DTU

DTU

Lazaros Nalpantidis

3D Point Cloud Processing - Clustering & Regression

Outline

- What are Clustering and Regression? A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?

- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

Outline

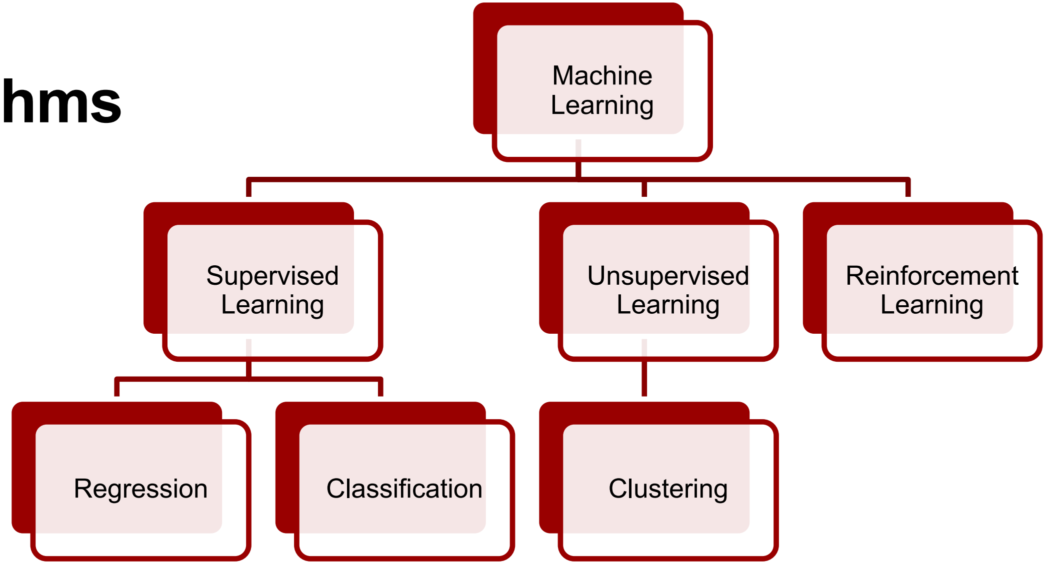
- What are Clustering and Regression? A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?

- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

What are Clustering and Regression?

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.

Outline

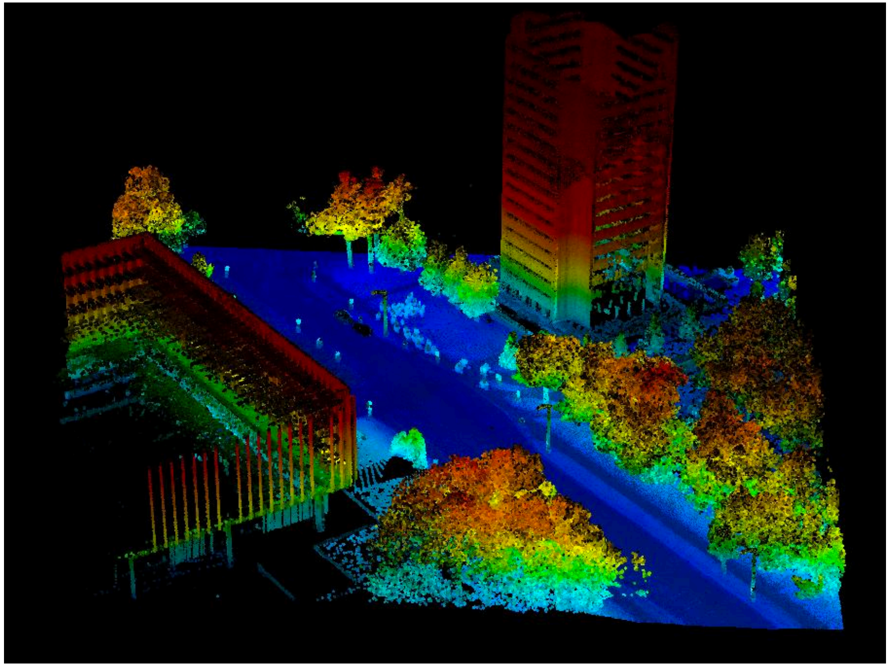
- What are Clustering and Regression? A taxonomy of ML algorithms
- <u>Why is ML important for Perception of Autonomous Systems?</u>

- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

Why is ML important for Perception of Autonomous Systems?

- Create initial object hypotheses
  – input to pose estimation?
- Abstract representation of scene
- …what else?



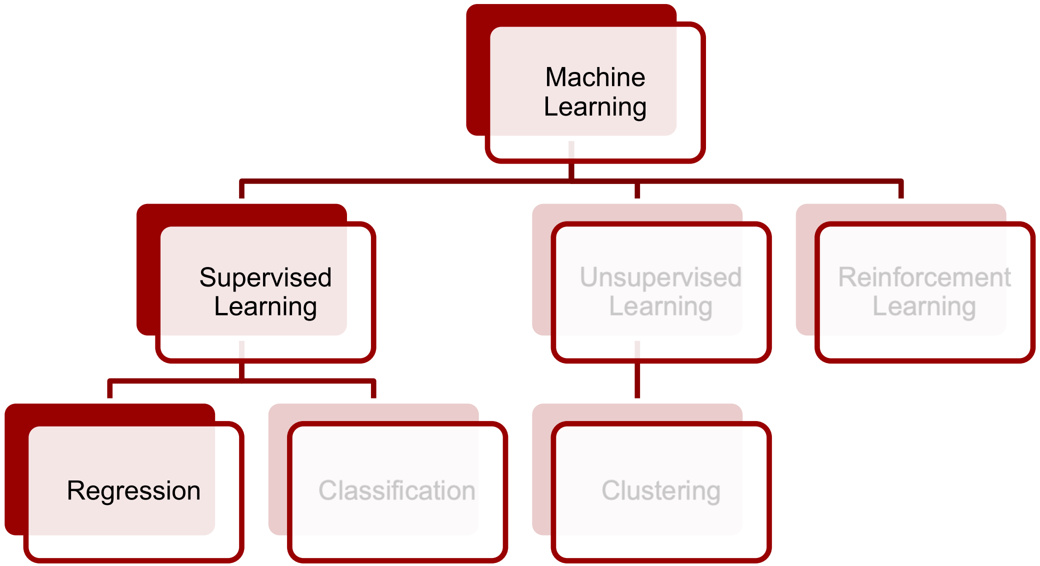
Outline

- What are Clustering and Regression? A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?

- <u>Regression</u>
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

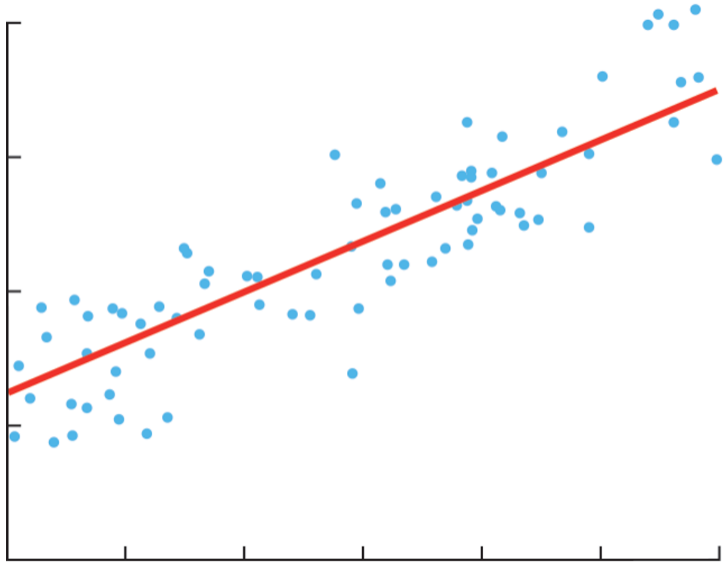
Regression



- **Supervised Learning** – The target values are known
  – Regression – The target value is numeric

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



Regression

- The goal is to find an equation *f( )* that models the relationship between input *x* and output *y* such as:

$$y = f(x) + \varepsilon$$

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

Regression

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

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

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

Regression

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

$$y = f(b, x) = b_1 x_1 + b_2 x_2 + b_3 x_3 + \ldots + b_n x_n + c = b^T x + c$$

  - 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.

Regression

- **Polynomial Regression**
  - In the case that the mapping between the inputs *x* and the output is not expressed well by a straight line we can use polynomial regression.
  - This is done by adding dimensions to the input data, where n is the degree of polynomial.

$$X = [x, x^2, x^3, \ldots, x^n]$$

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

Regression

- **Polynomial Regression**
  - In the case that the mapping between the inputs *x* and the output is not expressed well by a straight line we can use polynomial regression.
  - This is done by adding dimensions to the input data, where n is the degree of polynomial.

$$X = [x, x^2, x^3, \ldots, x^n]$$

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

Regression

- Can we use Regression on our 3D point clouds?
- What problems could we tackle?
  - Line fitting
  - Plane fitting
  - Fitting other (non-linear) surfaces
  - …what else?

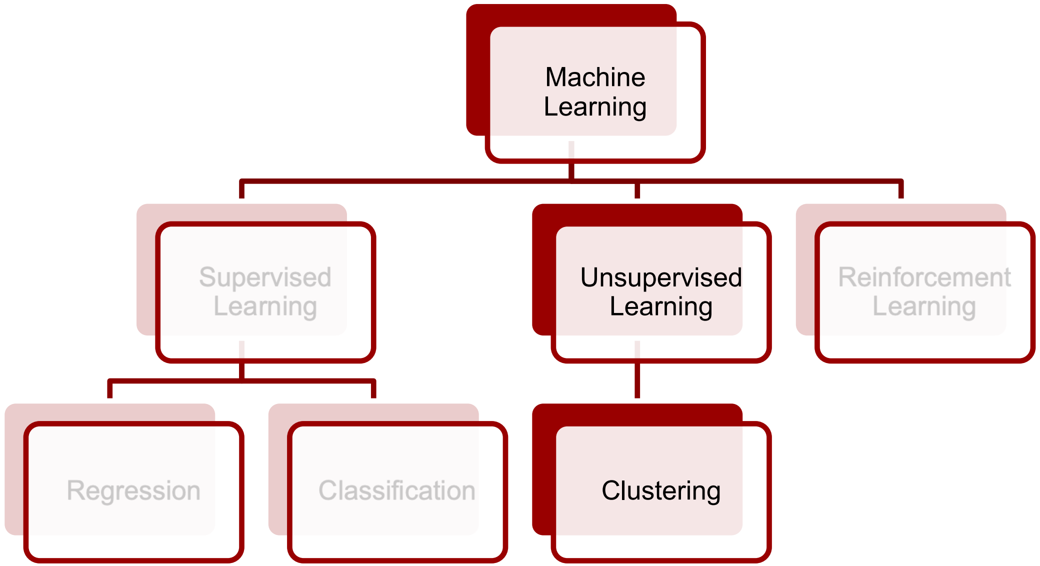
Outline

- What are Clustering and Regression? A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?

- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

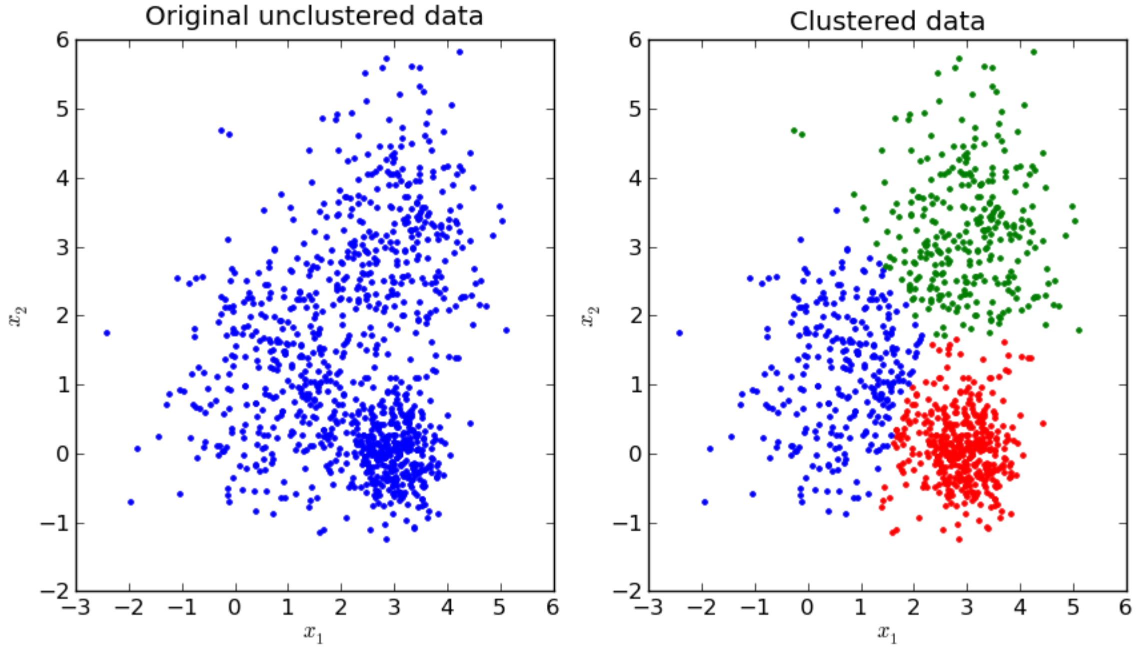
Clustering



- **Unsupervised Learning** – The target values are unknown
  - Clustering – Group together similar instances

Clustering

- Partitions a point cloud into groups of (similar) points, i.e. clusters,
- Finds the underlying structure of the point cloud.



Clustering – k-means

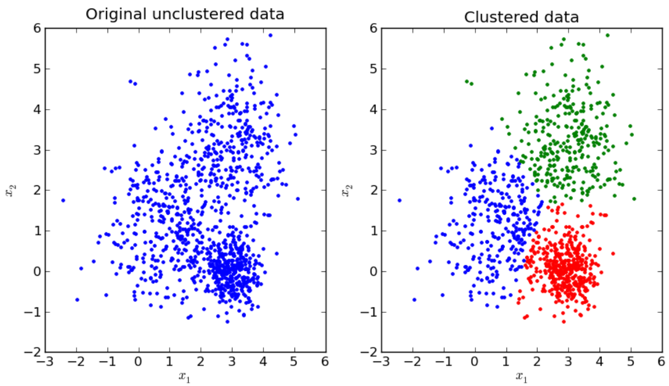
– Partitions a point cloud into k clusters, such that each point belongs to one cluster

– Assigns point to clusters, so as to minimize the Sum of Squared Euclidean Distances (Distortion) between points and their assigned cluster centers.

$$J = \sum_{k} \sum_{n \in k} d(x_n, \mu_k)^2$$

…where *x* is a vector representing the n^{th} data point *assigned to cluster k* and μ_k is the centroid the cluster *k* . The function d() is the Euclidean distance between *x* and μ .

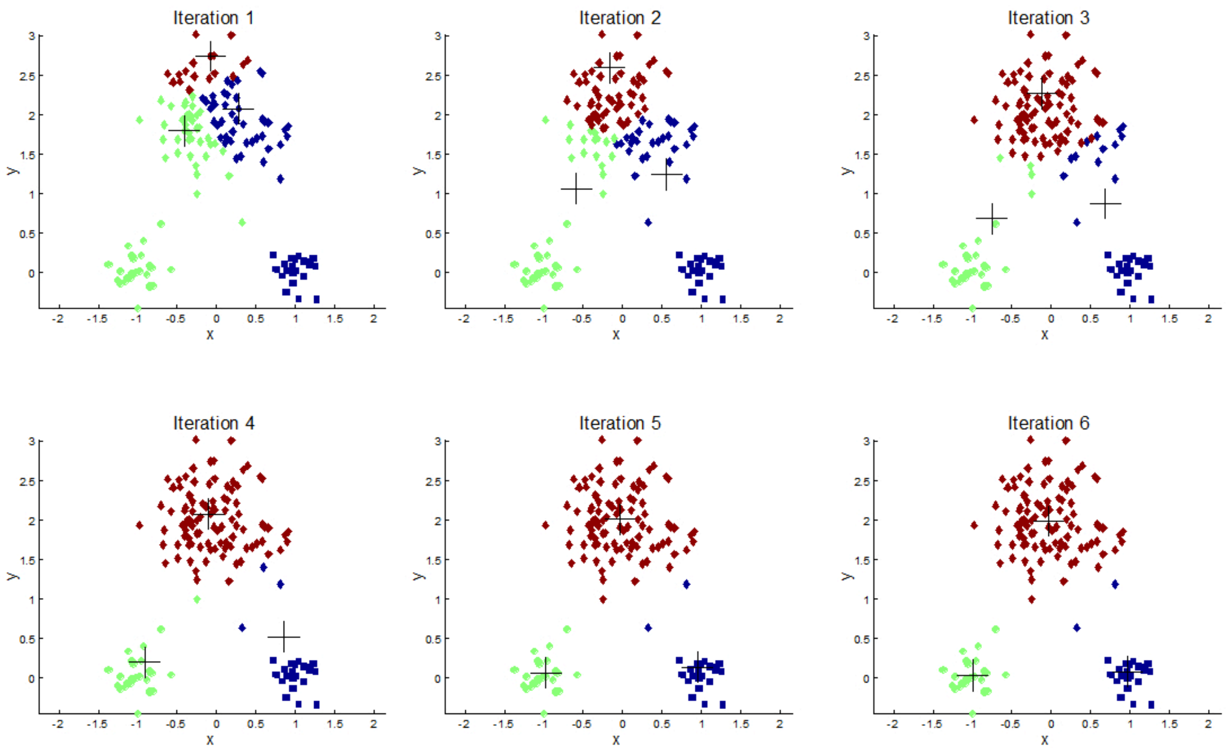
– **Important:** K-means requires known number of clusters *k*



Clustering – k-means

It is an iterative process:

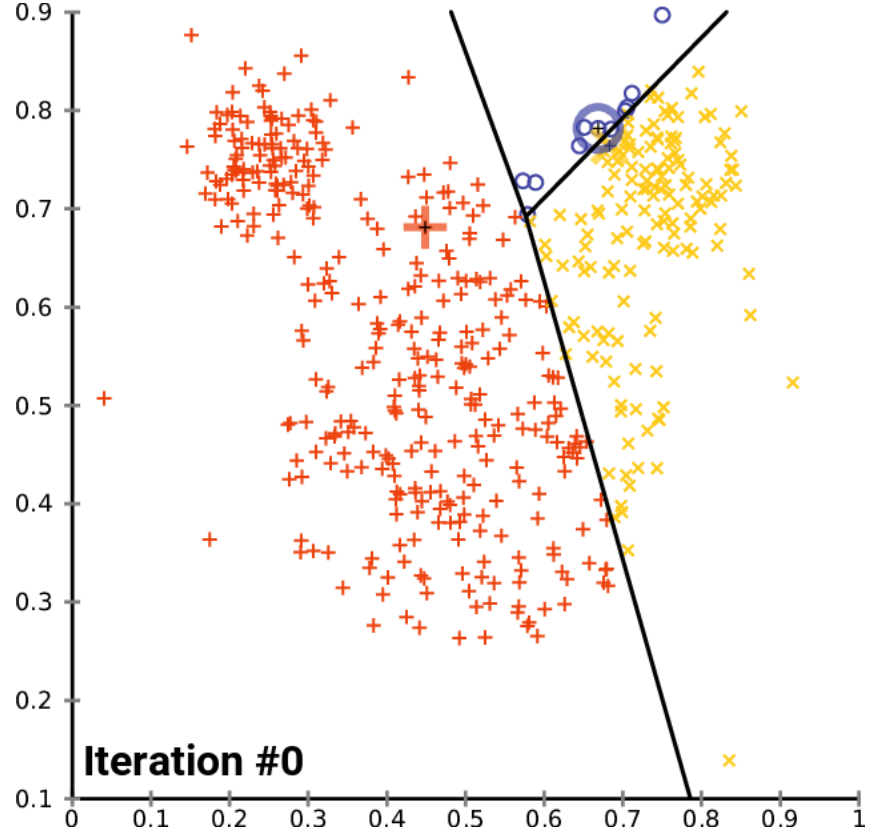
- Step 1:
  – Randomly initialize k cluster centers
- Step 2:
  – Assign each to its nearest cluster center
- Step 3:
  – Re-compute each cluster-center as the centroid of all points assigned to each cluster
- Step 4:
  – Check convergence/stop criterion, otherwise repeat Steps 2-4



Clustering – k-means

It is an iterative process:

- Step 1:
  - Randomly initialize k cluster centers
- Step 2:
  - Assign each to its nearest cluster center
- Step 3:
  - Re-compute each cluster-center as the centroid of all points assigned to each cluster
- Step 4:
  - Check convergence/stop criterion, otherwise repeat Steps 2-4



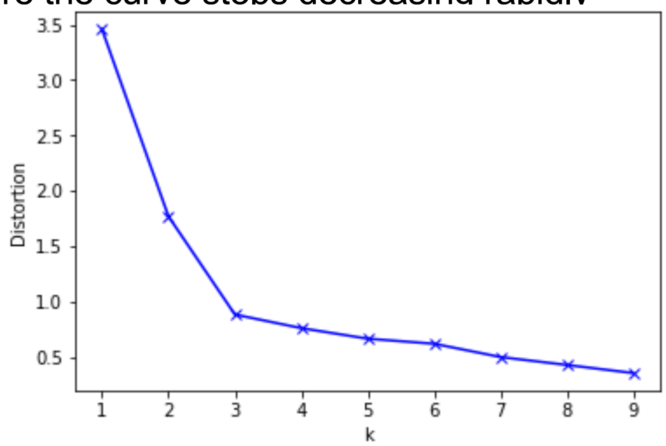
https://en.wikipedia.org/wiki/K-means_clustering

Clustering – k-means

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



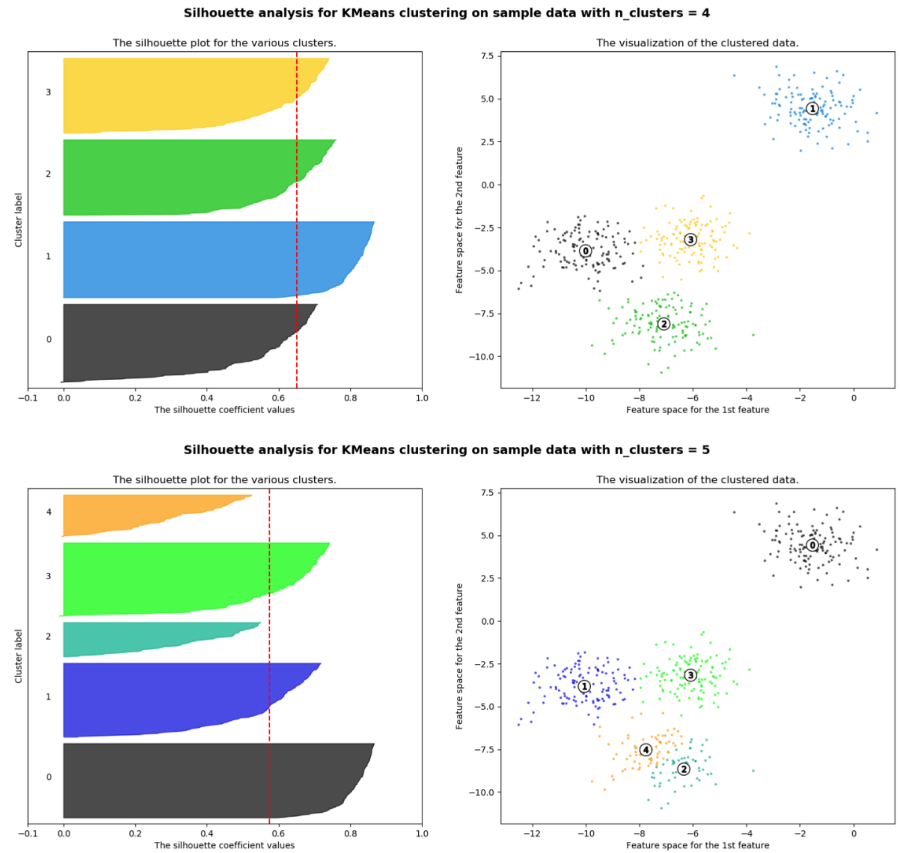
Clustering – k-means

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)

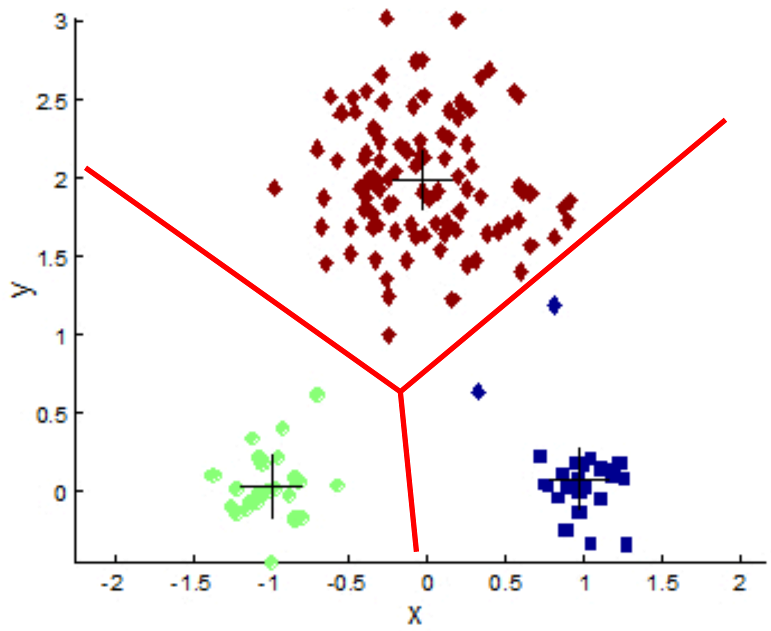


https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

Clustering – k-means

Issues

- Need to define *k*
- 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 cluster geometries
- Computationally demanding as the number of points and dimensions increase



3D Point Cloud Segmentation by k-means

Which space to apply Clustering? Is it your point cloud's 3D space?

What should be the dimensionality?

- dimensionality = 3 ? → X, Y, Z
- dimensionality = 6 ? → X, Y, Z, R, G, B
- dimensionality = 6 ? → X, Y, Z, L, U, V
- dimensionality = 6 ? → X, Y, Z, H, S, V
- dimensionality > 6 ? → X, Y, Z, H, S, V, gradients, texture, … ??

Clustering – Mean Shift

- **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. "bandwidth", *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!

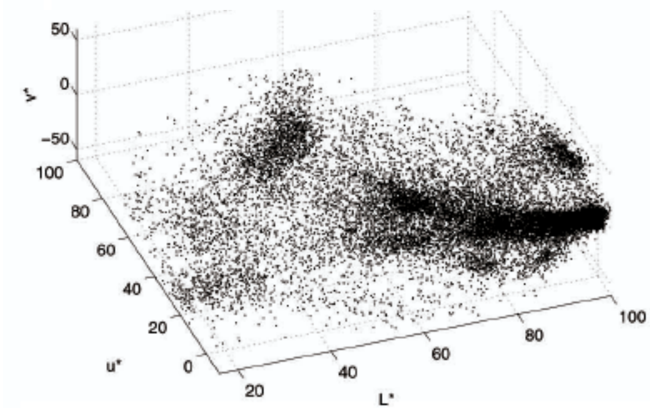
    – However, we need to define the bandwidth (and the kernel function).

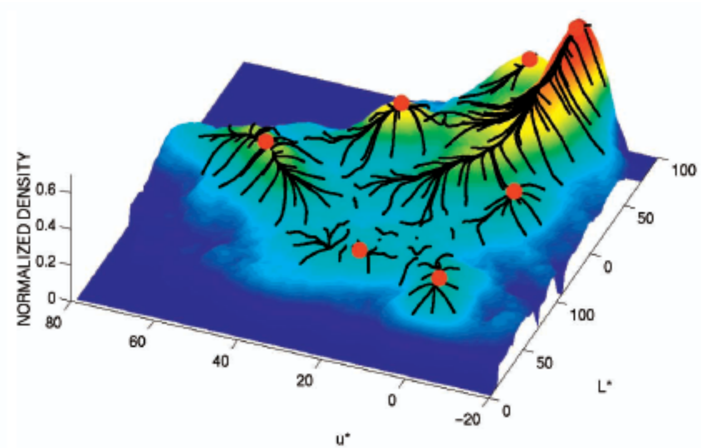
Dorin Comaniciu and Peter Meer, "Mean Shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002. pp. 603-619.

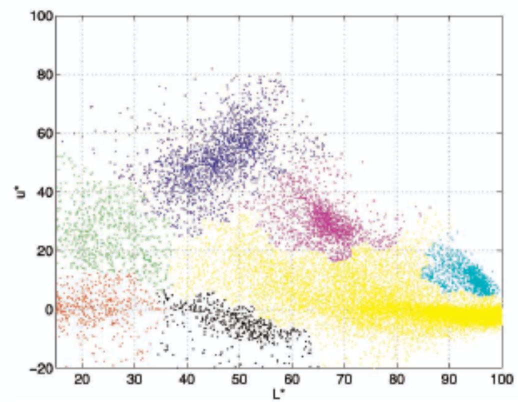
Yizong Cheng, "Mean shift, mode seeking, and clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, pp. 790–799, Aug 1995.

Mean shift algorithm

- Try to find *modes* of this non-parametric density







Kernel density estimation

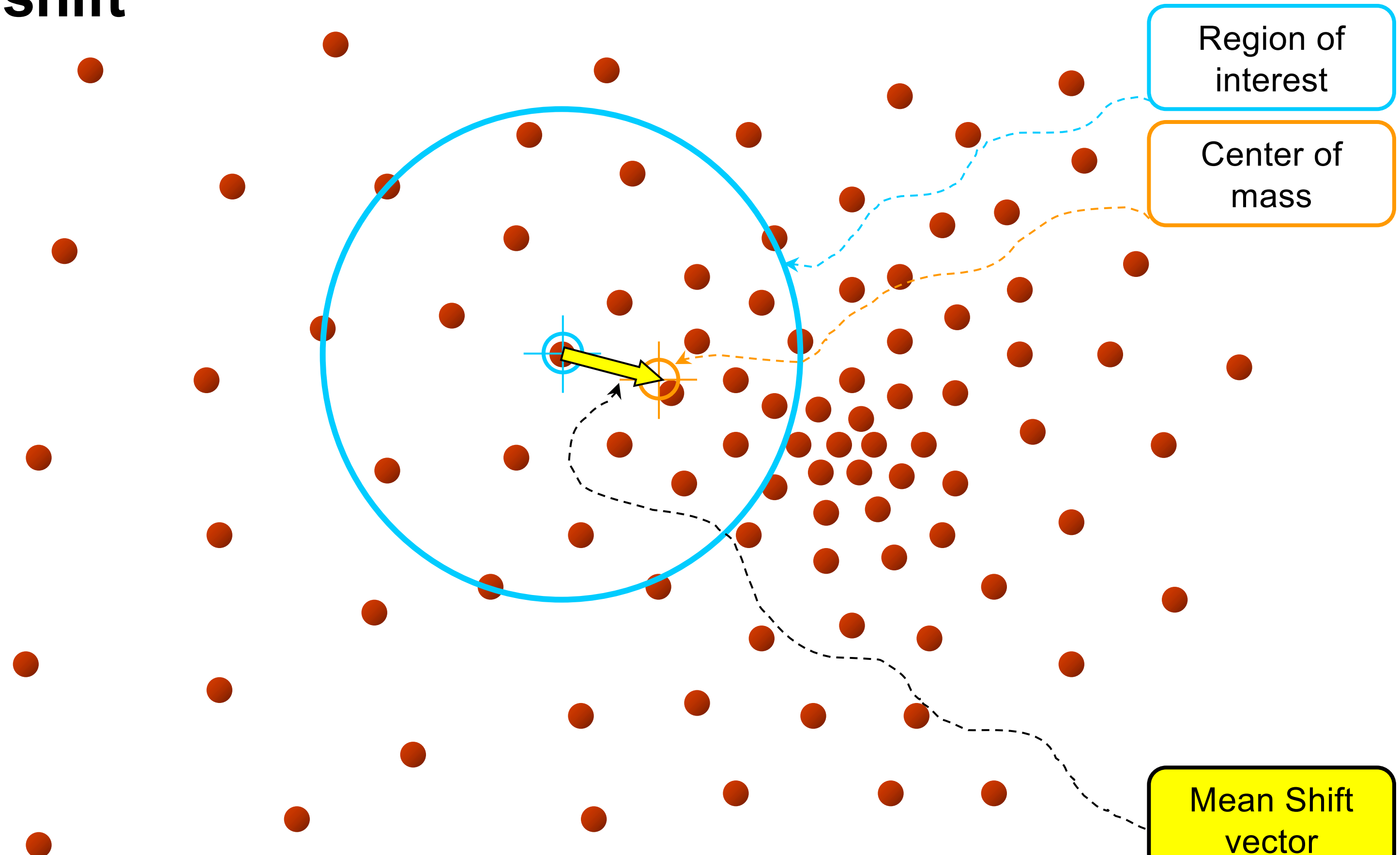
Kernel density estimation function

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

Gaussian kernel (typically used)

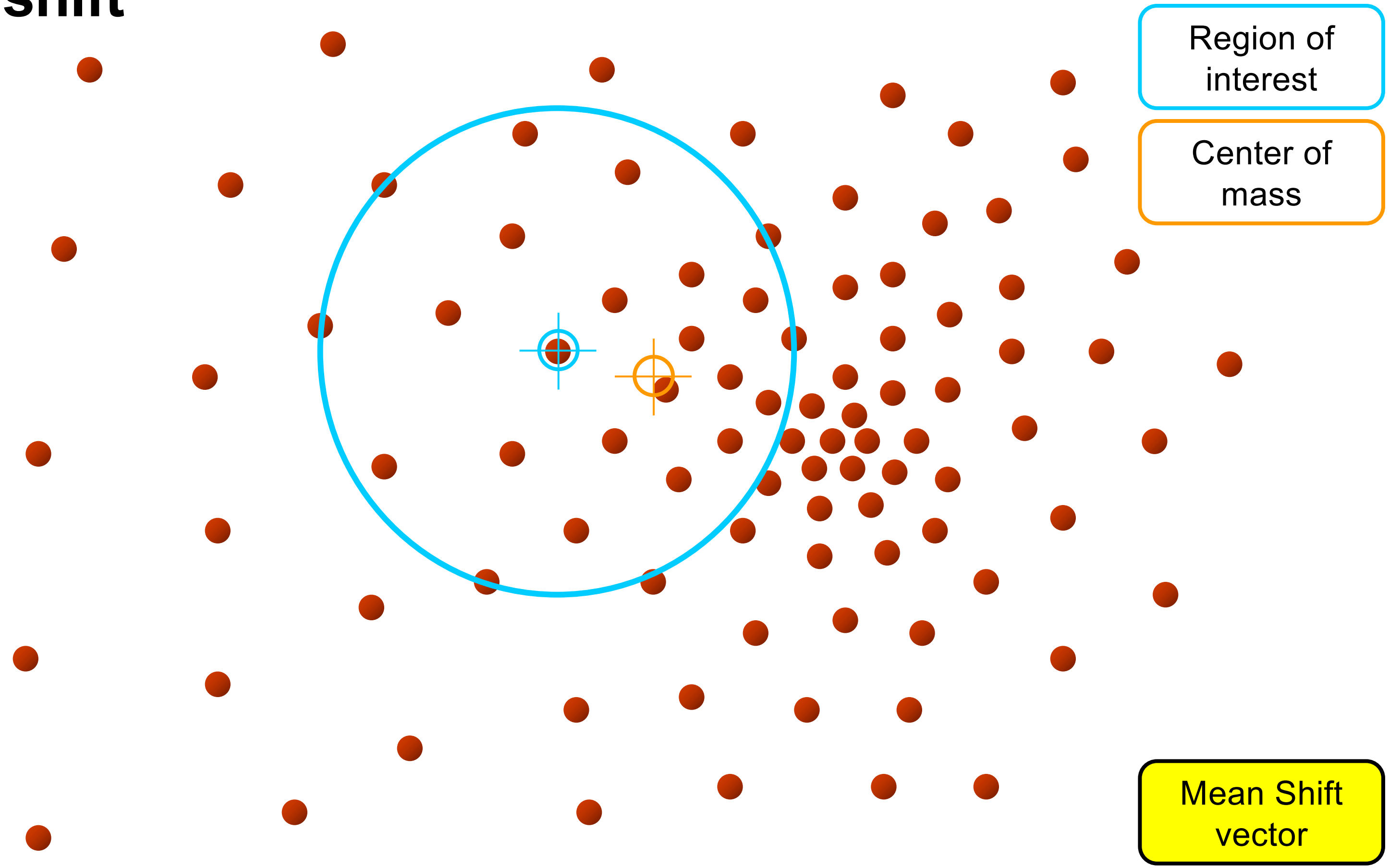
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift



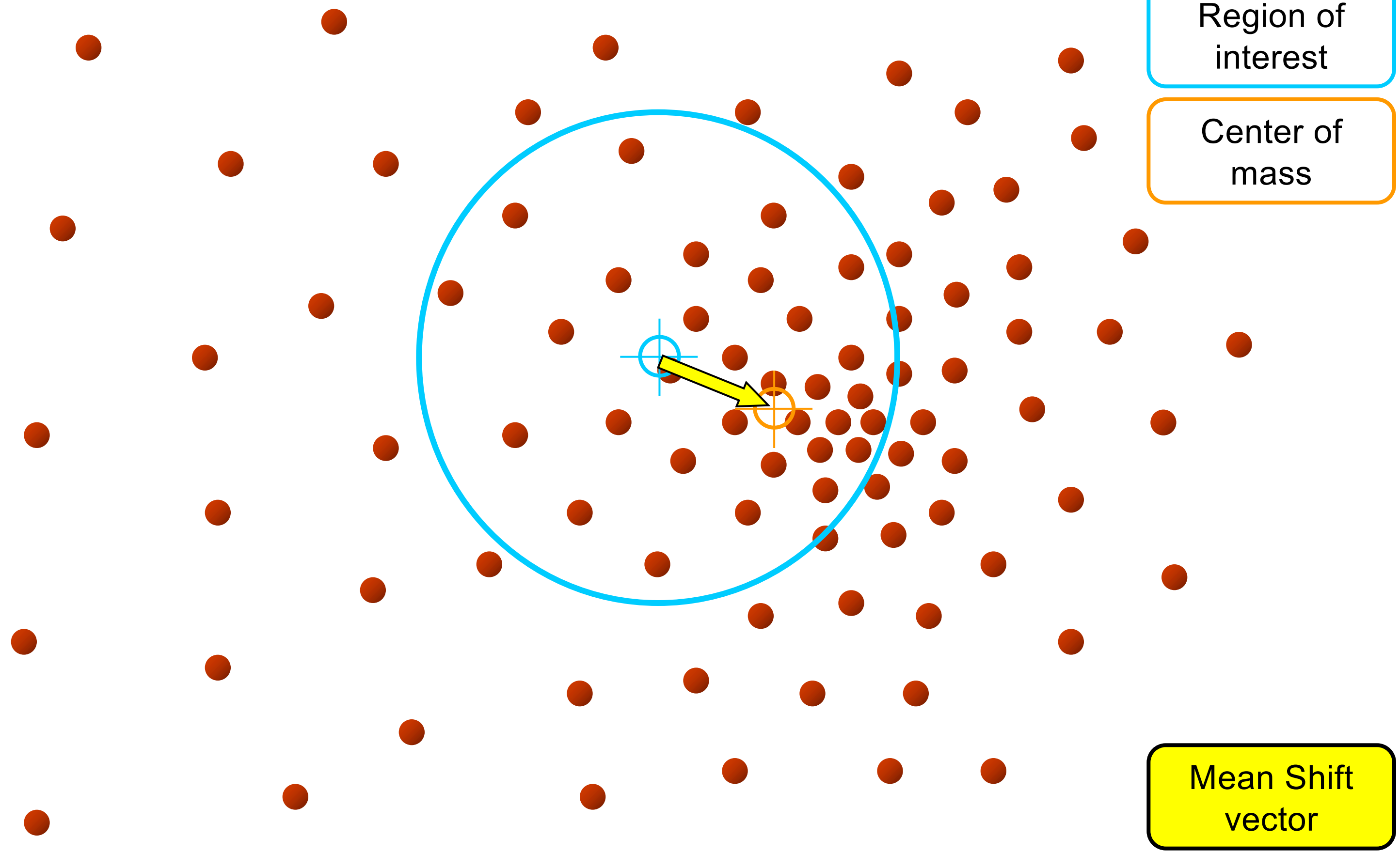
Slide by Y. Ukrainitz & B. Sarel

Mean shift



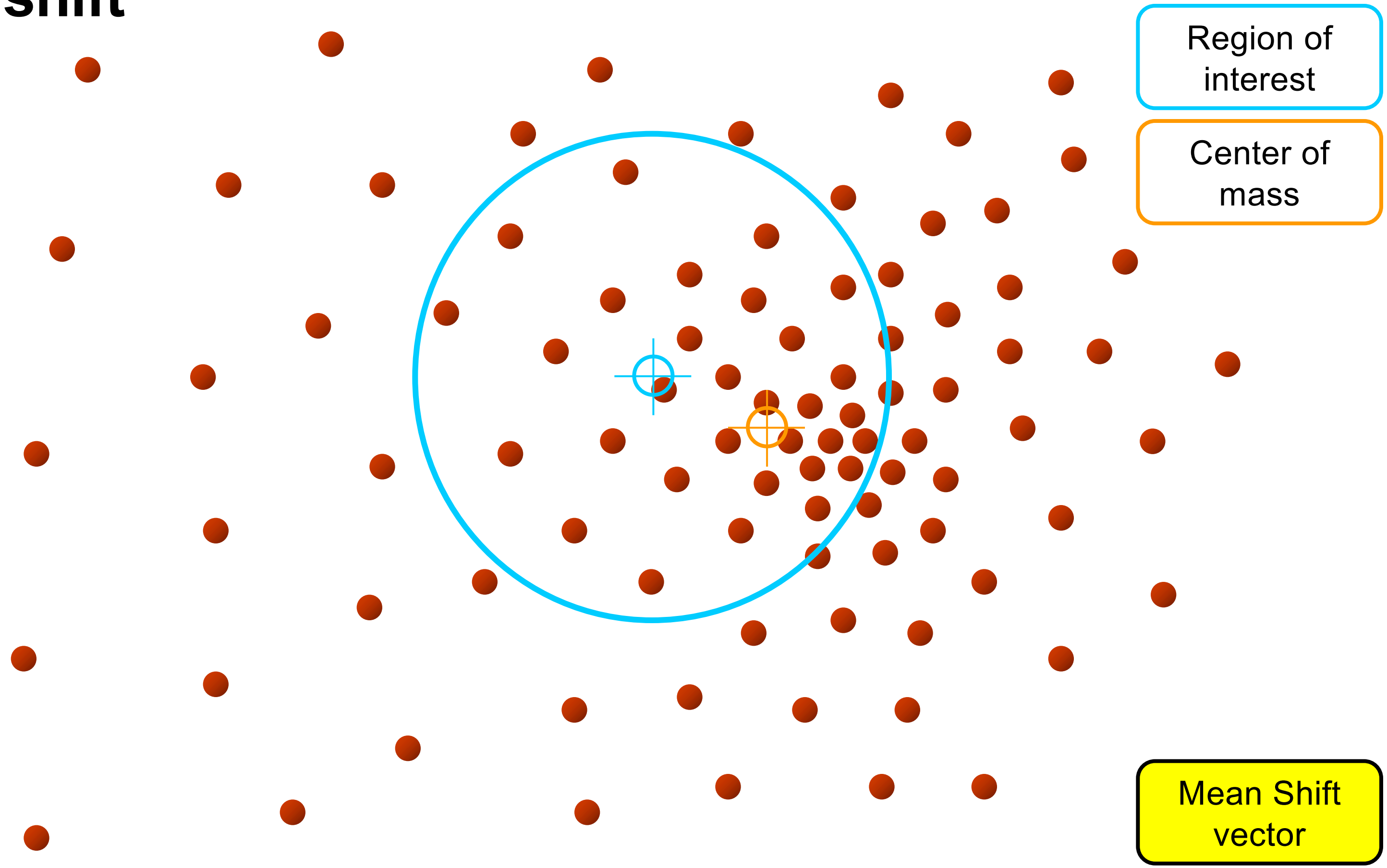
Slide by Y. Ukrainitz & B. Sarel

Mean shift



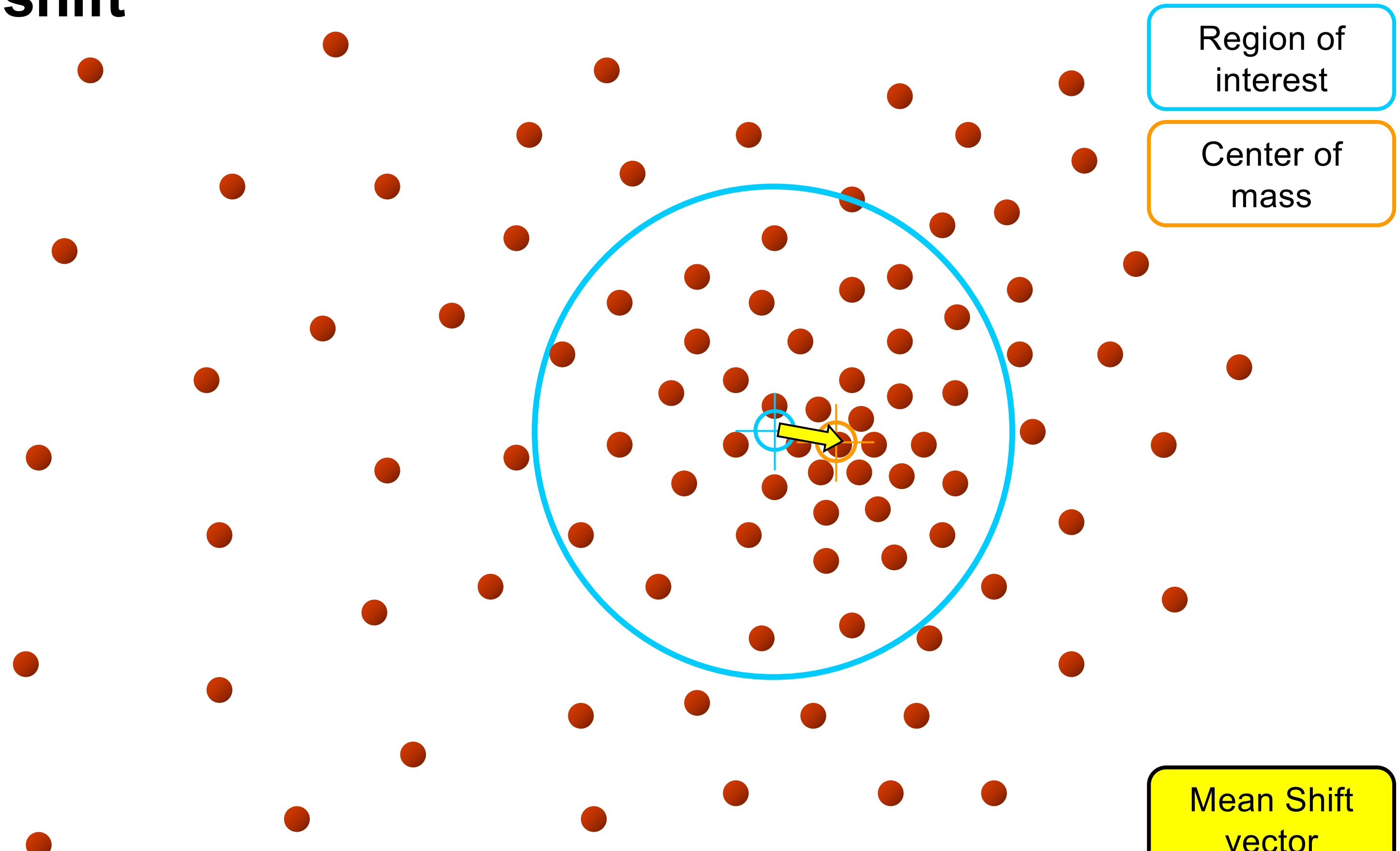
Slide by Y. Ukrainitz & B. Sarel

Mean shift



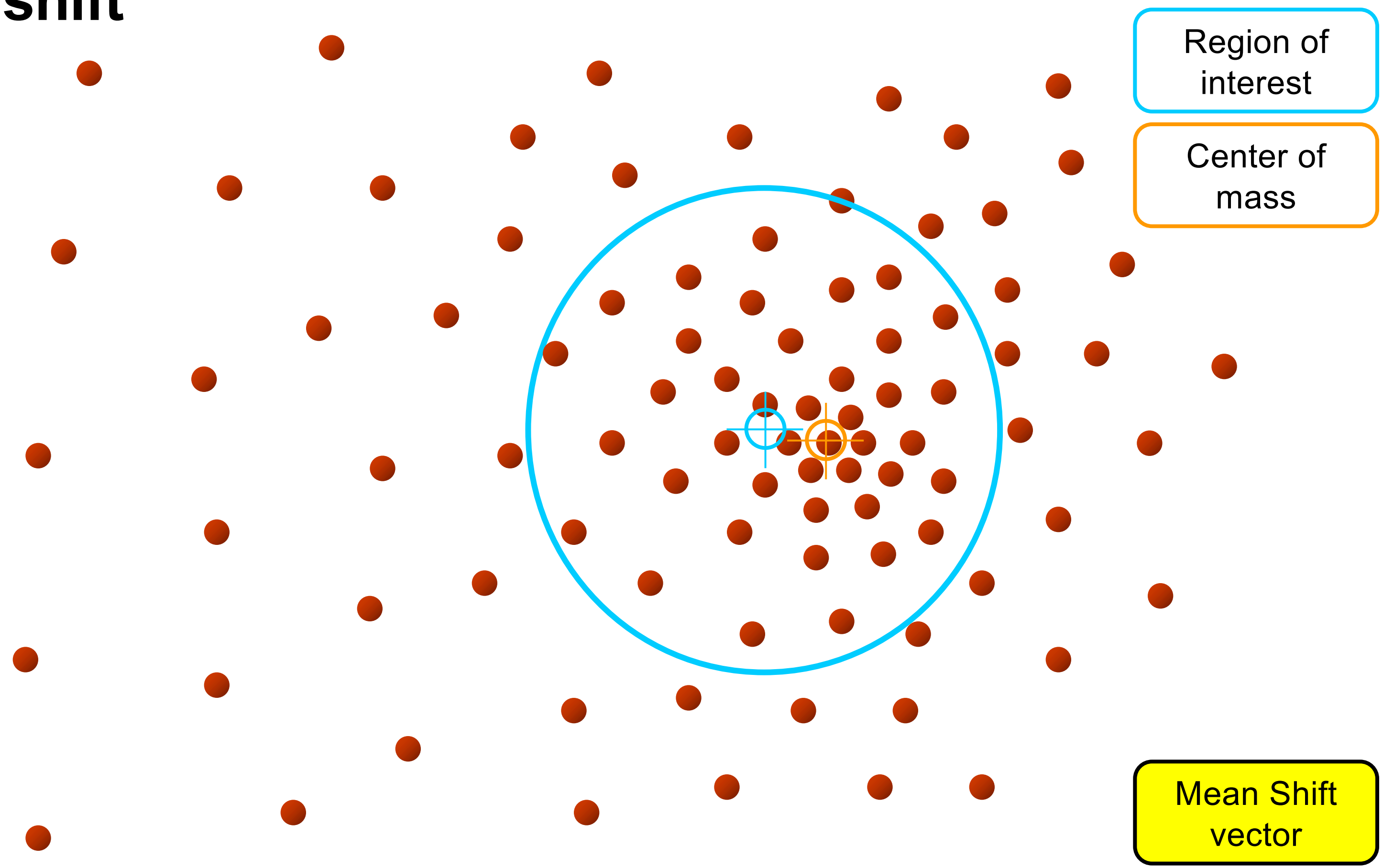
Slide by Y. Ukrainitz & B. Sarel

Mean shift



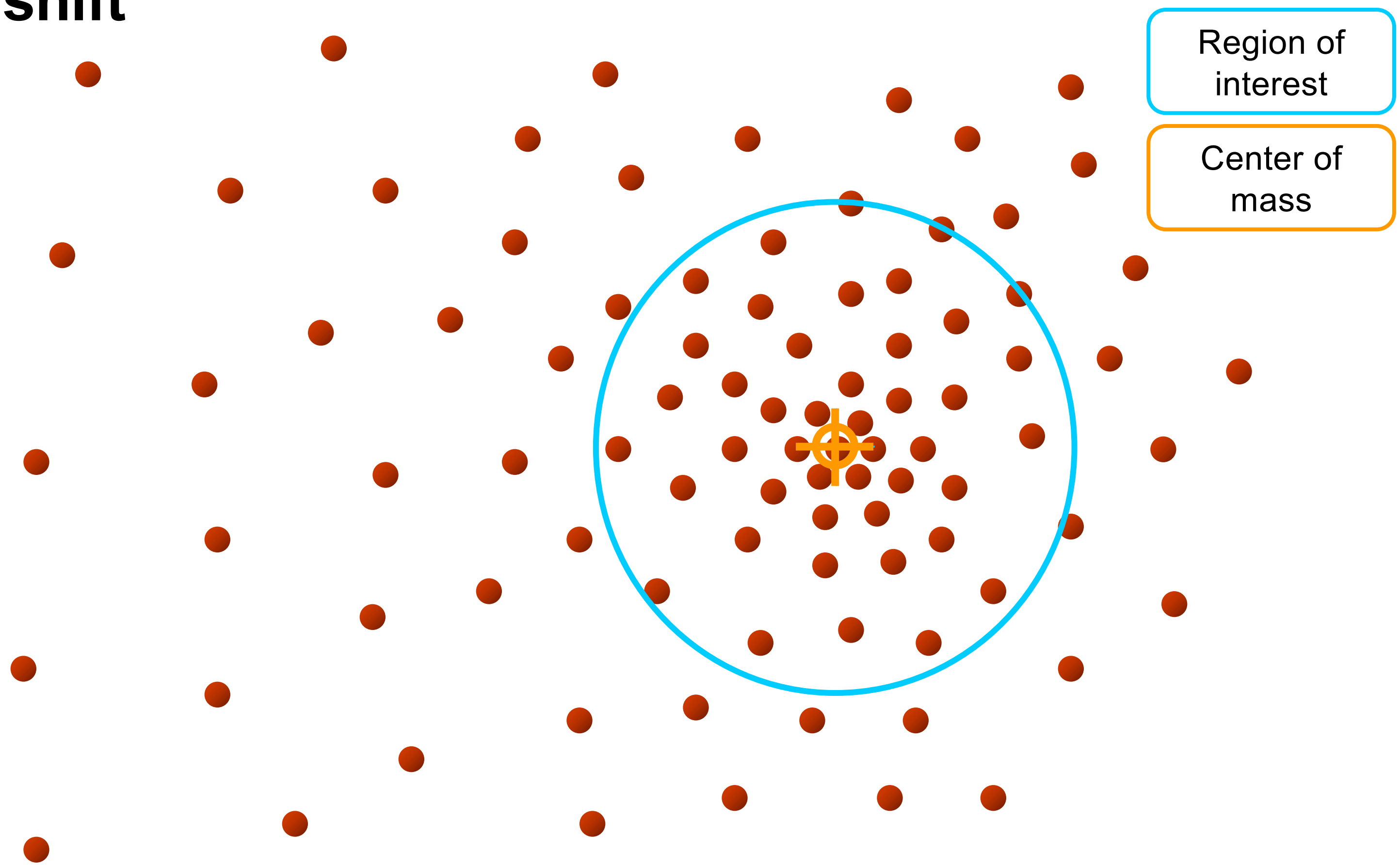
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

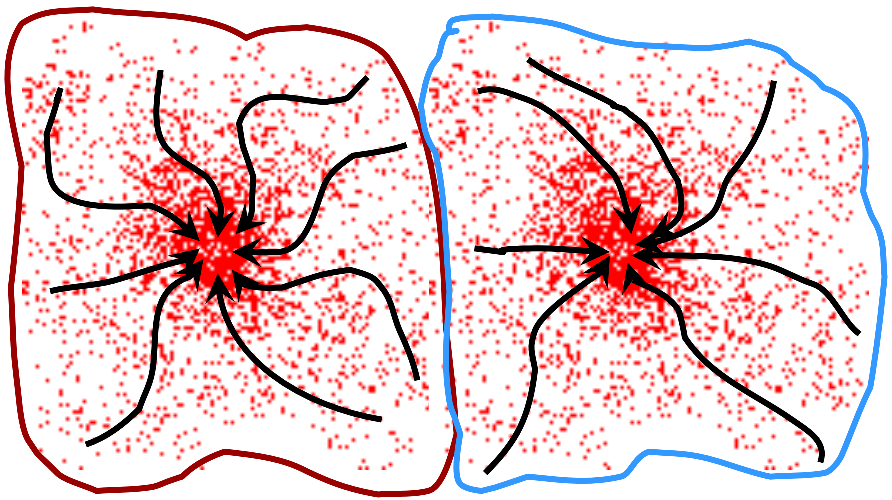
Mean shift



Slide by Y. Ukrainitz & B. Sarel

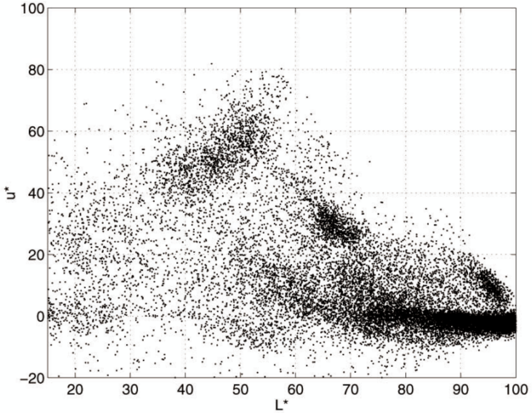
Attraction basin

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

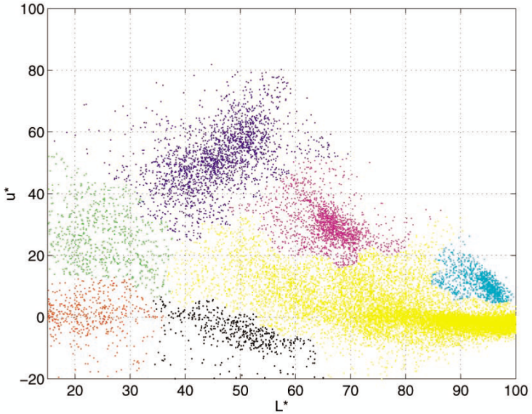


Slide by Y. Ukrainitz & B. Sarel

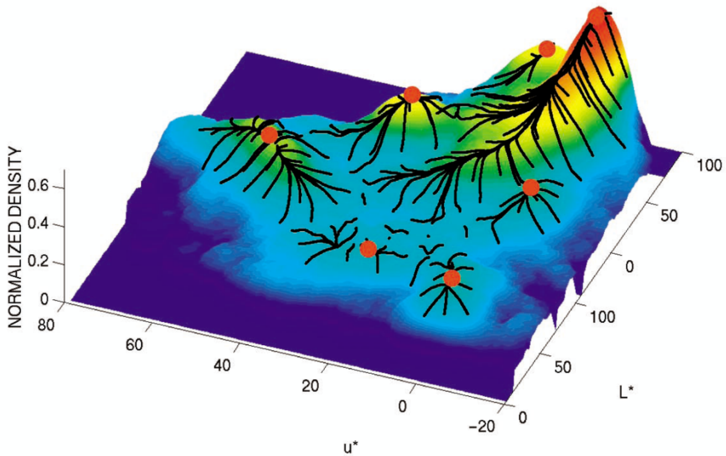
Attraction basin



(a)



(b)



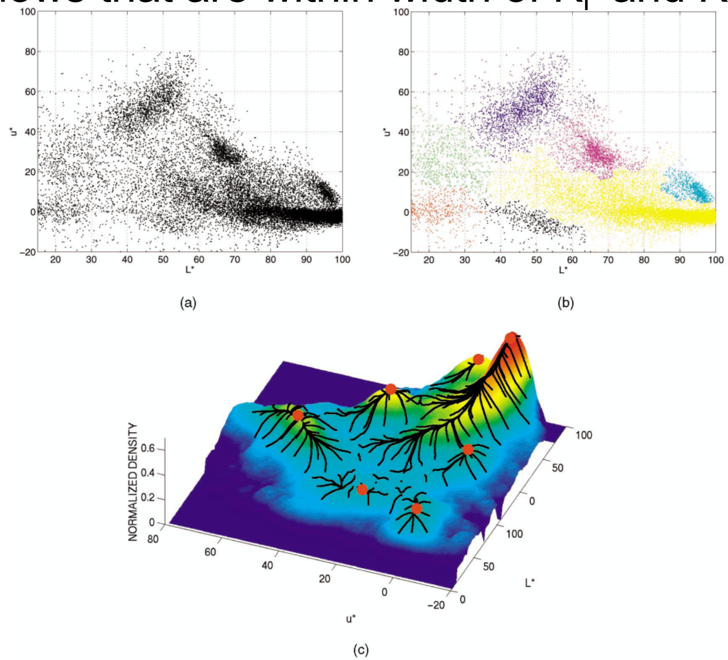
Mean shift clustering

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

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

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



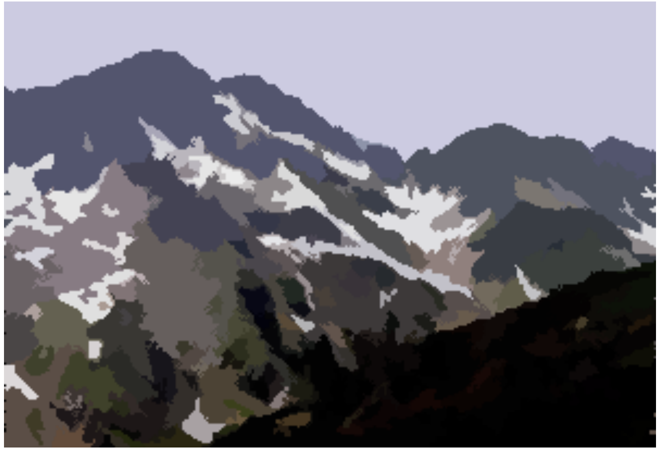
3D Point Cloud Segmentation by Mean Shift

Which space to apply Clustering? Is it your point cloud's 3D space?

What should be the dimensionality?

- dimensionality = 3 ? → X, Y, Z
- dimensionality = 6 ? → X, Y, Z, R, G, B
- dimensionality = 6 ? → X, Y, Z, L, U, V
- dimensionality = 6 ? → X, Y, Z, H, S, V
- dimensionality > 6 ? → X, Y, Z, H, S, V, gradients, texture, … ??

Mean shift segmentation results



Comaniciu and Meer 2002

Clustering - DBSCAN

- DBSCAN is a density-based clustering algorithm

- In density-based clustering:
  – we partition 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

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in International Conference on Knowledge Discovery and Data Mining, 1996.

Clustering - DBSCAN

Types of points:

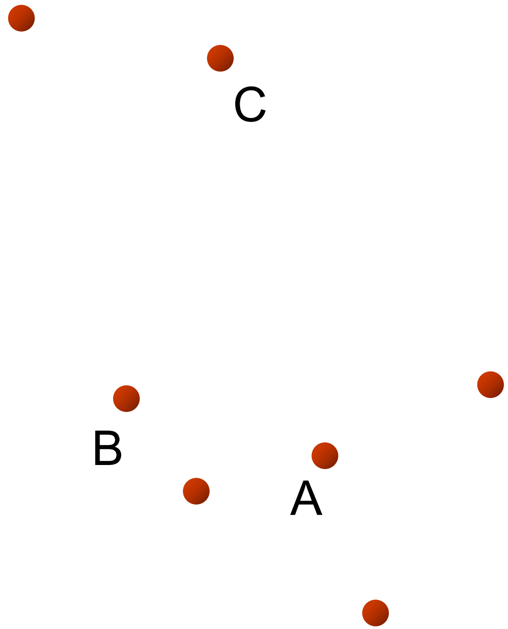
- A **core point** has more than a specified number of points (*MinPts*) within ε
- A **border point** has fewer than *MinPts* within ε, but is in the neighborhood of a core point.
- A **noise point** is any point that is not a core point or a border point.

Clustering - DBSCAN

Assume:

ε=1

MinPts=4



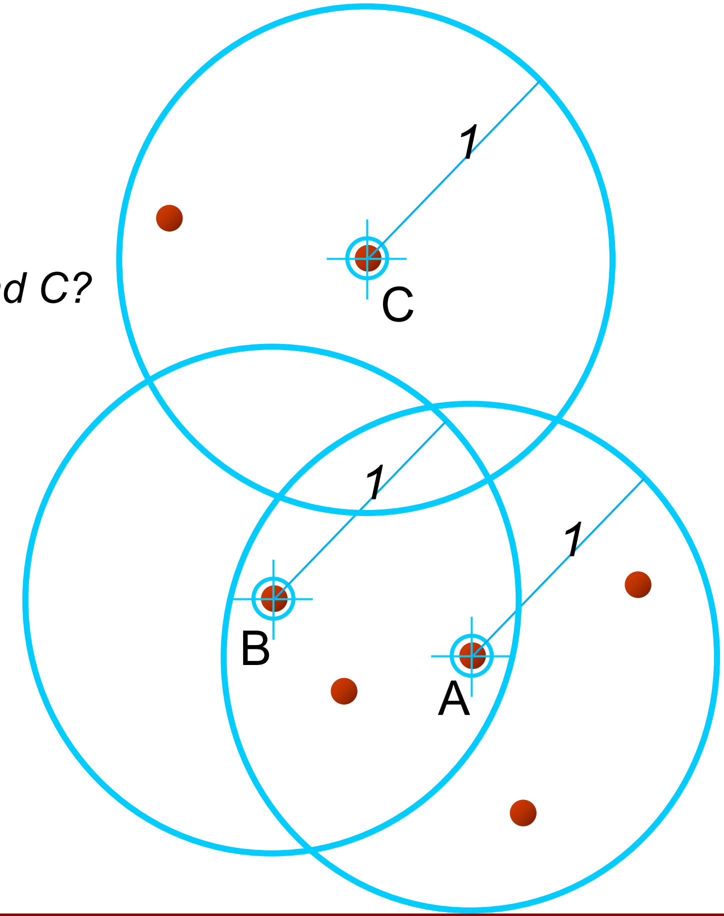
Clustering - DBSCAN

Assume:

$\varepsilon=1$

MinPts=4

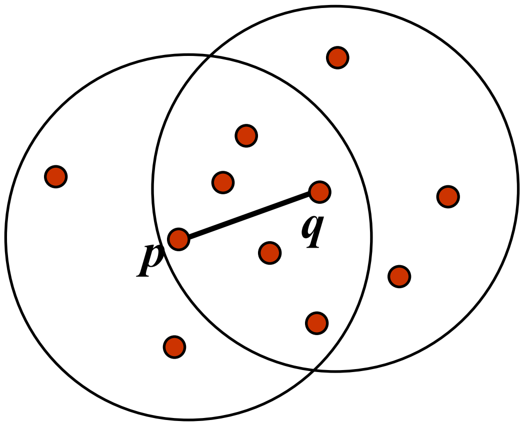
What is the type of points A, B and C?



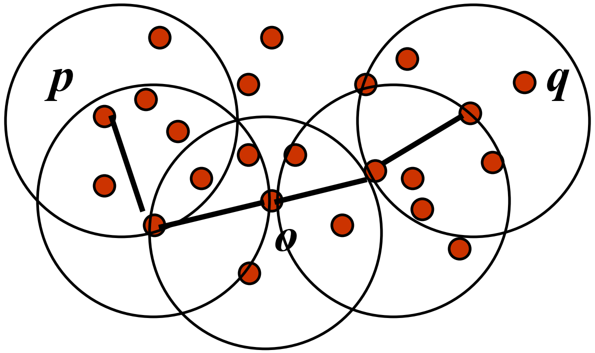
Clustering - DBSCAN

Some more definitions:

- **Density edge**
  - We place an edge between two core points *q* and *p* if they are within distance *ε*



- **Density-connected**
  - A point *p* is density-connected to a point *q* if there is a path of edges from *p* to *q*



Clustering - DBSCAN

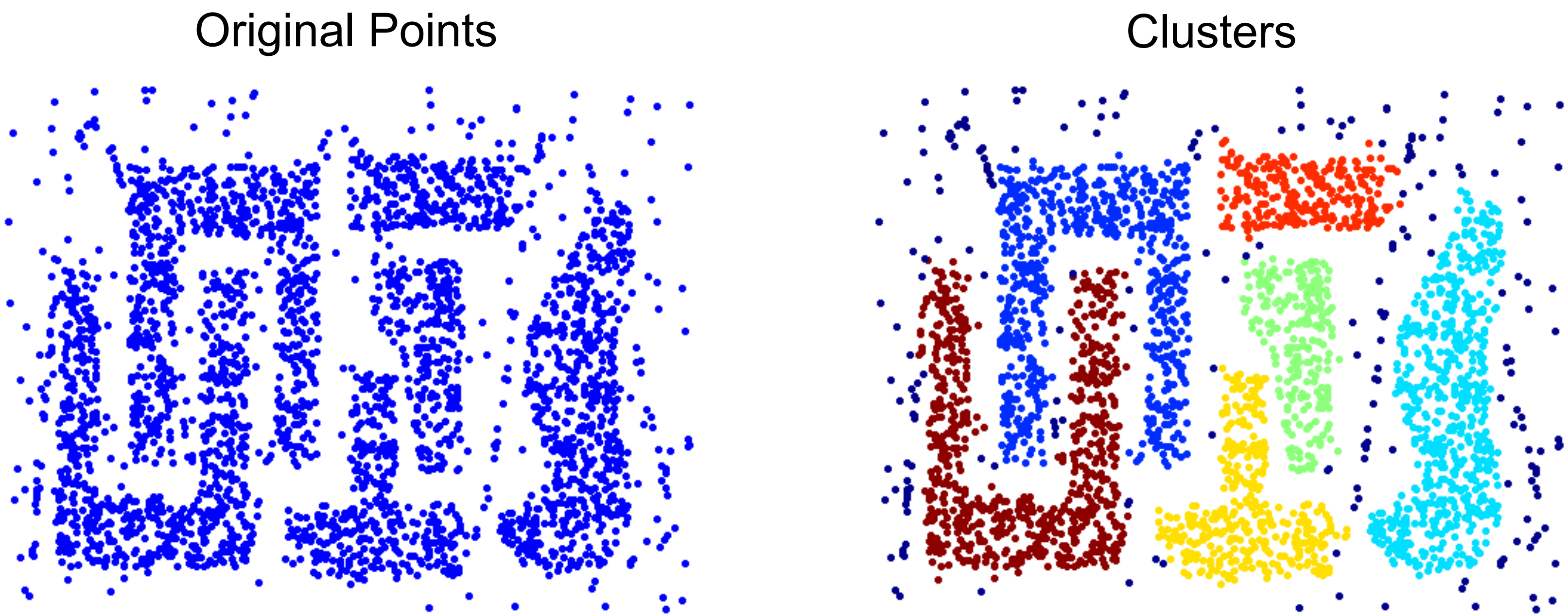
DBSCAN algorithm

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 points to the cluster of the closest core point.

A cluster contains:

- all core points that can be reached by following a sequence of density-connected core points
- border points that are closest to one of the above-clustered core points

Clustering - DBSCAN



Figures from: Jing Gao

Clustering - DBSCAN

Pros

- A cluster can have non-convex shape
- No need to define the number of clusters
- Resistant to noise

Cons

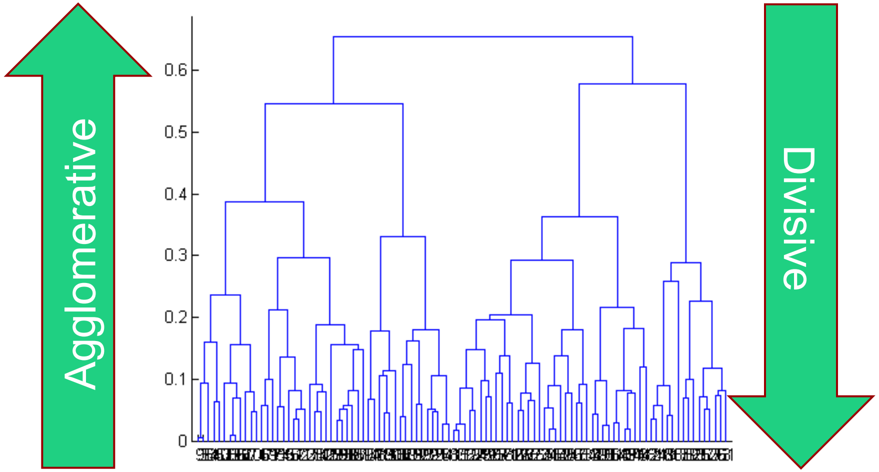
- Some points (noise points) are not assigned to any cluster (is this necessarily bad?)
- Need to define ε and *MinPts*
- Problems with point clouds that contain regions of varying densities

Clustering – Hierarchical Clustering

Builds a **hierarchy of clusters**, *i.e, clusters that consist of other smaller clusters*.

The 2 types of hierarchical clustering:

- **Agglomerative**: This is a "**bottom-up**" approach. *Each point starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy*.
- **Divisive**: This is a "**top-down**" approach. *All points start in one cluster, and splits are performed recursively as one moves down the hierarchy*.



Clustering – Hierarchical Clustering

Agglomerative / Bottom-up hierarchical clustering
is based only on distance (similarity) between the points.

Algorithm steps:

1. Start by assigning each point to its own cluster, obtaining as many clusters as points.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster.

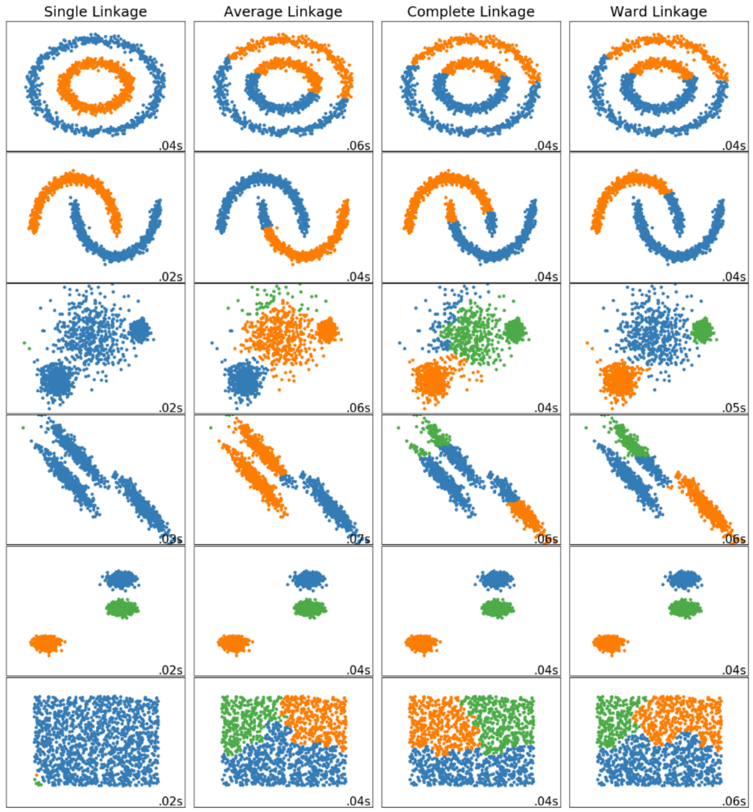
Agglomerative clustering is much more popular than Divisive hierarchical clustering.

Clustering – Hierarchical Clustering

How is distance (similarity) between clusters assessed?

Linkage methods:

- **Single-link**: the distance between two clusters is equal to the minimum distance from any member of one cluster to any member of the other cluster.
- **Complete-link**: the distance between two clusters is equal to the maximum distance from any member of one cluster to any member of the other cluster.
- **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.
- **Ward-link:** the distance between two clusters is equal to the sum of squared differences within any member of one cluster to any member of the other cluster.



https://scikit-learn.org/stable/modules/clustering.html

Clustering – Hierarchical Clustering

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

- Consists of many Π-shaped lines that connect data points in a hierarchical tree.
- 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

Outline

- What are Clustering and Regression? A taxonomy of ML algorithms
- Why is ML important for Perception of Autonomous Systems?

- Regression
- Clustering
  - k-means
  - Mean Shift
  - DBSCAN
  - Hierarchical Clustering

- Summary

Summary

- Machine Learning techniques can provide useful tools for:
  – finding the underlying structure or
  – fitting models
  to point clouds.

- Regression can fit models of lines, planes, non-linear surfaces

- Clustering is a very powerful tool.
  – No need for labeled data, as it belongs to the unsupervised learning algorithms
  – Many different methods for clustering with different pros and cons.

- 3D point clouds might need to be expanded in higher dimensional spaces to consider additional information (e.g. color, gradients, texture, …)
  – Machine Learning techniques still applicable in these higher dimensional spaces!

Source: https://en.wikipedia.org/wiki/Otsu%27s_method

Lazaros Nalpantidis

3D Point Cloud Processing - Clustering & Regression