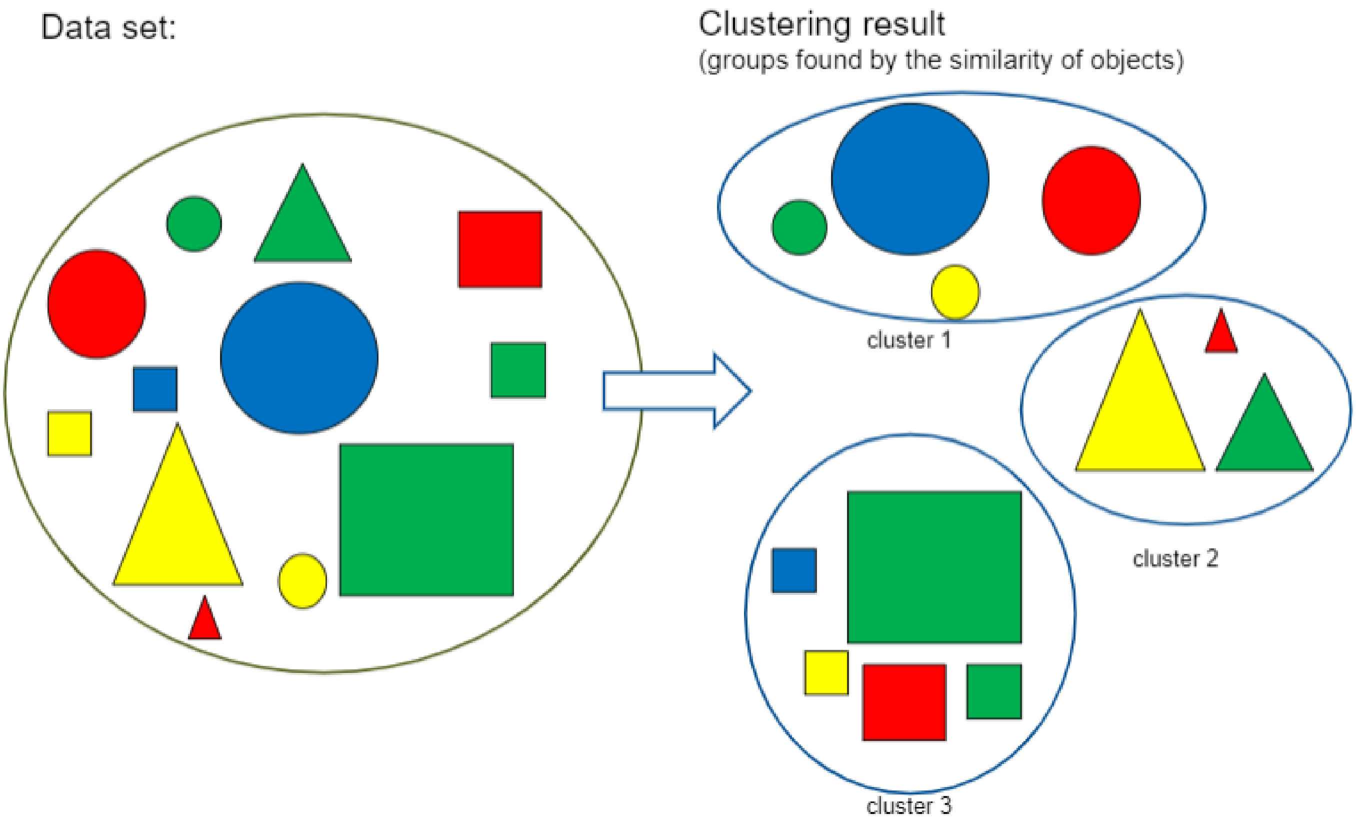
Find previously unkown groups (="clusters") in a data set, such that:

- data objects within one cluster are **similar to each other**
- data objects of different clusters are **dissimilar to each other**

Some applications of clustering:

- find different types of customers
  - e.g. an online shop could find groups of customers that are likely to spend a lot of money and customers that have not spend money for a long time
- detect communities in social networks
- group similar pixels in images (image processing)
- group similar documents (search engines, text mining)
- find similar recording from technical systems (automotive, automation)

- etc.



> The image could be clustered in different ways, for example shape, sizes or colors. Clustering is known as unsupervised learning because no information about classes is available for the instances.

The most common types of clustering methods are:

- partitioning methods
- hierarchical methods
- density-based methods
- grid-based methods

Partitioning method (k-means)

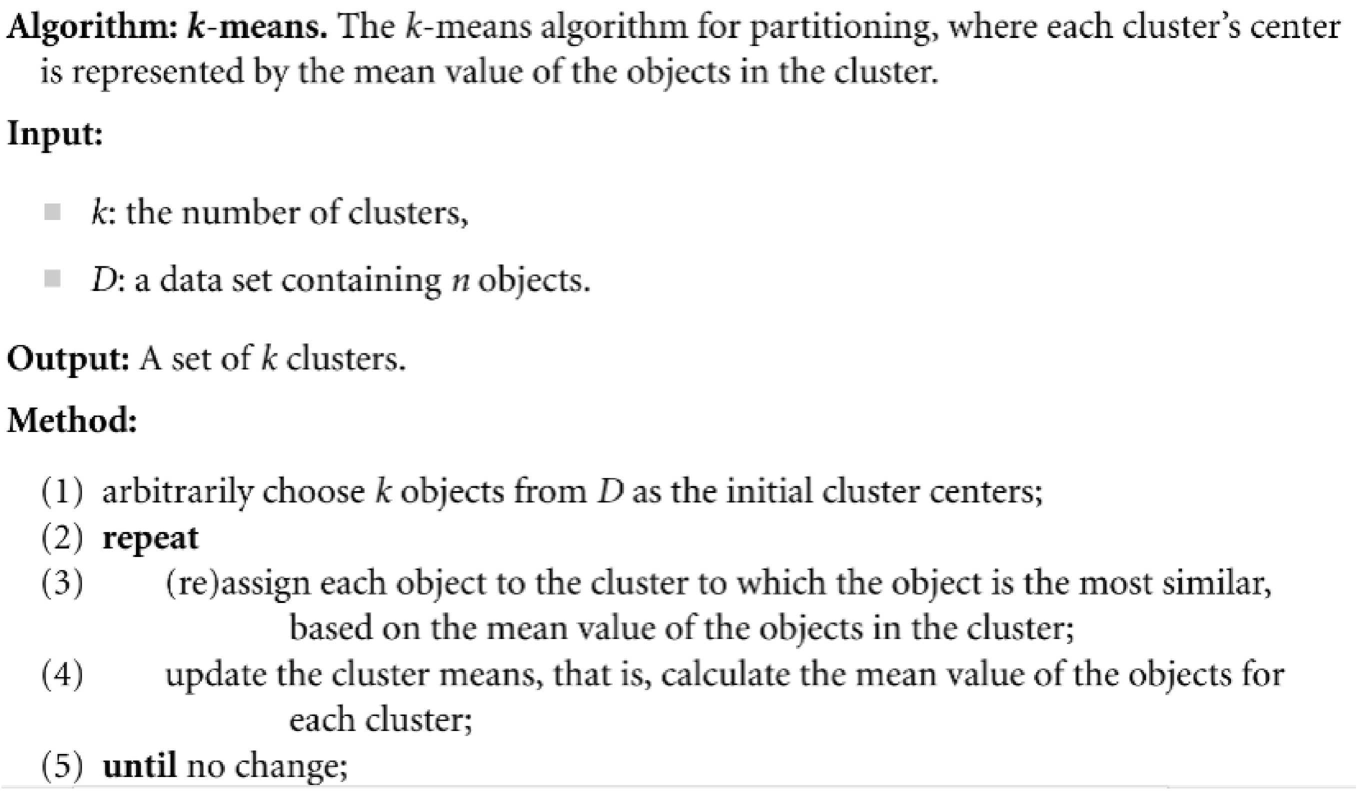
- find k clusters in the data set (k has to be pre-defined !)
- each cluster must contain >= 1 instances
- each instance must belong to exactly one cluster
- usually distance-based

> One distance based approach is to minimize the sum of squared euclidean distances (distortion)
>
> - $$J = \sum_{k} \sum_{n \in k} d(x_n, \mu_k)^2$$
> - where x is a vector representing the n^{th} data point assigned to cluster k and μ_k is the centroid of the cluster k. The function d() is the Euclidean distance between x and μ

Steps:

1. Creation of initial partitioning (e.g. randomly)
2. Iterative improvement of the partitioning by moving objects from one cluster to another. This is done by optimizing some criterion of what a "good" partitioning should look like.
3. Stop, if the partitioning quality criterion is satisfied.

Algorithm k-Means visualised



Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

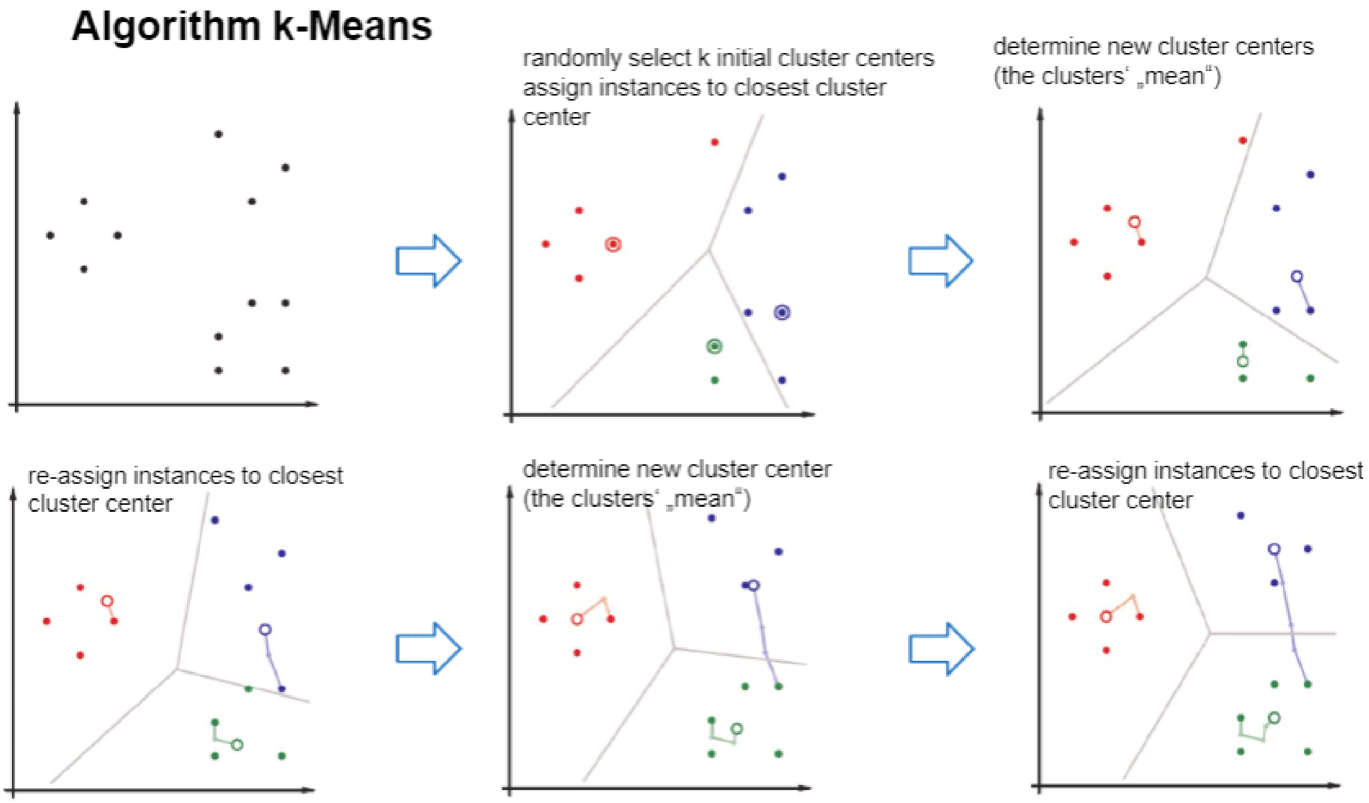
- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
(2) repeat
(3)     (re)assign each object to the cluster to which the object is the most similar,
            based on the mean value of the objects in the cluster;
(4)     update the cluster means, that is, calculate the mean value of the objects for
            each cluster;
(5) until no change;
```

Algorithm k-Means visualised



Algorithm k-Means

randomly select k initial cluster centers assign instances to closest cluster center

determine new cluster centers (the clusters' „mean")

re-assign instances to closest cluster center

determine new cluster center (the clusters' „mean")

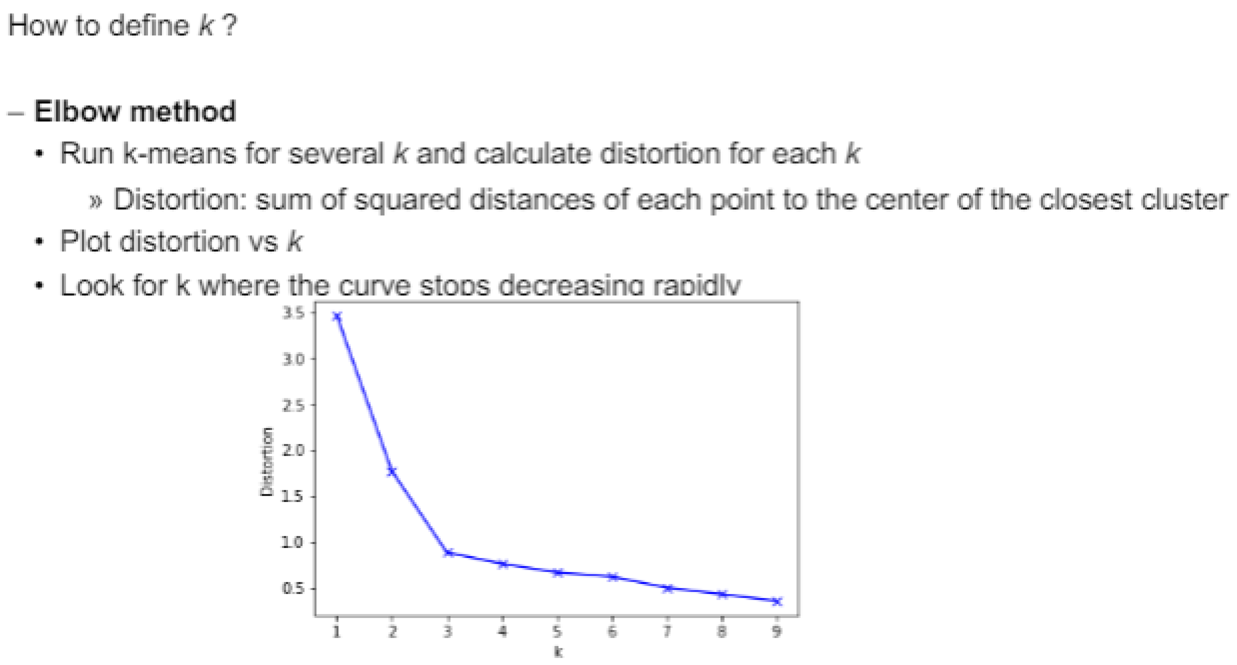
re-assign instances to closest cluster center

How to define k?

There are multiple ways to find the best k value for a given data set. However, two approaches are the most popular, namely:

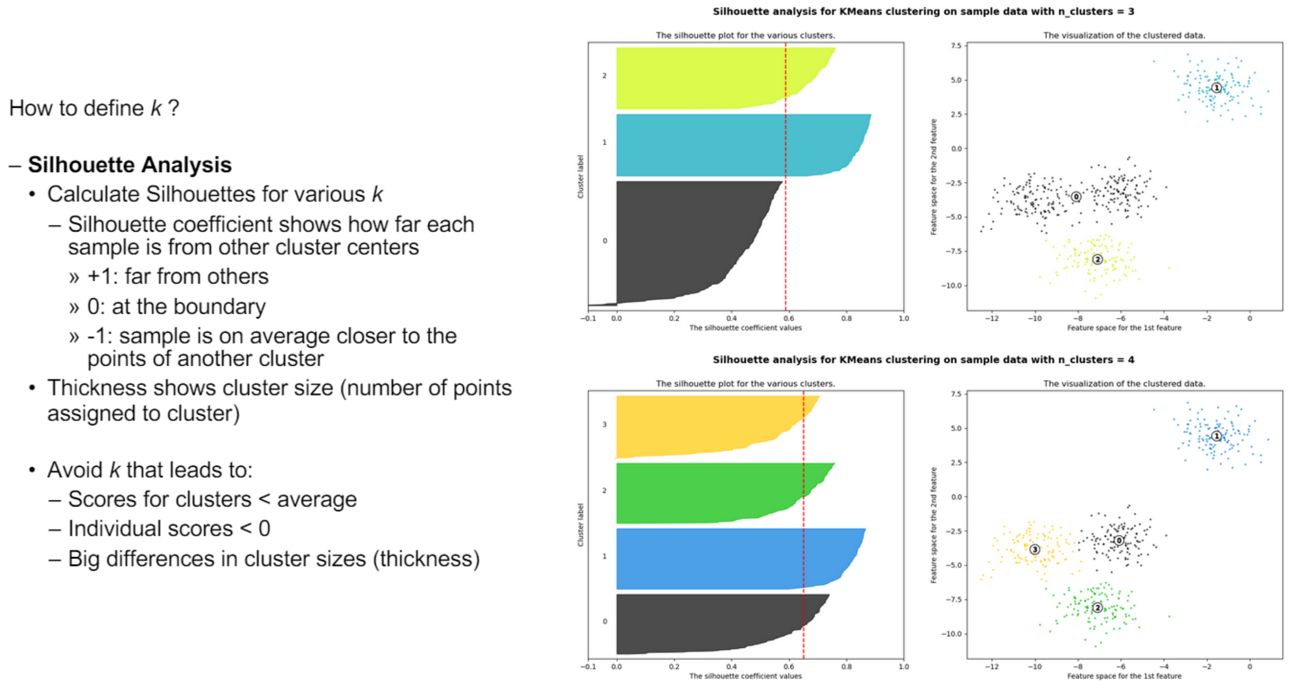
- Elbow method
- Silhouette Analysis

> The following illustrations elaborate the key points of those approaches, nicely.



How to define *k* ?

– **Elbow method**
- Run k-means for several *k* and calculate distortion for each *k*
  - » Distortion: sum of squared distances of each point to the center of the closest cluster
- Plot distortion vs *k*
- Look for k where the curve stops decreasing rapidly



How to define *k* ?

– **Silhouette Analysis**
- Calculate Silhouettes for various *k*
  - – Silhouette coefficient shows how far each sample is from other cluster centers
    - » +1: far from others
    - » 0: at the boundary
    - » -1: sample is on average closer to the points of another cluster
- Thickness shows cluster size (number of points assigned to cluster)
- Avoid *k* that leads to:
  - – Scores for clusters < average
  - – Individual scores < 0
  - – Big differences in cluster sizes (thickness)

Silhouette analysis for KMeans clustering on sample data with n_clusters = 3
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 1st feature
Feature space for the 2nd feature
Silhouette analysis for KMeans clustering on sample data with n_clusters = 4
The silhouette plot for the various clusters.
The visualization of the clustered data.
Cluster label
The silhouette coefficient values
Feature space for the 1st feature
Feature space for the 2nd feature

Limitations of k-means:

- K needs to be specifically defined

- Final cluster assignments depend on initialization
  - Cluster assignments may be different on different runs
  - K-means may not achieve the global optimum
- Can describe only convex clusters geometries
- Computationally demanding as the number of points dimensions increases

3D Point Cloud Segmentation by k-means Point Cloud Expert: Florent Poux Medium

My take on that is, that in terms of dimensionality it should be 3, for k means and mean shift respectively, since clustering methods usually act in a manner of dimensionality reduction. However, there might be exceptions of that rule? Or maybe, I'm wrong with my assumption altogether.

Clustering - Mean shift

Definition: Mean Shift is an algorithm that finds the maxima - the modes - of a density function given discrete data sampled from that function. Thus, it is a density hill climbing algorithm

- Every point gives rise to a cluster.

- Each cluster is defined by the radius, a.k.a "bandwith", h of its region, and a kernel function K that is used to calculate the contribution of the points included within this radius.

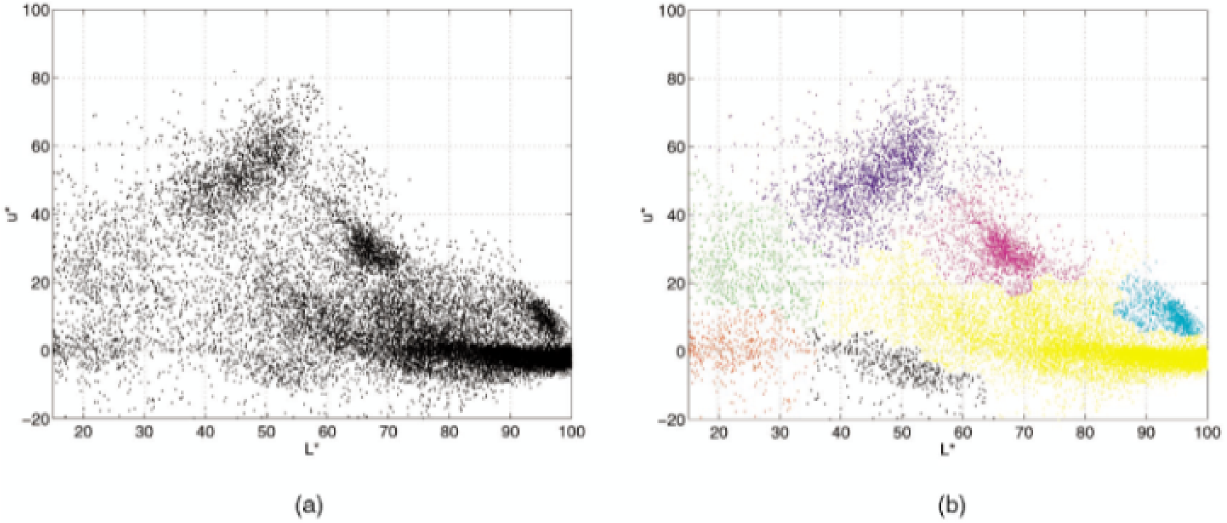
- In the end, only unique clusters are considered.

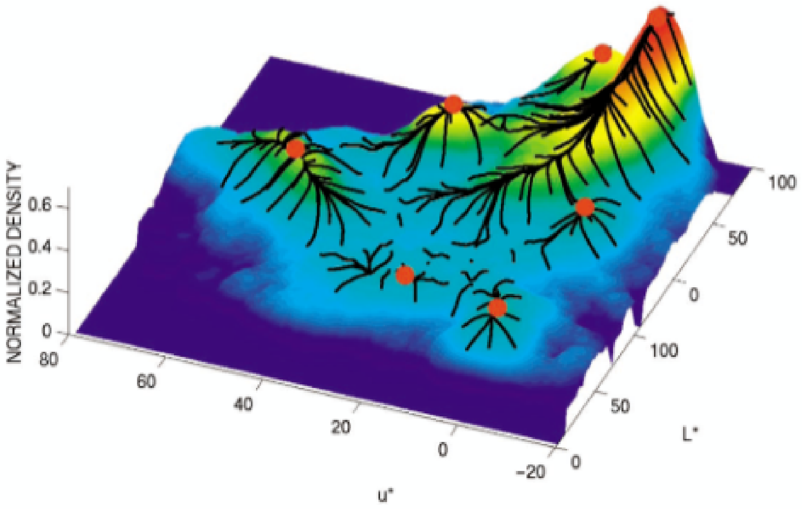
- So, we do not need to set the number of clusters!

  - However, we need to define the bandwith (and the kernel function).

Further definitions:

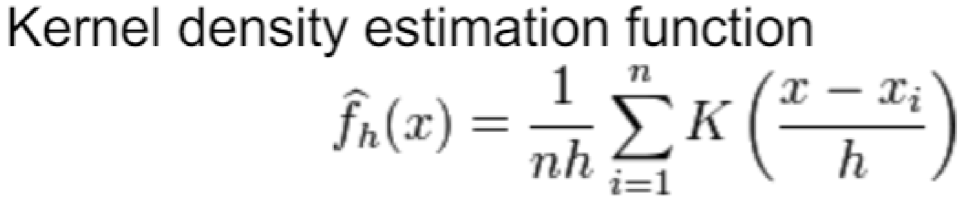
- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode





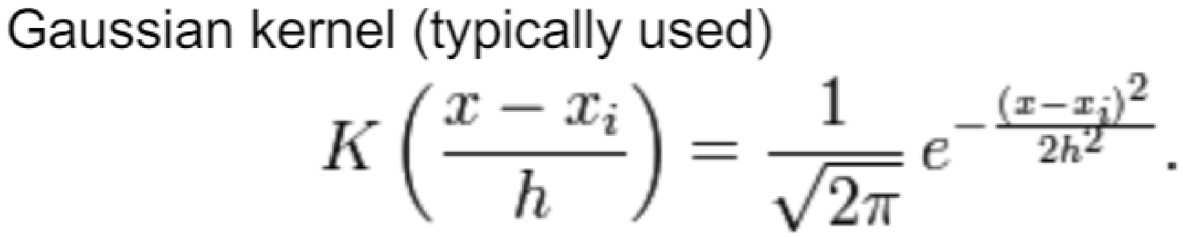
Pros	**Cons**
General, application-independent tool	Output depends on window size
Model-free, does not assume any prior shape (spherical, elliptical, etc.) on data clusters	Window size (bandwith) selection is not trivial
Just a single parameter (window size h)	Computationally (relatively) expensie (~2s/image)
Finds variable number of modes	Does not scale well with dimension of feature space
Robust to outliers	

Kernel density estimation



Kernel density estimation function

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$



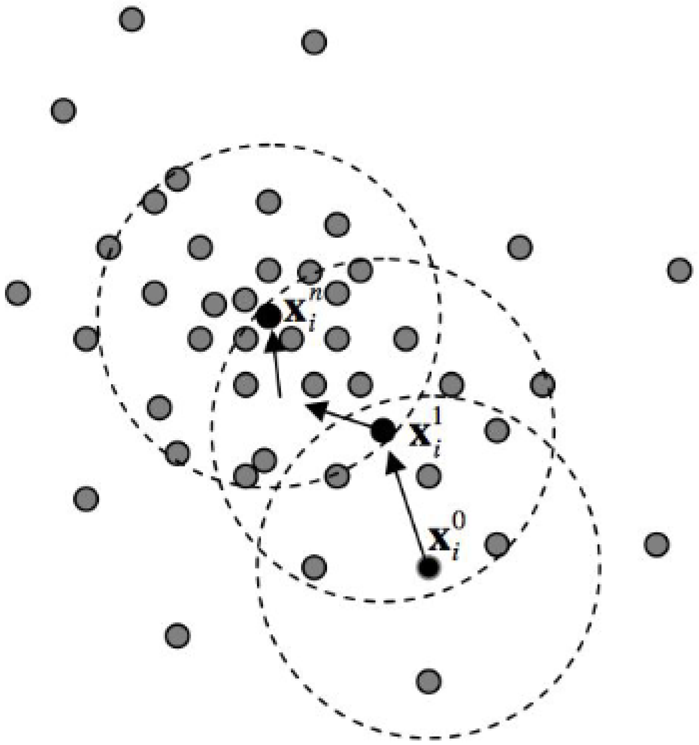
Gaussian kernel (typically used)

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift clustering

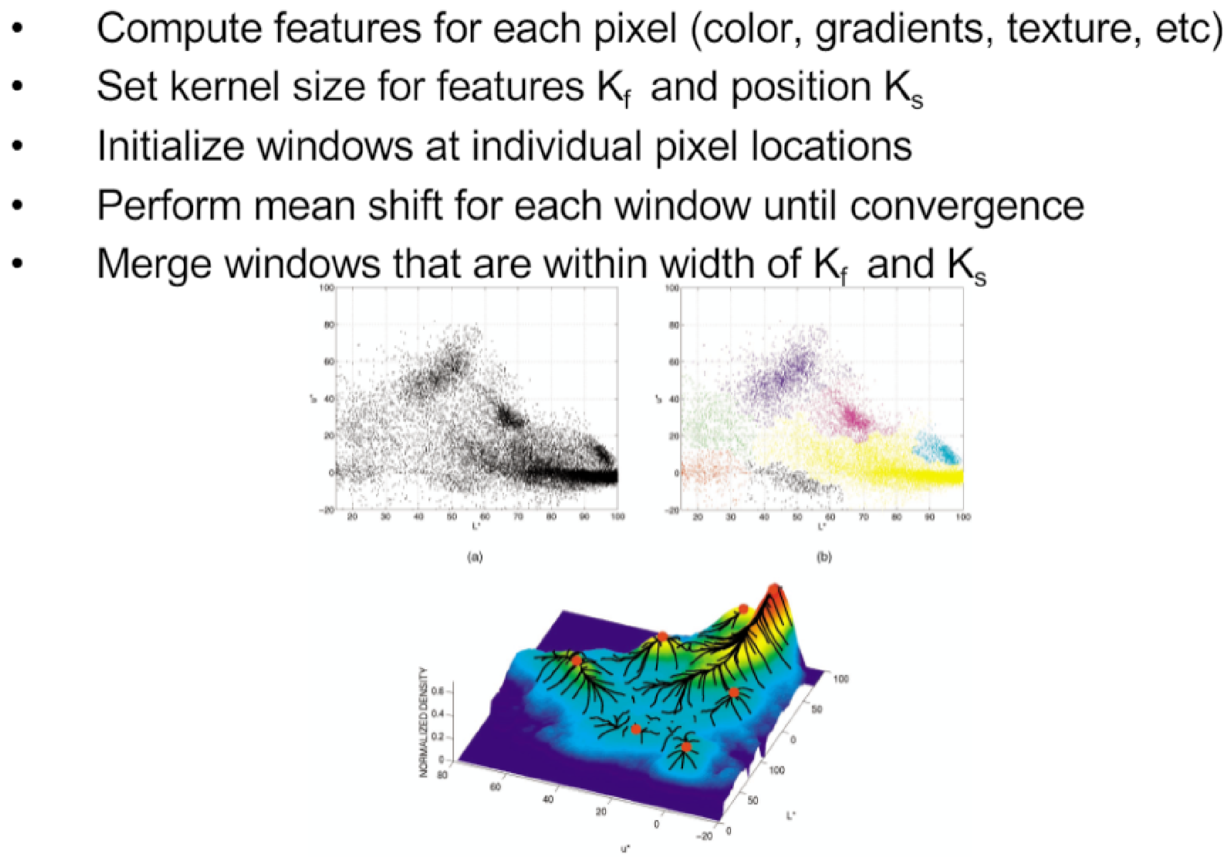
The mean shift algorithm seeks modes of the given set of points

1. Choose kernel and bandwidth
2. For each point:
    1. Center a window on that point
    2. Compute the mean of the data in the search window
    3. Center the search window at the new mean location
    4. Repeat (2,3) until convergence
3. Assign points that lead to nearby modes to the same cluster



> Mean shift procedure. Starting at data point x_i, run the mean shift procedure to find the stationary point of the density function. Superscripts denote the successive window centres, respectively, and the dotted circles denote the density estimation windows.

2D image Segmentation by Mean Shift



- Compute features for each pixel (color, gradients, texture, etc)
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that are within width of K_f and K_s

3D Point Cloud Segmentation by Mean ShiftPoint Cloud Expert: Florent Poux Medium

My take on that is, that in terms of dimensionality it should be 3, for k means and mean shift respectively, since clustering methods usually act in a manner of dimensionality reduction. However, there might be exceptions of that rule? Or maybe, I'm wrong with my assumption altogether.

Clustering - DBSCAN Article

Definition: DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise and is a unsupervised density-based clustering algorithm.

DBSCAN requires only two parameters: ϵ and minPoints. ϵ is the radius of the circle to be created around each data point to check the density and minPoints is the minimum number of data points required inside that circle for that data point to be classified as a Core point.

In density-based clustering:

- we partion points into dense regions separated by not-so-dense regions.
- a cluster is defined as a maximal set of density-connected points
- we can discover clusters of arbitrary shape

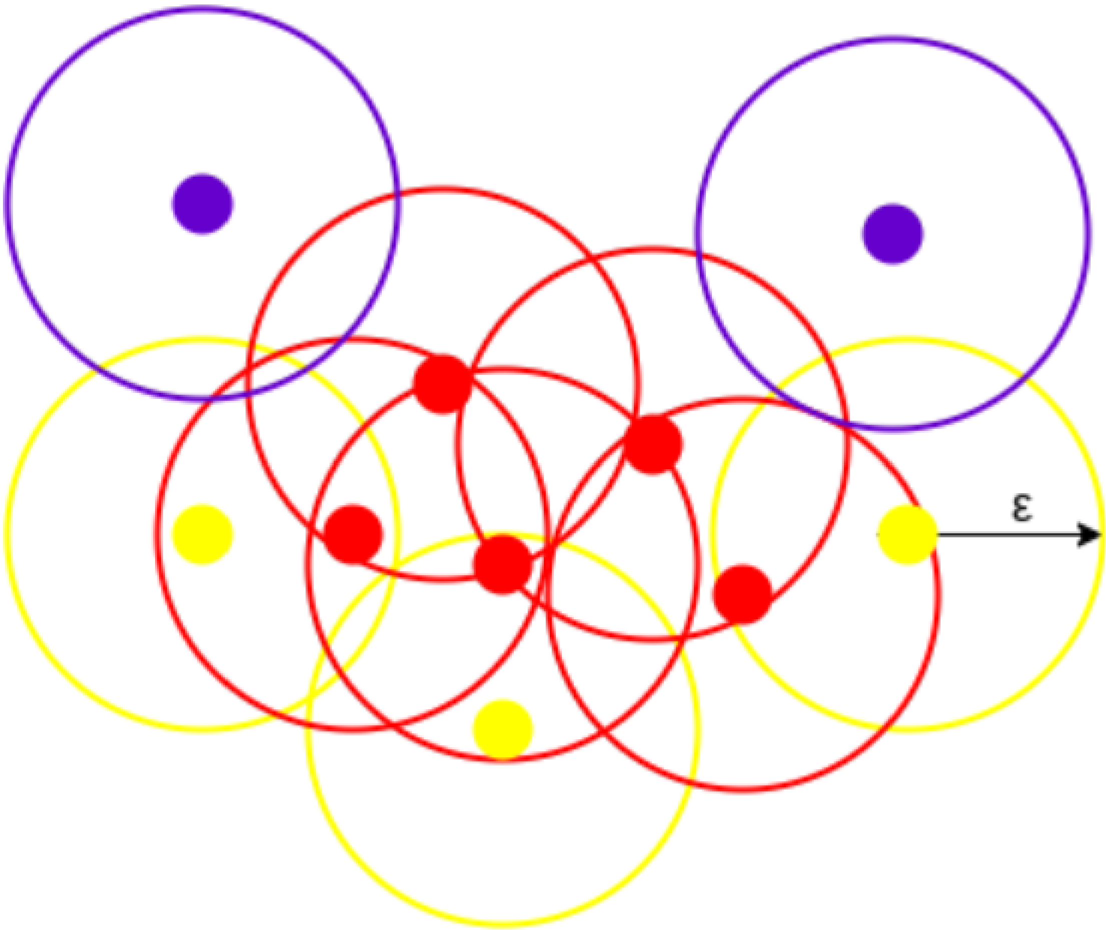
Density definition:

- Density at point p is defined as the number of points within a circle/sphere of radius ϵ
- A region is dense if the circle/sphere of radius ϵ contains at least MinPts points.

Types of points:

- A core point has more than a specified number of points (MinPts) within ϵ
- A border point has fewer than MinPts within ϵ, but is the neighborhood of a core point.
- A noise point is any point that is not a core point or a border point.

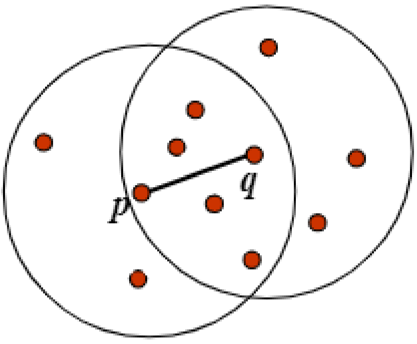
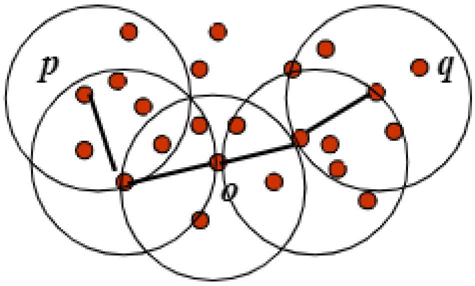
Example



> All the data points with at least 3 points (MinPnts = 2) in the circle including itself are considered as Core points represented by red color. All the data points with less than 3 but greater than 1 point in the circle including itself are considered as Border points. They are represented by yellow color. Finally, data points with no point other than itself present inside the circle are considered as Noise represented by the purple color.

Reachabillity and Connectivity

Reachability states if a data point can be accessed from another data point directly or indirectly, whereas Connectivity states whether two data points belong to the same cluster or not.

Density edge	Density-connected
	

Density edge	Density-connected
We place an edge between two core points q and p if they are within distance ϵ	A point p is density-connected to a point q if there is a path of edges from p to q

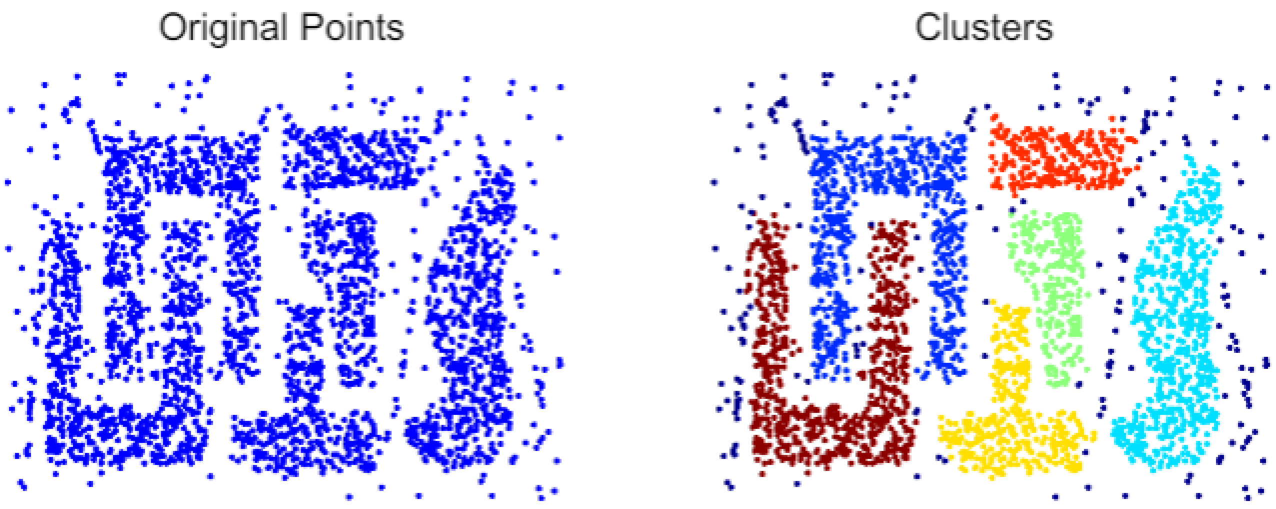
> There are other accessibility and connectivity metrics: e.g. direct density-reachable and density-reachable. See the linked article for further explanation.

DBSCAN algorithm

1. Label points as core, border and noise
2. Eliminate noise points
3. For every core point p that has not been assigned to a cluster
   - Create a new cluster with the point p and all the points that are density-connected to p.
4. Assign border point to the cluster of the closest core point.

> For locating data points in space, DBSCAN uses Euclidean distance, although other methods can also be used (like great circle distance for geographical data). It also needs to scan through the entire dataset once, whereas in other algorithms we have to do it multiple times.

Visualization



Pros	Cons
Can create clusters of arbitrary shapes	Some points (noise points) are not assigned to any cluster (is this necessarily bad?)
No need to define the number of clusters	Need to define ϵ and MinPts
Resistant to noise	Problems with point clouds that contain regions of varying densities

Hierarchical Clustering

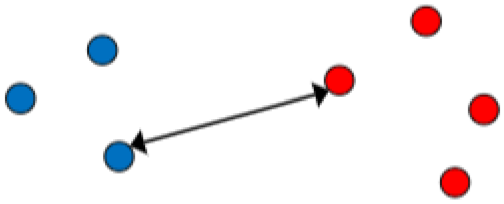
Hierarchical methods create a hierarchy of splits/merges of a dataset:

- **agglomerative methods (bottom-up):**
    1. start with each data object being one cluster
    2. iteratively merge the clusters
    3. stop when all clusters are merged or a stopping condition is met
- **divisive methods (top-down):**
    1. start with all data objects being one cluster
    2. iteratively split each cluster into smaller ones
    3. stop when each object forms its own cluster or a stopping condition is met

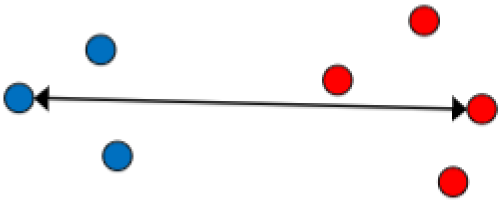
> - merging or splitting is done based on dissimilarities (= distances, so-called "linkages") or on densities
> - a merging or splitting step can not be undone

Linkage variants

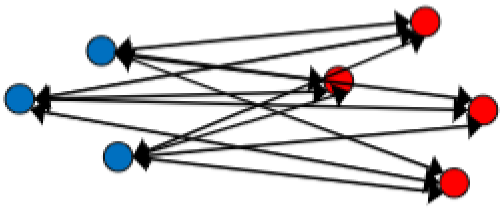
Single-link: the distance between two clusters is equal to the minimum distance (dissimilarity) between the two most similar data objects of the two clusters.



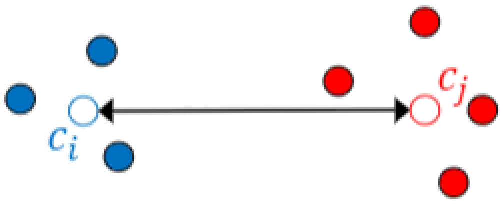
Complete-link: the distance between two clusters is equal to the maximum distance (dissimilarity) between the two most similar data objects of the two clusters.



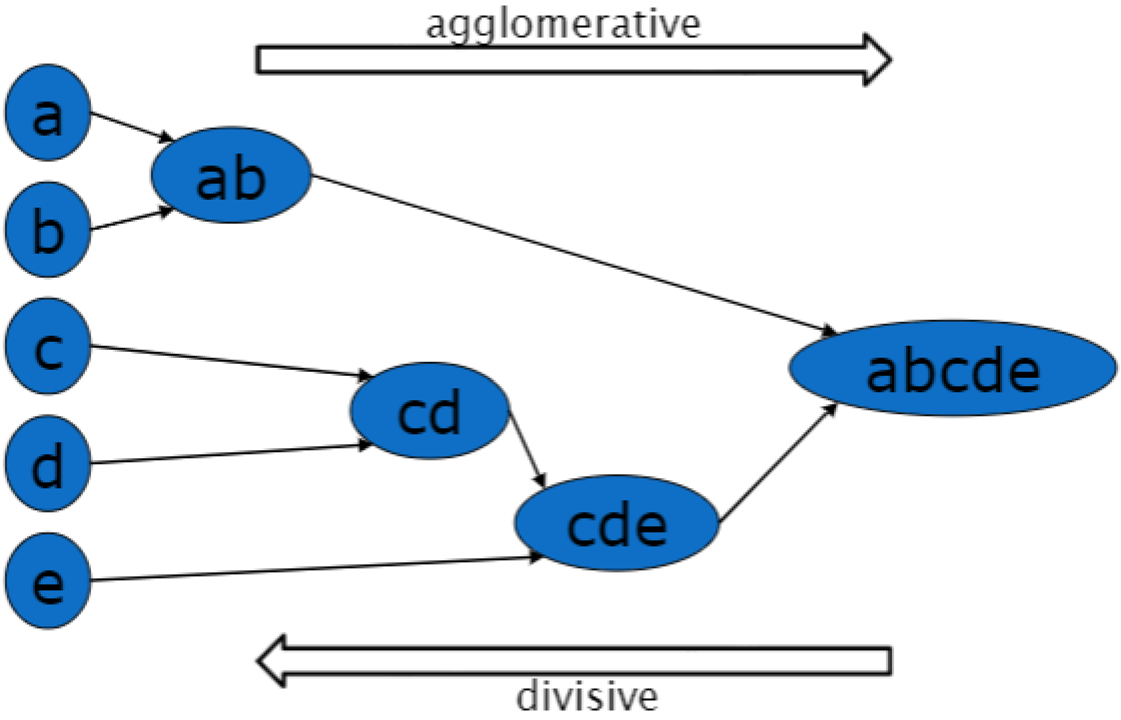
Average-link: the distance between two clusters is equal to the average distance from any member of one cluster to any member of the other cluster.



Centroid: the distance between two clusters is equal to the dissimilarity between the centroids, e.g. the mean value vectors of those two clusters.



The following figure illustrates the process of the hierarchical methods:



Hierarchical clustering representation: The output of the hierarchical clustering is a dendogram.

A dendogram:

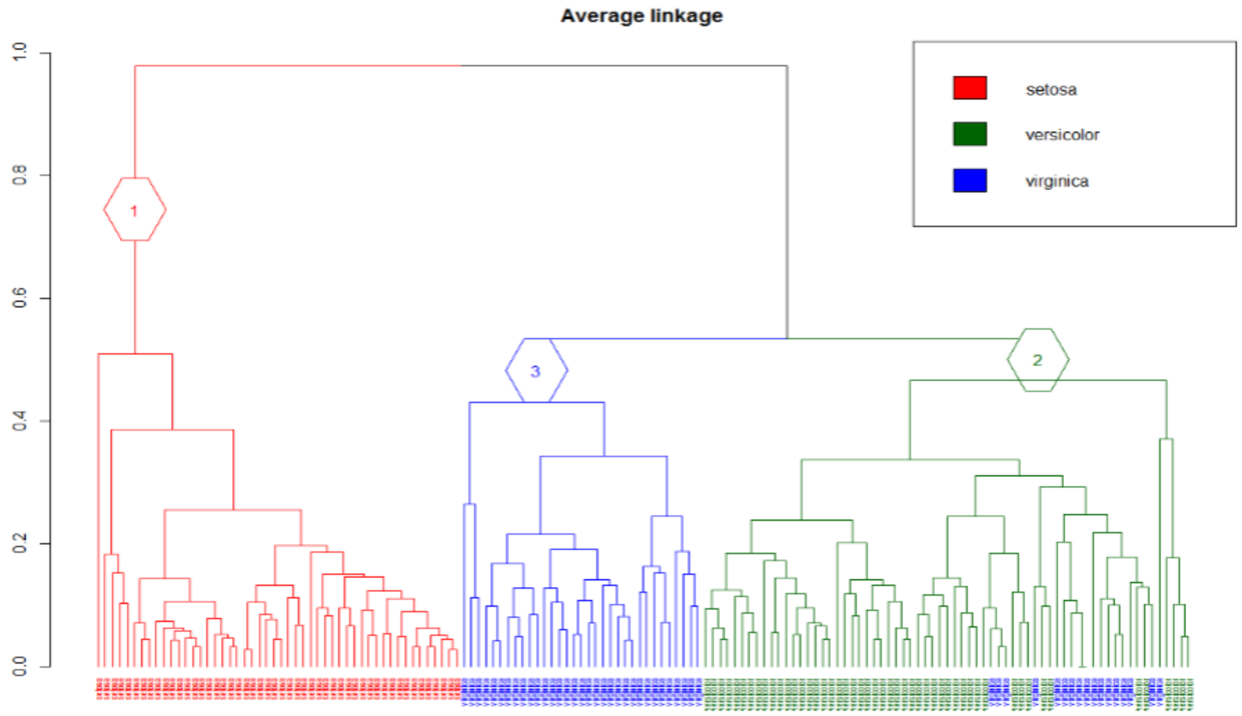
- Consists of many Π-shaped lines that connect data points in a hierarchical tree like structure.
- The height of each Π represents the distance between the two data points being connected.

Characteristics of Hierarchical Clustering:

- Any desired number of clusters can be obtained by 'cutting' the dendogram at the proper level
- They may correspond to meaningful taxonomies

Illustration of a dendogram

dendrogram of iris dataset
(with colours corresponding to cluster assignments, for 3 clusters)


Average linkage
setosa
versicolor
virginica