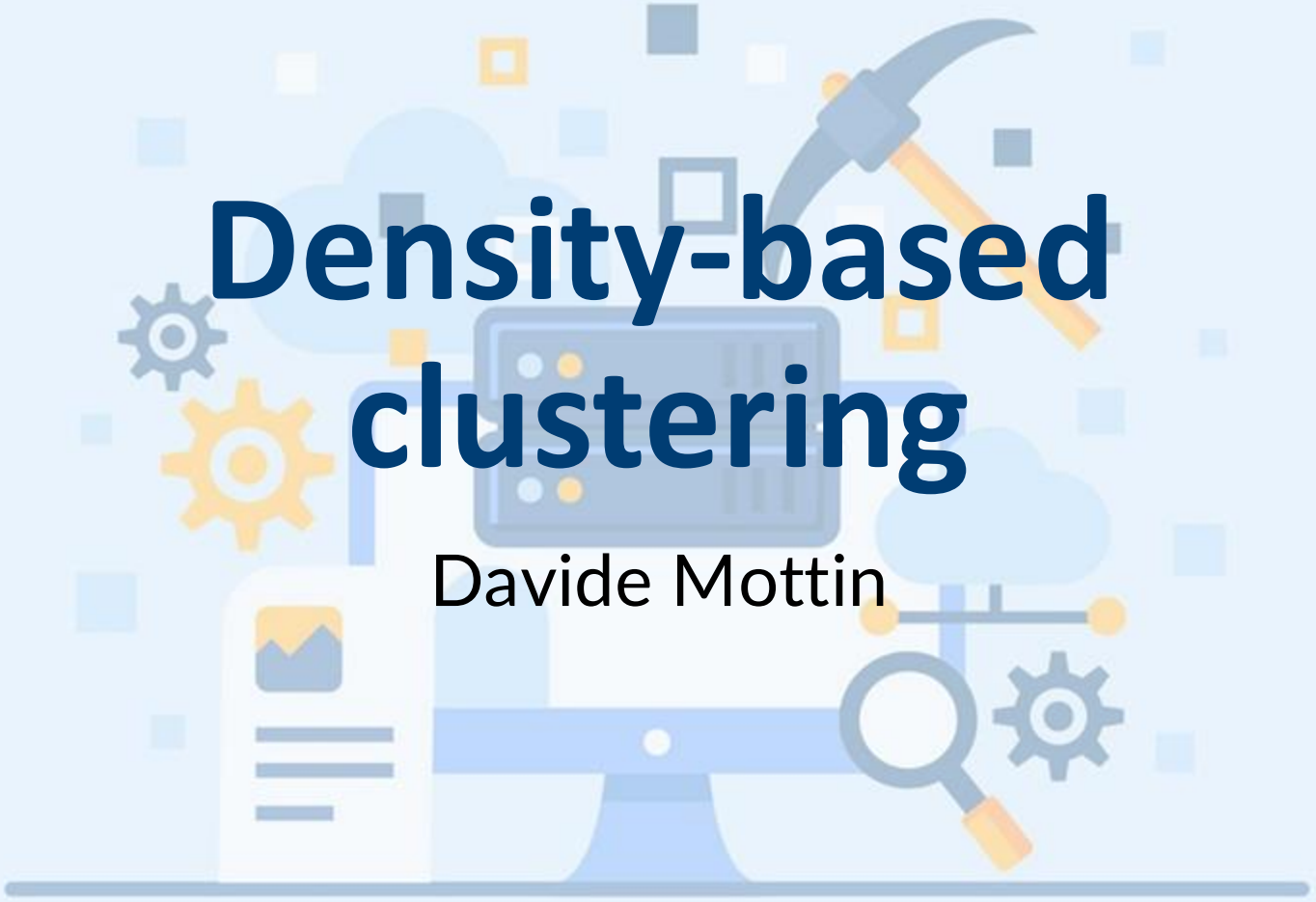


Density-based clustering

Davide Mottin

Data Mining

Review from last week

- Representative-based clustering
  - Clusters represented using mean, medoid
- Iterative refinement of initial clustering
  - Assignment of objects to representatives
  - Update of representatives
- K-means
- K-medoid
- EM

EM Algorithm

ClusteringByExpectationMaximization (point set *D*, int *k*)

Generate an initial model $M' = (C_1', \ldots, C_k')$

repeat

// assign points to clusters – **expectation step**

For each object $\mathbf{x}_j$ from D and for each cluster (= Gaussian) C_i

Compute $P(C_i|\mathbf{x_j}) = \mathrm{W_{ij}}$,

// compute the model - **maximization step**

For each Cluster $\boldsymbol{C_i}$

Compute a new model $M = \{C_1, \ldots, C_k\}$ by **recomputing** $\boldsymbol{P(C_i), \mu_{C_i}, \Sigma_{C_i}}$

$M' \leftarrow M$

until $\sum_{i=1}^{k} \left\| \mu_{C_i}^{t} - \mu_{C_i}^{t-1} \right\| \leq \varepsilon$

return M

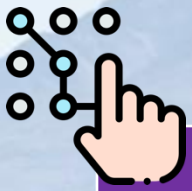
Roadmap

Clustering

- Representative-based
- Density-based
- Hierarchical and Subspace
- Outlier detection

Graph Mining

- Spectral Theory and clustering
- Community Detection
- Link Analysis
- Similarities and Graph Embeddings
- Graph Convolutional Networks



Pattern mining

- Frequent subgraph mining
- Frequent Items and Association Rules
- Sequence Mining
- Similarities and Stream Mining

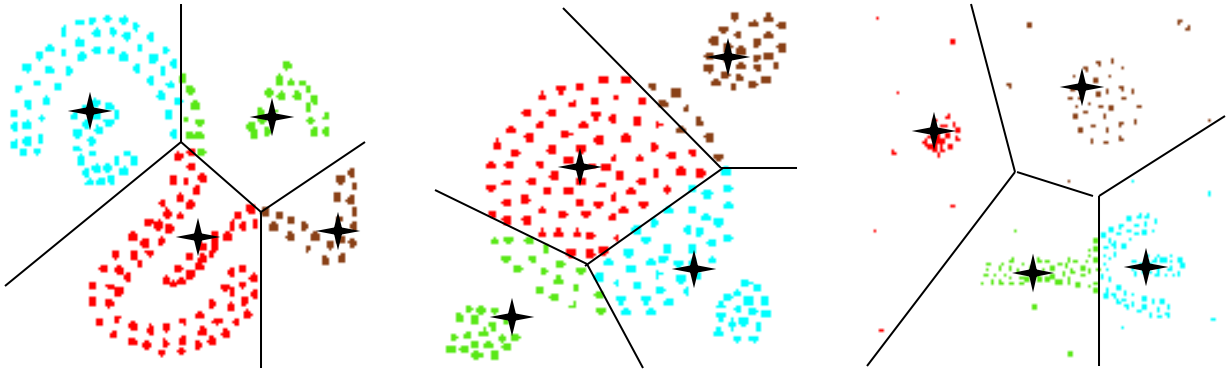
Icons made by Flat Icons, Freepik from www.flaticon.com

Learning goals for today

- What characterizes density-based approaches?
- Main algorithms: DBSCAN, DENCLUE
- Evaluation measures for clustering

Density-Based Clustering

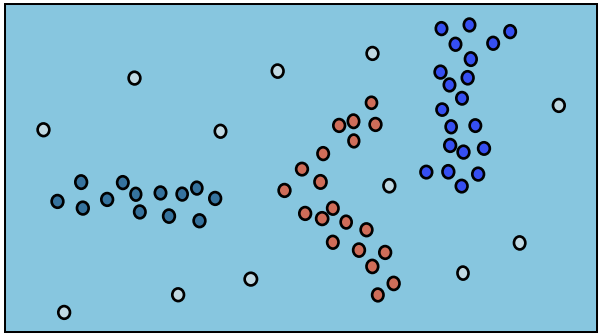
- Why Density-Based Clustering?



Results of a *k*-medoid algorithm for *k*=4

Basic Idea:

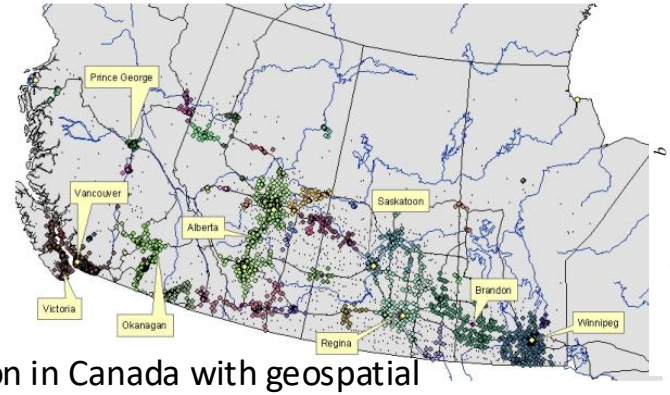
- Clusters are **dense regions** in the data space, separated by regions of lower object density



Different density-based approaches exist

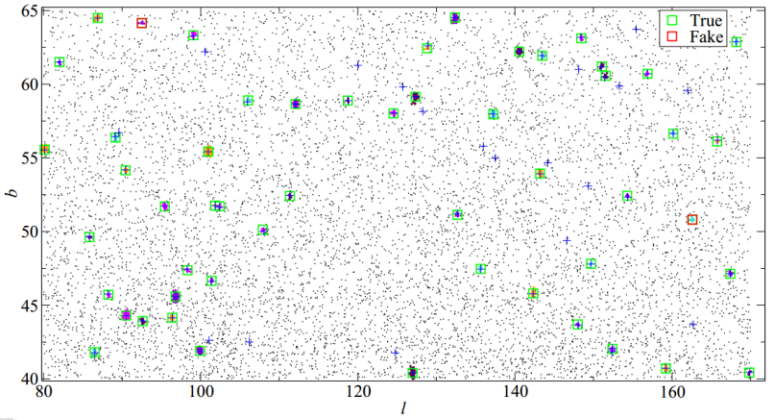
Next we discuss the ideas underlying the original DBSCAN algorithm

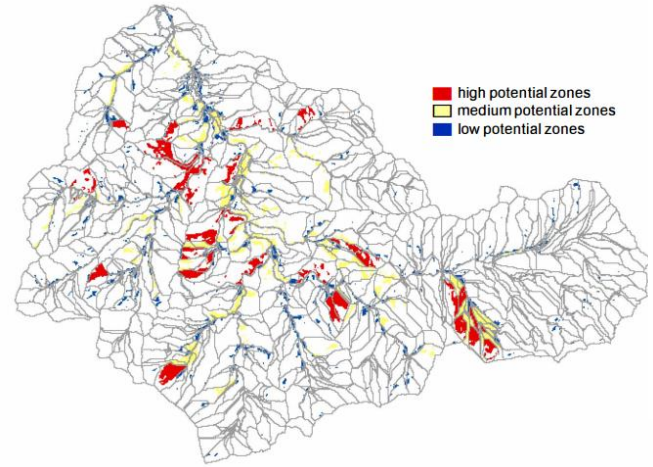
Applications



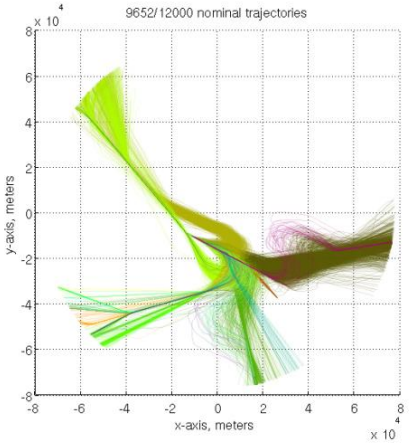
Population in Canada with geospatial constraints (Wang et al. 2004)

Arrival directions of photons in γ-ray astrophysical images (Tramacere et al. 2013)

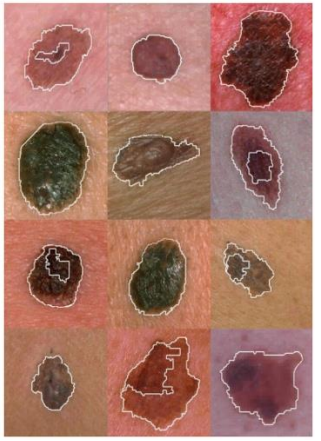




Detecting areas with landslide hazard in Taiwan (Hwang et al. 2012)



Monitoring of airspace moving trajectories (Gariel et al. 2010)



Identifying homogenous color regions in biomedical images (Celebi et al. 2005)

DBSCAN

Neighbours are friends

Density Based Clustering: Basic Concept

Intuition

- For any point in a cluster, the **local point density** around that point has to exceed some threshold
- The set of points from one cluster is spatially connected

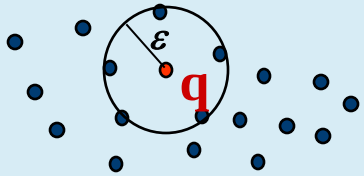
Local point density at a point *p* defined by two parameters

- ε : radius for the neighborhood of point q,

$$N_\varepsilon(q) := \{p \in D \mid dist(p,q) \leq \varepsilon\}$$

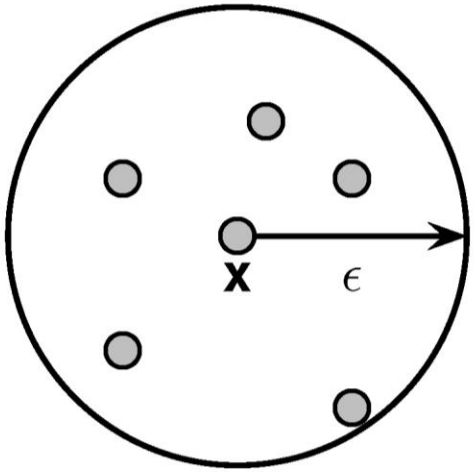
- ***MinPts***: minimum number of points in the given neighbourhood $N_\varepsilon(p)$

q is called a ***core object*** (or core point) w.r.t. ε, *MinPts* if $|N_\varepsilon(q)| \geq MinPts$

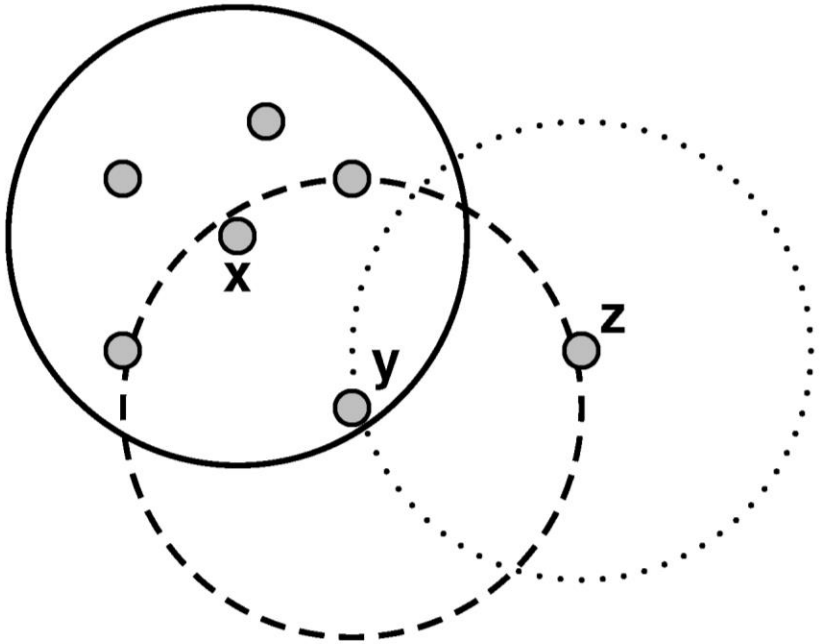


MinPts = 5 → **q** is a core object

Explanation of main notions



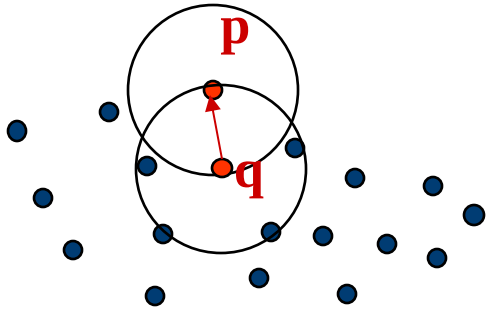
(a) Neighborhood of a Point



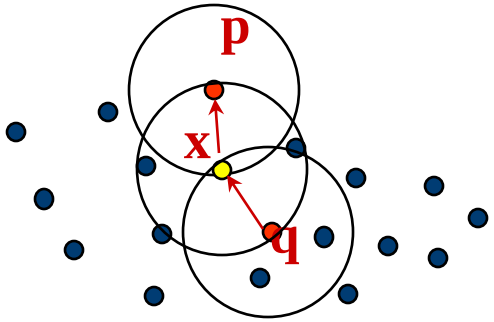
(b) Core, Border, and Noise Points

Density Based Clustering: Basic Definitions

- p **directly density-reachable** from q within ε, MinPts
  - $p \in N_\varepsilon(q)$
  - q is a core object w.r.t. ε, MinPts

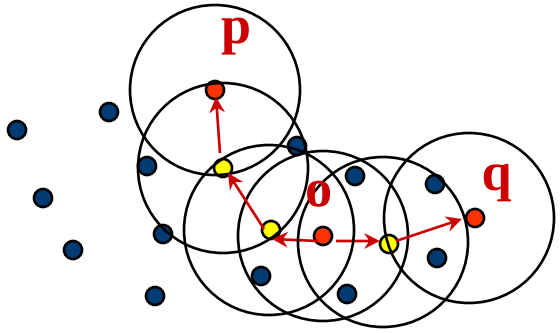


- **density-reachable:** transitive closure of directly density-reachable



Note that density-reachability is asymmetric

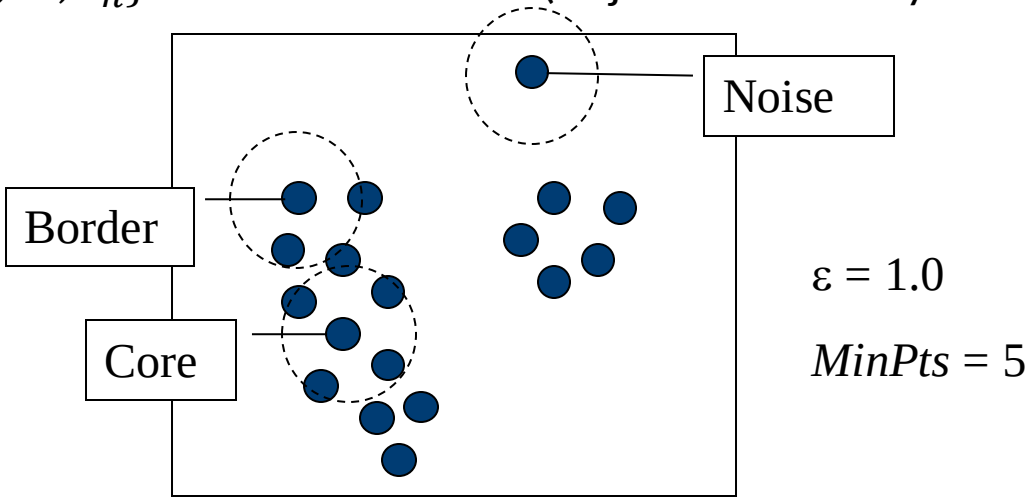
- p is **density-connected** to a point q w.r.t. ε, MinPts if there is a point o such that both, p and q are density-reachable from o w.r.t. ε, MinPts.



Density Based Clustering: Basic Definitions

- **Density-Based Cluster:** non-empty subset S of database D satisfying:
  1) Maximality: if p is in S and q is density-reachable from p then q is in S
  2) Connectivity: each object in S is density-connected to all other objects

- **Density-Based Clustering of a database** D : $\{S_1, \ldots, S_n; N\}$ where
  - $S_1, \ldots, S_n$: all density-based clusters in the database D
  - $N = D \setminus \{S_1, \ldots, S_n\}$ is called the **noise** (objects not in any cluster)



Density Based Clustering: DBSCAN Algorithm

Border points can be assigned to one or multiple clusters

Basic Theorem

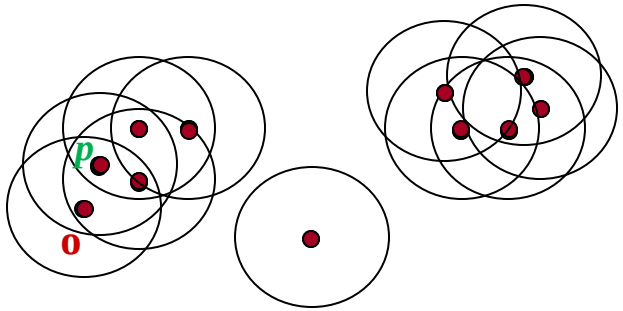
- Each object in a density-based cluster C is density-reachable **from any of its core-objects**
- Nothing else is density-reachable from core objects.

```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

- density-reachable objects are collected by performing successive ε-neighborhood queries.

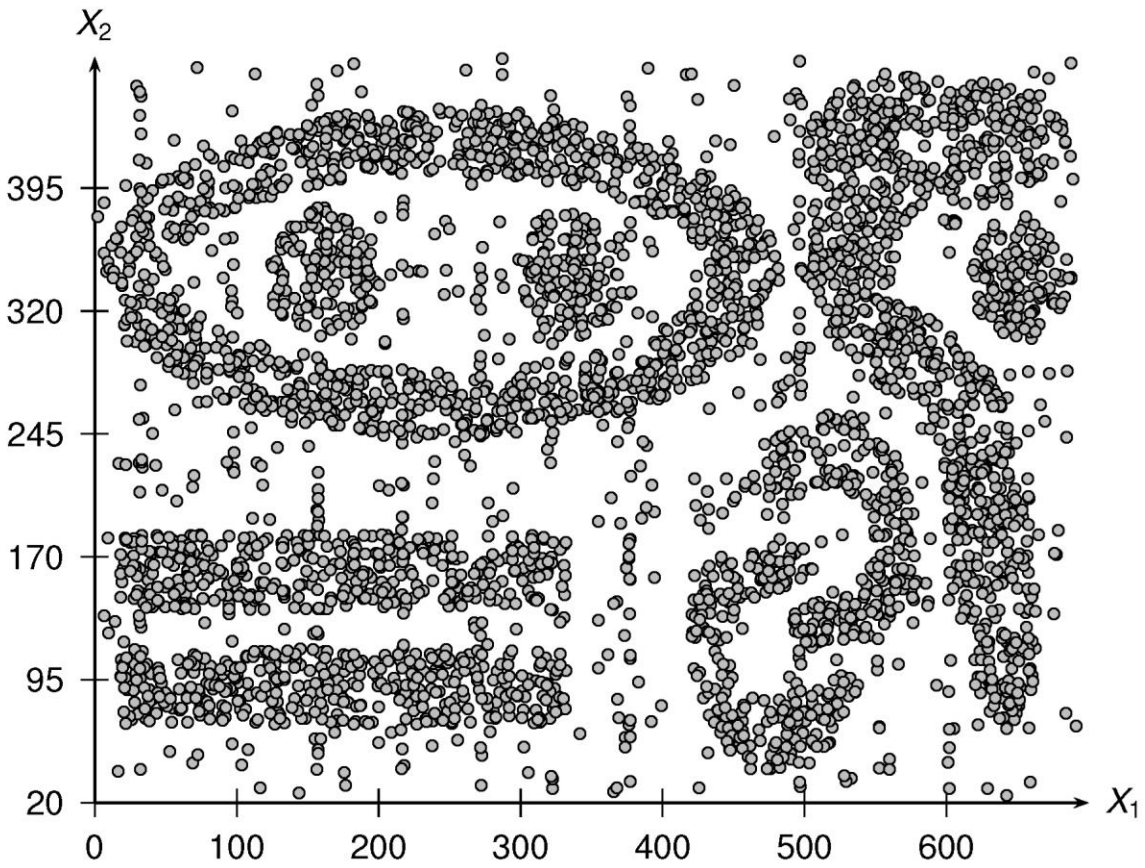
DBSCAN Algorithm: Example

- Parameters
  - $\varepsilon = 2.0$
  - MinPts = 3

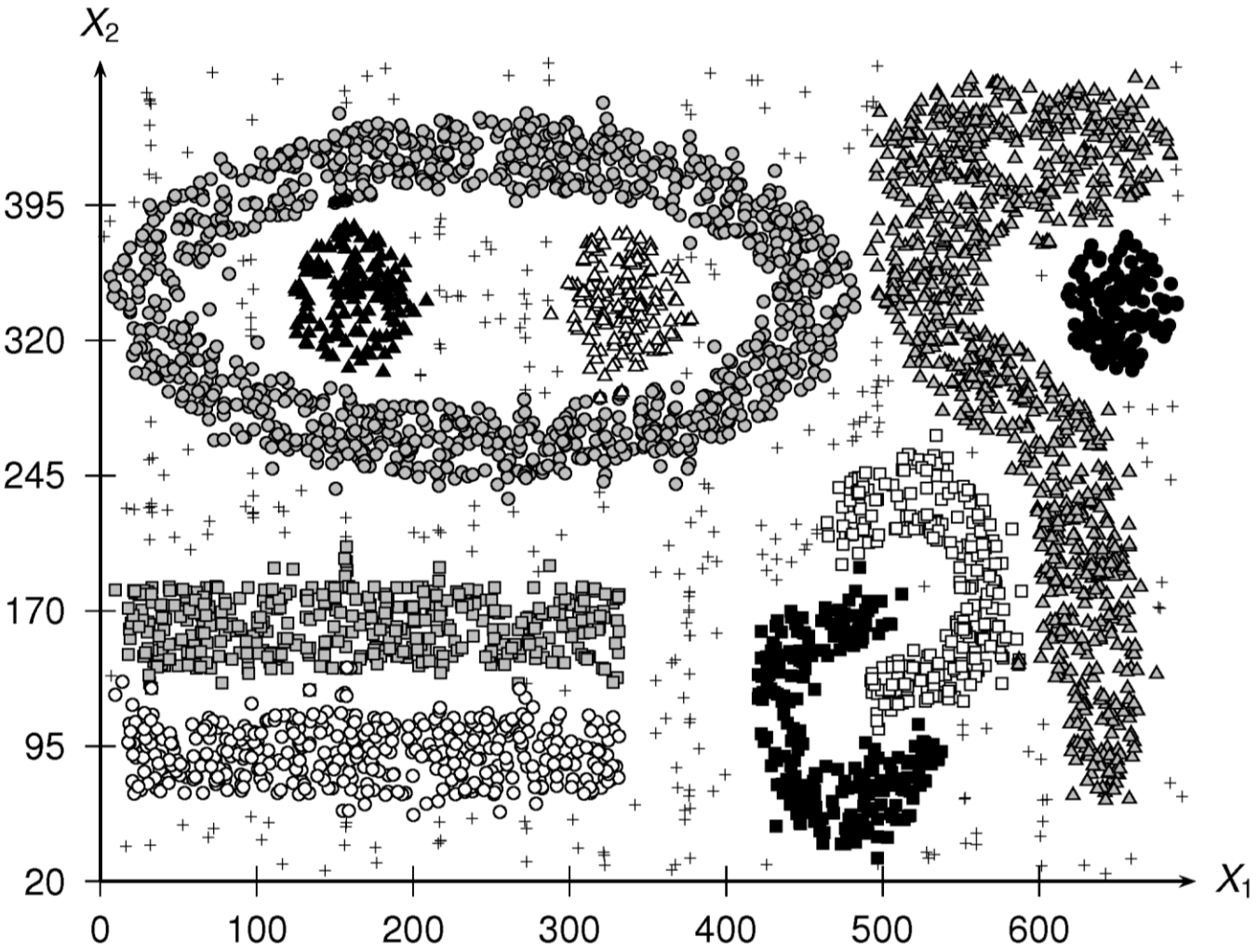


for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

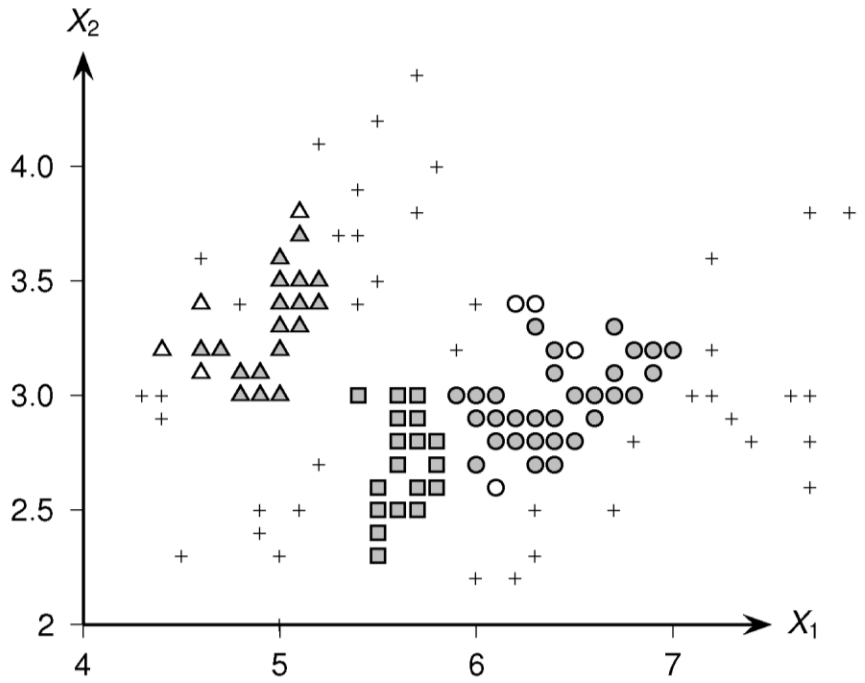
Synthetic data example in 2D for density-based clustering



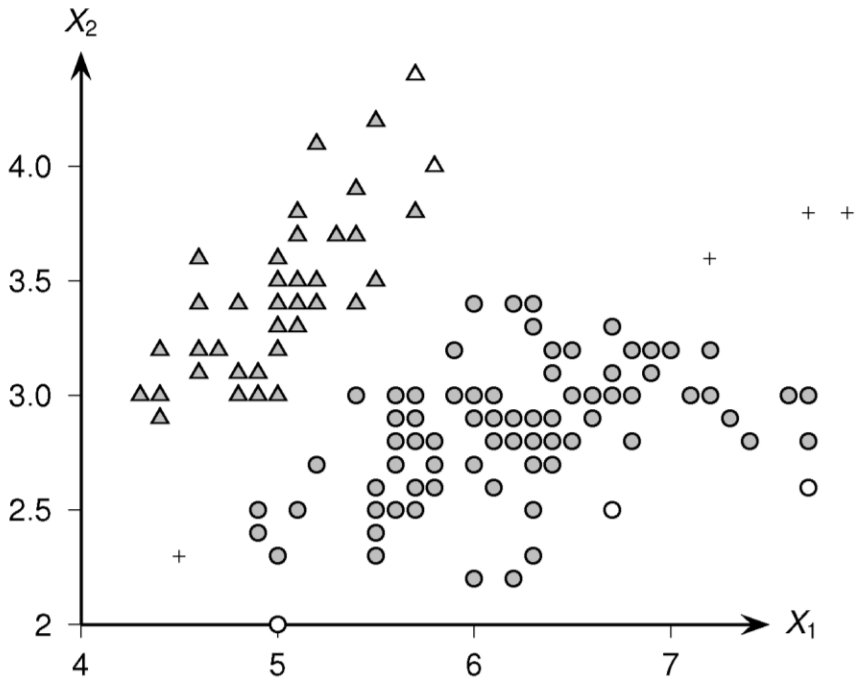
DBSCAN, ε=15, minpts=10



DBSCAN, Iris data (sepal length, sepal width)



(a) $\epsilon = 0.2$, $minpts = 5$



(b) $\epsilon = 0.36$, $minpts = 3$

DBSCAN Algorithm: Performance

- **Runtime complexity:** O(n * cost for neighborhood query)

	N_ε-query	DBSCAN
- without support (worst case):	$O(n)$	$O(n^2)$
- tree-based support (e.g. R*-tree) :	$O(\log(n))$	~~$O(n \log(n))$~~ *
- direct access to the neighborhood:	$O(1)$	$O(n)$

$\Omega(n^{\frac{4}{3}})$

When dimensionality is high DBScan is $O(n^2)$

Runtime Comparison:

- DBSCAN (+ R*-tree) $\leftrightarrow$ CLARANS ($O(k^3 + nk)$)

	Time (sec.)									
No. of Points	1,252	2,503	3,910	5,213	6,256	7,820	8,937	10,426	12,512	62,584
DBSCAN	3	7	11	16	18	25	28	33	42	233
CLARANS	758	3,026	6,845	11,745	18,029	29,826	39,265	60,540	80,638	?????

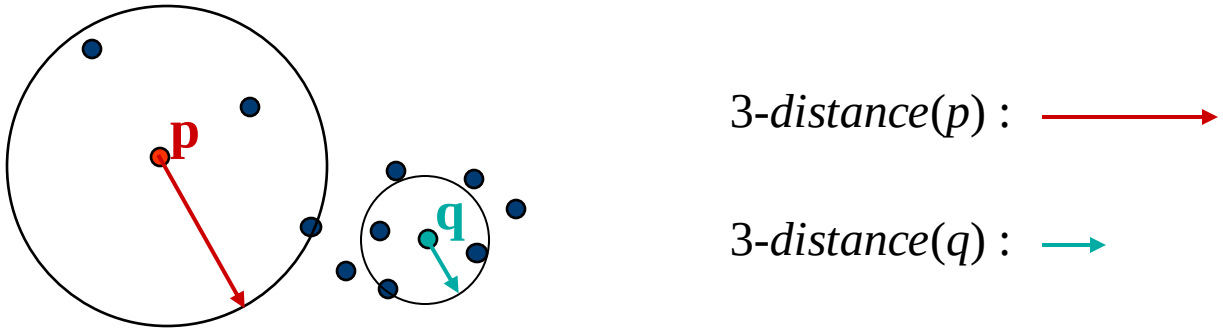
* Popular misclaim: Gan and Tao, "DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation"

Tune Parameters ε and *MinPts*

Cluster: Point density higher than specified by ε and *MinPts*

Idea: use the point density of the least dense cluster in the data set as parameters – but how to determine this?

Heuristic: look at the distances to the *k*-nearest neighbors

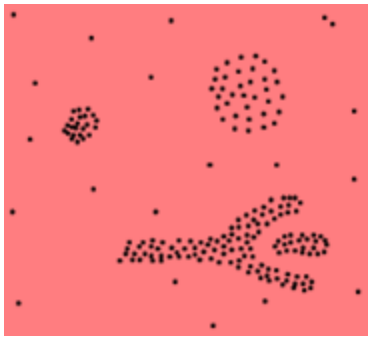


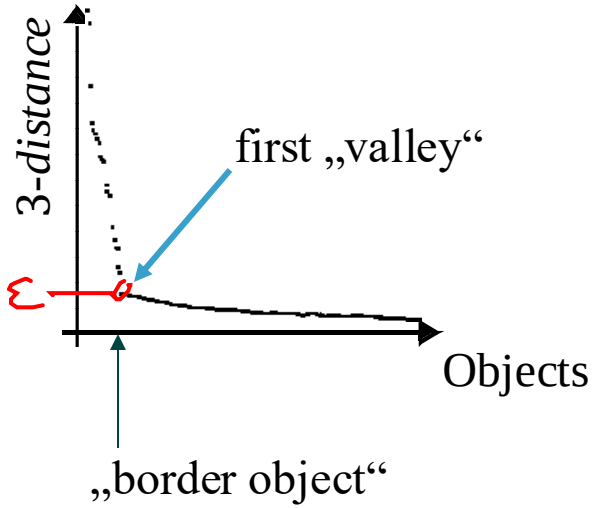
- Function *k-distance(p)*: distance from *p* to the its *k*-nearest neighbor
- *k-distance plot*: *k*-distances of all objects, sorted in decreasing order

Determining the Parameters ε and *MinPts*

- **Heuristic method:**
  - Fix a value for *MinPts*
  - (default: $2 \times d - 1$, d = dimension of data space)
  - User selects "border object" *o* from the *MinPts-distance* plot; ε is set to *MinPts-distance*(o)

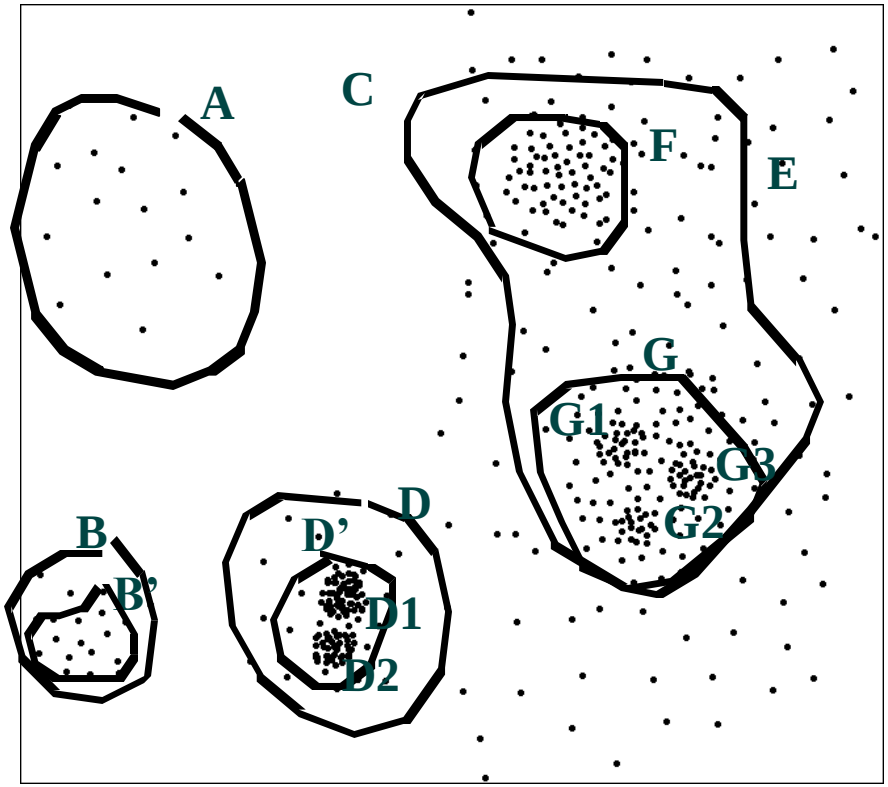
- Example *k*-distance plot
  1. dim = 2 $\rightarrow$ MinPts = 3
  2. Identify border object
  3. Set ε

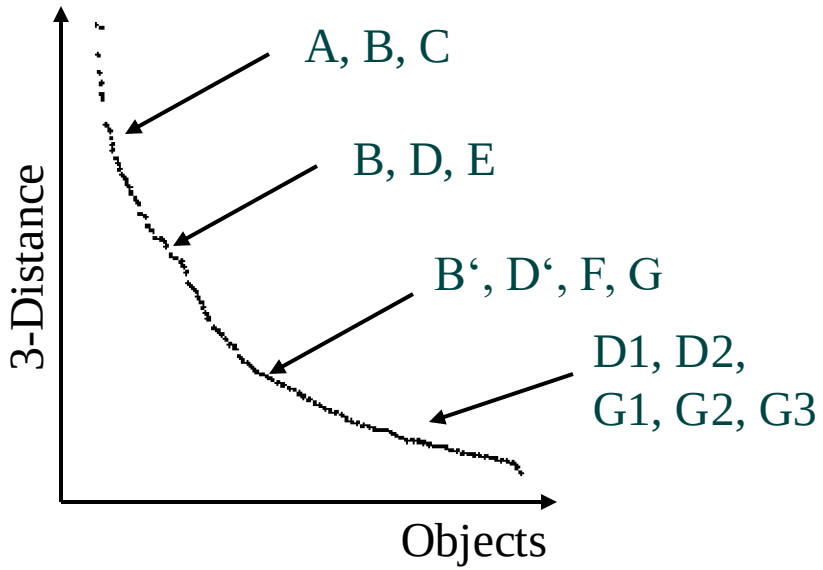




Determining the Parameters ε and *MinPts*

- Problematic example





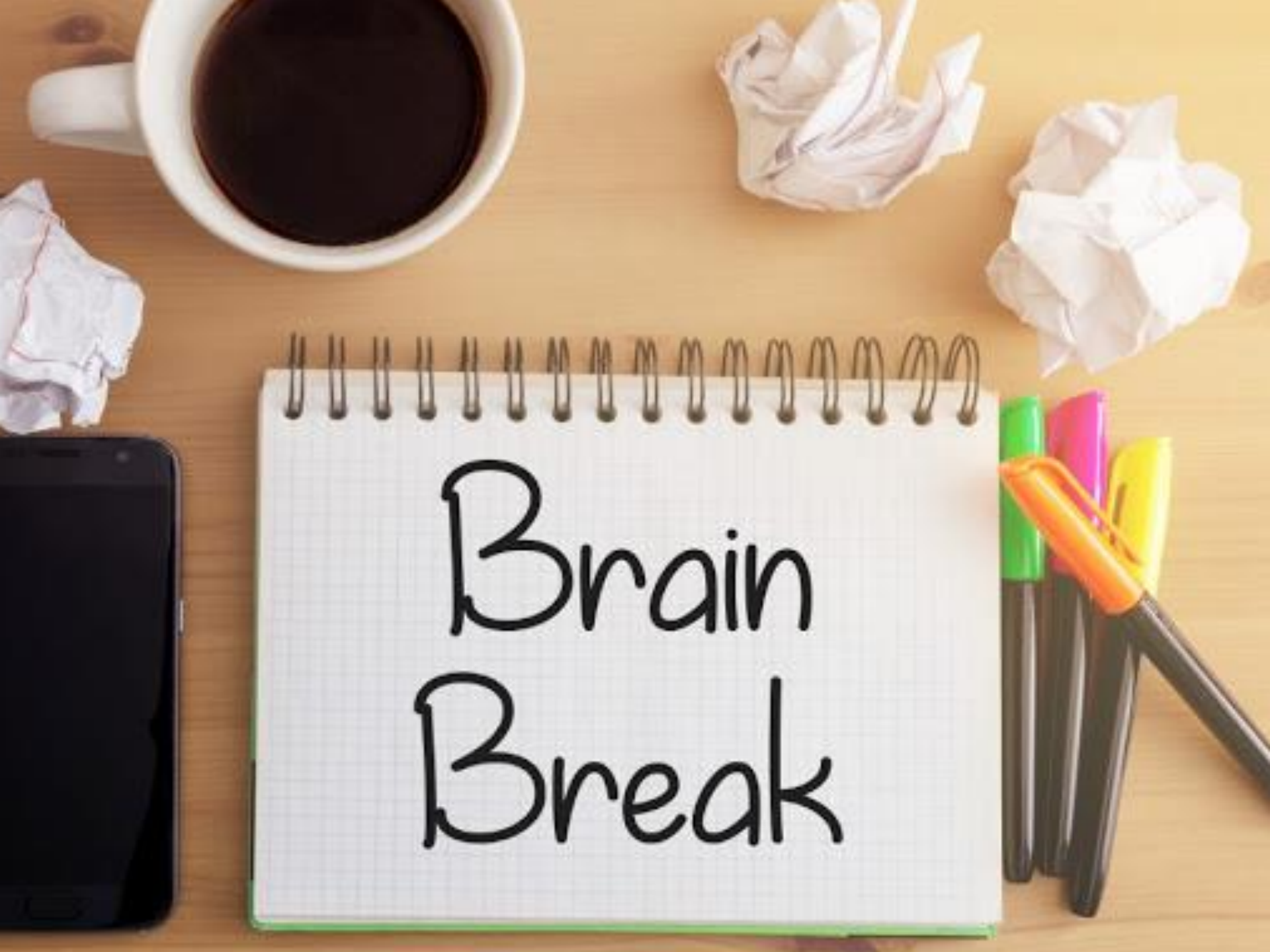
DBSCAN Discussion

Advantages

- Does not require to specify number of clusters beforehand.
- Performs well with arbitrary shapes clusters.
- DBSCAN is robust to outliers and able to detect them.

Disadvantages

- Determining an appropriate distance of neighborhood (ε) is not easy and it requires domain knowledge.
- If clusters are very different in terms of in-cluster densities, DBSCAN is not well suited to define clusters. DBSCAN cannot generalize well to clusters with much different densities.


Brain
Break

DENCLUE

Let's get rid of those nasty ε and minPts

Generalized density-based clustering

- Density assessment in DBSCAN uses a relatively simple model ☹
  - Neighborhood range parameter and MinPts threshold

Idea: DENCLUE generalizes this notion by considering the overall density distribution
  - Uses density estimation techniques

- **Density-estimation**
  - Determine the **unknown** probability density function
  - Nonparametric technique, does not assume fixed probability model of clusters
  - Tries to determine probability density at each point in the dataset
  - As we will see later, DBSCAN actually uses a simplified version of this approach

Kernel density estimation

Strong connection between density-based clustering and density estimation.

Density estimation:

- Determine unknown density function
- Locate dense regions of points
  - Used for clustering

Kernel density:

- **Non parametric** technique that infers density at each point
- Non parametric = no fixed probability model

Univariate density estimation

One dimension

- model the data as random variable X
- Data points are considered observations $x_1, x_2, \ldots, x_n$

- Estimate the **cumulative distribution function** by counting how many points are less than or equal to x

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

  - Where I is an indicator function

- **Density function** is estimated by taking the derivative

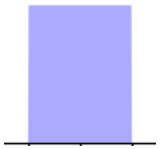
$$\hat{f}(x) = \frac{\hat{F}\left(x + \frac{h}{2}\right) - \hat{F}\left(x - \frac{h}{2}\right)}{h} = \frac{k/n}{h} = \frac{k}{nh}$$

  - Where k is the number of points in a window of width h centered at point x
  - Density estimate is ratio of points in window (k/n) to volume of window (h)

Kernel Estimator

- Use kernel function K
  - non-negative, symmetric, integrates to 1
  - $K(x) \geq 0, K(-x) = K(x)$ for all values x, and $\int K(x)dx = 1$
- **Discrete kernel**

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



Basically a probability distribution function

where $|z| = \left|\frac{x - x_i}{h}\right|$ for window of width h

- Density estimator can be rewritten as

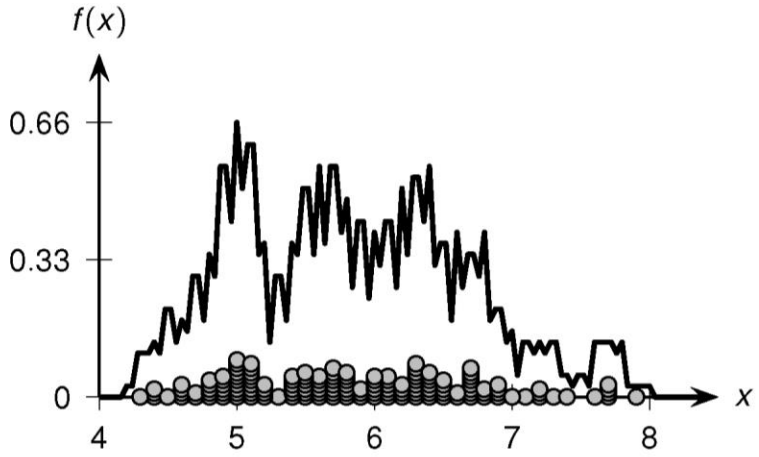
$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

To derive that remember

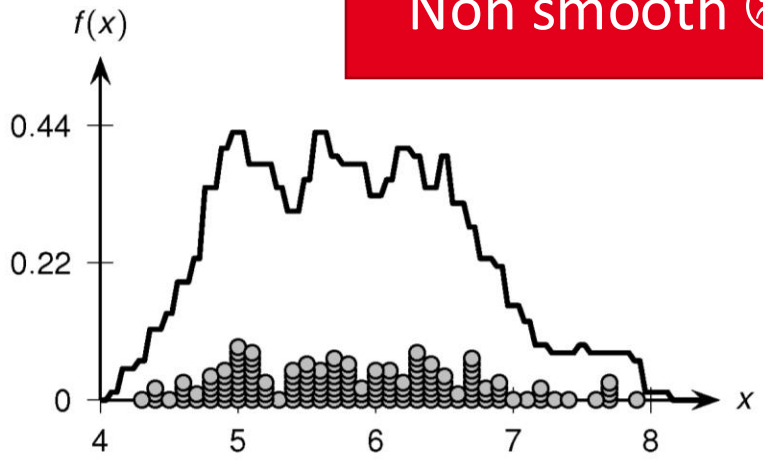
$$|z| = \left|\frac{x - x_i}{h}\right| \leq \frac{1}{2}$$

Density Estimation: discrete kernel (Iris 1D, sepal length)

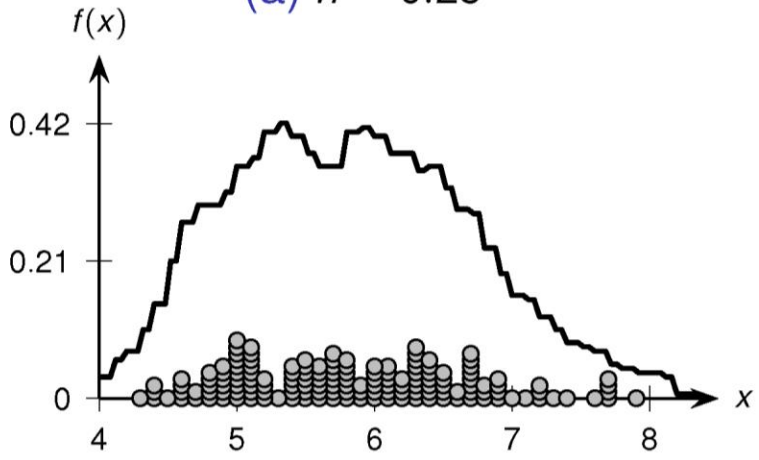
Non smooth ☹



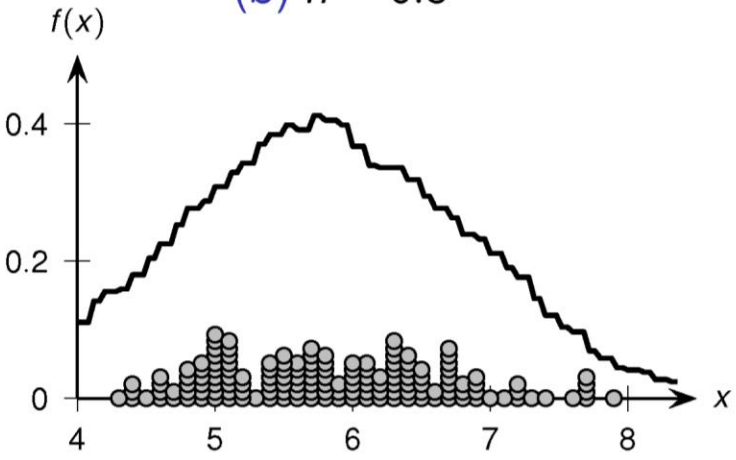
(a) $h = 0.25$



(b) $h = 0.5$



(c) $h = 1.0$



(d) $h = 2.0$

Gaussian Kernel: A smooth kernel

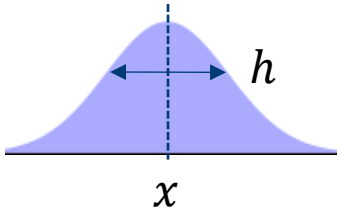
Discrete kernel
- Non-smooth estimation
- **Note:** This is the model underlying DBSCAN
  - Data points within the neighborhood range contribute **fully**, those outside not at all

DENCLUE uses Gaussian kernels for smoothed data point influence

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

- This yields

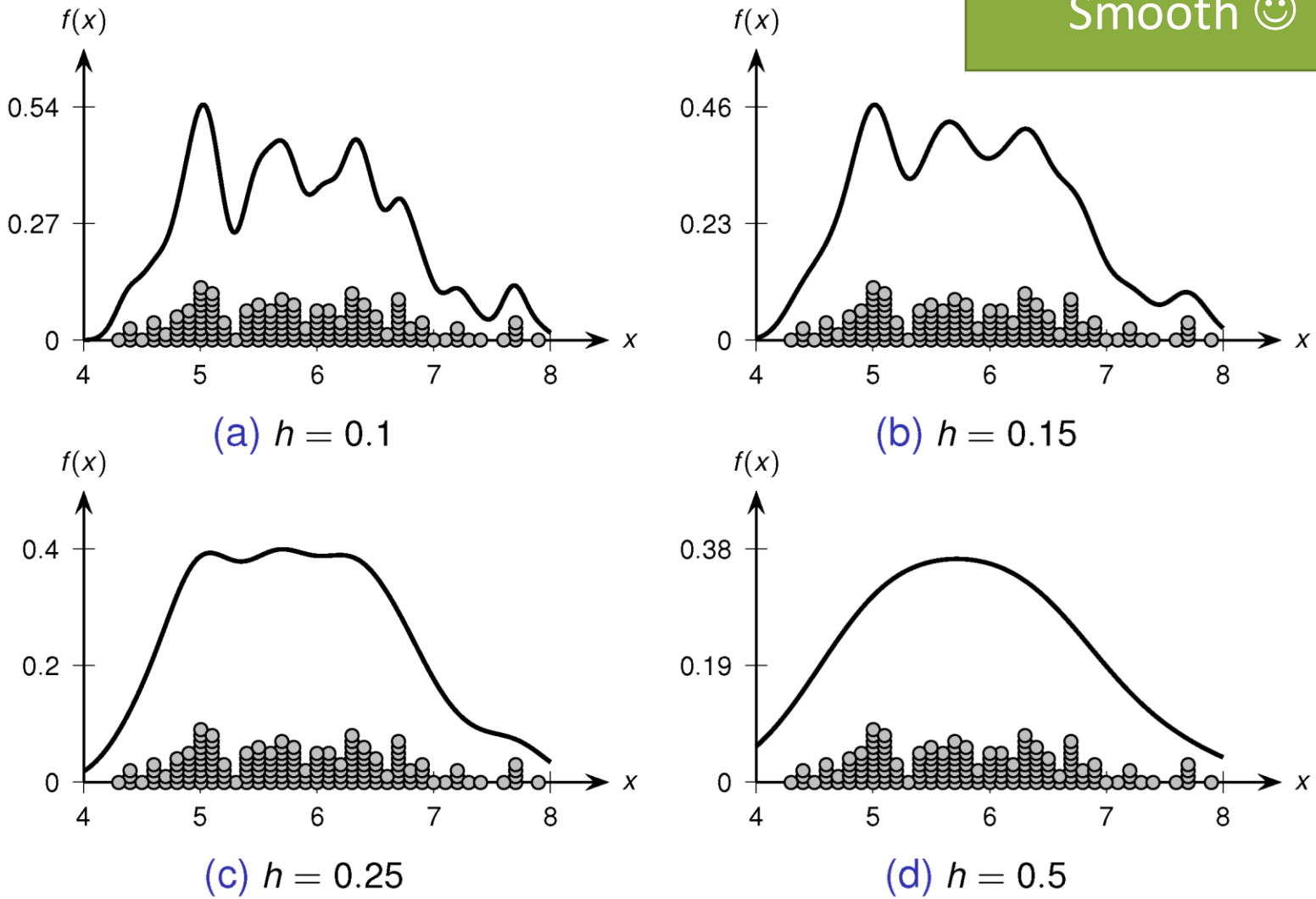
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - x_i)^2}{2h^2}\right\}$$



x at the center of the window plays the role of the **mean**, and h of the **standard deviation**

Density Estimation: Gaussian kernel (Iris 1D, sepal length)

Smooth ☺



Multivariate Density Estimation

- For d-dimensional data $x = (x_1, x_2, \dots, x_d)$
- Window becomes a **hypercube centered at x** with edge length h
  - Volume

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

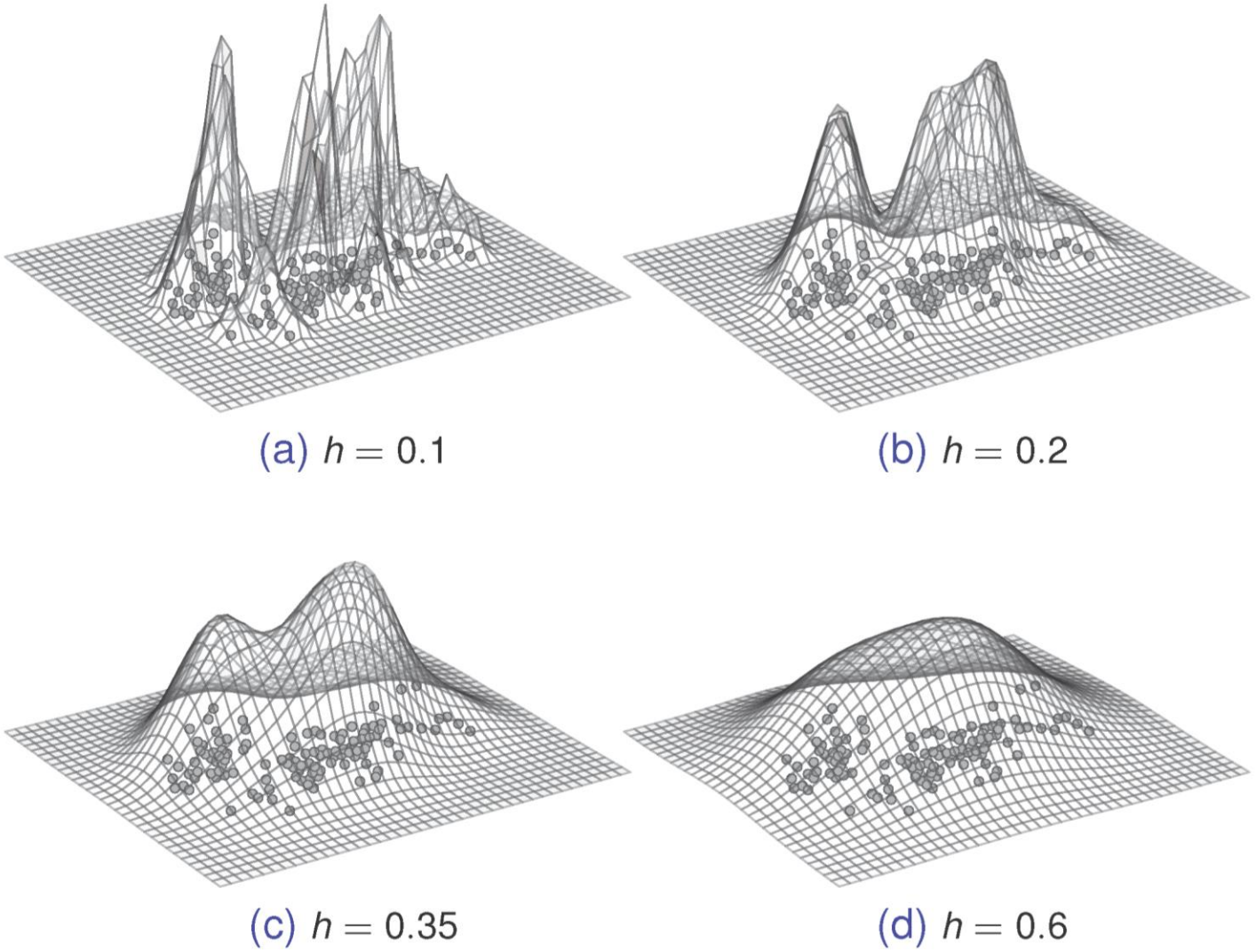
- Density estimation as before fraction of point weight within window, divided by volume

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

Where the multivariate kernel satisfies

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

Density Estimation: Gaussian kernel (Iris 1D, sepal length)



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

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

Multivariate Kernels

- **Discrete kernel in d dimensions**

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

- **Gaussian kernel in d dimensions**, assuming identity matrix as covariance matrix

$$K(\mathbf{z}) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{\mathbf{z}^\top \mathbf{z}}{2}\right\}$$

Remember what is the covariance matrix!

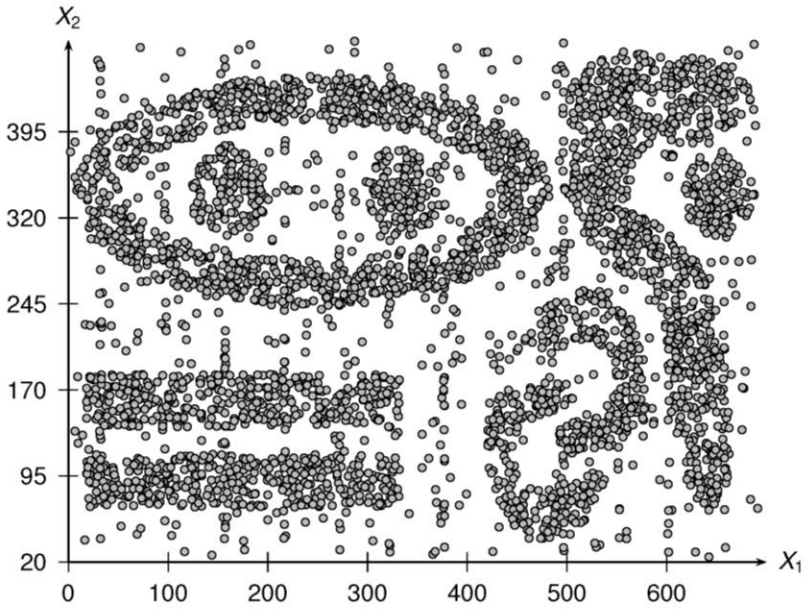
Nearest Neighbor Estimation

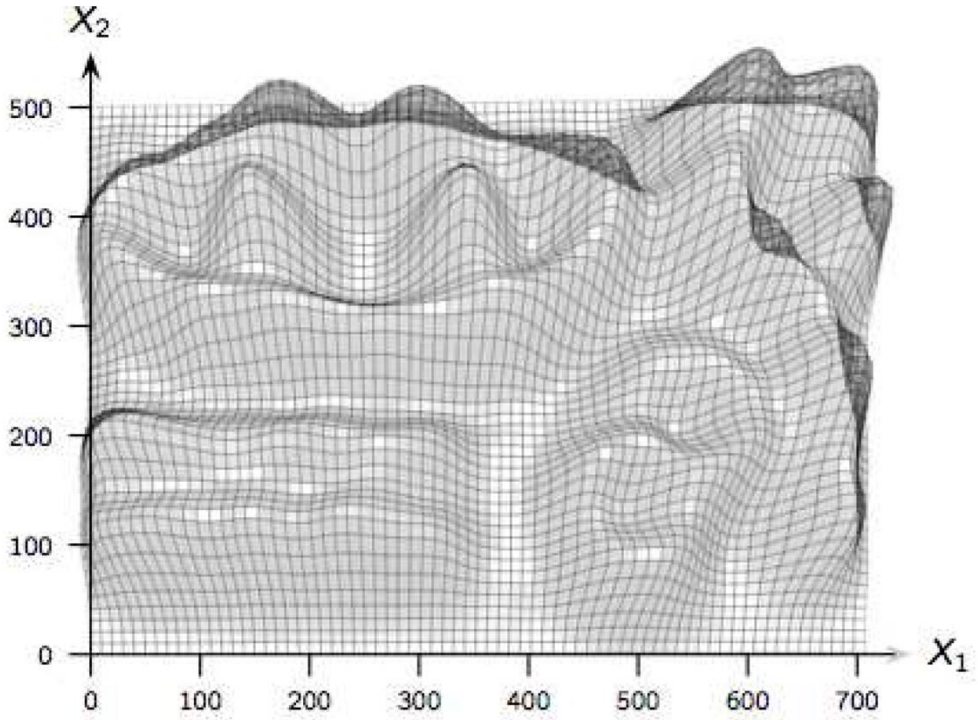
- **Kernel density estimation**
  1. **Fix the volume** h of dense region
  2. Kernel function finds the number or weight of points inside the region with fix volume
- **K-Nearest neighbor**
  - **Fix k**, the number that defines a "dense" region
  - The volume varies with k
  - Given k, estimate density at $\mathbf{x}$

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

  - $h_{\mathbf{x}}$: distance from x to its kth nearest neighbor
  - $\mathrm{vol}(S_d(h_{\mathbf{x}}))$: the volume of the d-dimenstional hypershere centered at $\mathbf{x}$

Density Estimation: Gaussian kernel on density-based data example,h=20





DENCLUE: Density attractor

- x^*: **local maximum** of the probability density distribution

Gradient-based approach

- Gradient: multivariate derivative of the probability density estimate
  - i.e., How does the probability density estimate varies with **x**?

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

- For Gaussian kernel, gradient at $\mathbf{x}$

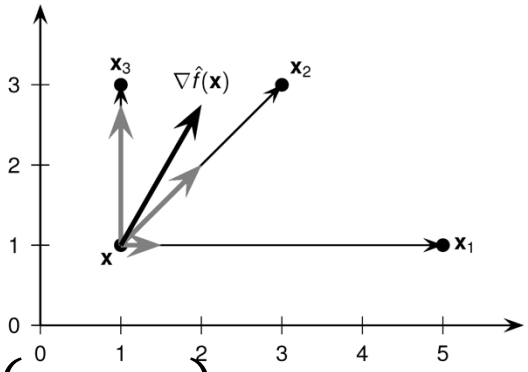
$$\frac{\partial}{\partial \mathbf{x}} K(\mathbf{z}) = \frac{1}{(2\pi)^{\frac{d}{2}}} \exp\left\{-\frac{\mathbf{z}^\top \mathbf{z}}{2}\right\} \cdot (-\mathbf{z}) \cdot \frac{\partial \mathbf{z}}{\partial \mathbf{x}} = K(\mathbf{z})(-\mathbf{z}) \frac{\partial \mathbf{z}}{\partial \mathbf{x}}$$

Setting $\mathbf{z} = \frac{\mathbf{x}-\mathbf{x}_i}{h}$, we get $\frac{\partial}{\partial \mathbf{x}} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) = \frac{1}{n} \frac{1}{h} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)$

$$\frac{1}{nh^{d+2}} \sum_{i=1} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \cdot (\mathbf{x}_i - \mathbf{x})$$

where vector $(\mathbf{x}_i - \mathbf{x})$ of direction for each $\mathbf{x}_i$ and weight $K\left(\frac{x-x_i}{h}\right)$

DENCLUE: Density attractor



- x^*: **local maximum** of the probability density distribution
  - For Gaussian kernel, gradient at x

$$\nabla\hat{f}(\mathbf{x}) = \frac{1}{nh^{d+2}} \sum_{i=1}^{n} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \cdot (\mathbf{x}_i - \mathbf{x})$$

with

- vector $(\mathbf{x}_i - \mathbf{x})$ of direction for each $\mathbf{x}_i$
- scaled by the weight $K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$
  - i.e., the further $\mathbf{x}_i$ from x the less influence it has

x^* density attractor for x

- hill climbing $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \eta\nabla\hat{f}(\mathbf{x})$ starting at x converges to x^*
- i.e., exists sequence of points $x = \mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \cdots \rightarrow \mathbf{x}_m$ with $\|\mathbf{x}_m - \mathbf{x}^*\| \leq \epsilon$

Find attractors: Instead of hill-climbing

Hill-climbing might have slow convergence ☹

- **Idea:** solve for $\nabla \hat{f}(\mathbf{x}) = 0$

$$\frac{1}{nh^{d+2}} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \cdot (\mathbf{x}_i - \mathbf{x}) = 0$$

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

x is in both sides → use as an iterative rule $\mathbf{x}_{t+1} = \cdots$

DENCLUE: Density-based Cluster

- Set C of data points from data set D is **density-based cluster** with respect to density threshold ξ if there exists a set of density attractors $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_m^*$
  1. Each point x in C is attracted to some attractor $\mathbf{x}_i^*$
  2. Each density attractor $\mathbf{x}_i^*$ exceeds density threshold ξ: $\hat{f}(x_i^*) \geq \xi$
  3. Any two density attractors $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ are density reachable
     - i.e., there exists a path from $\mathbf{x}_i^*$ to $\mathbf{x}_j^*$ such that for all points **y** on the path $\hat{f}(\mathbf{y}) \geq \xi$

- Extends DBSCAN notion
  - Points are either dense or within neighborhood of dense points
  - Any two points are density reachable using a path of dense points with mutual neighborhood inclusion

DENCLUE algorithm

Complexity $O(n^2 t)$
- n points
- t iterations

DENCLUE $(\mathbf{D}, h, \xi, \epsilon)$:

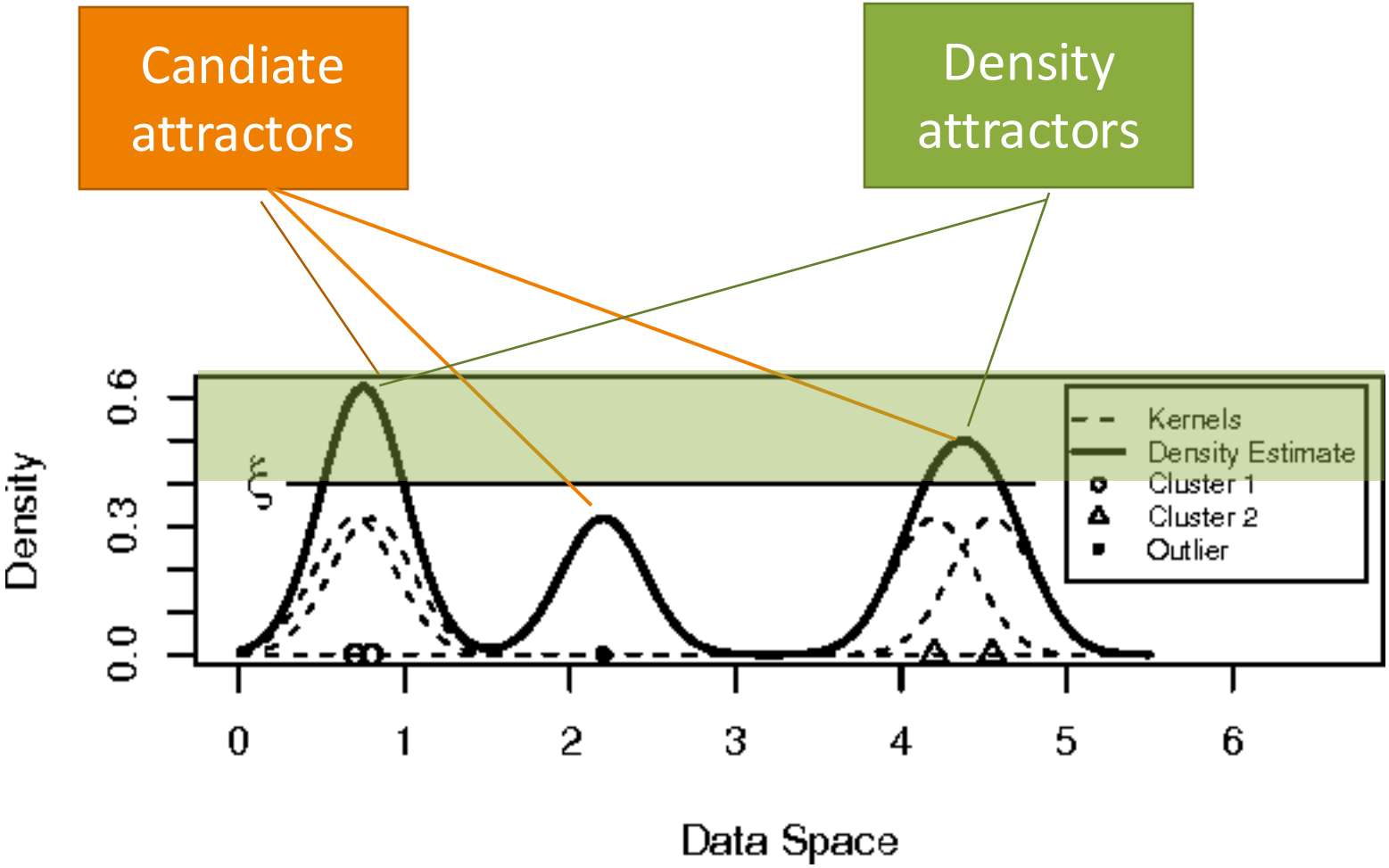
1 $\mathcal{A} \leftarrow \emptyset$
2 **foreach** $\mathbf{x} \in \mathbf{D}$ **do** // find density attractors
4 $\quad \mathbf{x}^* \leftarrow$ FINDATTRACTOR$(\mathbf{x}, \mathbf{D}, h, \epsilon)$
5 $\quad$ **if** $\hat{f}(\mathbf{x}^*) \geq \xi$ **then**
7 $\quad\quad \mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{x}^*\}$
9 $\quad\quad R(\mathbf{x}^*) \leftarrow R(\mathbf{x}^*) \cup \{\mathbf{x}\}$
11 $\mathcal{C} \leftarrow \{\text{maximal } C \subseteq \mathcal{A} \mid \forall \mathbf{x}_i^*, \mathbf{x}_j^* \in C, \mathbf{x}_i^* \text{ and } \mathbf{x}_j^* \text{ are density reachable}\}$
12 **foreach** $C \in \mathcal{C}$ **do** // density-based clusters
13 $\quad$ **foreach** $\mathbf{x}^* \in C$ **do** $C \leftarrow C \cup R(\mathbf{x}^*)$
14 **return** $\mathcal{C}$

If attractor $\mathbf{x}^*$'s density $\geq \xi$ then it is added to the set of attractors and
x is added to the attractor's set

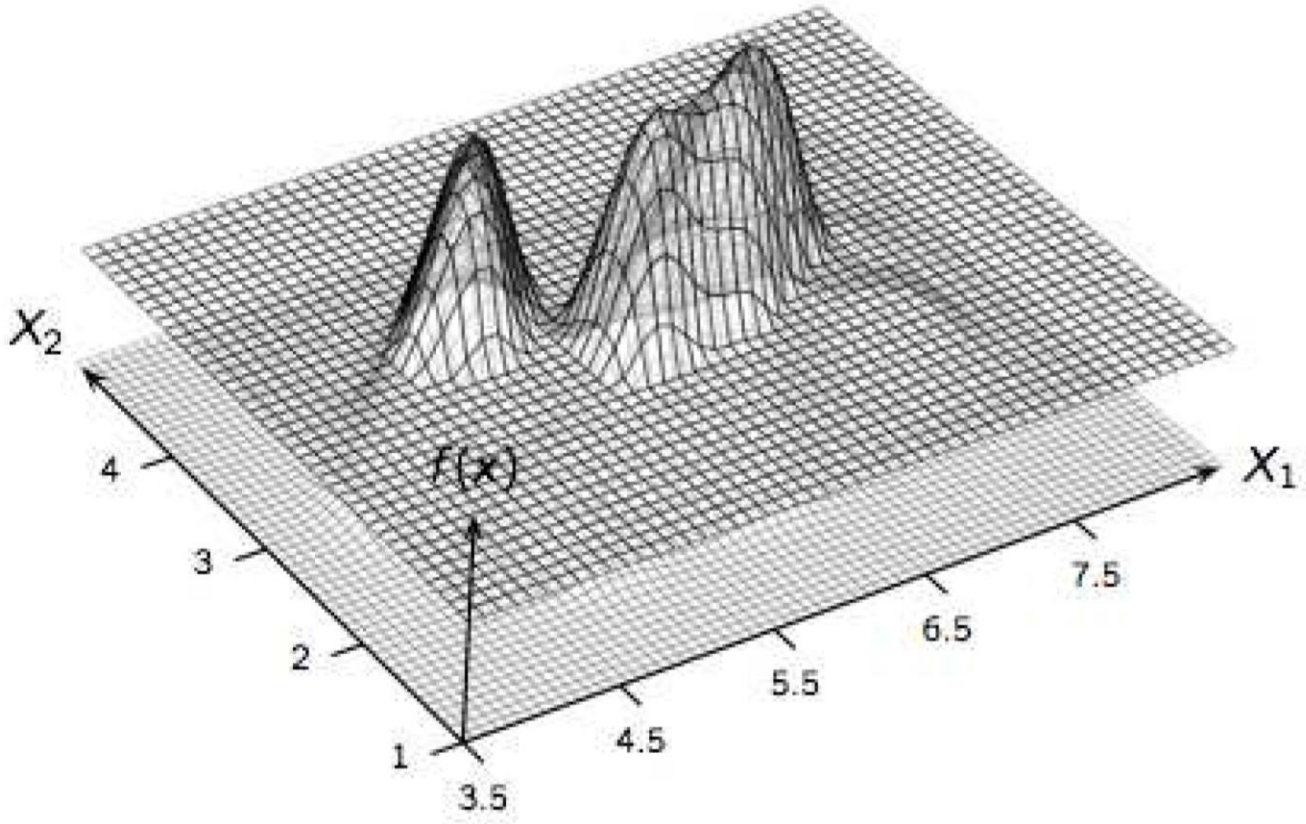
FINDATTRACTOR $(\mathbf{x}, \mathbf{D}, h, \epsilon)$:

2 $t \leftarrow 0$
3 $\mathbf{x}_t \leftarrow \mathbf{x}$
4 **repeat**
6 $\quad \mathbf{x}_{t+1} \leftarrow \frac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_t}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$
7 $\quad t \leftarrow t + 1$
8 **until** $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$
10 **return** $\mathbf{x}_t$

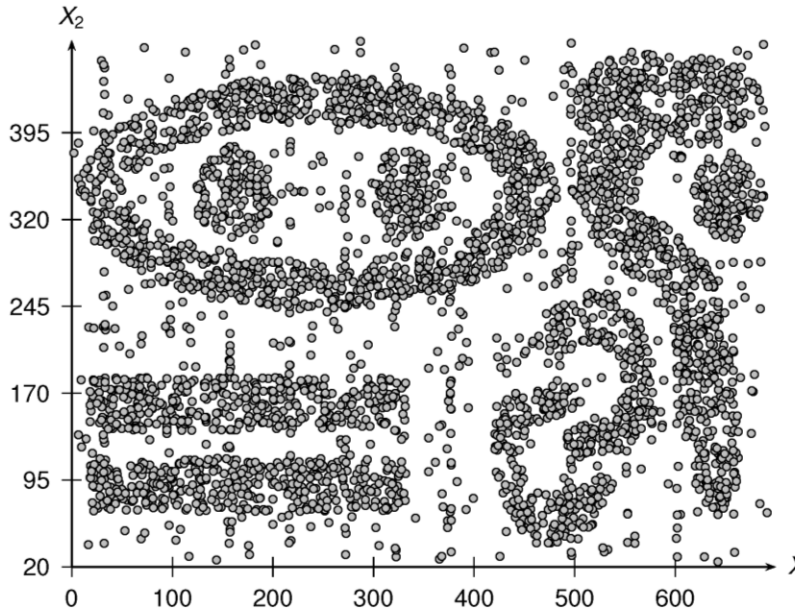
DENCLUE: A visual explanation

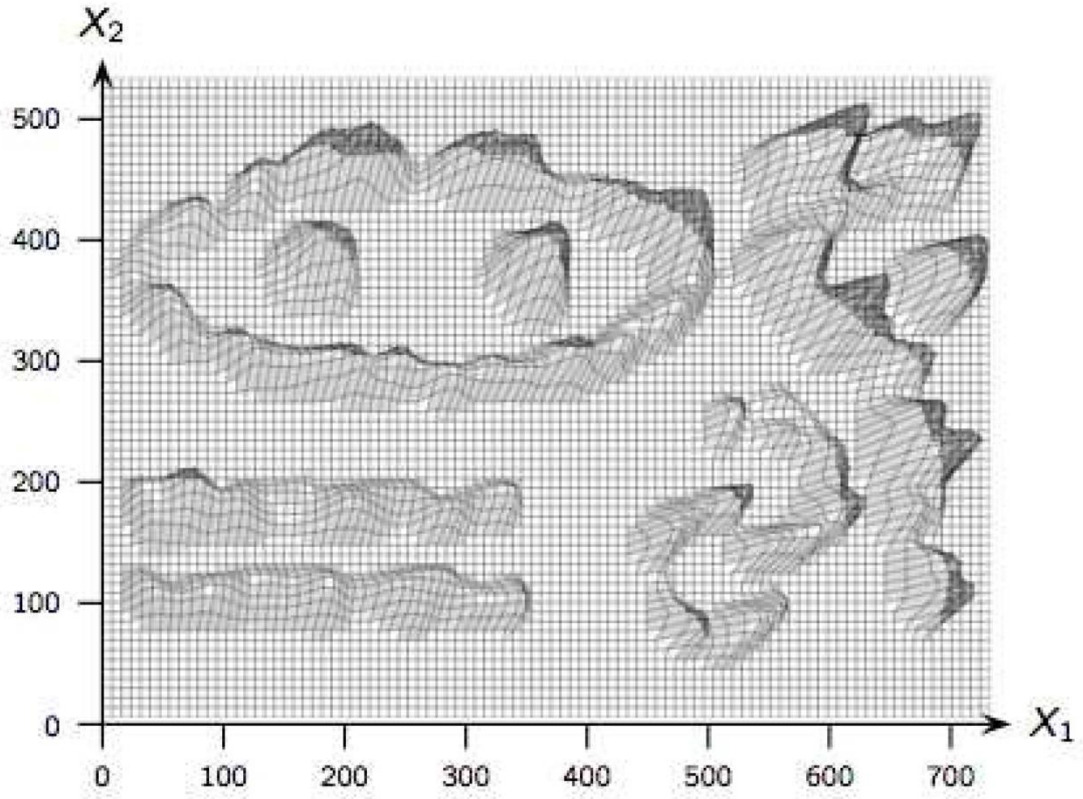


DENCLUE: Iris 2D data (sepal length, sepal width), h=0.2, ξ=0.008



DENCLUE: Density based data set example





DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

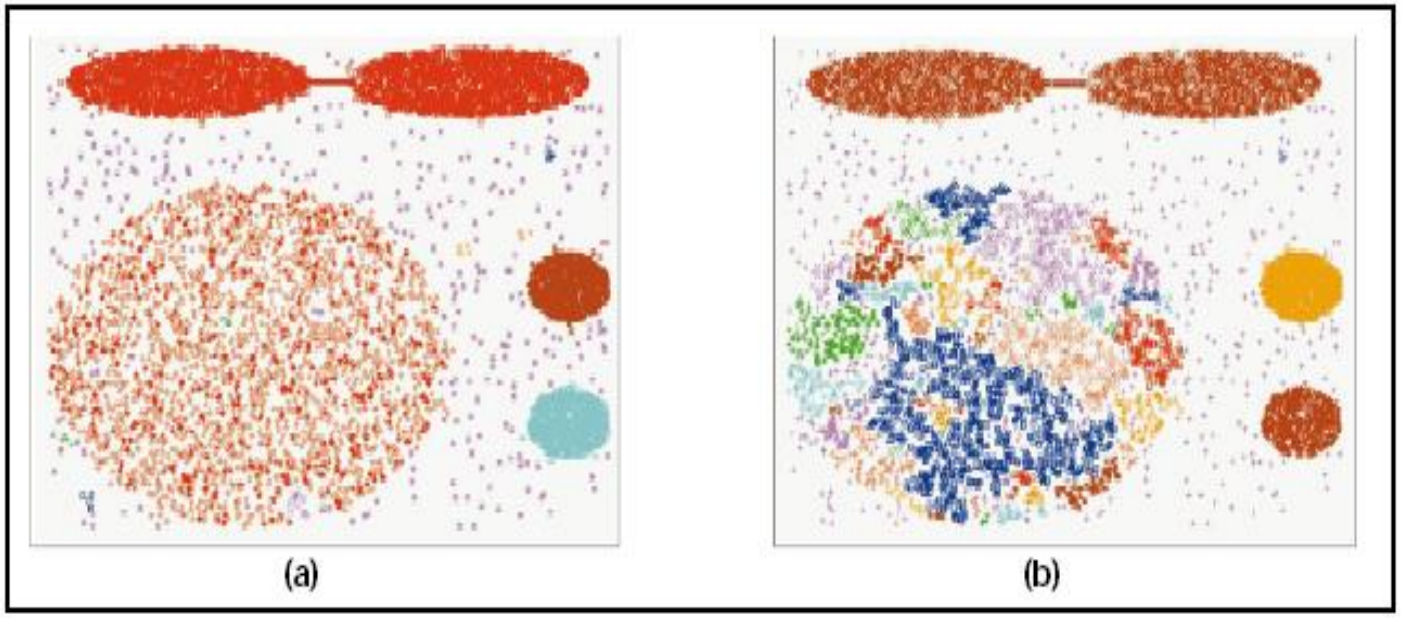
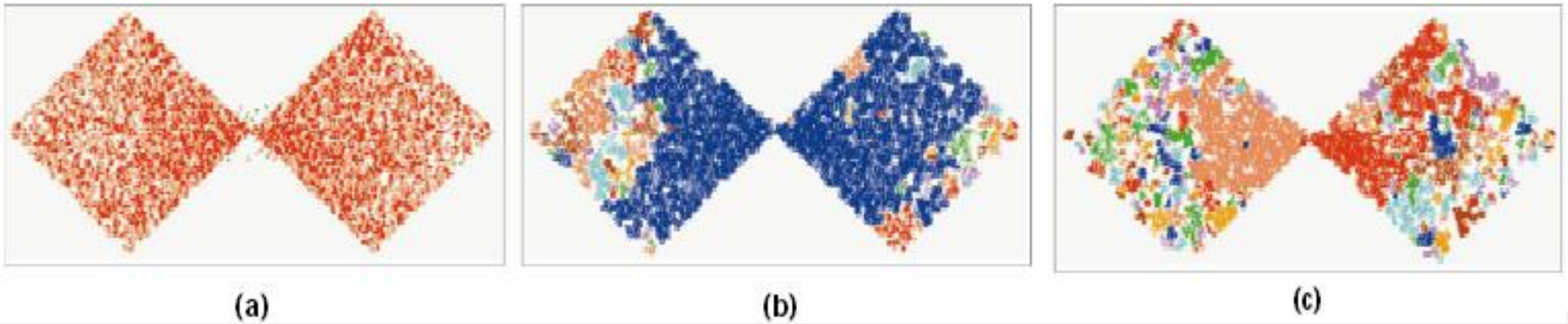


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.



Density Based Clustering: Discussion

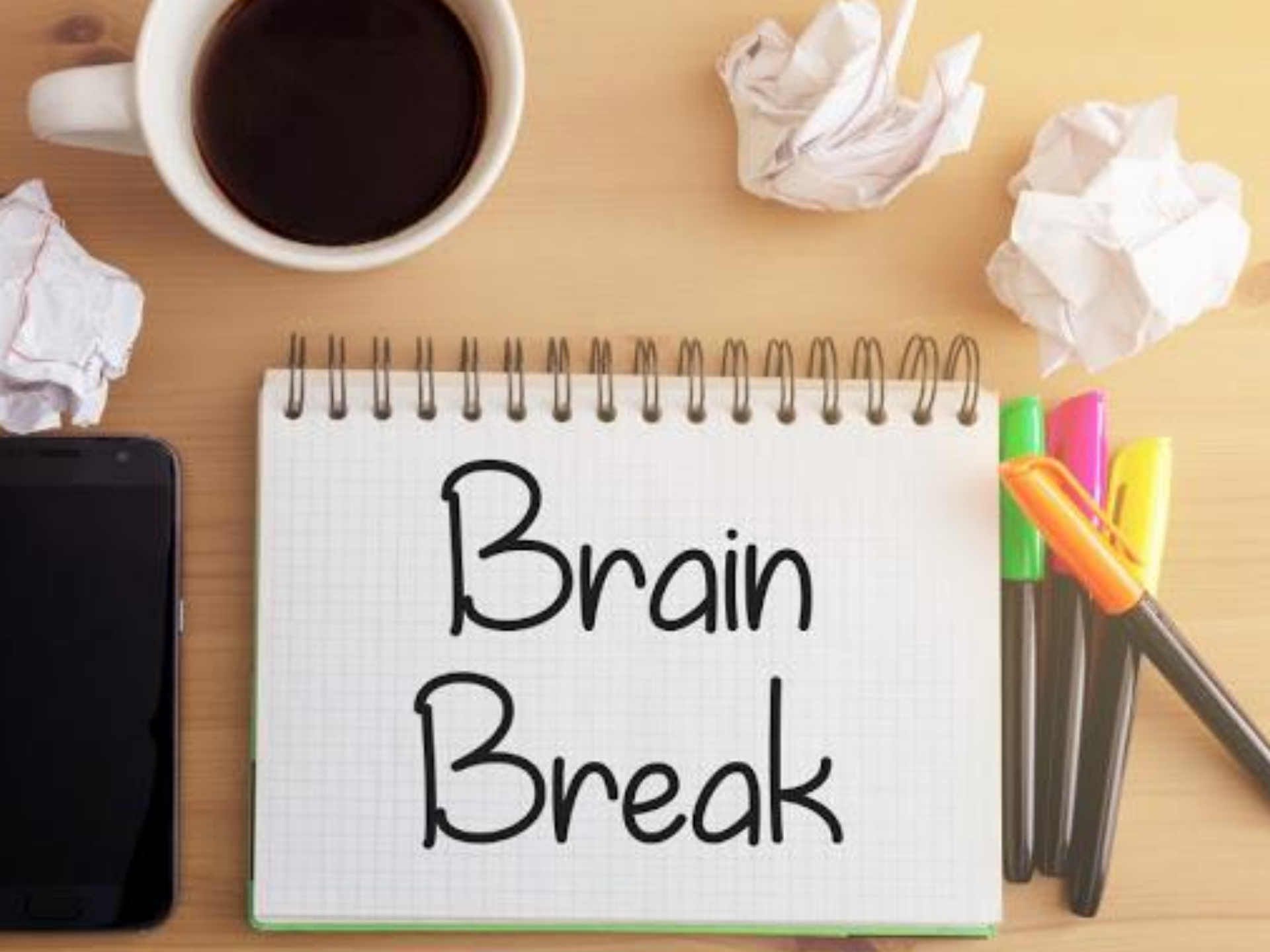
- DBSCAN corresponds to discrete kernel, **more efficient**
- DENCLUE uses Gaussian kernel density based attractors, **smooth model**

Advantages

- Clusters can have arbitrary shape and size, i.e. clusters are not restricted to have convex shapes
- Number of clusters is determined automatically
- Can separate clusters from surrounding noise
- Can be supported by spatial index structures

Disadvantages

- Input parameters may be difficult to determine
- In some situations very sensitive to input parameter setting


Brain
Break

Clustering evaluation

Am I really returning better clusters?

Clustering evaluation and validation

- **External measures:**
  - Take criteria into account that are not part of the clustering data
    - E.g. class labels
- **Internal measures:**
  - Based only on clustering data
    - E.g. distances as in TD or silhouette coefficient
- **Measures can be relative**
  - Compare two clusterings instead of obtaining ”objective” goodness value
  - Some measures, e.g. silhouette can be used in both ways

External measures (using ground-truth clustering)

Correct or ground-truth clustering is known *a priori*

- $y_i \in \{1,2,\dots,k\}$ ground-truth cluster membership
  - $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ ground-truth clustering
  - $T_j = \{\mathbf{x_i} \in \boldsymbol{D} | y_i = j\}$
- $C = \{C_1, \dots, C_r\}$ a clustering in r clusters
  - $\hat{y}_i \in \{1,2,\dots,r\}$ cluster label for $\mathbf{x_i}$

Contingency table

- $r \times k$ table $\mathbf{N}$ induced by clustering $\mathcal{C}$ and ground truth $\mathcal{T}$

$$\mathbf{N}(i,j) = \mathrm{n_{ij}} = |\mathrm{C_i} \cap T_j|$$

- n_{ij} : number of points **common** to cluster C_i and ground-truth T_j
- $n_i = |C_i|$ number of points in cluster C_i
- $m_j = |T_j|$ number of points in ground-truth T_j
- Computed in $O(n)$ time
  - Examine the ground-truth and cluster labels $y_i, \hat{y}_i$ for each point $x_i \in D$
  - Increment the count $n_{y_i,\hat{y}_i}$

Purity

- Quantifies the extent to which a cluster C_i contains entities from **only** one partition

$$purity_i = \frac{1}{n_i} \max_{j=1\ldots k} \{n_{ij}\}$$

- **Purity of clustering $\mathcal{C}$**: weighted sum of each cluster's purity

$$purity = \sum_{i=1}^{r} \frac{n_i}{n} purity_i = \frac{1}{n} \sum_{i=1}^{r} \max_{j=1\ldots k} \{n_{ij}\}$$

Maximum matching

Select mapping between clusters and ground-truth, such that the sum of the number of common points is maximized

Computation

- Create a **weighted bipartite graph**
  - $V = \mathcal{C} \cup \mathcal{T}$ vertex set
  - $E = \{(C_i, T_j\}\}$ edge set
  - $w(C_i, T_j) = n_{ij}$
- Matching M: subset of pairwise nonadjacent edges from E (i.e., no common vertex)
- Maximum weight matching
  - Computed , e.g. .with Hungarian algorithm for assignment problem

$$match = \arg\max_{M} \left\{ \frac{\sum_{e \in M} w(e)}{n} \right\}$$

F-measure

- j_i ground-truth cluster with max number of points from C_i, $j_i = \arg\max_{j=1}^{k}\{n_{ij}\}$
- **Precision**
  - Same as purity

$$prec_i = \frac{1}{n_i}\max_{j=1}^{k}\{n_{ij}\} = \frac{n_{ij_i}}{n_i}$$

- **Recall**

$$recall_i = \frac{n_{ij_i}}{|T_{j_i}|} = \frac{n_{ij_i}}{m_{j_i}}$$

  - $m_{j_i} = |\mathrm{T_{j_i}}|$
- **F-measure**
  - Harmonic mean of precision and recall for each cluster

$$F_i = \frac{2}{\frac{1}{prec_i} + \frac{1}{recall_i}} = \frac{2 \cdot prec_i \cdot recall_i}{prec_i + recall_i} = \frac{2n_{ij_i}}{n_i + m_{j_i}}$$

- F-measure of clustering $\mathcal{C}$ is the mean clusterwise F-measure:

$$F = \frac{1}{r}\sum_{i=1}^{r} F\_i$$

Conditional entropy

- Entropy of a clustering

$$H(\mathcal{C}) = -\sum_{i=1}^{r} p_{C_i} \log p_{C_i}$$

Where $p_{C_i} = \frac{n_i}{n}$ probability of cluster C_i

- Entropy of ground-truth $H(\mathcal{T})$ is similarly defined
- Cluster specific entropy of $\mathcal{T}$

$$H(\mathcal{T}|C_i) = -\sum_{j=1}^{k} \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}$$

- Conditional entropy of clustering

$$H(\mathcal{T}|\mathcal{C}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} \frac{n_{ij}}{n} \log \frac{n_{ij}}{n_i} = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{p_{C_i}}$$

  - Perfect clustering $H(\mathcal{T}|\mathcal{C}) = 0$
  - Worst entropy $\log k$

Mutual-Information

Quantify the amount of shared information between the clustering $\mathcal{C}$ and the ground-truth $\mathcal{T}$

$$I(\mathcal{C},\mathcal{T}) = -\sum_{i=1}^{r}\sum_{j=1}^{k} p_{ij}\log\left(\frac{p_{ij}}{p_{C_i}p_{T_j}}\right)$$

- Measures the dependence between the observed joint probability p_{ij} of the clustering and the gropund-truth, and the expected joint probability $p_{C_i}p_{T_j}$ **under independence assumption**
  - If C and T are independent → $p_{C_i}p_{T_j} = \mathrm{p_{ij}}$ → $I(\mathcal{C},\mathcal{T}) = 0$
- No upper bound on $I(\mathcal{C},\mathcal{T})$!!!

Normalized version

$$NMI(\mathcal{C},\mathcal{T}) = \frac{I(\mathcal{C},\mathcal{T})}{\sqrt{H(\mathcal{C})H(\mathcal{T})}}$$

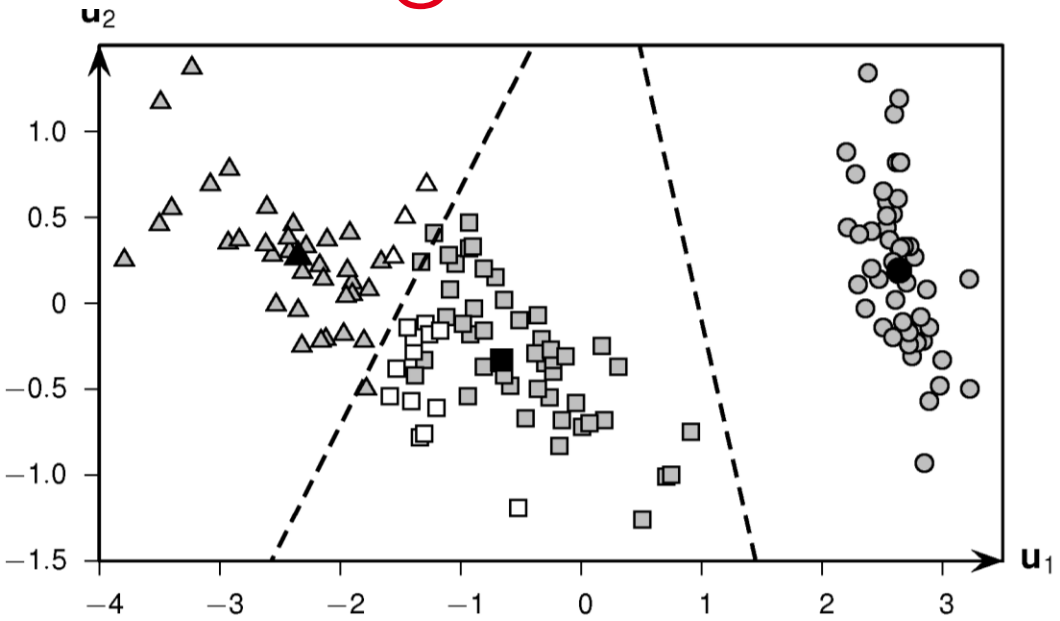
- NMI is in the range [0,1] with 1 indicate a good clustering!

Iris prinpiple component
Good clustering

Purity 0.887

Match 0.887

F1 0.85



Contingency table:

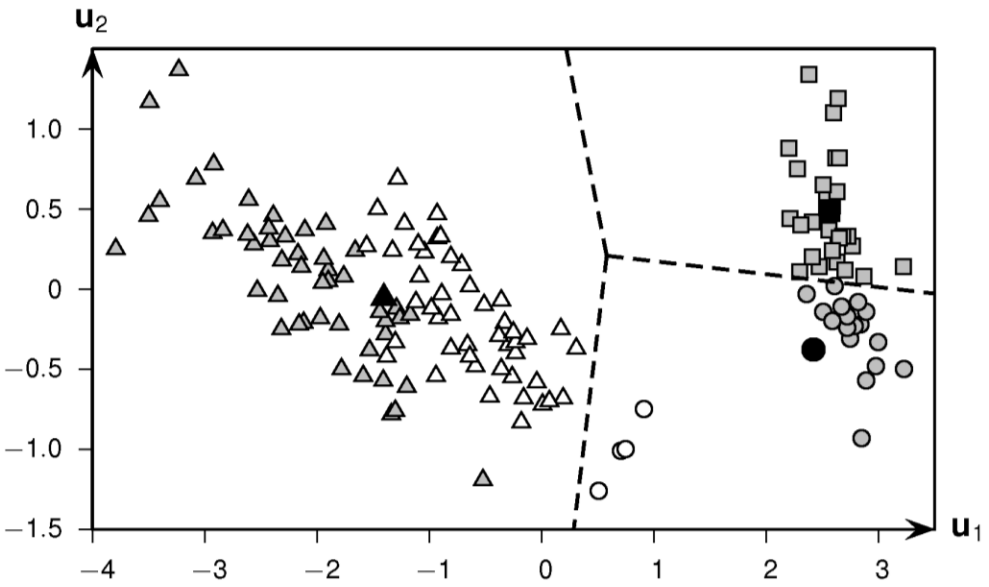
	iris-setosa T_1	iris-versicolor T_2	iris-virginica T_3	n_i
C_1 (squares)	0	47	14	61
C_2 (circles)	50	0	0	50
C_3 (triangles)	0	3	36	39
m_j	50	50	50	$n = 100$

Iris principle component data
Poor clustering

Purity 0.667

Match 0.560

F 0.658



Contingency table:

	iris-setosa T_1	iris-versicolor T_2	iris-virginica T_3	n_i
C_1 (*squares*)	30	0	0	30
C_2 (*circles*)	20	4	0	24
C_3 (*triangles*)	0	46	50	96
m_j	50	50	50	$n = 150$

Evaluation measures - Discussion

- **Internal measures:**
  - Make no assumptions about ground truth
  - In unsupervised learning this is very often the case in practice
- **External measures:**
  - Make use of external information such as domain expertise
  - Sometimes to be generated for testing sample

Different measures capture different aspects of a good clustering

- This is actively discussed in the research community
- For example, can a cluster be “matched” to two ground truth clusters (or the other way around? Should this be punished somehow?

Summary

- Generalized density-based clustering
  - DBSCAN uses discrete kernel with non-smooth behavior
  - DENCLUE uses kernel density estimation
- Clustering evaluation
  - Internal measures
  - External measures
- **Next week**
  - Hierarchical clustering: clustering at different levels of resolution
  - Subspace clustering: clustering in different subspaces

What was today's lecture about?

- Density-based clustering has the following general idea…
- The main difference between DBSCAN and DENCLUE is that…
- We evaluate clusters through …

Acknowledgements

- Slides for book Data Mining and Analysis by Mohammed J Zaki and Wagner Meira Jr
- Slides for book Data Mining: Concepts and Techniques, 2nd ed. by Jiawei Han and Micheline Kamber
- Slides for course on Knowledge Discovery in Databases by Jörg Sander and Martin Ester
- Slides for course Data Mining Algorithms by Thomas Seidl
- Slides for course Advanced Data Mining Algorithms by Ira Assent
- Slides AnyDBC by Mai Thai Son

References

- J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
- M. H. Dunham. Data Mining: Introductory and Advanced Topics. Prentice Hall, 2003.
- U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
- T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of the ACM, 39:58-64, 1996.
- G. Piatetsky-Shapiro, U. Fayyad, and P. Smith. From data mining to knowledge discovery: An overview. In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining, 1-35. AAAI/MIT Press, 1996.
- G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991.
- D. Hand, H. Mannila, P. Smyth. Principles of Data Mining. MIT Press, 2001.