TEXAS A&M UNIVERSITY
Engineering

ECEN 758 Data Mining and Analysis: Lecture 11, Density-based Clustering

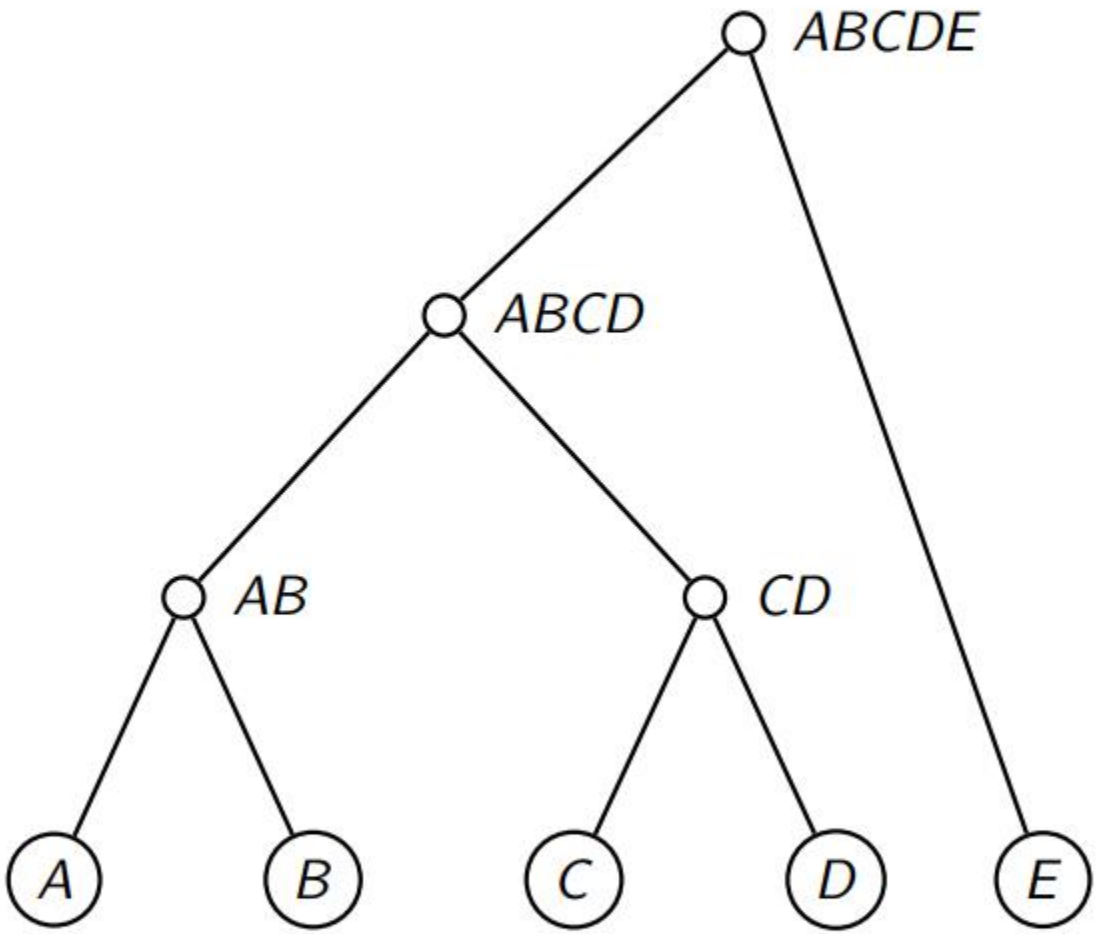
Joshua Peeples, Ph.D.
Assistant Professor
Department of Electrical and Computer Engineering

Announcements

- Assignment #2 due this Friday (09/27)
  - Q1.4: Only need to show FP-tree and conditional pattern base for final solution

Last Lecture

- Hierarchical Clustering



Gif from: Expectation-maximization algorithm, Wikipedia

Today

- Density-based Clustering
- Reading: ZM Chapter 15

Clustering Overview

- We will discuss several variants of clustering
  - Representative-based Clustering
  - Hierarchical Clustering
  - **Density-based Clustering**

Density-based Clustering Overview

Shortcomings of k-Means

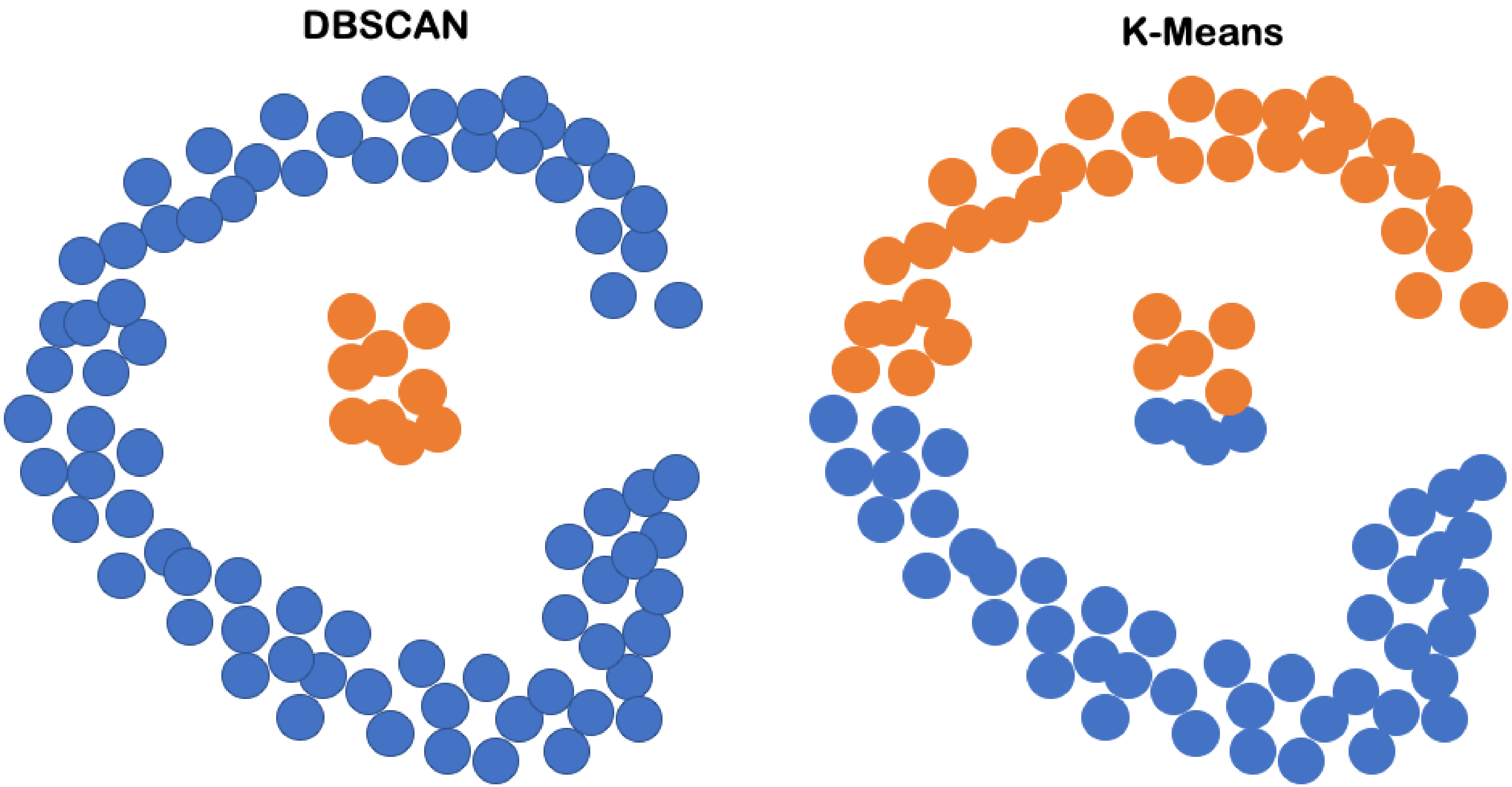
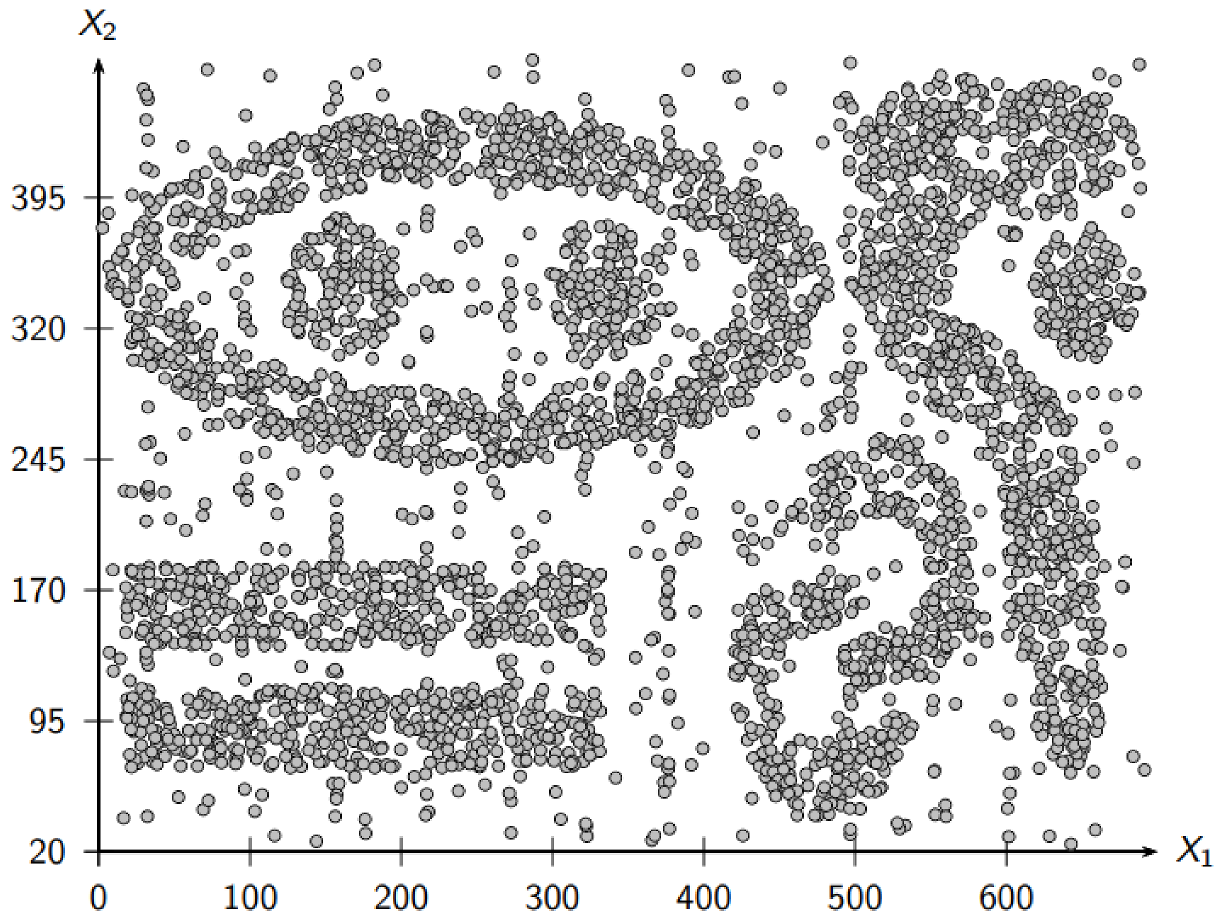


Image from: Yufeng, Understanding DBSCAN and Implementation with Python

Density-based Clustering

TEXAS A&M UNIVERSITY Engineering

- Able to find nonconvex cluster
- Distance-based method may have trouble with nonconvex clusters

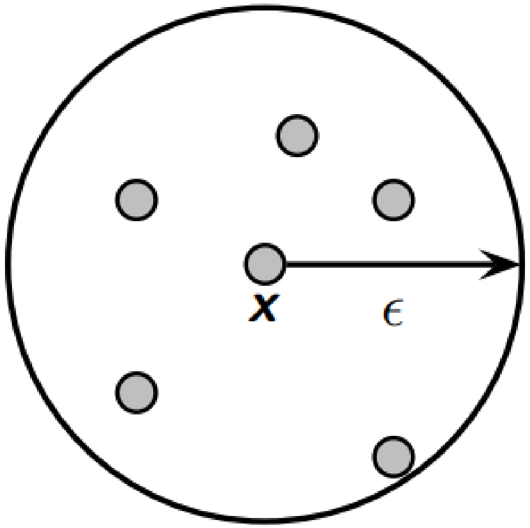


Density-based spatial clustering of applications with noise (DBSCAN) Algorithm

DBSCAN Approach

- Define a ball with radius ϵ around a point **x**
  - ϵ neighborhood of **x**
- δ represents distance function

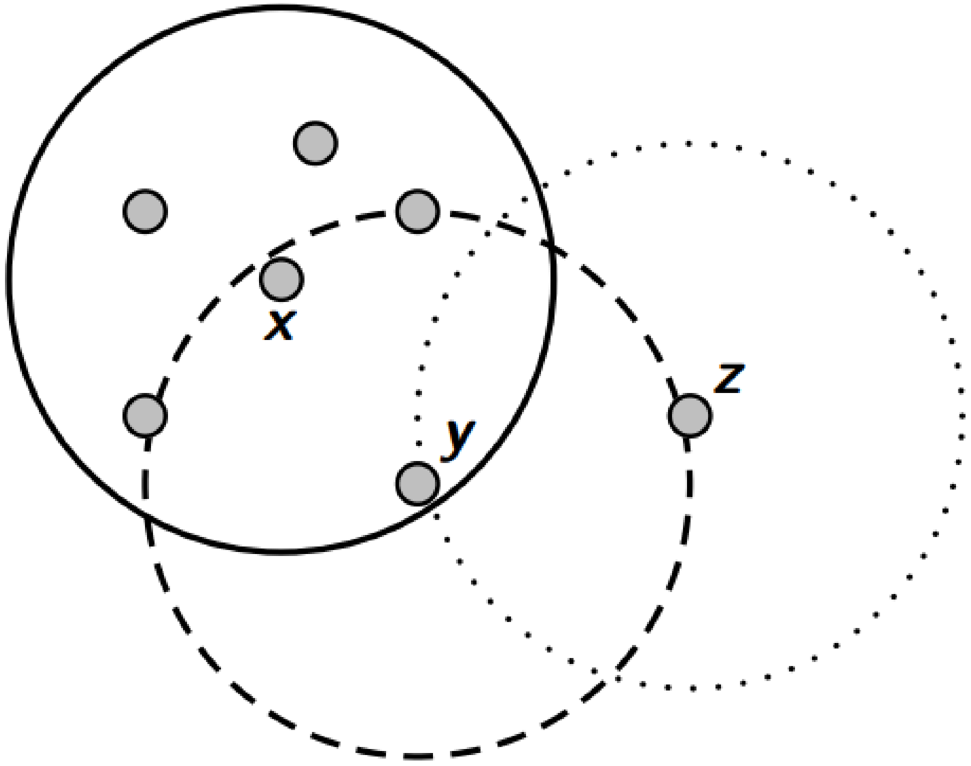
$$N_\epsilon(\mathbf{x}) = B_d(\mathbf{x}, \epsilon) = \{\mathbf{y} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$$



(a) Neighborhood of a Point

DBSCAN Approach

- Core point:
  - There are at least *minpts* in its ϵ-neighborhood
- Border point:
  - Does not meet the *minpts* threshold but is in ϵ-neighborhood of core point
- Noise point
  - Neither core or border point



(b) Core, Border, and Noise Points

DBSCAN Approach

- Directly density reachable:
  - If in a point’s ϵ-neighborhood and is a core point
- Density reachable:
  - If there are a set of core of points leading from **v** to **u**
- Density connected
  - If there exist a core point **m** such that both **v** and **u** are density reachable
- Density-based cluster defined for maximal set of density connected points

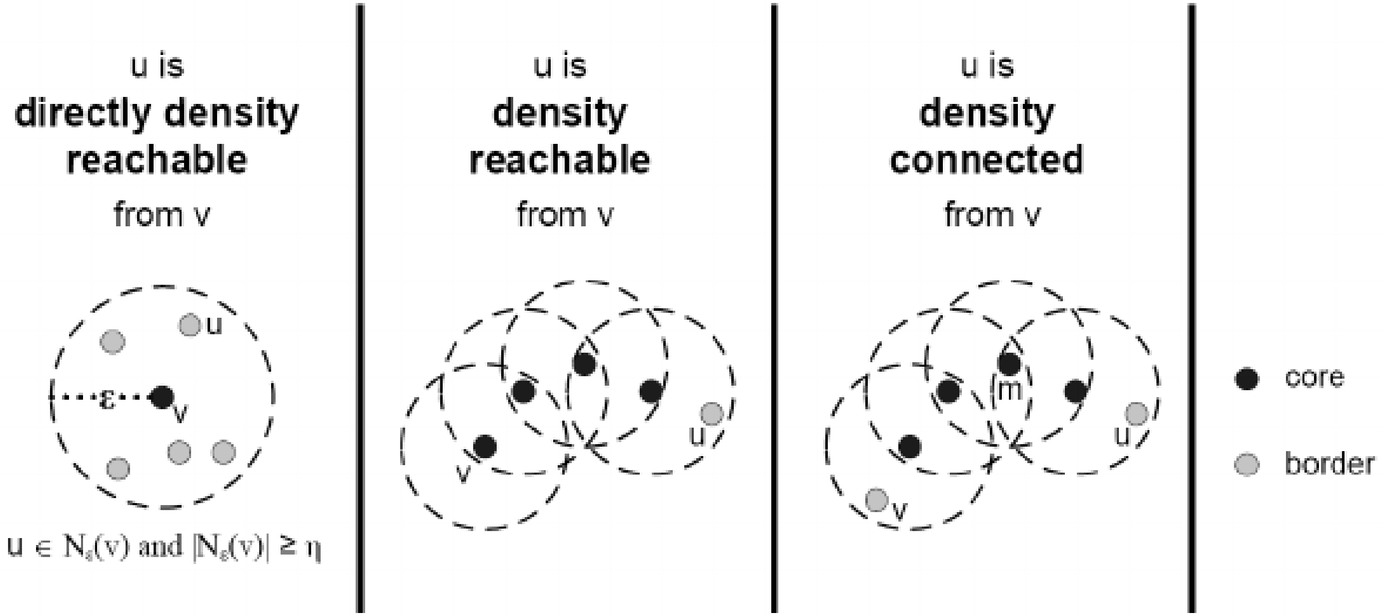
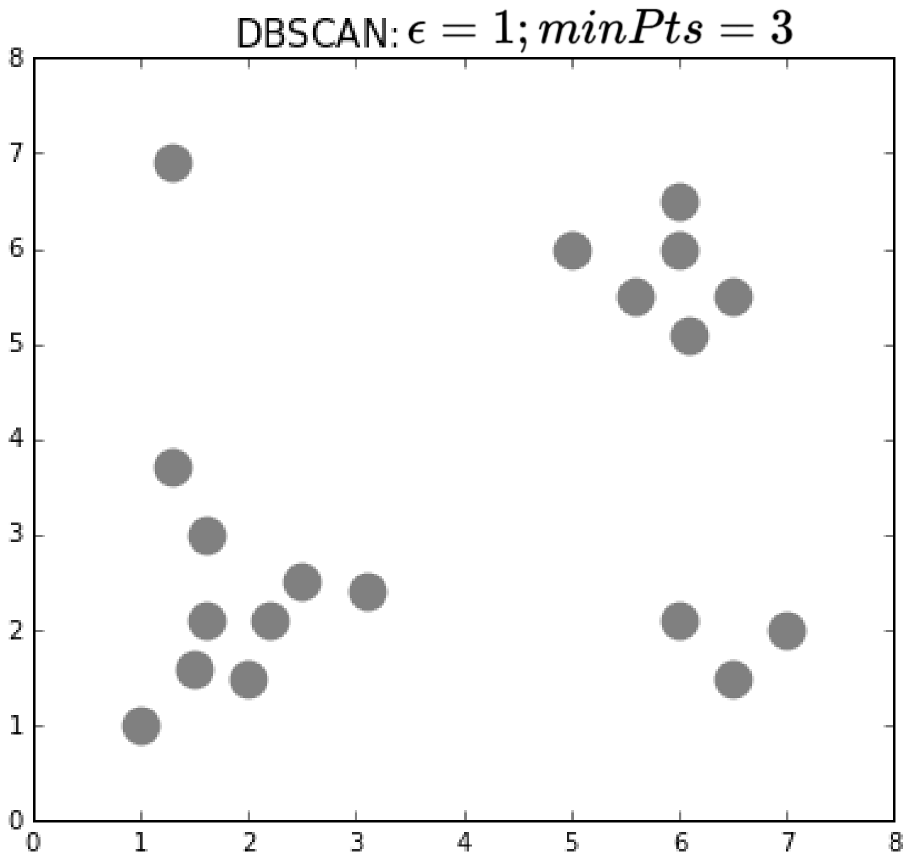


Image from: N. Schlitter , DenGraph-HO: Density-based Hierarchical Community Detection for Explorative Visual Network Analysis

DBSCAN Algorithm

- Compute ϵ-neighborhood for each point
- Check if this is a core point
- Recursively find all density connected point and assign to same cluster
- Border points may be reachable by core points in multiple clusters
  - Assign to one or to all overlapping clusters (“soft”)



Gif from: D. Sheehan , Clustering with Scikit with GIFs

DBSCAN Algorithm

dbscan ($\boldsymbol{D}$, ϵ, *minpts*):

$Core \leftarrow \emptyset$

foreach $\boldsymbol{x}_i \in \boldsymbol{D}$ **do** `// Find the core points`

  Compute $N_\epsilon(\boldsymbol{x}_i)$

  $id(\boldsymbol{x}_i) \leftarrow \emptyset$ `// cluster id for` $\boldsymbol{x}_i$

  if $N_\epsilon(\boldsymbol{x}_i) \geq minpts$ **then** $Core \leftarrow Core \cup \{\boldsymbol{x}_i\}$

$k \leftarrow 0$ `// cluster id`

foreach $\boldsymbol{x}_i \in Core$, *such that* $id(\boldsymbol{x}_i) = \emptyset$ **do**

  $k \leftarrow k + 1$

  $id(\boldsymbol{x}_i) \leftarrow k$ `// assign` $\boldsymbol{x}_i$ `to cluster id` k

  DensityConnected $(\boldsymbol{x}_i, k)$

$\mathcal{C} \leftarrow \{C_i\}_{i=1}^{k}$, where $C_i \leftarrow \{\boldsymbol{x} \in \boldsymbol{D} \mid id(\boldsymbol{x}) = i\}$

$Noise \leftarrow \{\boldsymbol{x} \in \boldsymbol{D} \mid id(\boldsymbol{x}) = \emptyset\}$

$Border \leftarrow \boldsymbol{D} \setminus \{Core \cup Noise\}$

return $\mathcal{C}, Core, Border, Noise$

DensityConnected ($\boldsymbol{x}$, k):

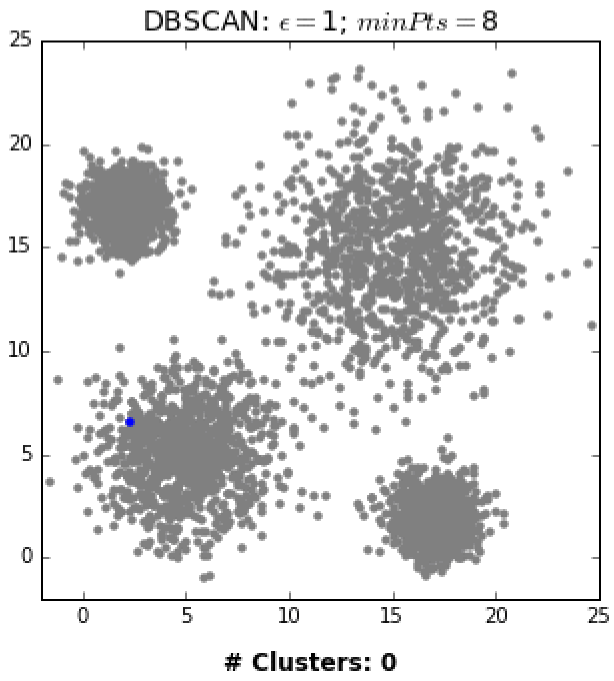
foreach $\boldsymbol{y} \in N_\epsilon(\boldsymbol{x})$ **do**

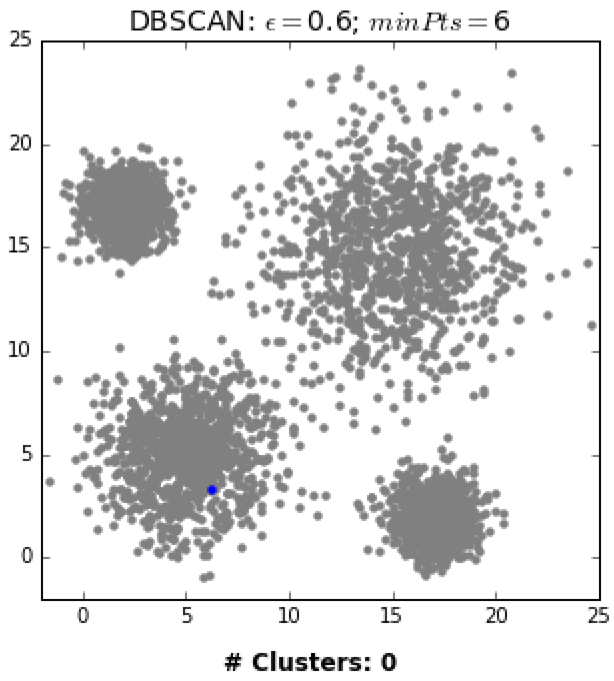
  $id(\boldsymbol{y}) \leftarrow k$ `// assign` $\boldsymbol{y}$ `to cluster id` k

  if $\boldsymbol{y} \in Core$ **then** DensityConnected $(\boldsymbol{y}, k)$

DBSCAN Algorithm

- Advantages
  - Find nonconvex clusters
  - Do not need to set the number of cluster
- Disadvantages
  - Very sensitive to selection of ϵ and minPts
  - Difficult for data with clusters of varying density





Gif from: D. Sheehan , Clustering with Scikit with GIFs

Extensions of DBSCAN

- Ordering points to identify the clustering structure (OPTICS)
- Hierarchical DB (HDBSCSAN)

OPTICS

- Account for clustering of varying densities and shapes
- Creates an ordered list of points through reachability plot
- Reachability: measure of how easy it is to reach point

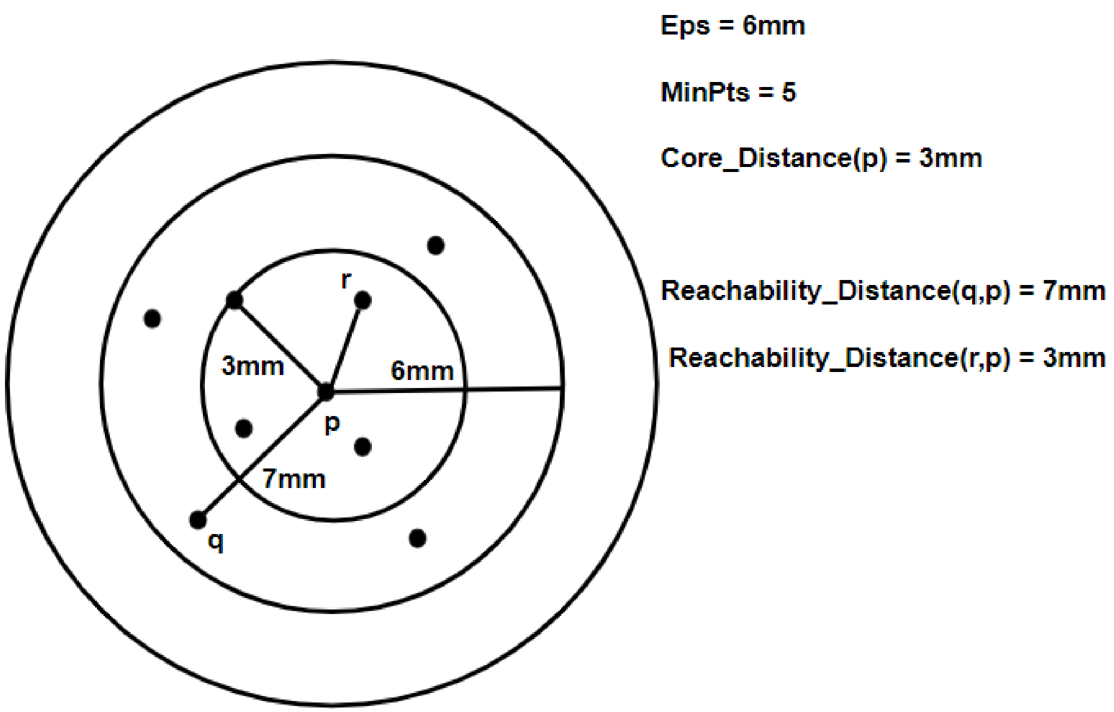


Image from: ML | OPTICS Clustering Explanation

HDBSCAN

- Transform space based on density-sparsity
- Build minimum spanning tree of distance weighted graph
- Construct hierarchy of connected components
- Condense hierarchy based on minimum cluster size
- Extract stable clusters from condensed tree

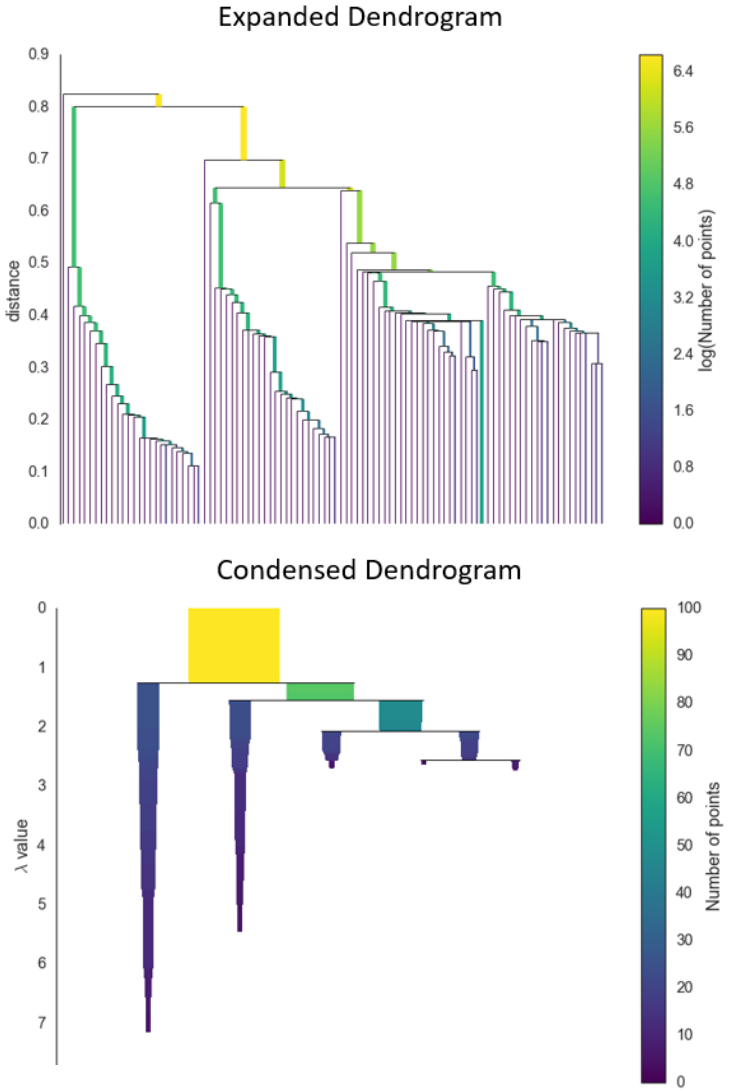


Image from: T. Dorfer, Density-Based Clustering: DBSCAN vs. HDBSCAN

Kernel Density Estimation

Kernel Density Estimation (KDE)

- Close connection between density-based clustering and density estimation
- KDE seeks to determine unknown PDF by finding dense regions of points

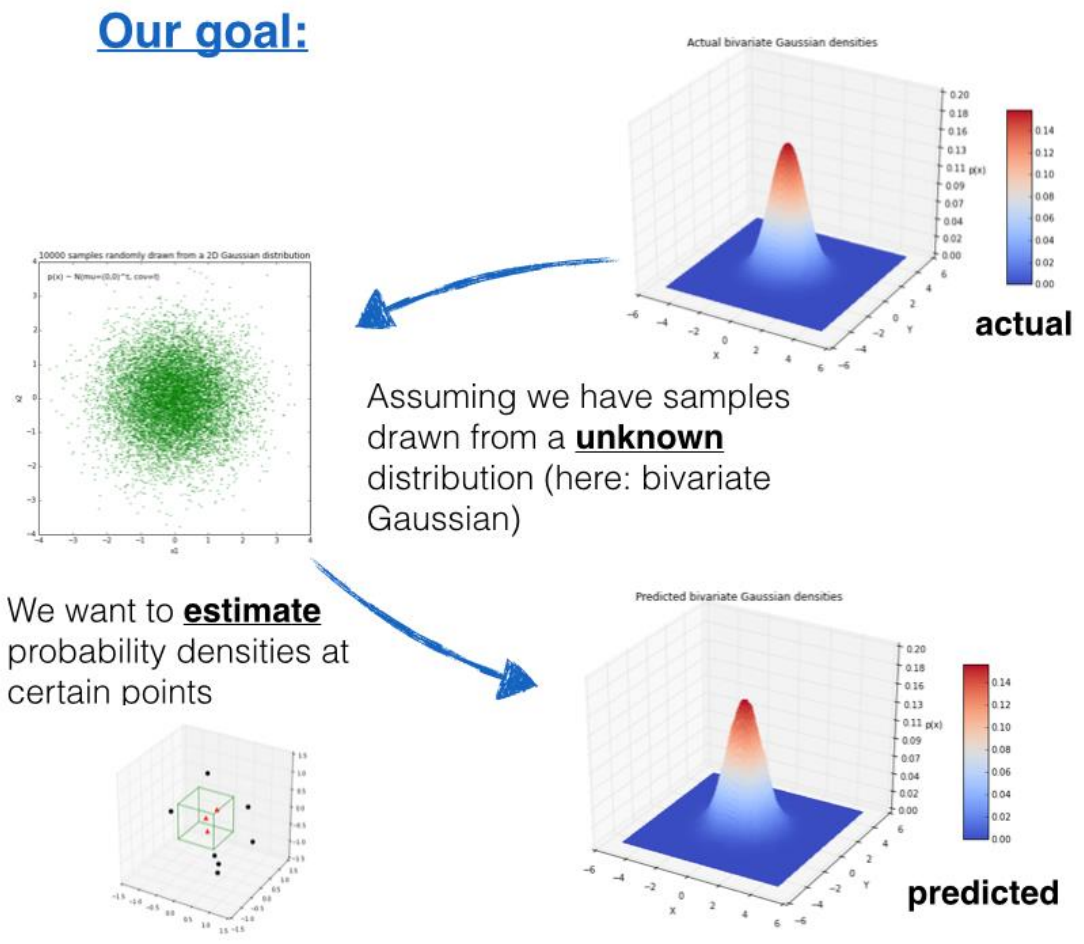


Image from: S. Raschka, Kernel density estimation via the Parzen-Rosenblatt window method– explained using Python

Univariate Density Estimation

- Can use CDF to estimate PDF
- k is the number of points that lie in a window of width h centered on x
- Density estimation is ratio of fraction of points in window and volume of window

$$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x)$$

$$\hat{f}(x) = \frac{\hat{F}\left(x+\frac{h}{2}\right) - \hat{F}\left(x-\frac{h}{2}\right)}{h} = \frac{k/n}{h} = \frac{k}{nh}$$

Kernel Estimator

TEXAS A&M UNIVERSITY Engineering

- Need to define kernel function
- Need to be non-negative and integrates to 1 for all values
- Discrete kernel uses indicator function

$$K(z) = \begin{cases} 1 & \text{If } |z| \leq \frac{1}{2} \\ 0 & \text{Otherwise} \end{cases}$$

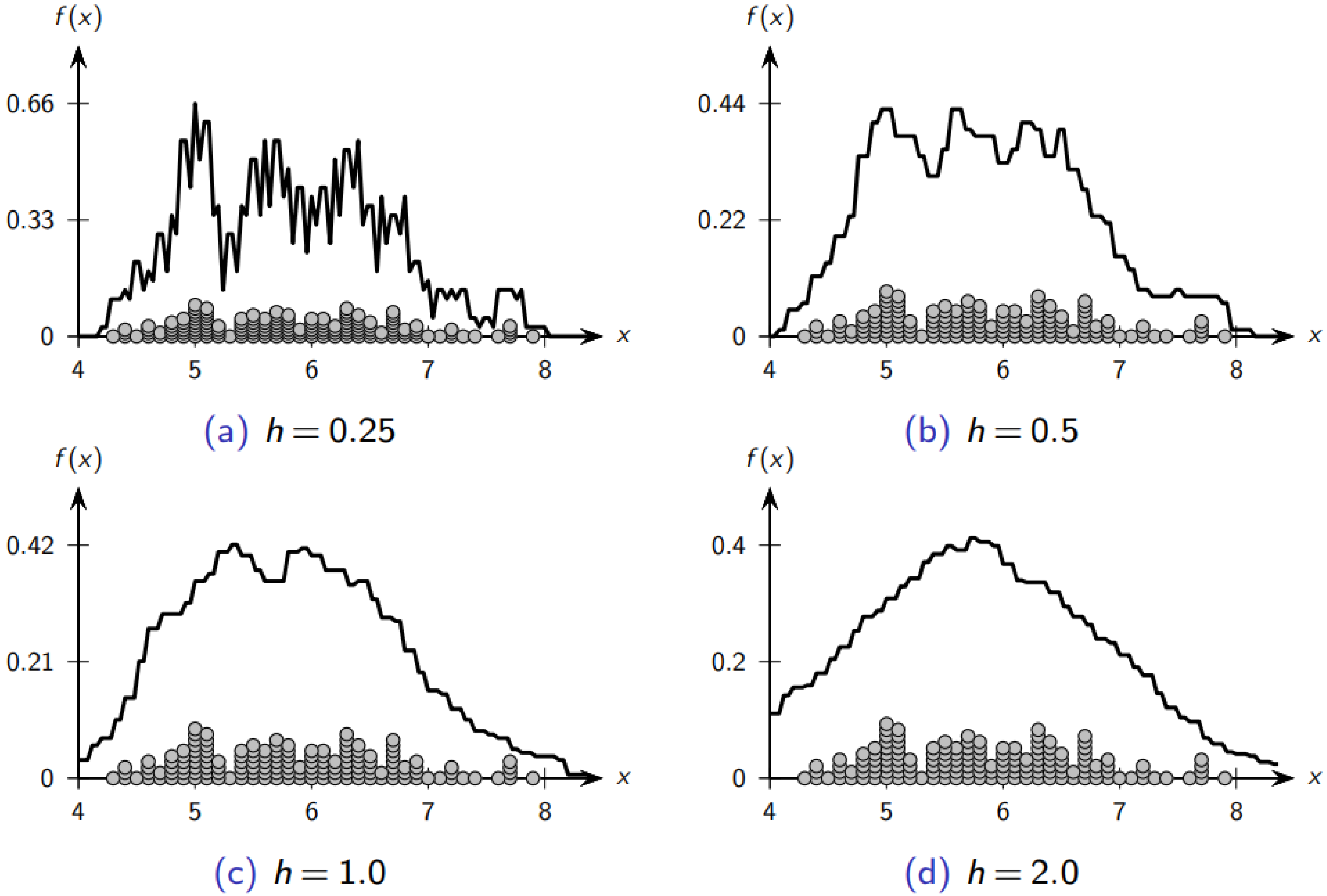
Discrete kernel

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

Density estimate

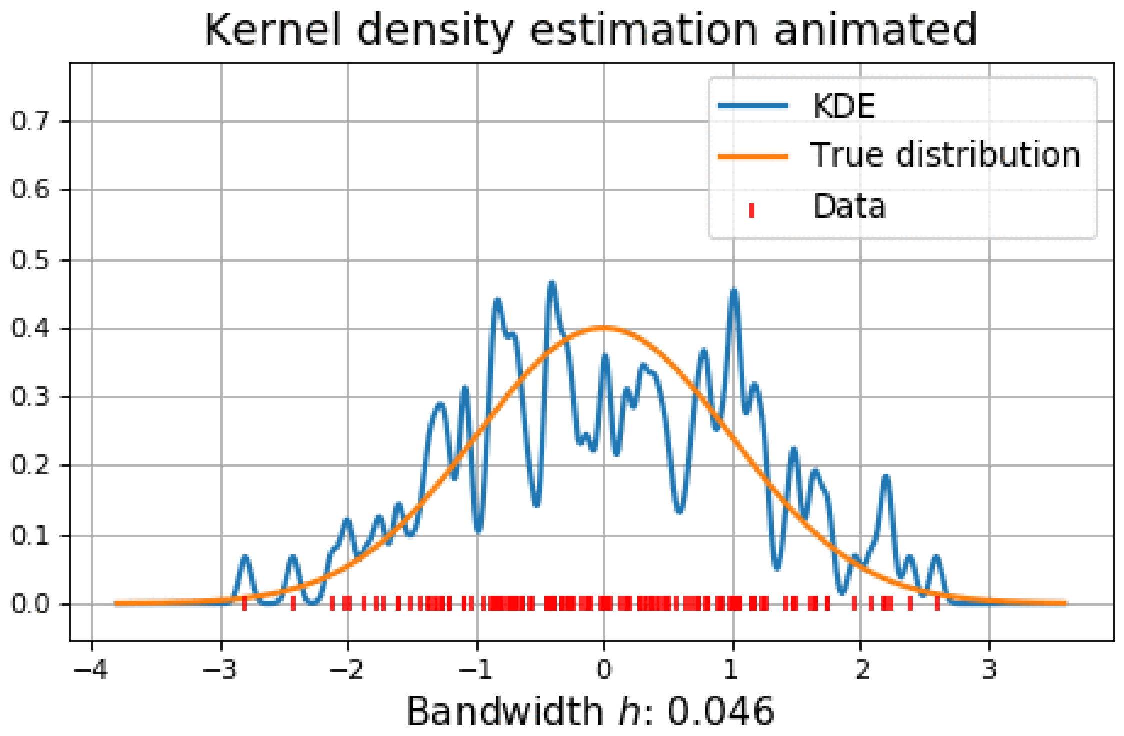
Images from: P. Sharma, What is Hierarchical Clustering in Python?

Discrete KDE Example



Gaussian Kernel Estimator

- Width parameter *h* controls spread or smoothness of estimate
- Discrete kernel function has abrupt changes
- Gaussian kernel provides smooth transistion



Gif from: KDEpy

Gaussian Kernel Estimator

TEXAS A&M UNIVERSITY Engineering

- Width parameter *h* controls spread or smoothness of estimate
- Discrete kernel function has abrupt changes
- Gaussian kernel provides smooth transistion

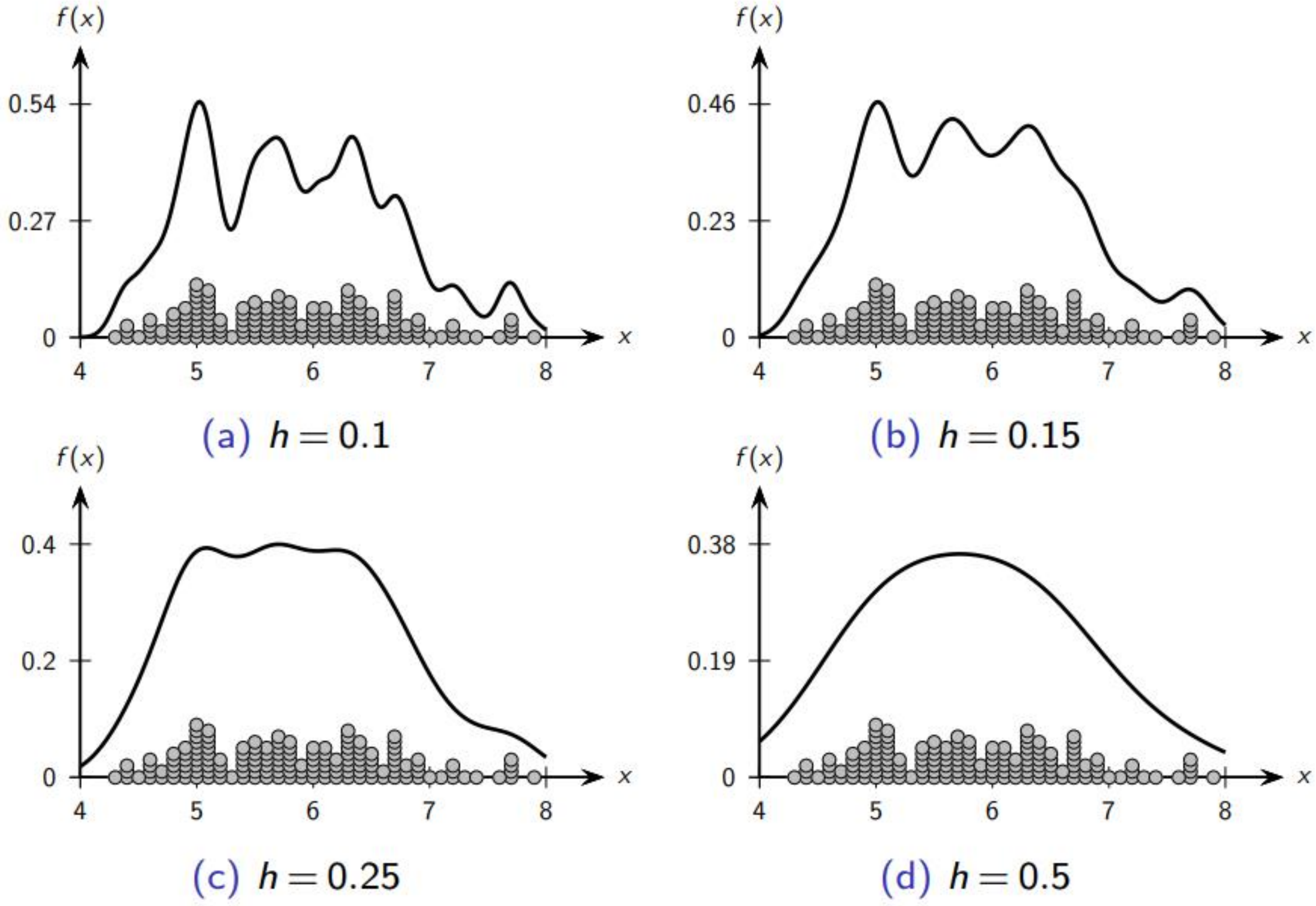
$$K(z) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{z^2}{2}\right\}$$

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

Density estimate

Images from: P. Sharma, What is Hierarchical Clustering in Python?

Discrete KDE Example



Multivariate Density Estimation

TEXAS A&M UNIVERSITY Engineering

- Define d-dimension "window" as hypercube with edge length *h*
- Density estimation is ratio of fraction of points in window and volume of window
- Kernel function still must integrate to 1

$$\mathrm{vol}(H_d(h)) = h^d$$

Volume of hypercube

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

Density estimate

$$\int K(\mathbf{z})d\mathbf{z} = 1$$

Multivariate Kernel Estimators

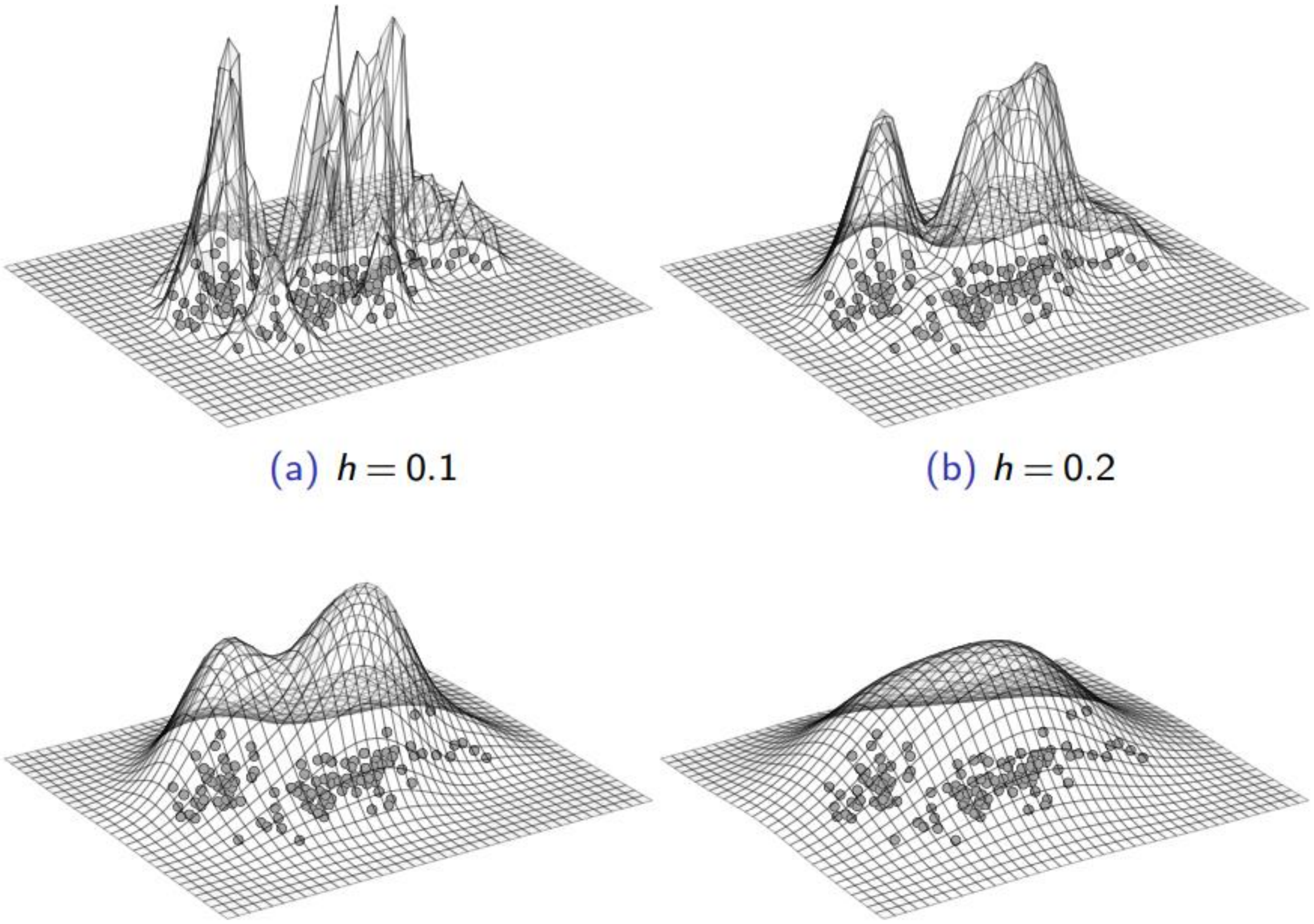
$$K(\mathbf{z}) = \begin{cases} 1 & \text{If } |z_j| \leq \frac{1}{2}, \text{ for all dimensions } j = 1, \ldots, d \\ 0 & \text{Otherwise} \end{cases}$$

Discrete kernel

$$K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{(\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)}{2h^2}\right\}$$

Gaussian kernel

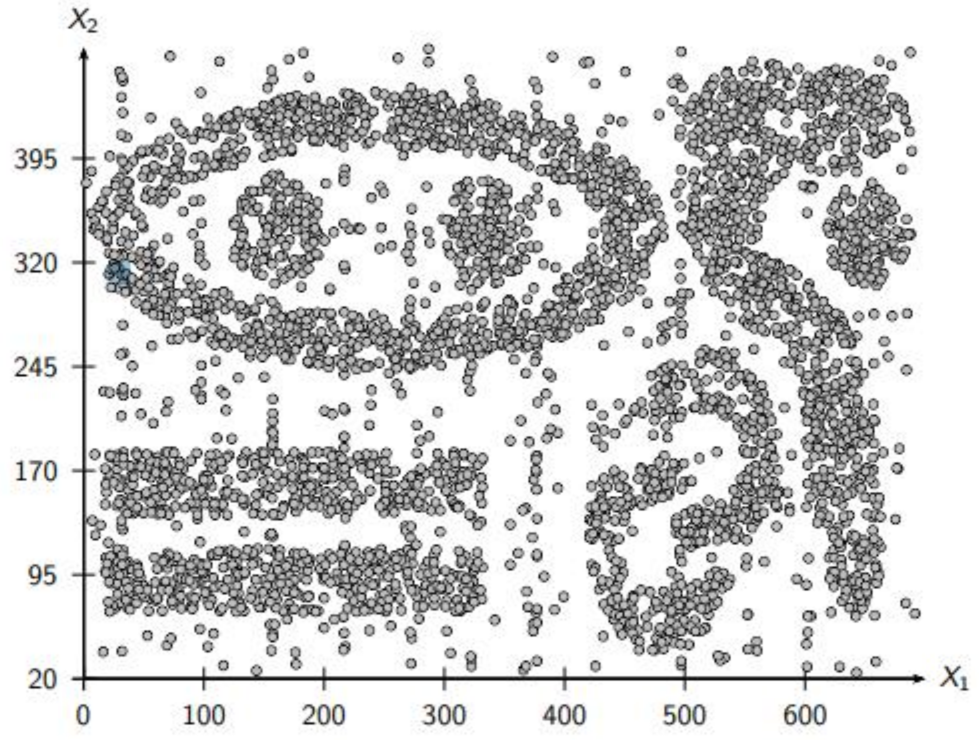
Gaussian KDE (Iris 2D)



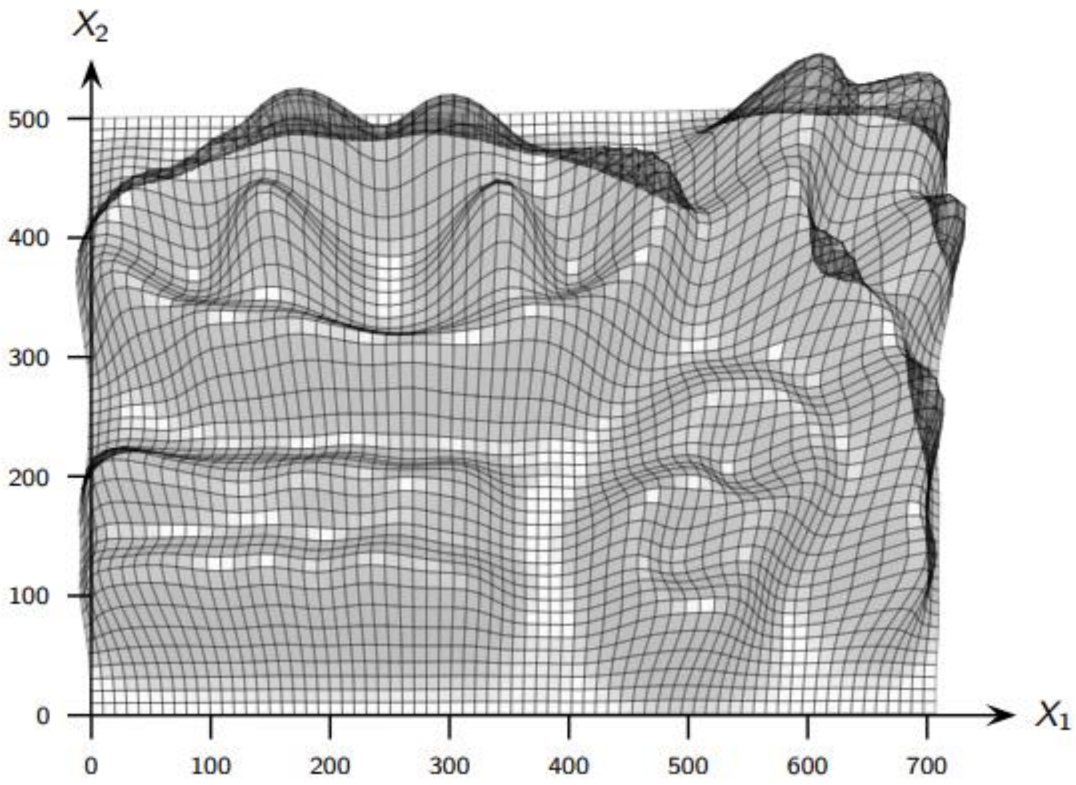
(a) $h = 0.1$ (b) $h = 0.2$

(c) $h = 0.35$ (d) $h = 0.6$

Gaussian KDE



(a) Original Points



(b) Gaussian Density Estimation

Nearest Neighbor Density Estimation

- Previously, fixed volume by fixing *h*
- Alternative: fix number of points and adapt volume
- *k* is number of neighbors

$$\hat{f}(\mathbf{x}) = \frac{k}{n\,\mathrm{vol}(S_d(h_{\mathbf{x}}))}$$

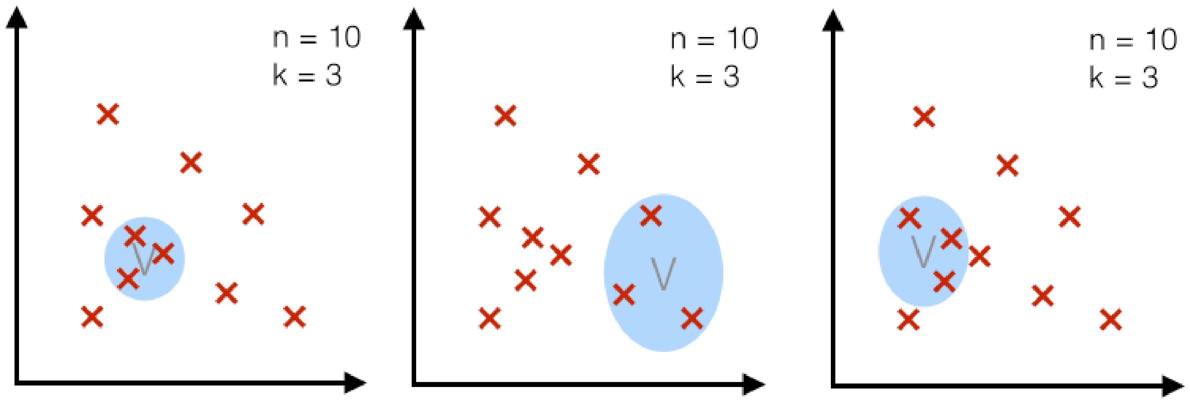


Image from: S. Raschka, Kernel density estimation via the Parzen-Rosenblatt window method– explained using Python

Nearest Neighbor Density Estimation

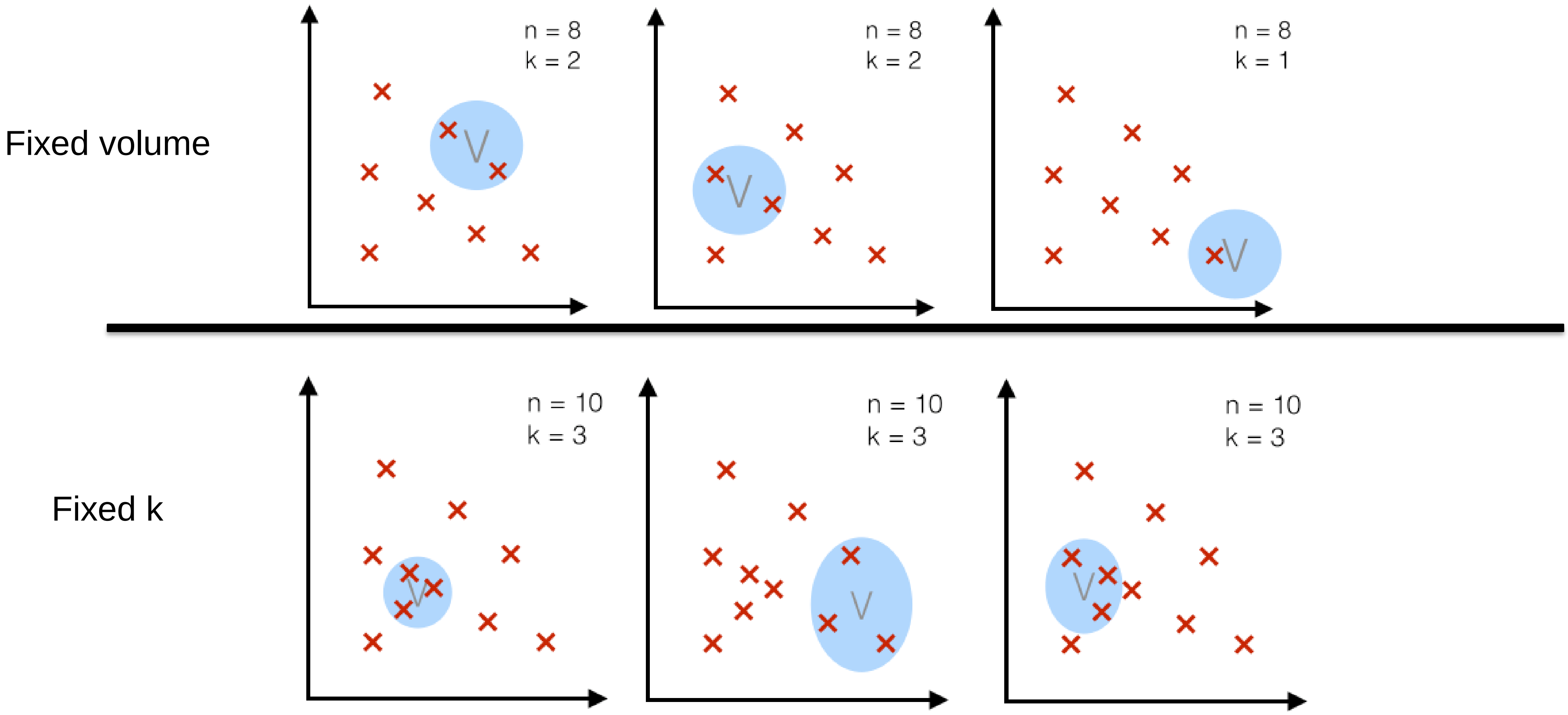
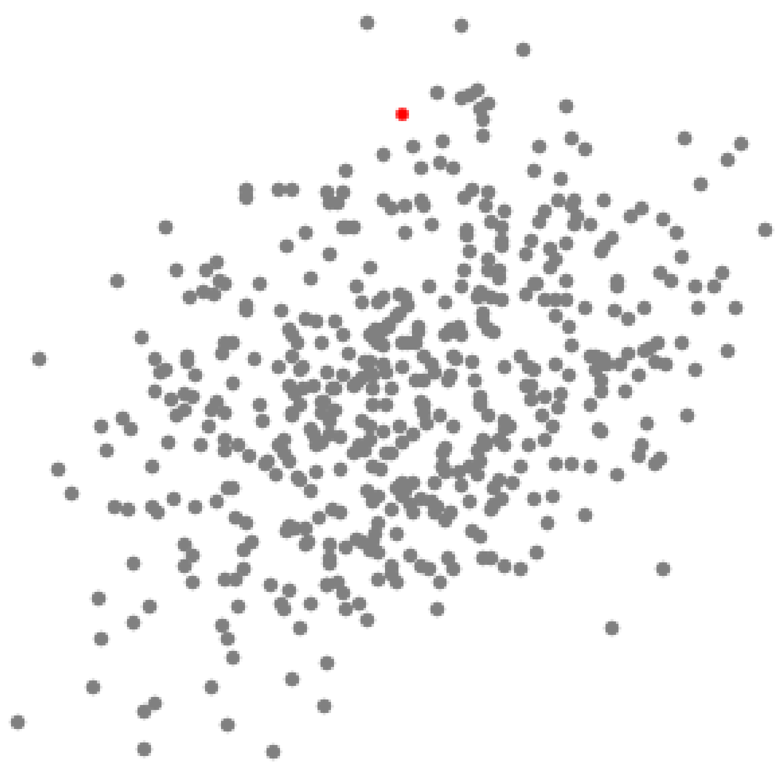


Image from: S. Raschka, Kernel density estimation via the Parzen-Rosenblatt window method– explained using Python

Mean Shift Algorithm

Mean Shift Algorithm

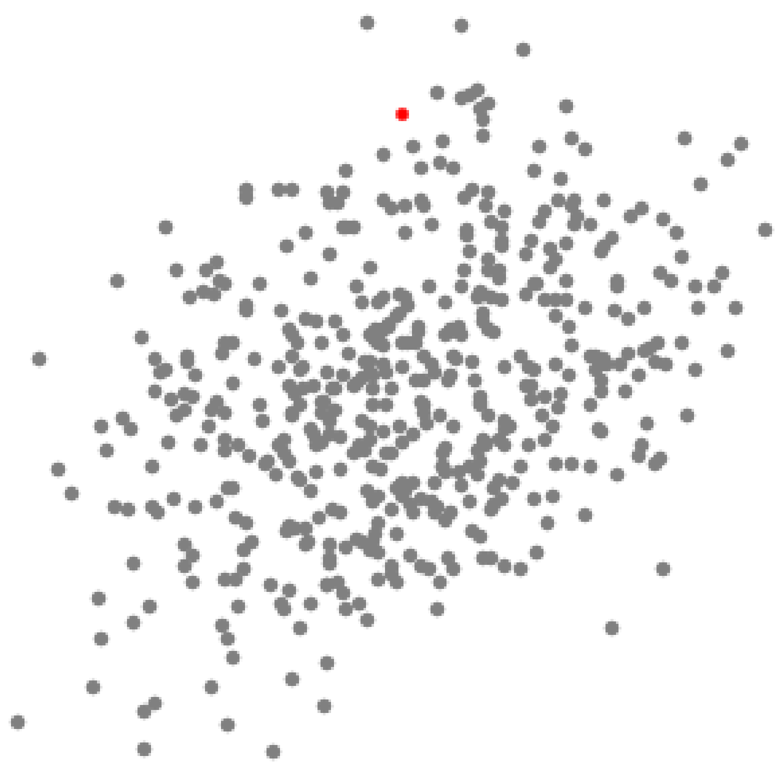
- Useful for clusters with arbitrary shapes and not well-separated data
- Idea: shift each data toward the mode of the distribution of points within a given radius



Gif from: D. Sheehan , Clustering with Scikit with GIFs

Mean Shift Algorithm Steps

- Initialize data points as cluster centers
- Repeat until convergence
  - Compute the mean of all data points within a certain radius (kernel)
  - Shift the data to the mean
  - Identify the cluster centroids as points that have not moved
  - Return cluster assignments

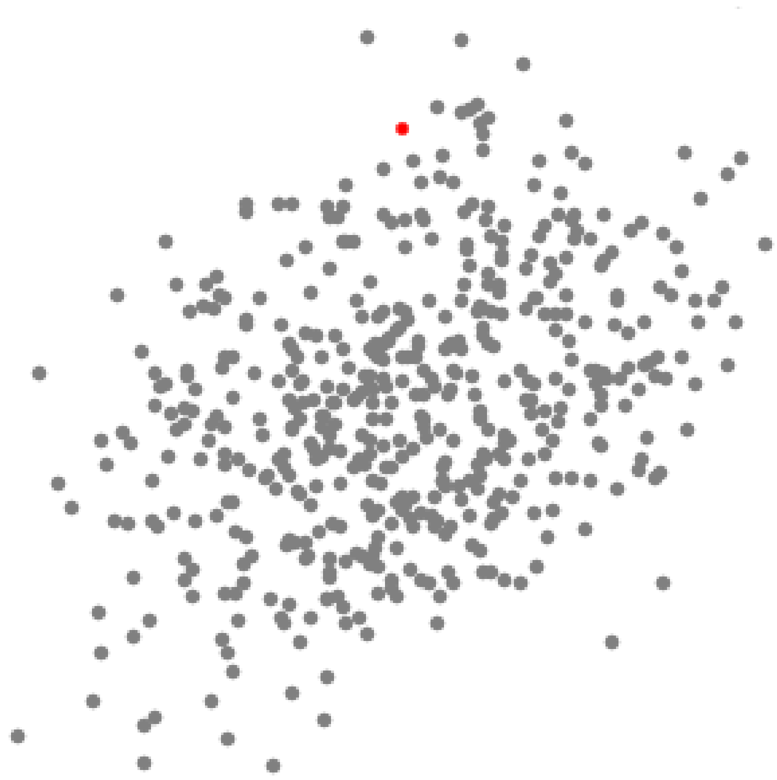


Gif from: D. Sheehan , Clustering with Scikit with GIFs

Mean Shift Algorithm Steps

TEXAS A&M UNIVERSITY Engineering

- Initialize data points as cluster centers
- Repeat until convergence
  - **Compute the mean of all data points within a certain radius (kernel)**
  - Shift the data to the mean
  - Identify the cluster centroids as points that have not moved
  - Return cluster assignments



Gif from: D. Sheehan , Clustering with Scikit with GIFs

Mean Shift Algorithm: Kernel Function

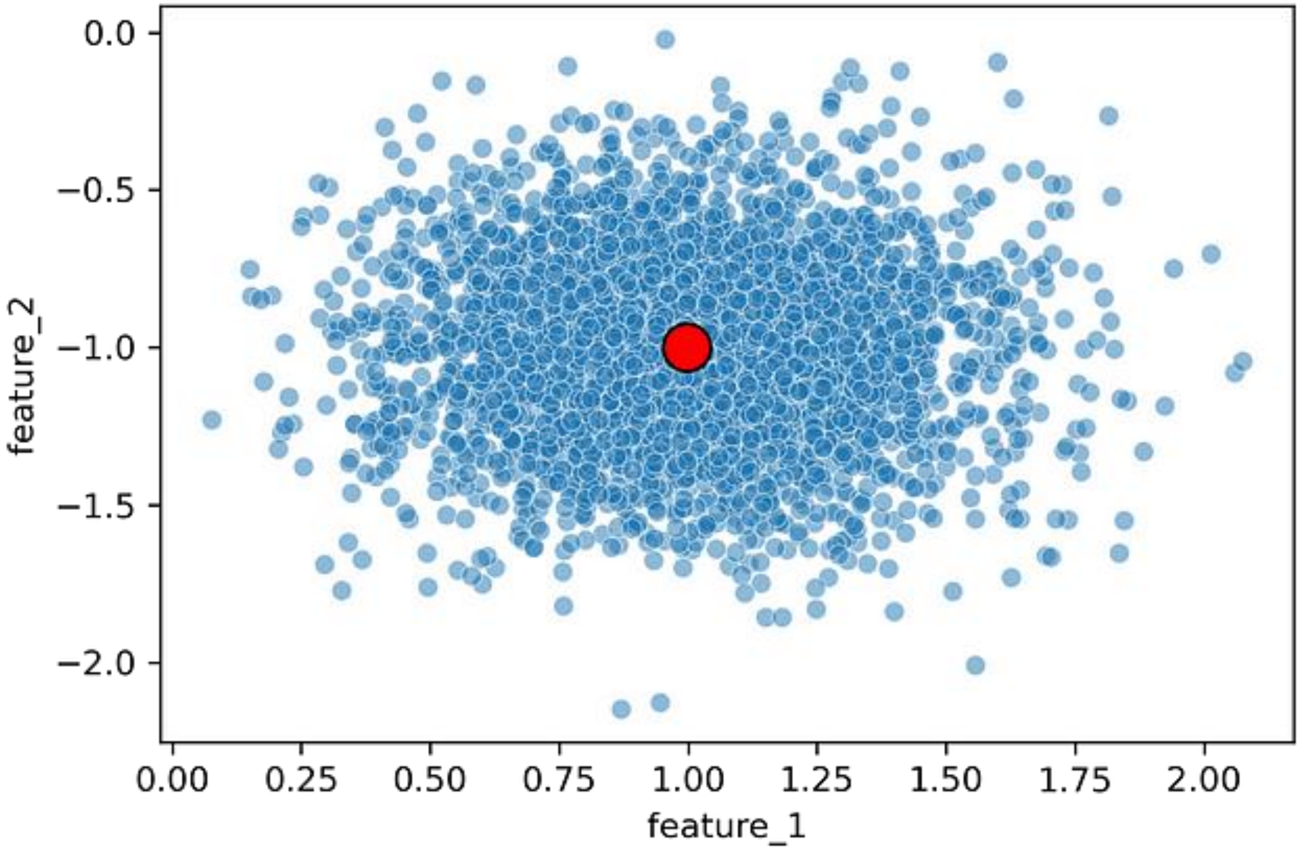


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Kernel Function

- Easily compute mean of data
- Will be more meaningful if weighted mean

$$M_W = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

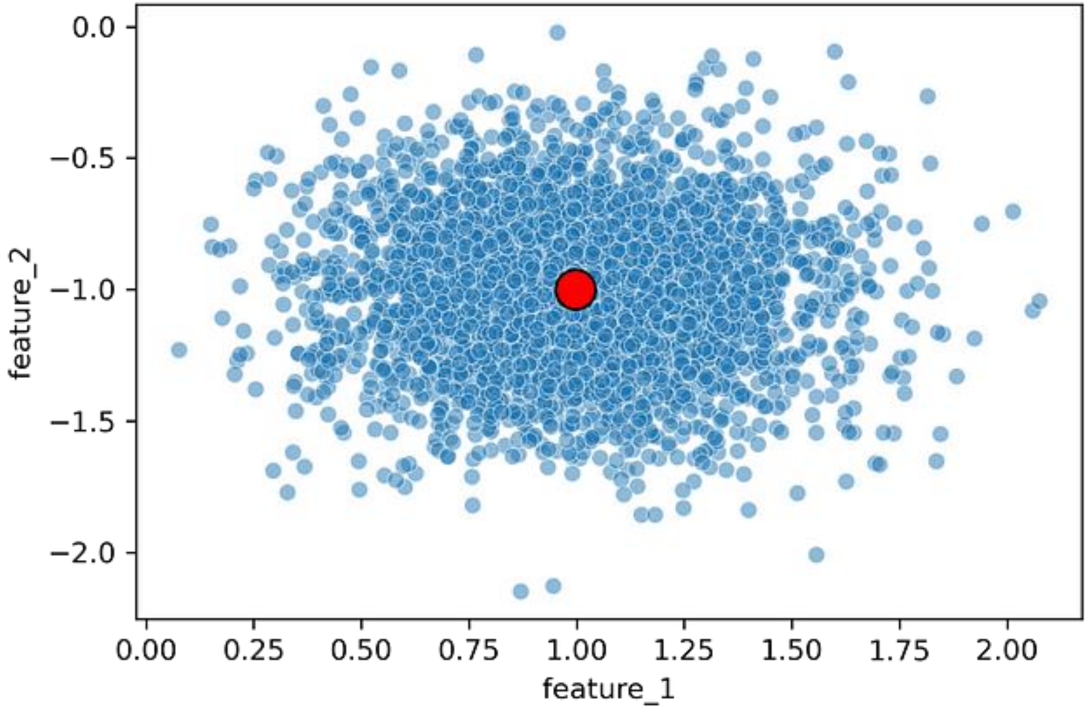
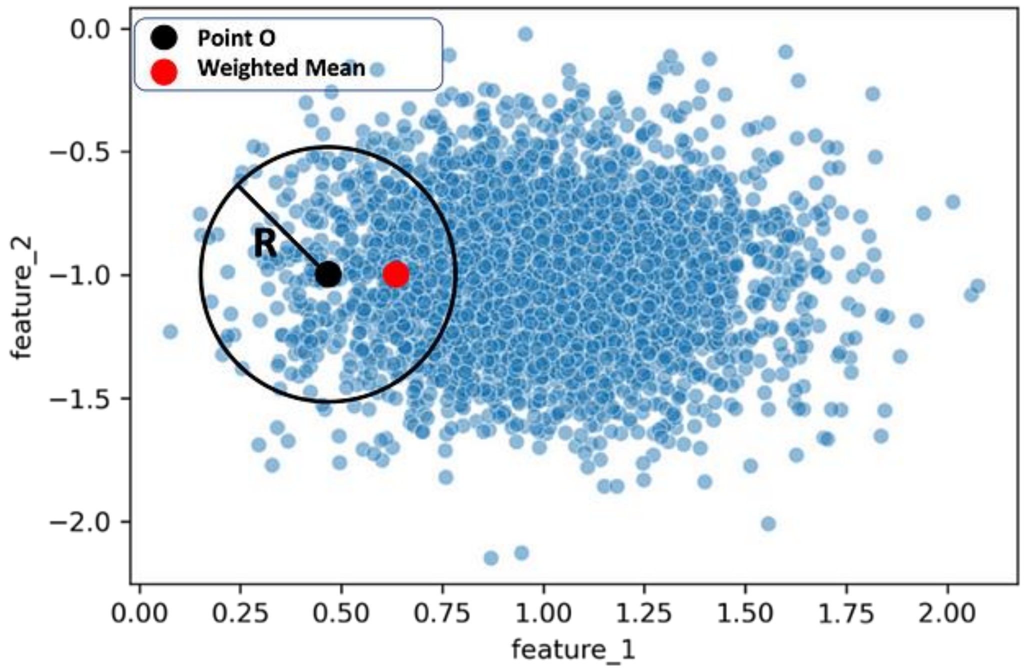


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Kernel Function

- Can use “flat” or Gaussian kernel for weights



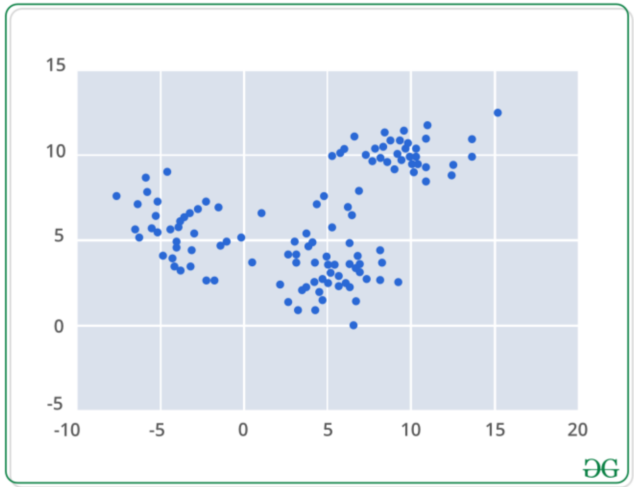
$$w(d) = \begin{cases} 1, & \text{if } d \leq R \\ 0, & \text{if } d > R \end{cases}$$

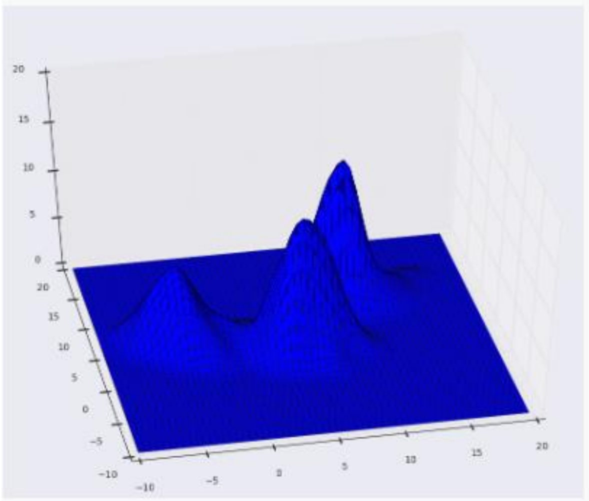
Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Kernel Function

TEXAS A&M UNIVERSITY Engineering

- Can use “flat” or Gaussian kernel for weights



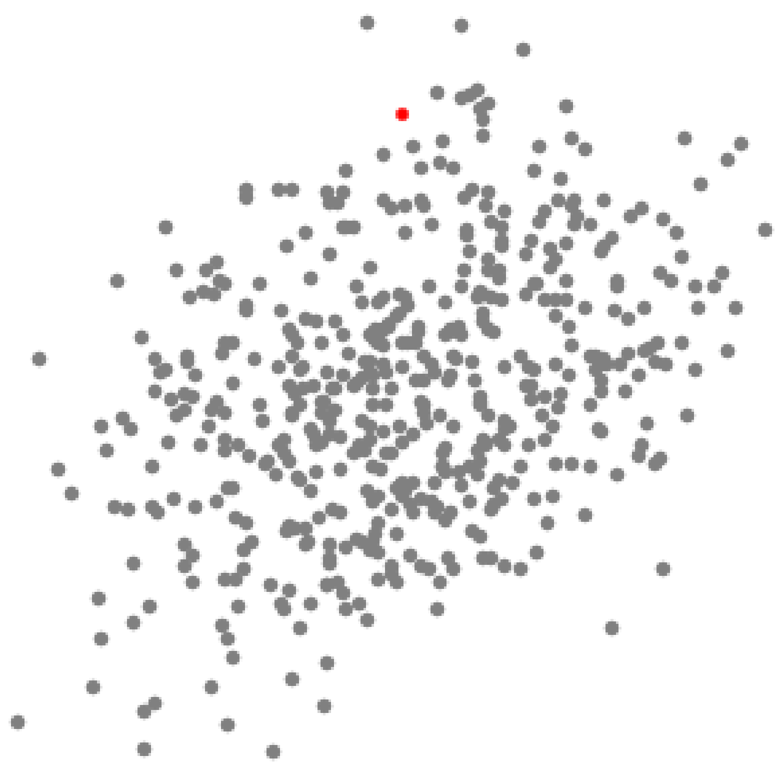


$$w(d) = e^{-\frac{d}{2\sigma^2}}$$

Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python and ML | Mean-Shift Clustering

Mean Shift Algorithm Steps

- Initialize data points as cluster centers
- Repeat until convergence
  - Compute the mean of all data points within a certain radius (kernel)
  - **Shift the data to the mean**
  - Identify the cluster centroids as points that have not moved
  - Return cluster assignments



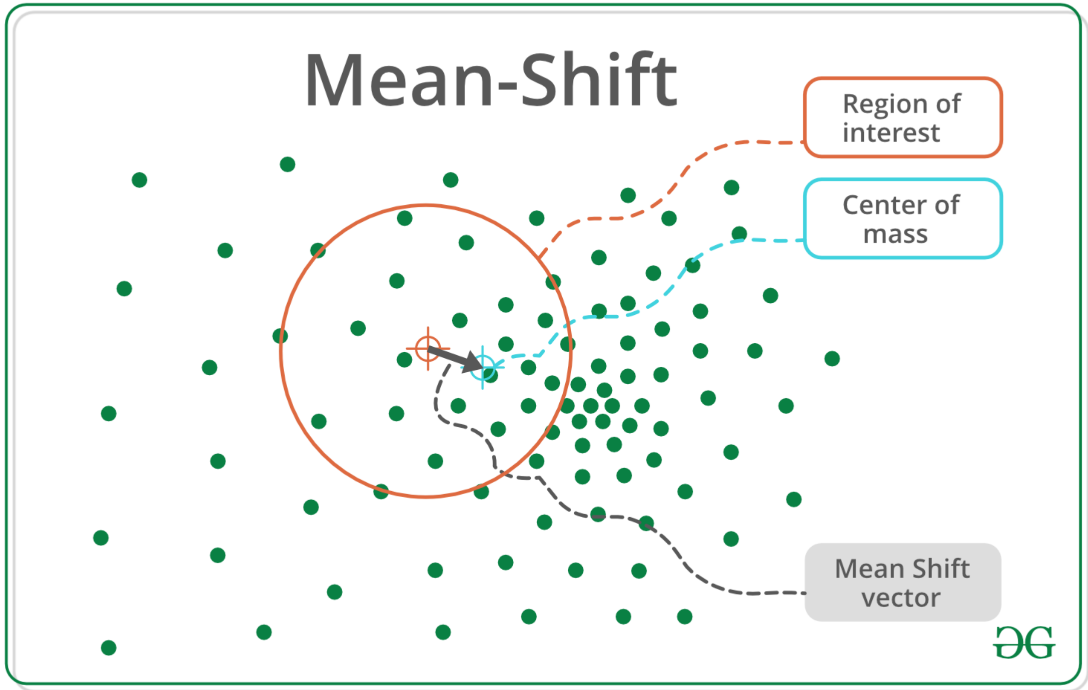
Gif from: D. Sheehan , Clustering with Scikit with GIFs

Mean Shift Algorithm

- “Shift” local mean based on neighborhood defined by kernel

$$x^{t+1} = x^t + m(x^t)$$

Update centroid location



$$m(x) = \frac{1}{|N(x)|} \sum_{x_j \in N(x)} x_j - x = \frac{\sum_{x_j \in N(x)} K(x_j - x) x_j}{\sum_{x_j \in N(x)} K(x_j - x)} - x$$

Mean-shift vector

Image from: ML | Mean-Shift Clustering

Mean Shift Algorithm: Shifting

TEXAS A&M UNIVERSITY Engineering

- Compute weighted mean based on area defined by kernel

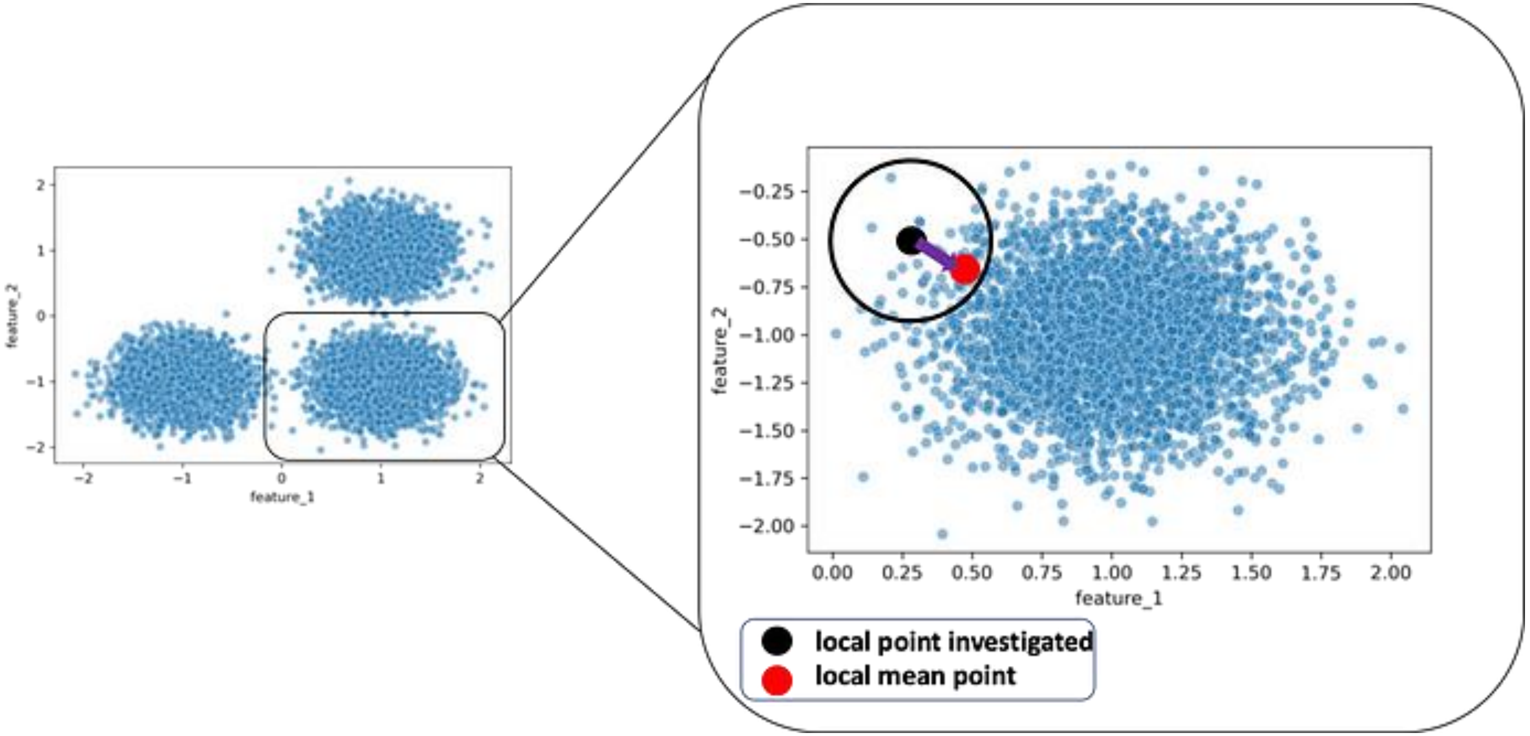


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Shifting

- Repeat procedure for new position (i.e., neighborhood)

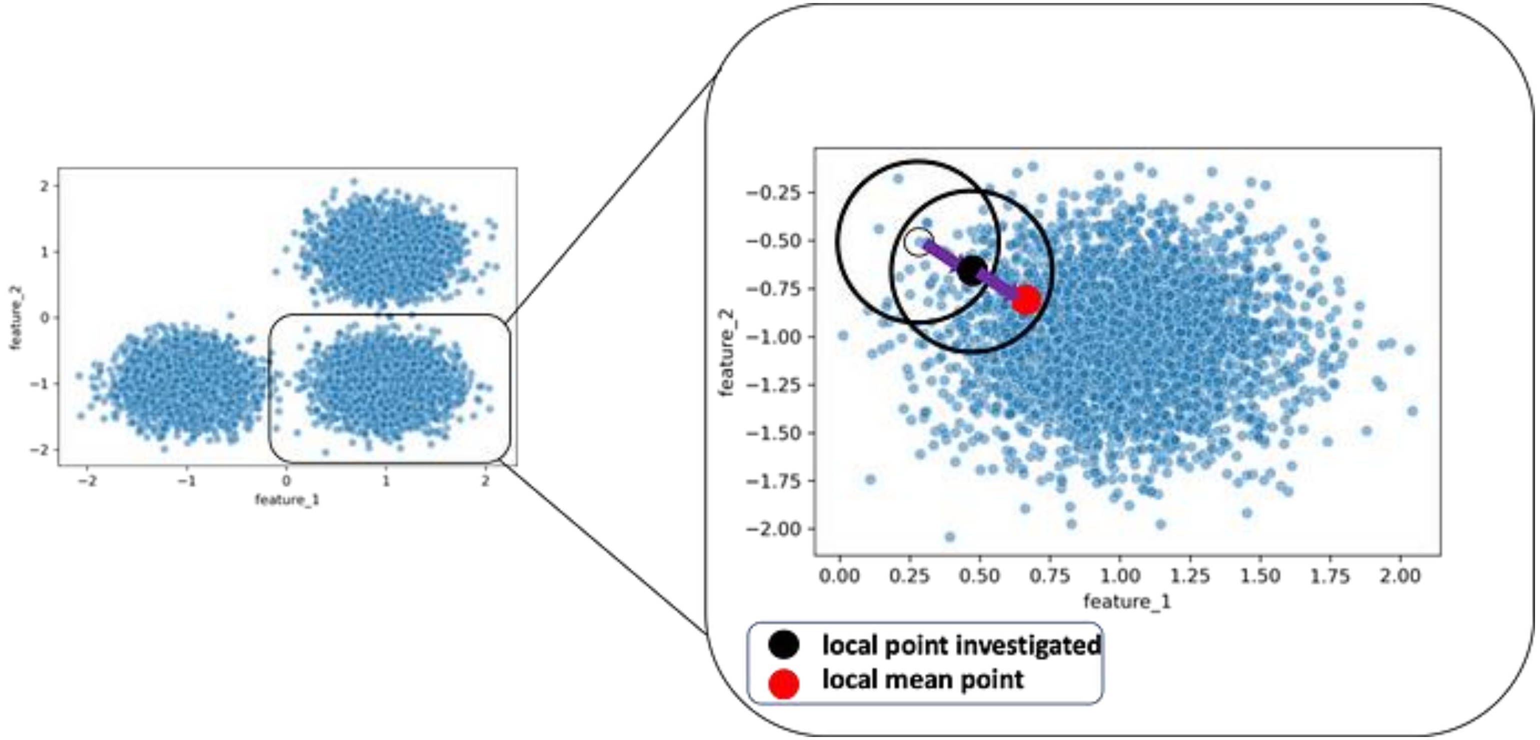


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Shifting

- Terminate after enough iterations or until the number of points within a cluster no longer increase (convergence)

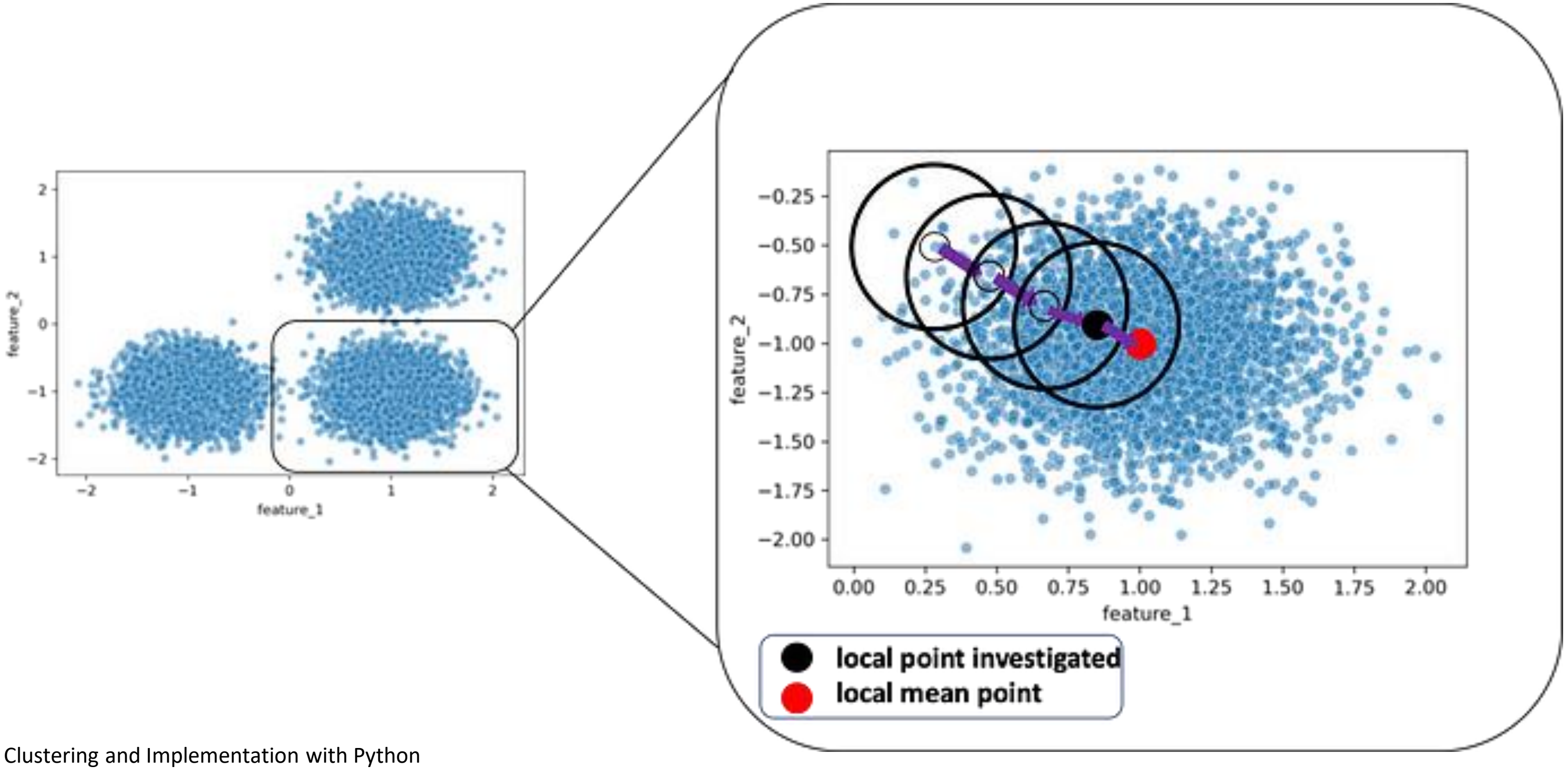
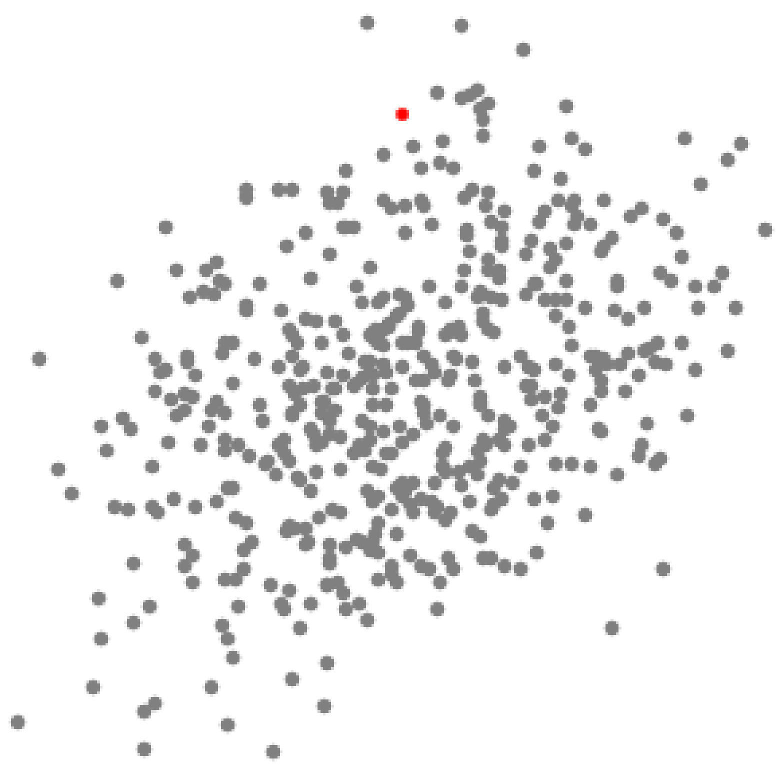


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm Steps

- Initialize data points as cluster centers
- Repeat until convergence
  - Compute the mean of all data points within a certain radius (kernel)
  - Shift the data to the mean
  - **Identify the cluster centroids as points that have not moved**
  - **Return cluster assignments**



Gif from: D. Sheehan , Clustering with Scikit with GIFs

Mean Shift Algorithm Cluster Assignments

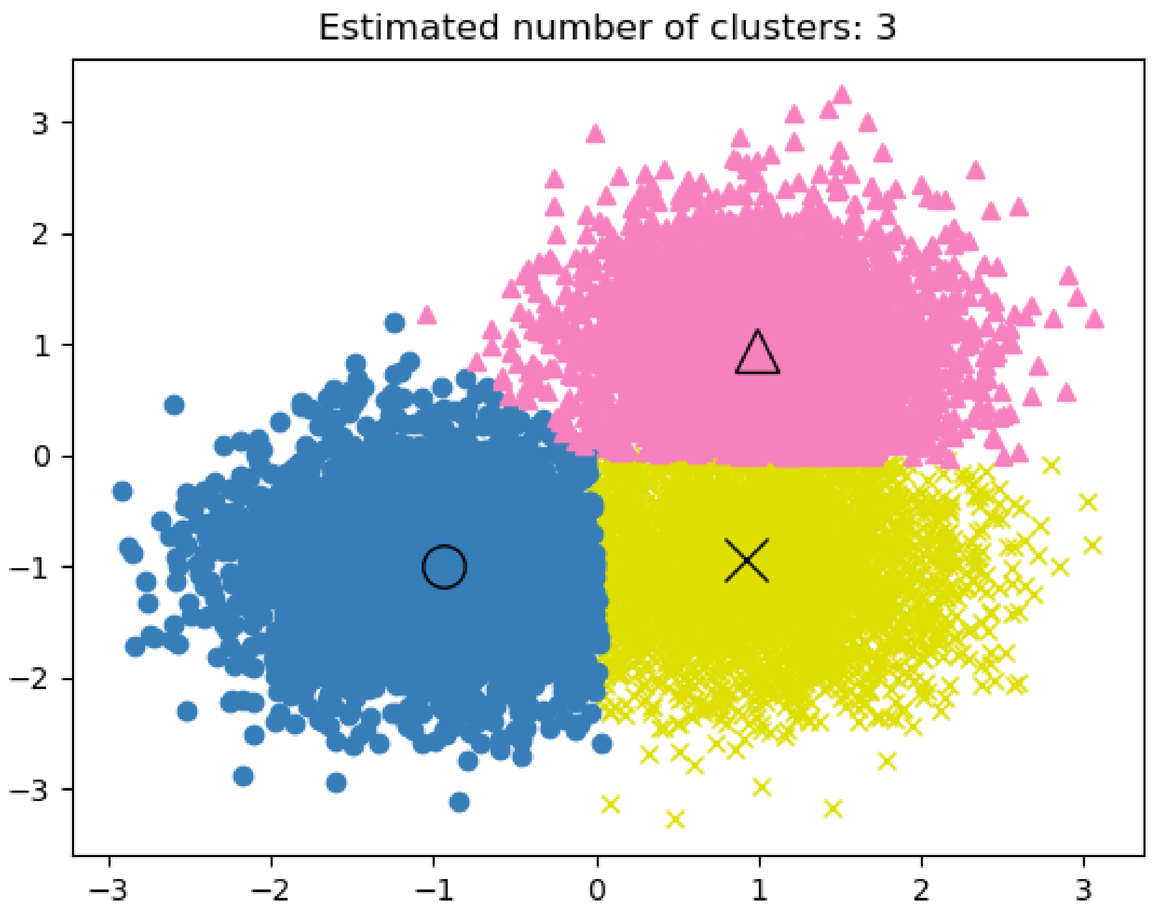


Image from: Sklearn, Mean Shift

Mean Shift Algorithm: Bandwidth

- Too small of a bandwidth can lead to “non-sense” clusters

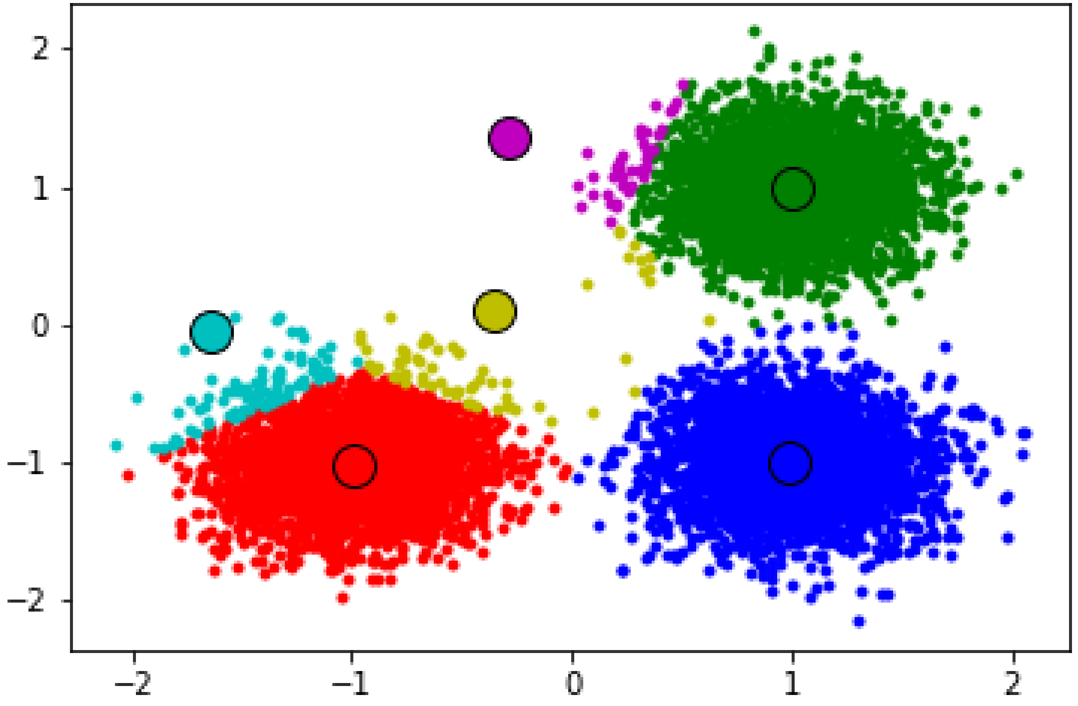
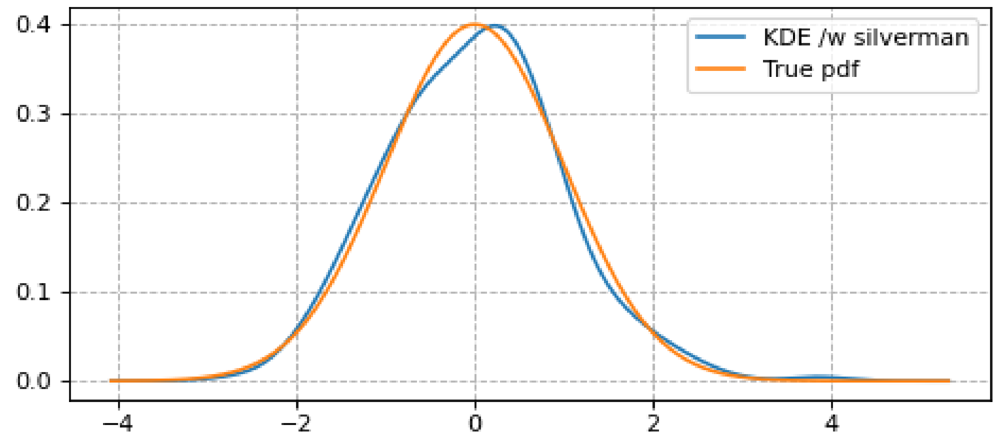


Image from: Yufeng, Understanding Mean Shift Clustering and Implementation with Python

Mean Shift Algorithm: Bandwidth Selection

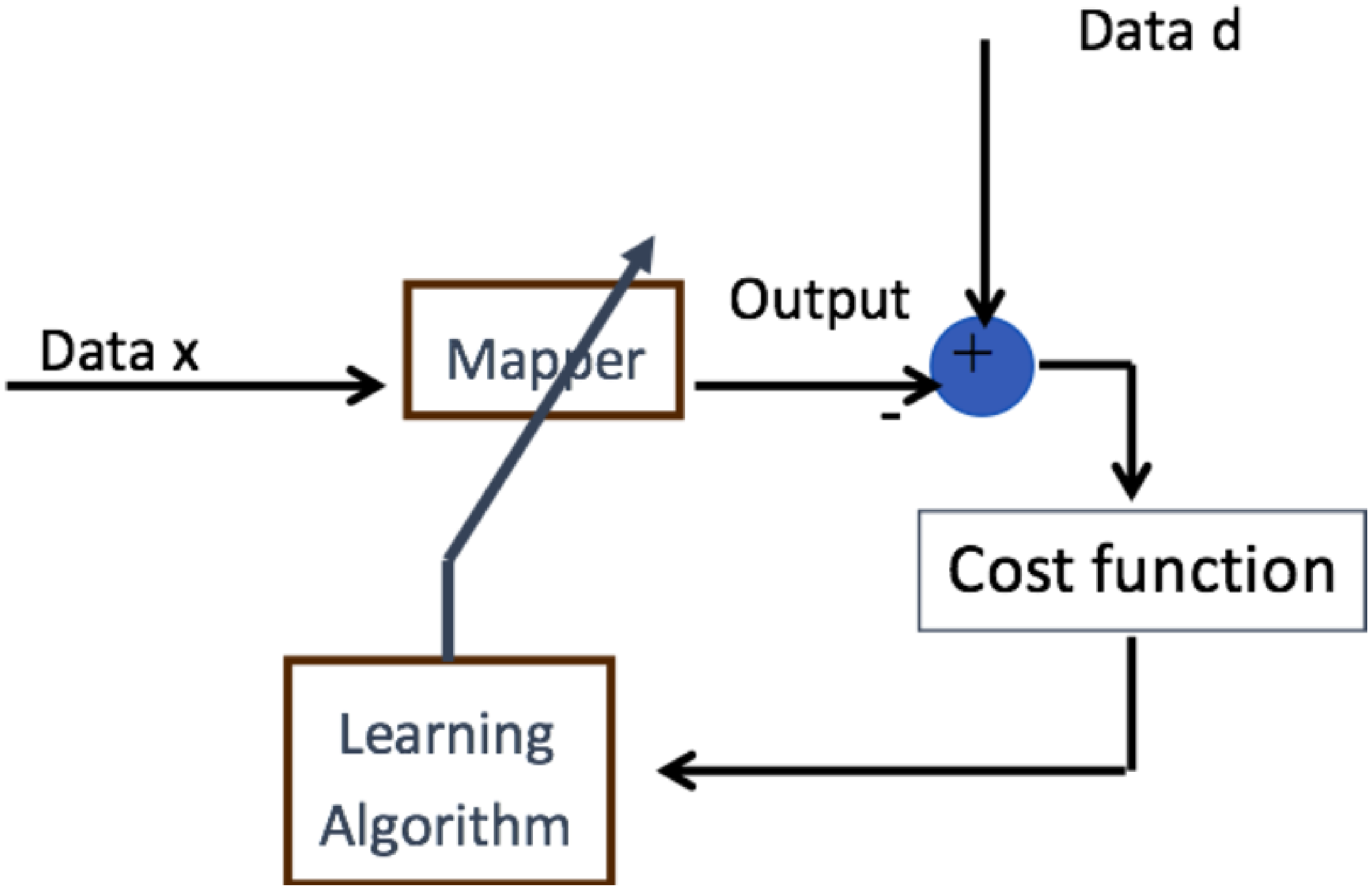
- Sklearn built-in function to estimate bandwidth based on data
- Silverman's rule
  - Assumes Gaussian distribution
- Empirically
- Etc.



$$h = 0.9 \min\left(\hat{\sigma}, \frac{IQR}{1.34}\right) n^{-\frac{1}{5}}$$

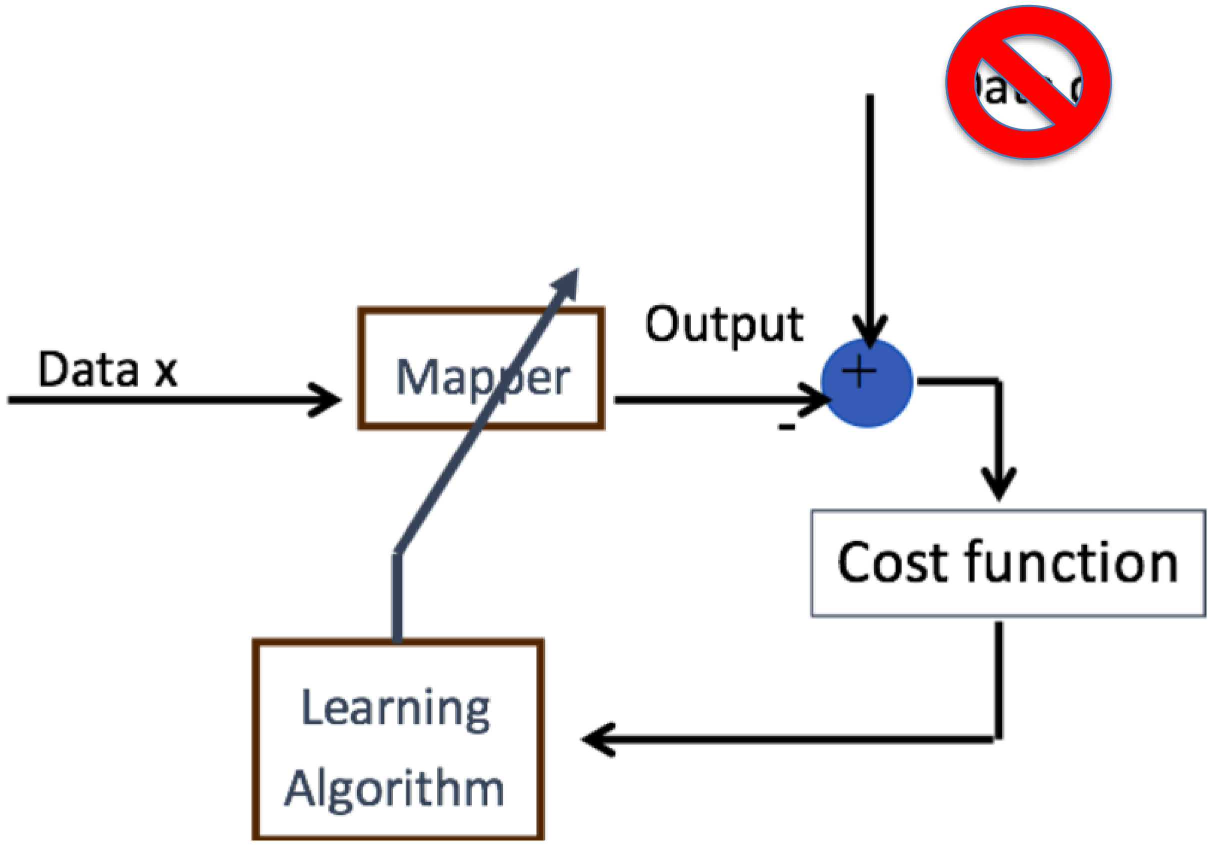
Image from: KDEpy

Density-based Clustering Machine Learning Model



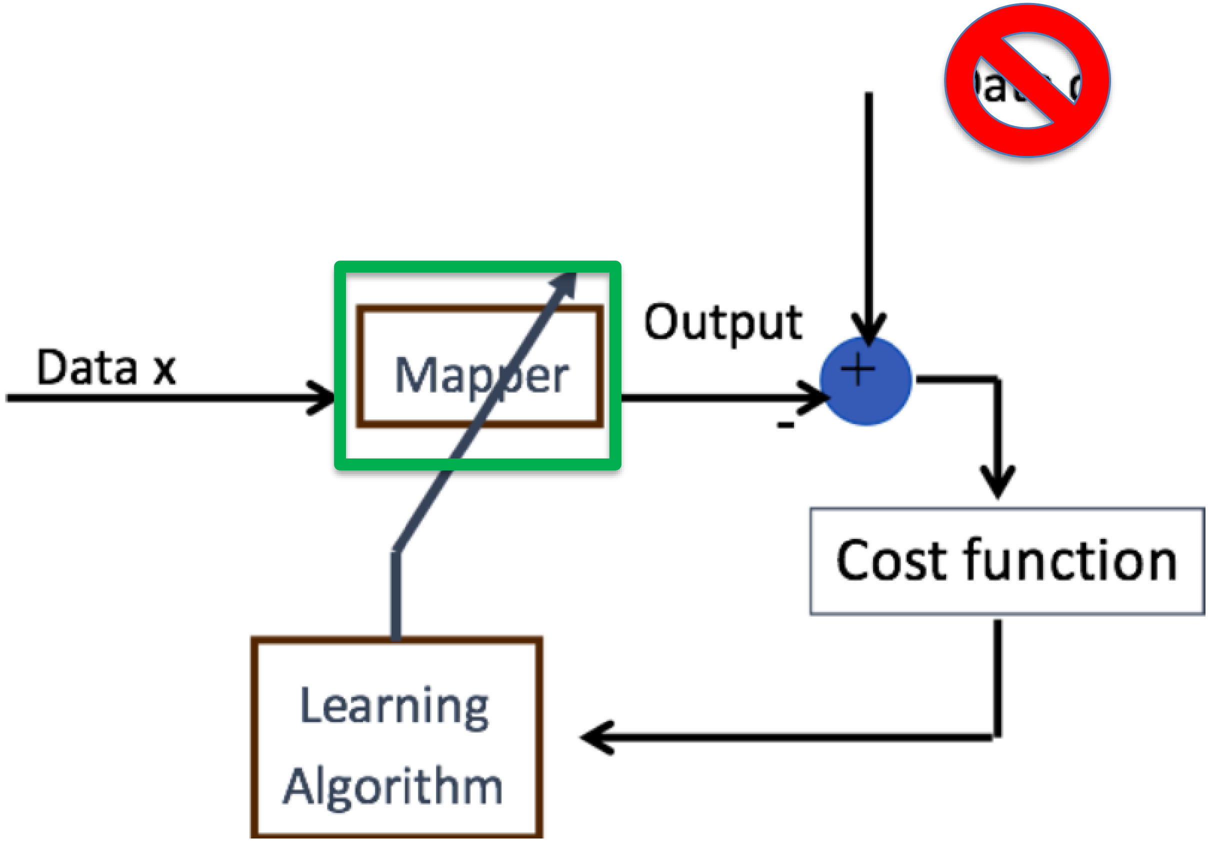
Density-based Clustering Machine Learning Model

- Unsupervised: No labels, *d*



Density-based Clustering Machine Learning Model

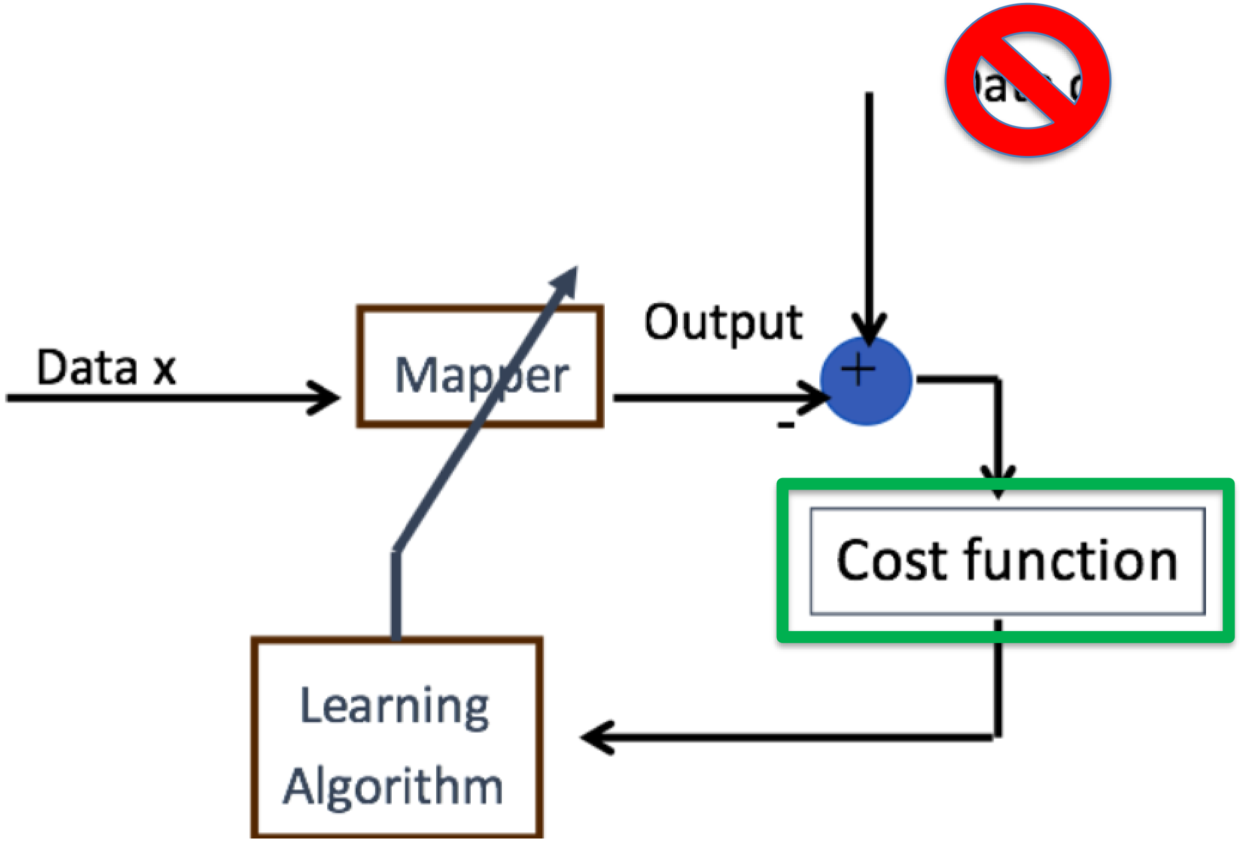
- Unsupervised: No labels, *d*
- **Mapper:**
  - Density-based Clustering
  - Takes input data and groups into *k* clusters



Density-based Clustering Machine Learning Model

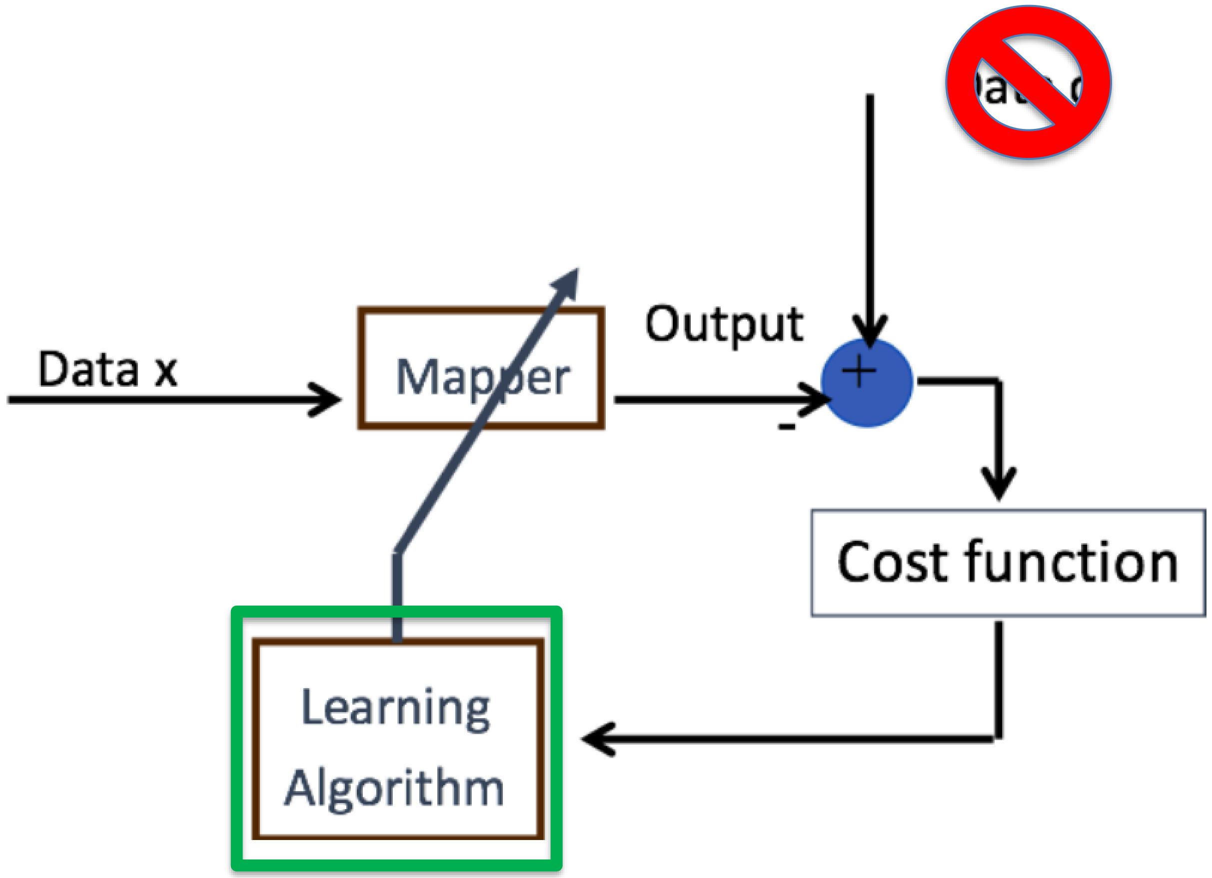
- Unsupervised: No labels, *d*
- Mapper:
  - Density-based Clustering
  - Takes input data and groups into *k* clusters
- **Cost function:**
  - Kernel function

$$\frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$



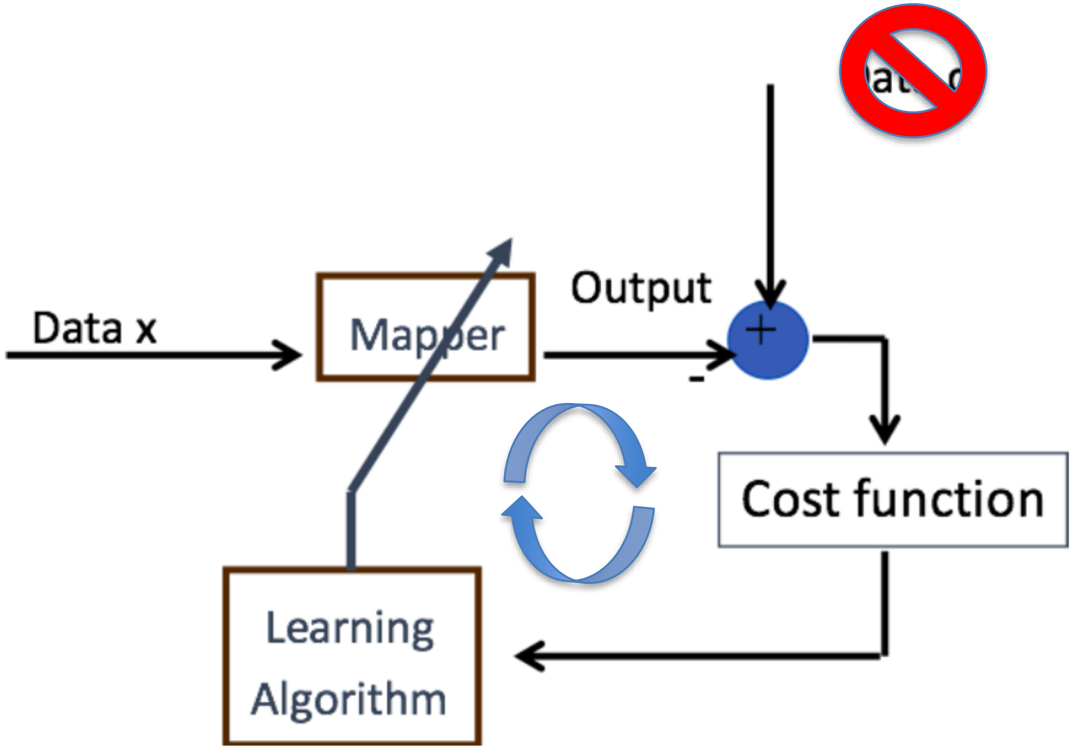
Density-based Clustering Machine Learning Model

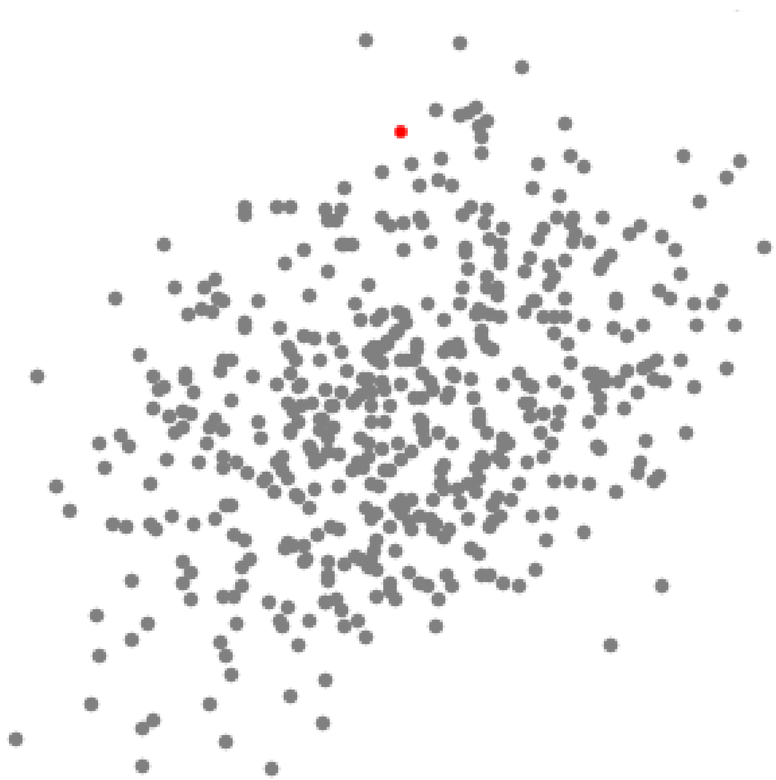
- Unsupervised: No labels, *d*
- Mapper:
  - Density-based
  - Takes input data and groups into *k* clusters
- Cost function:
  - Kernel function
- **Learning algorithm**
  - Maximal density
    - DBSCAN: Density connected points
    - MeanShift: Modes



Density-based Clustering Machine Learning Model

TEXAS A&M UNIVERSITY Engineering

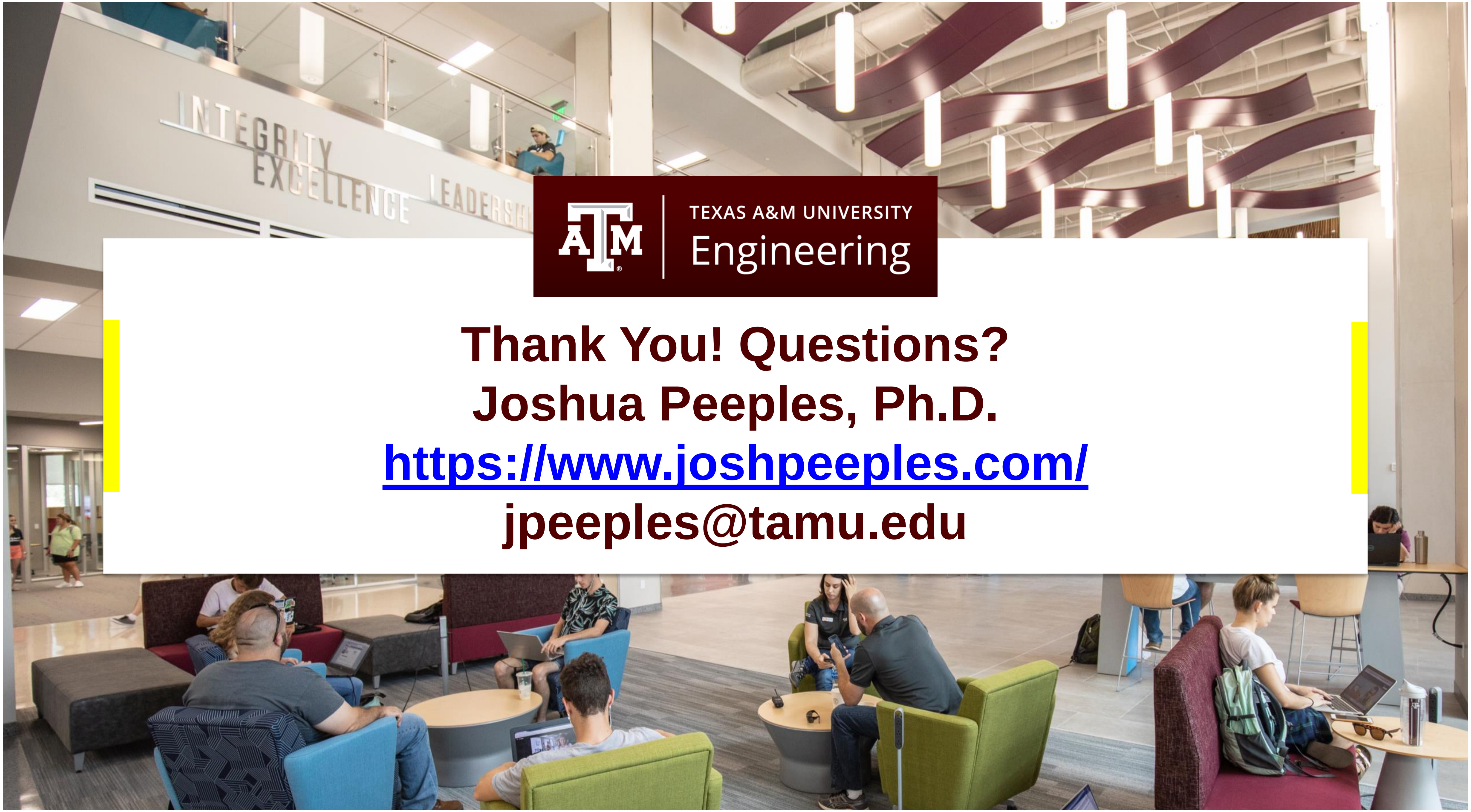




Gif from: D. Sheehan , Clustering with Scikit with GIFs

Next class

- Bayesian and Nearest Neighbor Classification



TEXAS A&M UNIVERSITY Engineering

Thank You! Questions?

Joshua Peeples, Ph.D.

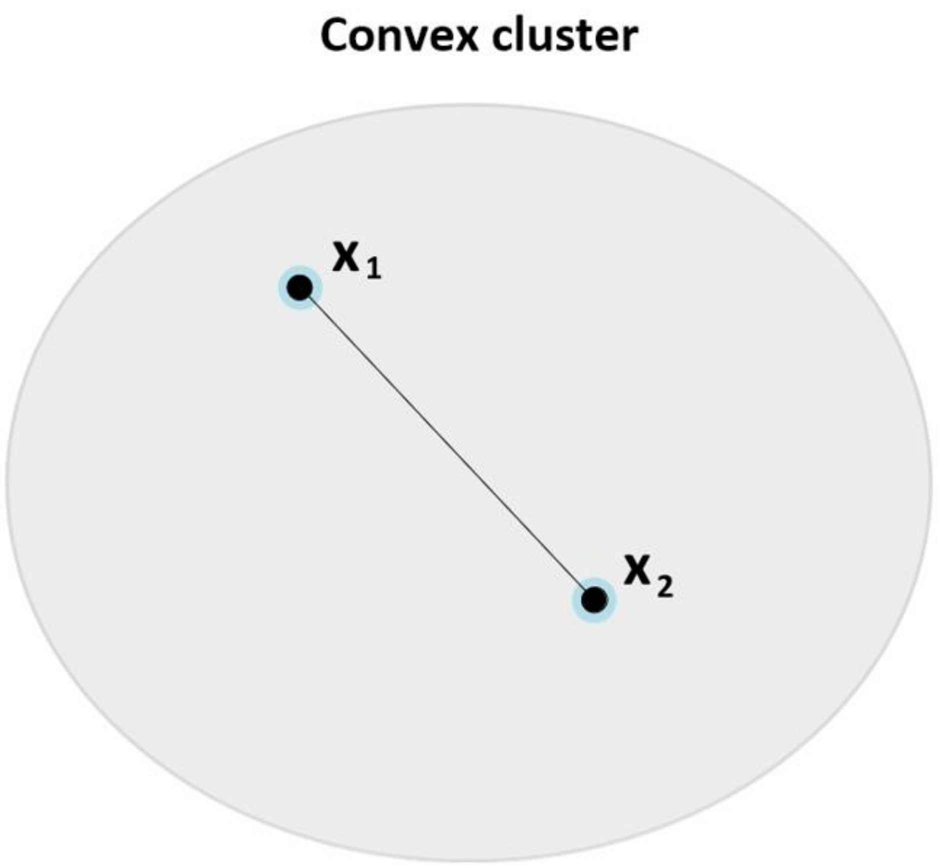
https://www.joshpeeples.com/

jpeeples@tamu.edu

TEXAS A&M UNIVERSITY
Engineering

Supplemental Slides

Convex vs Non-Convex Clusters



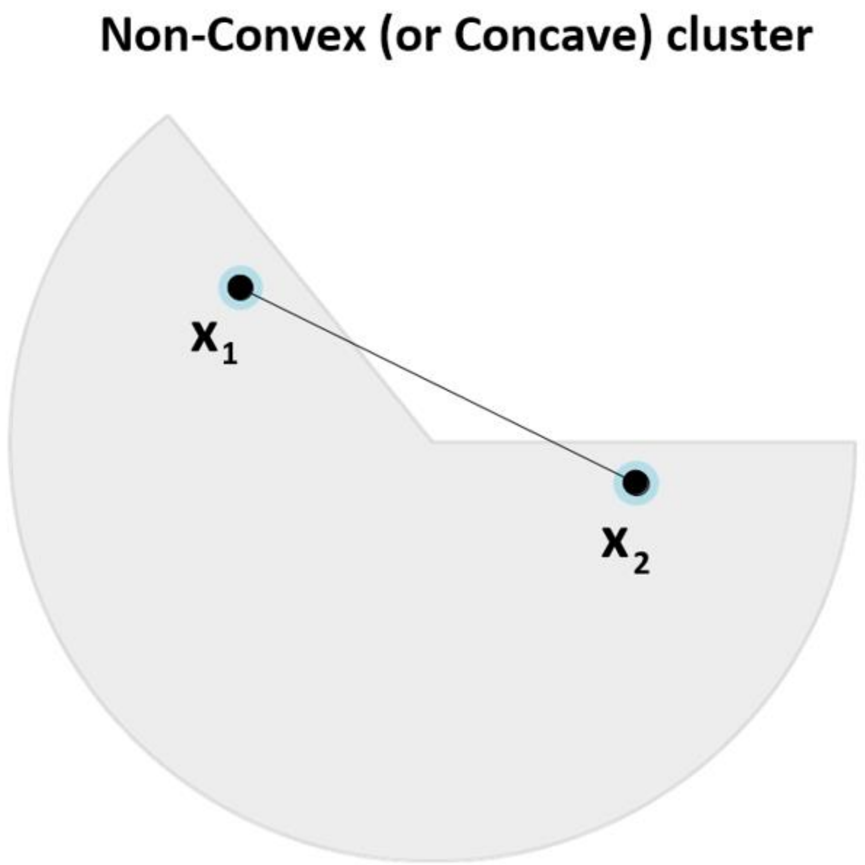


Image from: Deep Learning Bible: Clustering Fundamentals

Useful Links

- [StatQuest: DBSCAN]
- [Mean Shift Clustering Summary]
- [Kernel Density Estimation]
- [HDBSCAN]
- [OPTICS]