Mean Shift Algorithm

Presented By:

Eldridge Fernandes

Rakkshab Iyer

Ashvinjeet Singh

Instructor: Dr. Yaseer Alginahi

Date: November 3rd, 2023

Table of Contents

What is Clustering?

- Clustering is the process of finding a pattern in a collection of unlabeled data based on the similarity feature.
- It reduces a set of infinite values into finite values
- Similarity Feature measures similarity between examples and compresses them into a relevant cluster with its respective cluster ID
- It helps deals with challenges like outliers, missing data, and categorical versus numerical data

Cluster formation of similar features

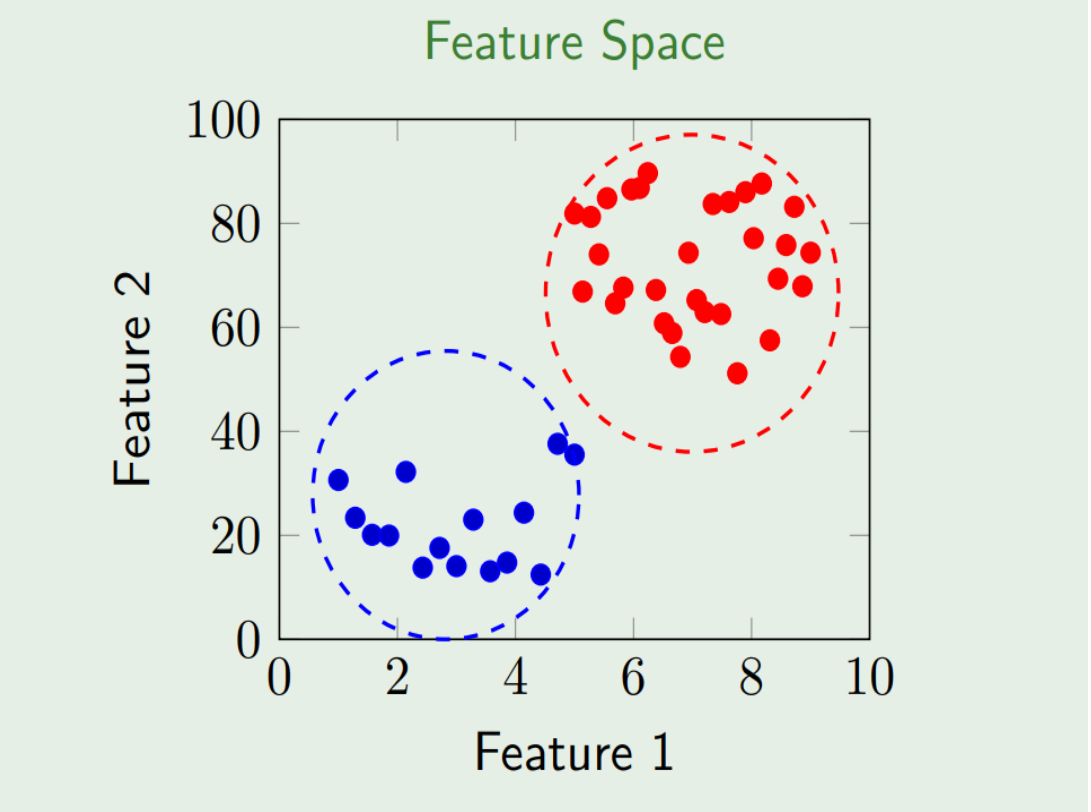


Image Source: Taken from [1]

Types of Clustering

	Supervised	unsupervised
Discrete	• Classification ○ K-NN ○ Logistic Regression ○ SVM	• Clustering ○ K-means ○ Mean-shift ○ GMM • Association Analysis ○ Apriori
Continuous	• Regression ○ Linear regression ○ Polynomial regression ○ Decision Trees ○ Random Forests	• Dimensionality Reduction ○ PCA ○ SVD ○ LDA

Image Source: Taken from [1]

Challenges in Clustering

- Algorithm should scale to your dataset
- Number of clusters may be unknown
- Datasets have millions of records and clustering algorithms will not be able to scale it efficiently as the runtime increases as the square of the number of examples ,i.e., $O(n^2)$

Mean-Shift Algorithm

- It is a non-parametric feature-space mathematical analysis technique
- Mean Shift algorithm calculates maxima and minima of a density function
- Assigns data-points to the clusters iteratively by shifting points towards the mode (highest density data points)
- The kernel is shifted iteratively to a higher density region until convergence and the shift is defined by a mean shift vector

Gradient Ascent

- Gradient ascent is based on the principle of locating the greatest point on a function and then moving in the direction of the gradient
- In simpler terms, moving uphill, from a region of lower density to a higher density point

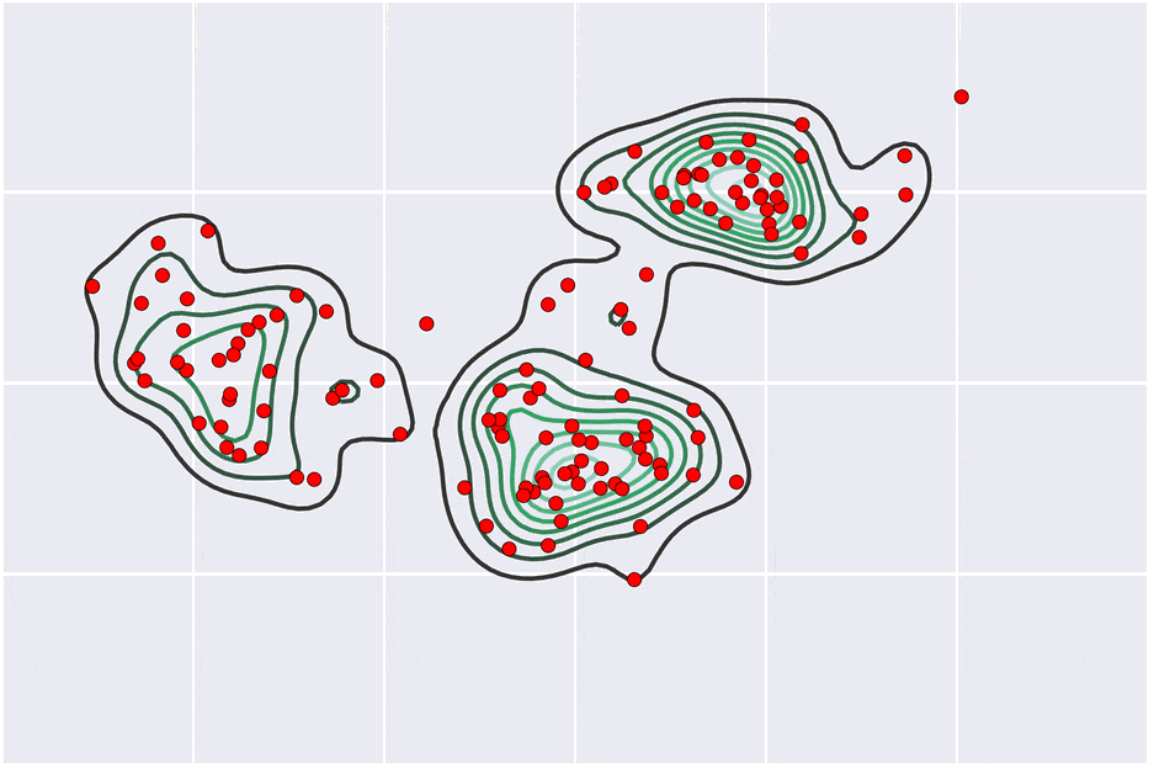


Image Source: Taken from [1]

Basic Idea

- A flexible approach to density estimation ("how many data points are in a certain region?")
- Find the local modes of this density
- All points that "belong" or "lead" to the same mode form a cluster

How Does the Mean-Shift Algorithm Work?

Initialization of a Window

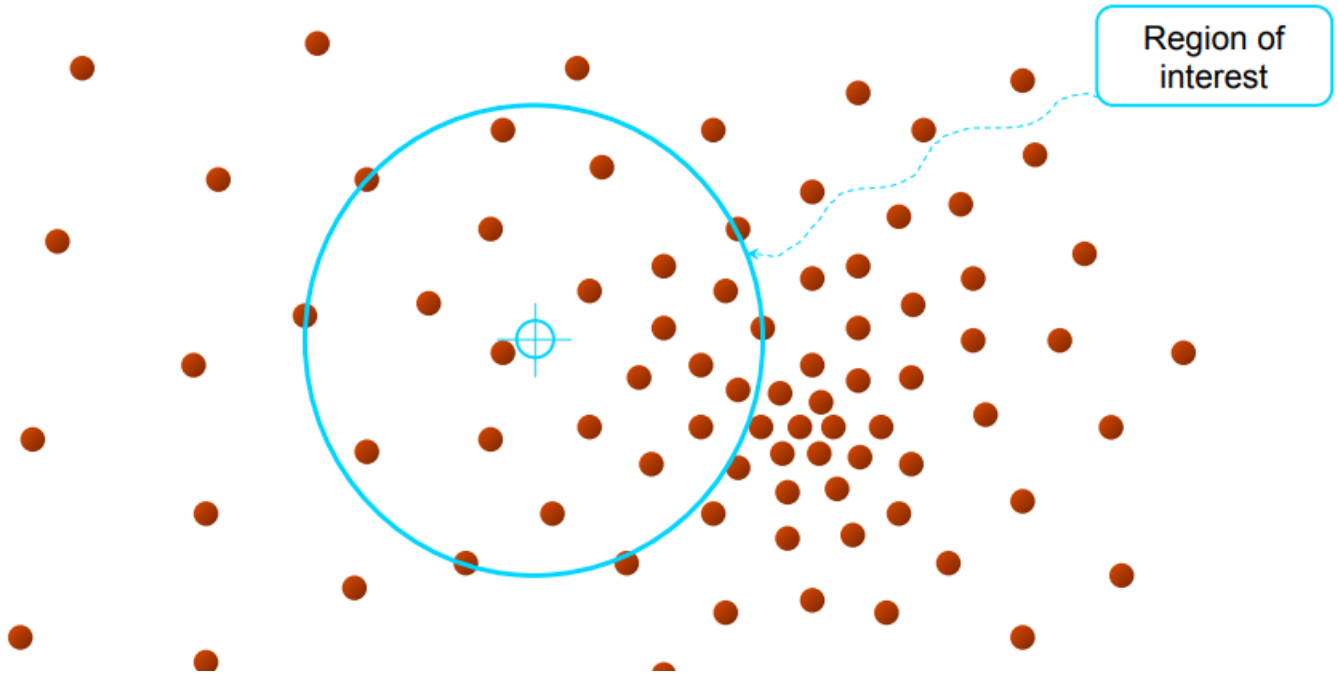


Image Courtesy: https://shorturl.at/exGT2

- Initialize a random seed (choose any data point)
- With the coordinates of this point as the centroid, consider an imaginary window (size=d)
- Window size should be optimal; not too large, not too small

Mean-Calculation Phase

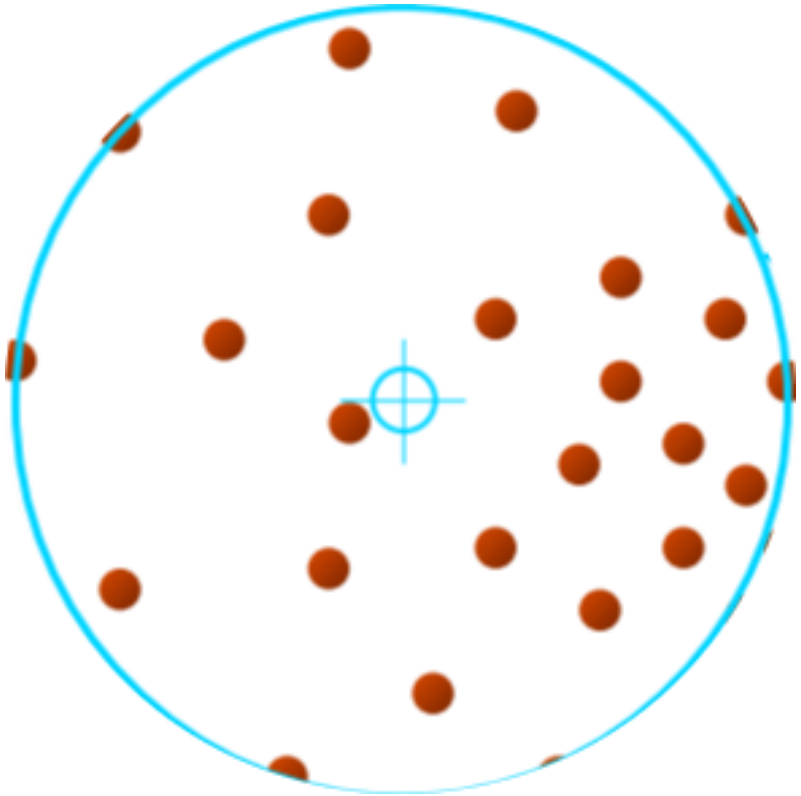


Image Courtesy: https://shorturl.at/exGT2

- Calculate the mean coordinates considering all of the data points lying inside the window
- This can be a simple mean or weighted mean

Weighted Mean Calculation

- Weighted mean is the mean calculated by assigning weights to the neighboring data based on their distance from the sample point.
- Neighbors near the point will have a larger weight as compared to the ones farther away.
- Usually weighted mean is preferred since it reduces the impact of the data points farther away from the sample point.

Kernel Density Estimation

- Kernel Density Estimation can be defined as a technique that estimates a real valued function as the weighted average of neighboring observed data.
- The weights (kernels) assigned to the features are calculated based on this method.

Weight/Kernel Calculation

Variable Summary:

- x_s – Coordinates of the sample point
- x_n – Coordinates of the neighbors
- d – Size of Window
- $\bar{x}$ – Normalized Distance
- $K(x)$ – Kernel Function (Gaussian)
- x_C – Centroid Coordinates

Weight/Kernel Calculation (cont.)

- The most basic method of weight calculation is with the help of a gaussian function:

$$K(x) = \exp\left(-\frac{1}{2} * \|\bar{x}\|\right)$$

- Where, $\bar{x}$ is the normalized distance of the sample point (x_s) from the neighbor (x_n)

$$\bar{x} = \frac{x_s - x_n}{d}$$

Centroid Calculation

- The weighted x-coordinate of a single data can be given as:

$$\mathrm{K(x)} = \exp\left(-\frac{1}{2} * \left\|\frac{x_s - x_n}{d}\right\|\right)$$

- The centroid can be calculated by,

$$x_C = \frac{\sum_n \left(x_n * K\left(\overline{x_n}\right)\right)}{n}$$

Centroid Calculation (cont.)

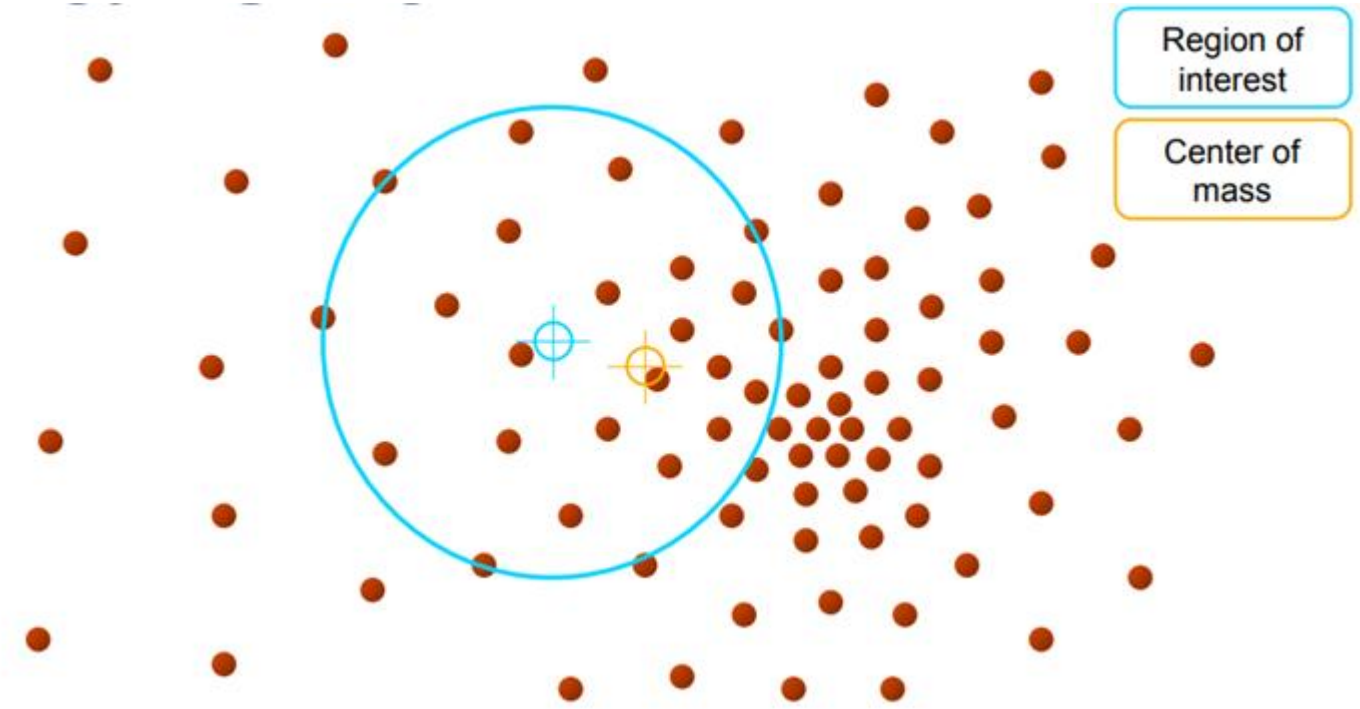
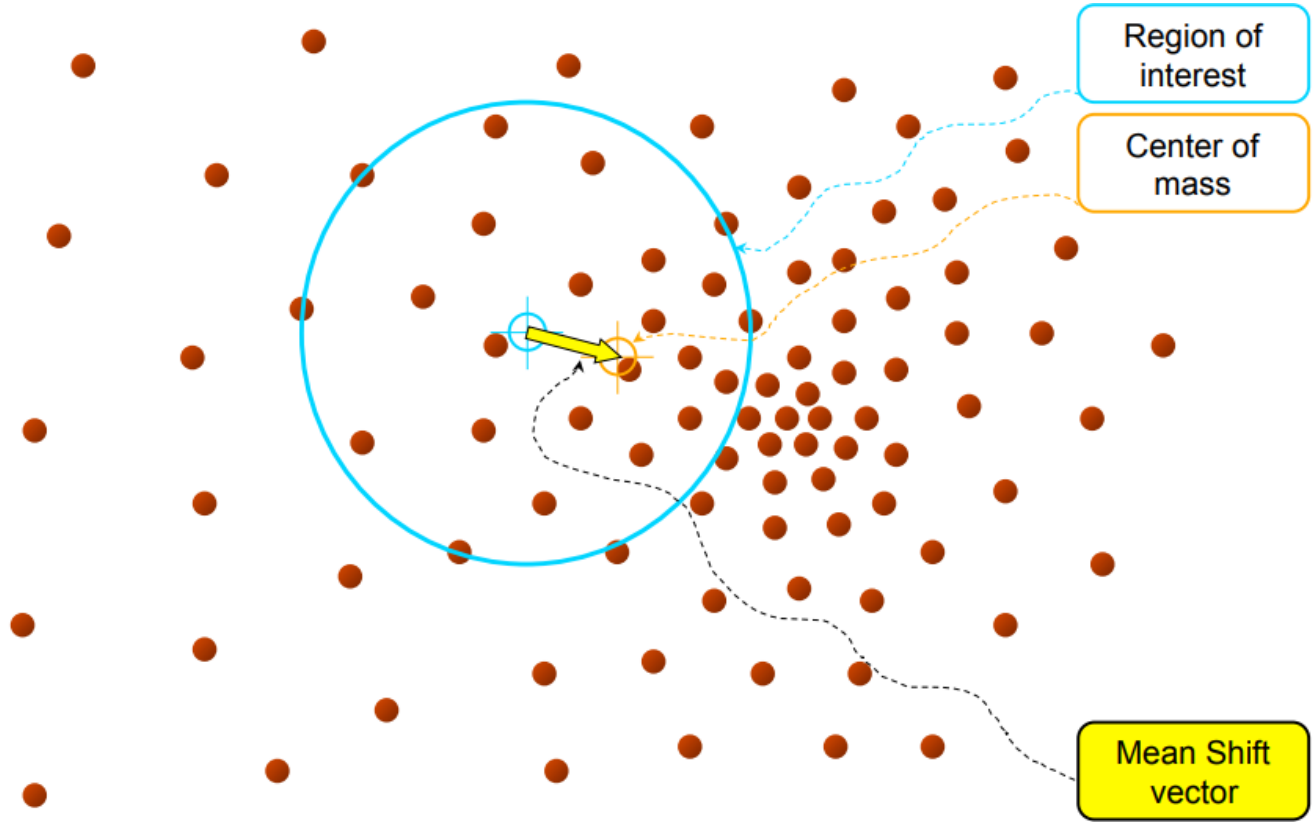


Image Courtesy: https://shorturl.at/exGT2

- Similar procedure is followed to calculate the centroids for all other coordinates
- Higher dimensional data would require much more complex calculation and suitable hardware for execution

Shifting the Window



- The window is shifted to these centroid coordinates so as to make these coordinates the new center
- “Mean-Shift Vector” signifies the window shift
- It is a unit vector, representing the direction of shift

Image Courtesy: https://shorturl.at/exGT2

Shifting the Window (cont.)

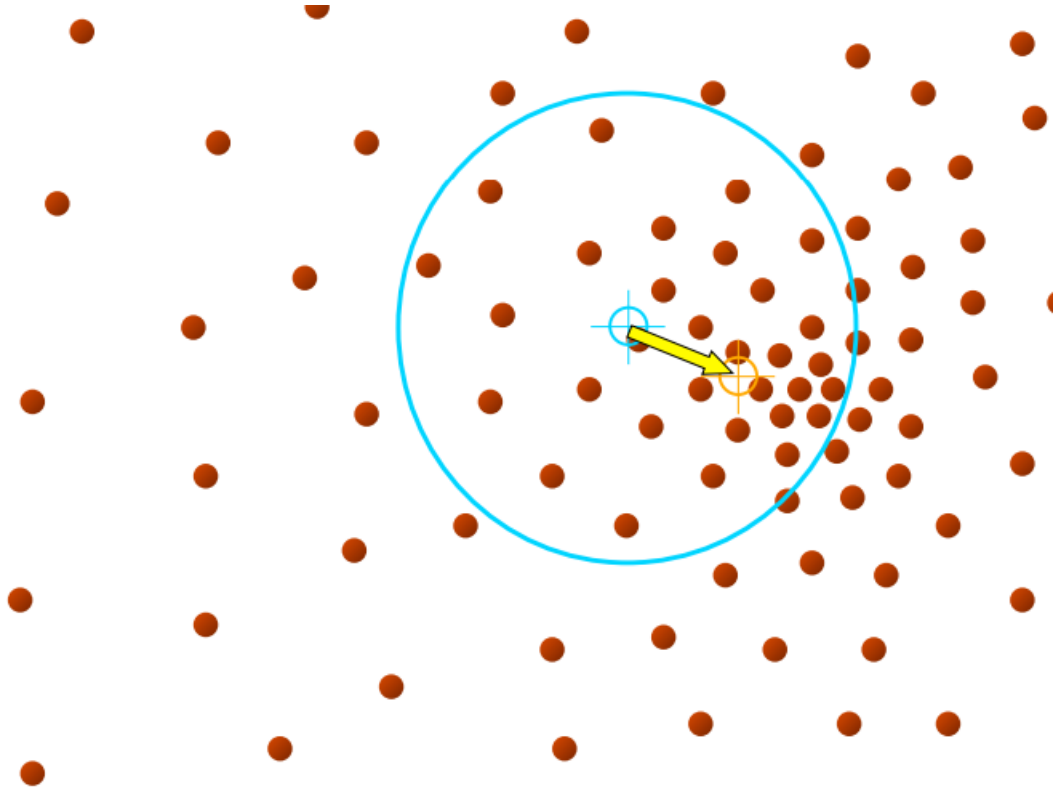


Image Courtesy: https://shorturl.at/exGT2

- Again, the new centroid for the shifted window is calculated
- The window is shifted toward this centroid
- This process repeats until convergence

When to Stop?

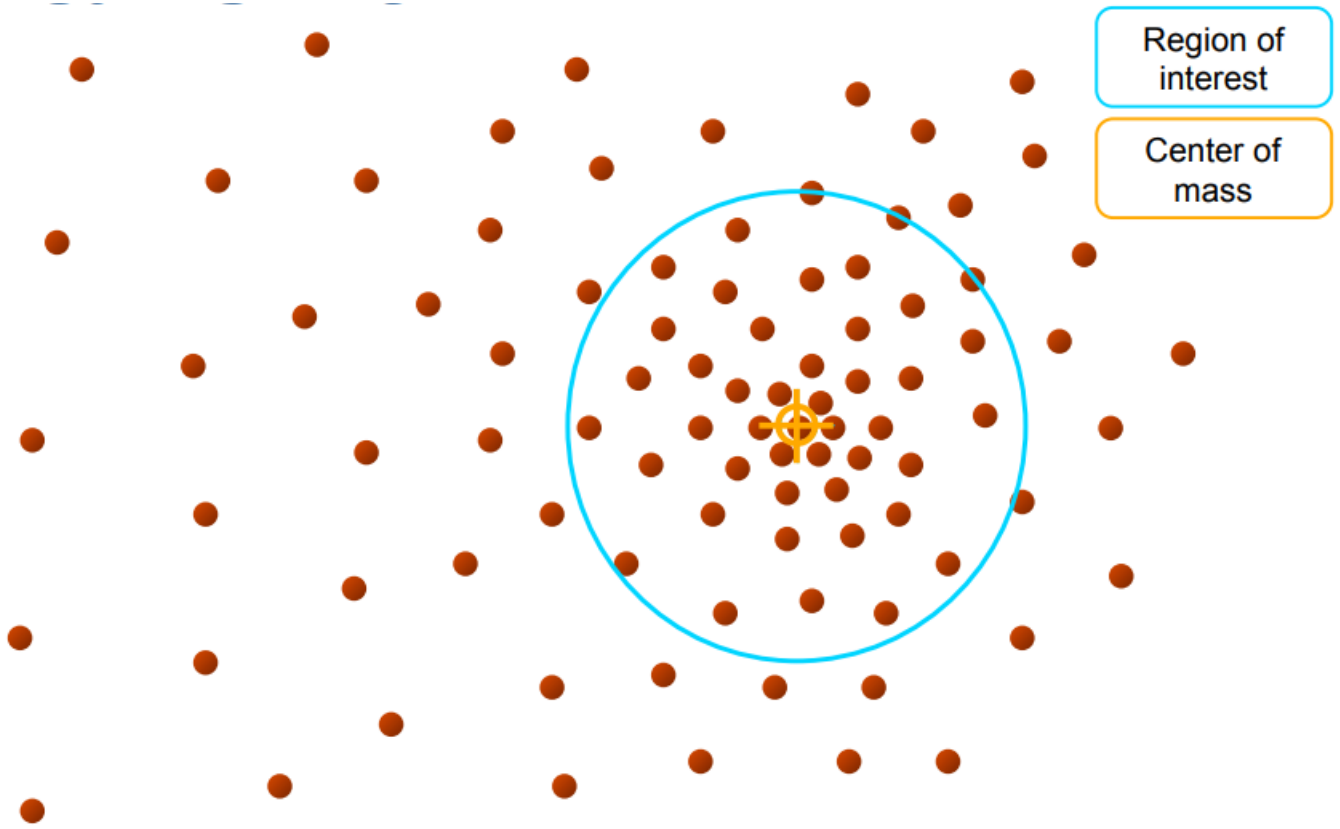


Image Courtesy: https://shorturl.at/exGT2

- At some point, the window's centroid stops changing or more accurately, the changes are insignificant
- It can be said that, the window has reached a point with high data density (local maxima) and this point can be treated as a cluster label
- This is known as convergence and signifies end of the loop

Algorithm Summary

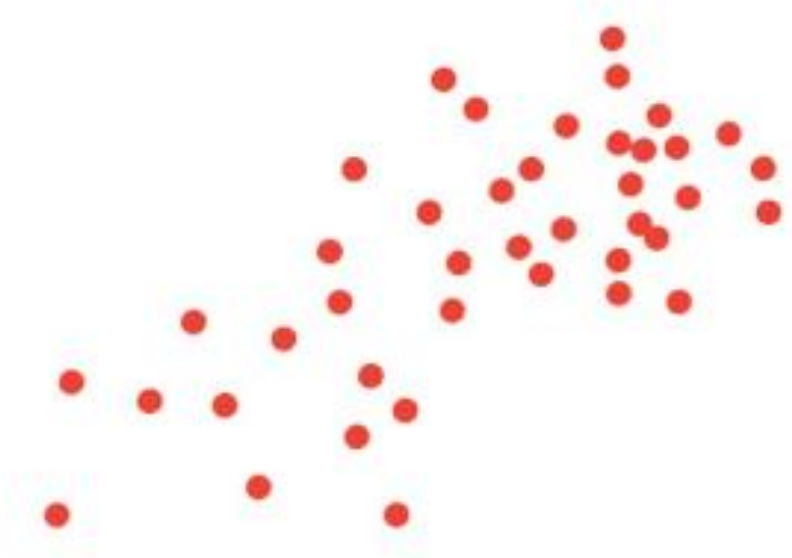


Image Courtesy: https://shorturl.at/aditM

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

Pros and Cons

Pro's	Con's
No model assumption is made like K-means	Doesn't perform well in high dimensional cases
Application-independent tool and its model doesn't assume prior shape	Computational expensive and no control on specific number of clusters
The number of clusters is determined by the algorithm with respect to the data	Cannot differentiate between meaningful and meaningless modes
Local minima issues like K-means are averted	Inapproriate window size can cause modes to be merged or shallow modes
Robust to Outlier problems	Often requires adaptive window size

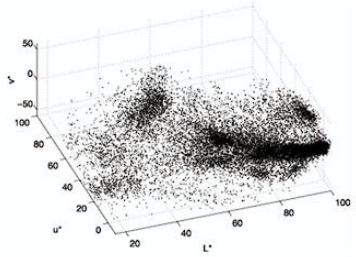
Comparison with Other Clustering Methods

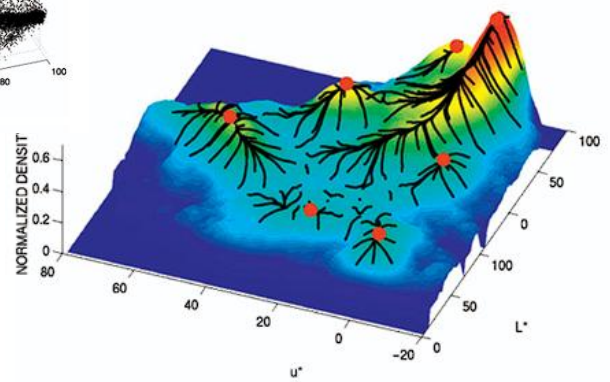
Algorithm	Description	Parameters	Pros	Cons
K-Means	Partitions the data into K clusters, each represented by the mean of the samples in the cluster.	n_clusters, init, n_init, max_iter	Simple and fast, works well on large datasets.	Assumes clusters are spherical and equally sized, which may not always be the case.
Spectral Clustering	Uses the eigenvectors of a similarity matrix to reduce the dimensionality of the data before clustering in a lower dimensional space.	n_clusters, eigen_solver, random_state, n_init, gamma, affinity, n_neighbors, eigen_tol, assign_labels	Can find arbitrarily shaped clusters.	Does not work well with large datasets or when clusters have different sizes.
Gaussian Mixture Models (GMM)	Models the data as a collection of Gaussian distributions.	n_components, covariance_type, tol, reg_covar, max_iter, n_init, init_params, weights_init, means_init, precisions_init	Works well with data that is distributed as a mixture of Gaussians.	Requires the number of Gaussian components to be specified.
Agglomerative Hierarchical Clustering	Builds a hierarchy of clusters by merging or splitting existing clusters.	n_clusters, affinity, memory, connectivity, compute_full_tree, linkage, distance_threshold	Does not require the number of clusters to be specified, can produce a hierarchy of clusters.	Can be slow on large datasets, sensitive to the choice of linkage criteria.
Mean Shift Clustering	Identifies blobs in a smooth density of samples.	bandwidth, seeds, bin_seeding, min_bin_freq, cluster_all, n_jobs	Does not require the number of clusters to be specified.	Computationally expensive on large datasets.

Image Source: https://shorturl.at/ekGPQ

Applications of Mean-Shift Algorithm

- Video Object Tracking
- Anomaly detection
- Image segmentation
- Image denoising
- Spatial Data Analysis
- Document categorization and topic modelling
- Remote sensing





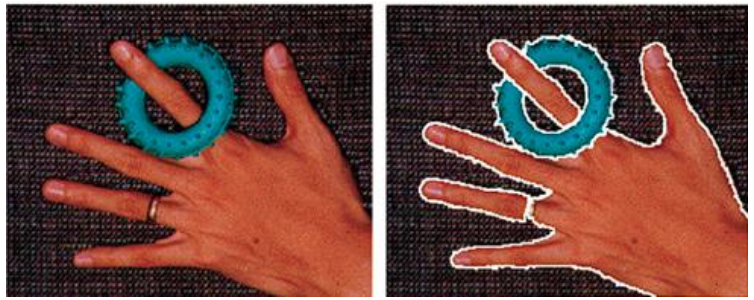


Image Source: https://shorturl.at/ekGPQ

Implementation of Mean shift Algorithm

Implementation of a machine learning model to precisely segment features and objects in high-resolution aerial photos taken by unmanned aerial vehicles (UAVs) in order to facilitate environmental monitoring, object detection, and precise land analysis
Code reference:-https://tinyurl.com/Google-collab
Article reference:-Medium

```
import numpy as np
import pandas as pd
import cv2
import collections
import seaborn as sns
import matplotlib.pyplot as plt
from skimage import filters
from skimage import util
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import MeanShift, estimate_bandwidth
import warnings
from google.colab import files
warnings.filterwarnings("ignore")
```

Here we have imported all the libraries required for the implementation

```
[ ]  uploaded=files.upload()
```

```
Choose Files  No file chosen        Upload widget is only available when the cell has been e
Saving UAV_image.jpg to UAV_image.jpg
```

```
img=cv2.imread('UAV_image.jpg')
img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7a905757f160>
```

Imported a local image to collab notebook of a drone shot a field

Mean shift builds on kernel density estimation. Kernel density estimation estimate PDF(probability density function) for a set of data. t works by placing a kernel on each point in the data set. A kernel has a mathematical algorithm for weighting function generally user in convolution. There are many different kernels. But the most popular one is the **Gaussian** kernel.

```
bandwidth_1 = estimate_bandwidth(nd_1, quantile=.04, n_jobs=-1)
bandwidth_2 = estimate_bandwidth(nd_2, quantile=.04, n_jobs=-1);
```

Here are we calculating the best bandwidth which is specified by feature selection

Here the bandwidths are calculated as 3 and 5, respectively. calculated that 0.13559591178830535 and 0.034407035660636826.

Next step is clustering,
cluster_all parameter can be set False. That means that if the dataset have noise depends on clusters, these point never passing values. That properties is quite useful if you want to eliminate anomaly of the image.

```
ms_1 = MeanShift(bandwidth = bandwidth_1 , n_jobs=-1, bin_seeding=True, cluster_all=True).fit(nd_1)
ms_2 = MeanShift(bandwidth = bandwidth_2 , n_jobs=-1, bin_seeding=True, cluster_all=True).fit(nd_2)
```

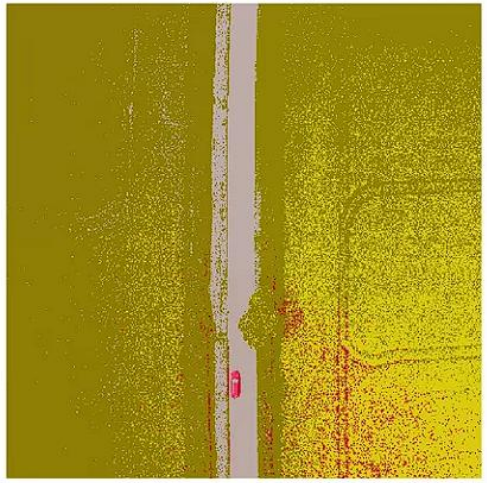
Bin seeding helps to speed up the algorithm on a backend

```
plt.figure(1)
plt.clf()
plt.axis('off')
plt.title('Original image', loc='center')
plt.imshow(image)
plt.figure(2)
plt.clf()
plt.axis('off')
plt.title('Pixels without their location ({} colors, Mean-Shift)'.format(len(ms_2.cluster_centers_)), loc='center')
plt.imshow(recreate_image(ms_1.cluster_centers_[:, 2:], ms_1.labels_, width, height));
plt.figure(3)
plt.clf()
plt.axis('off')
plt.title('Pixels without their location ({} colors, Mean-Shift)'.format(len(ms_2.cluster_centers_)), loc='center')
plt.imshow(recreate_image(ms_2.cluster_centers_, ms_2.labels_, width, height));
```

Here the final code to plot all the images with different bandwidths

Original image

Pixels with their location image (8 colors, Mean-Shift)



Pixels without their location (23 colors, Mean-Shift)

References

[1] Robust Analysis of Feature Space: Color Image Segmentation," by D. Comaniciu and P.Meer, CVPR 1997, pp. 750-755.

[2] https://developers.google.com/machine-learning/clustering/overview

[3] https://prutor.ai/clustering-algorithms-mean-shift-algorithm/

[4] https://medium.com/@sina.nazeri/comparing-the-state-of-the-art-clustering-algorithms-1e65a08157a1

[5] https://medium.com/@sina.nazeri/comparing-the-state-of-the-art-clustering-algorithms-1e65a08157a1

[6] https://www.preprints.org/manuscript/202108.0140/v1

[7] https://www.freecodecamp.org/news/8-clustering-algorithms-in-machine-learning-that-all-data-scientists-should-know/

[8] https://github.com/EmirKorkutUnal/A-Comparison-of-Clustering-Algorithms-K-means-MeanShift-DBSCAN-in-Python