Data Mining and Analysis: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

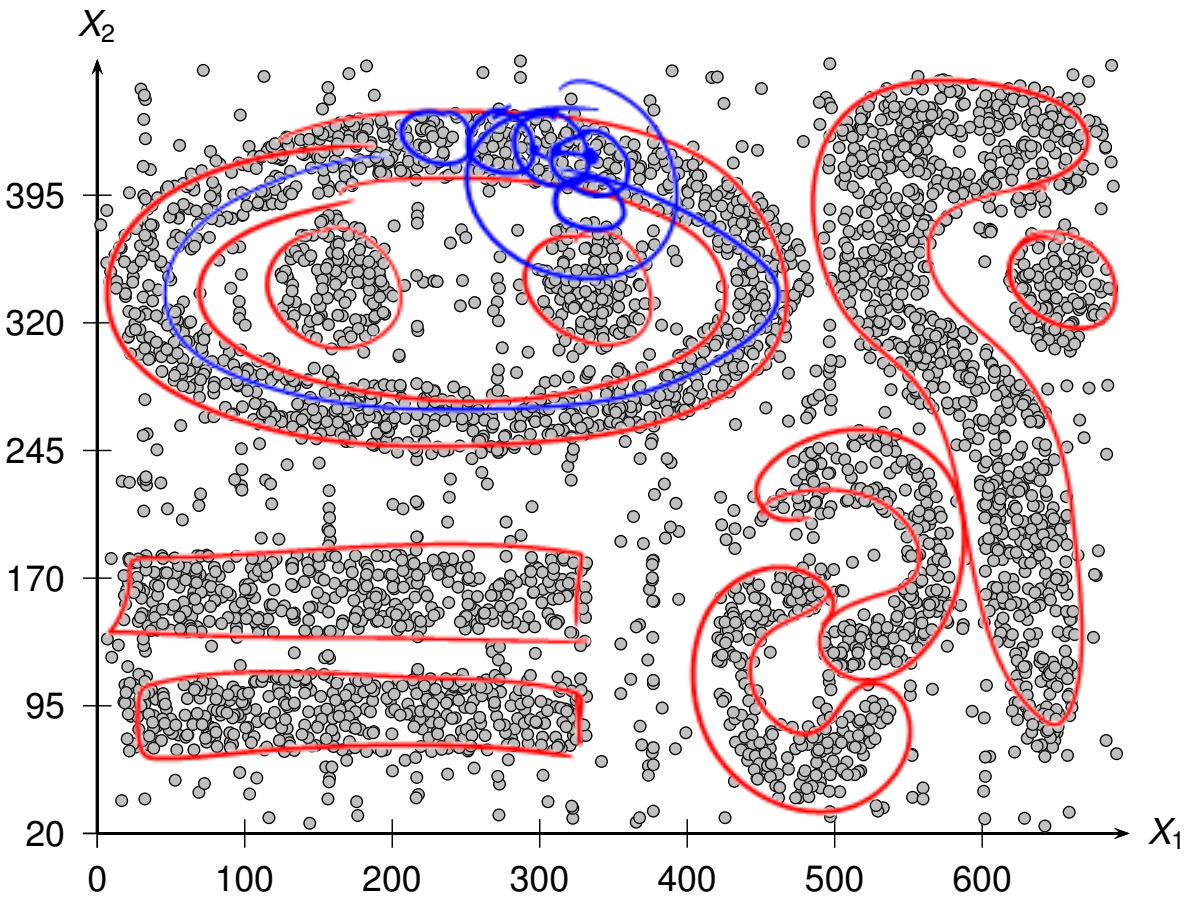
[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 15: Density-based Clustering

Density-based Clustering

Density-based methods are able to mine nonconvex clusters, where distance-based methods may have difficulty.



$\varepsilon \leftarrow$ radius

$N_\varepsilon \leftarrow$ ball of radius ε

$|N_\varepsilon| \geq$ minpts

The DBSCAN Approach: Neighborhood and Core Points

Define a ball of radius ϵ around a point $\mathbf{x} \in \mathbb{R}^d$, called the ϵ-*neighborhood* of $\mathbf{x}$, as follows:

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

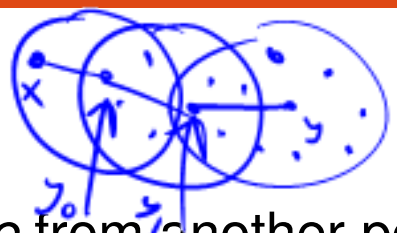
Here $\delta(\mathbf{x}, \mathbf{y})$ represents the distance between points $\mathbf{x}$ and $\mathbf{y}$. which is usually assumed to be the Euclidean

We say that $\mathbf{x}$ is a *core point* if there are at least *minpts* points in its ϵ-neighborhood, i.e., if $|N_\epsilon(\mathbf{x})| \geq$ *minpts*.

A *border point* does not meet the *minpts* threshold, i.e., $|N_\epsilon(\mathbf{x})| <$ *minpts*, but it belongs to the ϵ-neighborhood of some core point $\mathbf{z}$, that is, $\mathbf{x} \in N_\epsilon(\mathbf{z})$.

If a point is neither a core nor a border point, then it is called a *noise point* or an outlier.

The DBSCAN Approach: Reachability and Density-based Cluster



A point **x** is *directly density reachable* from another point **y** if $\mathbf{x} \in N_\epsilon(\mathbf{y})$ and **y** is a core point.

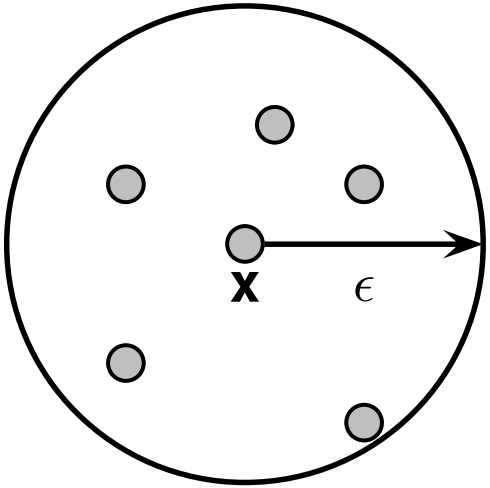
A point **x** is *density reachable* from **y** if there exists a chain of points, $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_l$, such that $\mathbf{x} = \mathbf{x}_0$ and $\mathbf{y} = \mathbf{x}_l$, and $\mathbf{x}_i$ is directly density reachable from $\mathbf{x}_{i-1}$ for all $i = 1, \ldots, l$. In other words, there is set of core points leading from **y** to **x**.

Two points **x** and **y** are *density connected* if there exists a core point **z**, such that both **x** and **y** are density reachable from **z**.

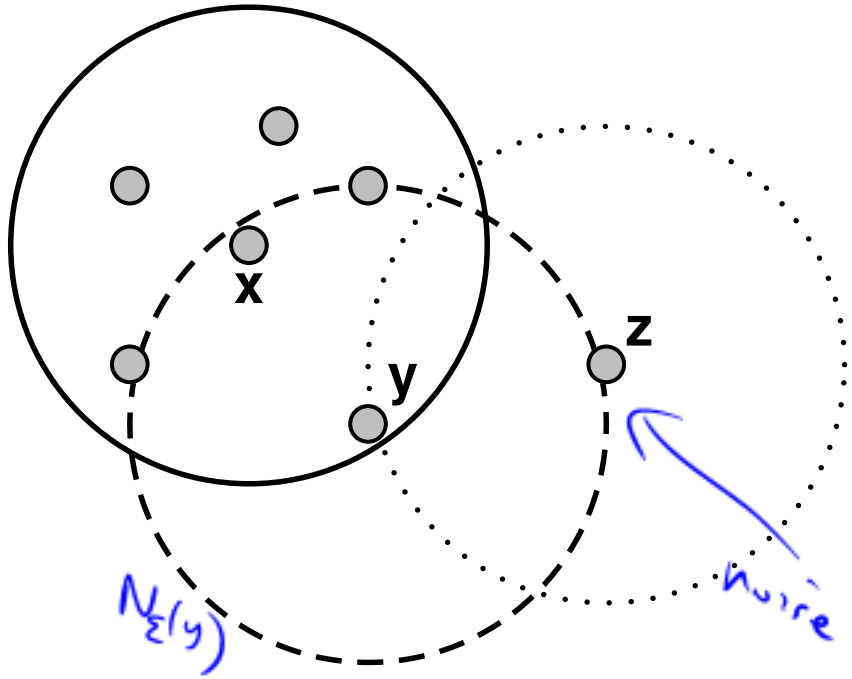
A *density-based cluster* is defined as a maximal set of density connected points.

Core, Border and Noise Points

Core $minpts = 6$



(a) Neighborhood of a Point



(b) Core, Border, and Noise Points

$|N_\epsilon(y)| = 5 < minpts$

y is not core, y is border

DBSCAN Density-based Clustering Algorithm

DBSCAN computes the ϵ-neighborhood $N_\epsilon(\mathbf{x}_i)$ for each point $\mathbf{x}_i$ in the dataset $\mathbf{D}$, and checks if it is a core point. It also sets the cluster id $id(\mathbf{x}_i) = \emptyset$ for all points, indicating that they are not assigned to any cluster.

Starting from each unassigned core point, the method recursively finds all its density connected points, which are assigned to the same cluster.

Some border point may be reachable from core points in more than one cluster; they may either be arbitrarily assigned to one of the clusters or to all of them (if overlapping clusters are allowed).

Those points that do not belong to any cluster are treated as outliers or noise.

Each DBSCAN cluster is a maximal connected component over the core point graph.

DBSCAN is sensitive to the choice of ϵ, in particular if clusters have different densities. The overall complexity of DBSCAN is $O(n^2)$.

DBSCAN Algorithm

DBSCAN (D, ϵ, *minpts*):

1 $Core \leftarrow \emptyset$
2 **foreach** $\mathbf{x}_i \in \mathbf{D}$ **do** // Find the core points
3 Compute $N_\epsilon(\mathbf{x}_i)$
4 $id(\mathbf{x}_i) \leftarrow \emptyset$ // cluster id for $\mathbf{x}_i$
5 **if** $N_\epsilon(\mathbf{x}_i) \geq minpts$ **then** $Core \leftarrow Core \cup \{\mathbf{x}_i\}$
6 $k \leftarrow 0$ // cluster id
7 **foreach** $\mathbf{x}_i \in Core$, *such that* $id(\mathbf{x}_i) = \emptyset$ **do**
8 $k \leftarrow k + 1$
9 $id(\mathbf{x}_i) \leftarrow k$ // assign $\mathbf{x}_i$ to cluster id k
10 DENSITYCONNECTED $(\mathbf{x}_i, k)$
11 $\mathcal{C} \leftarrow \{C_i\}_{i=1}^k$, where $C_i \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = i\}$
12 $Noise \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = \emptyset\}$
13 $Border \leftarrow \mathbf{D} \setminus \{Core \cup Noise\}$
14 **return** $\mathcal{C}, Core, Border, Noise$

DENSITYCONNECTED (x, *k*):

15 **foreach** $\mathbf{y} \in N_\epsilon(\mathbf{x})$ **do**
16 $id(\mathbf{y}) \leftarrow k$ // assign $\mathbf{y}$ to cluster id k
17 **if** $\mathbf{y} \in Core$ **then** DENSITYCONNECTED $(\mathbf{y}, k)$

$O(n^2)$

$O(V+E)$

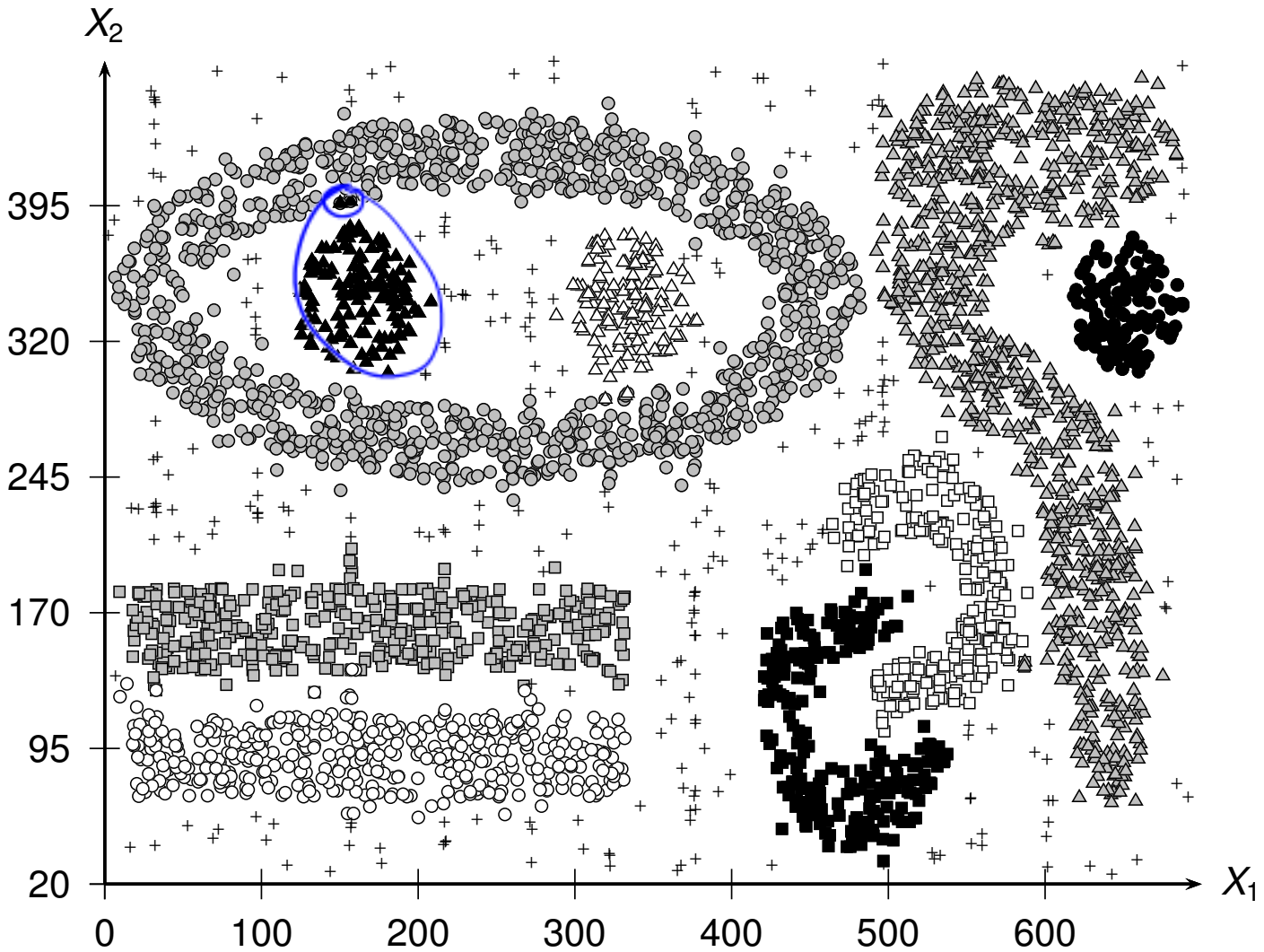
(1) find connected components of core points

(2) add back all border points for each core

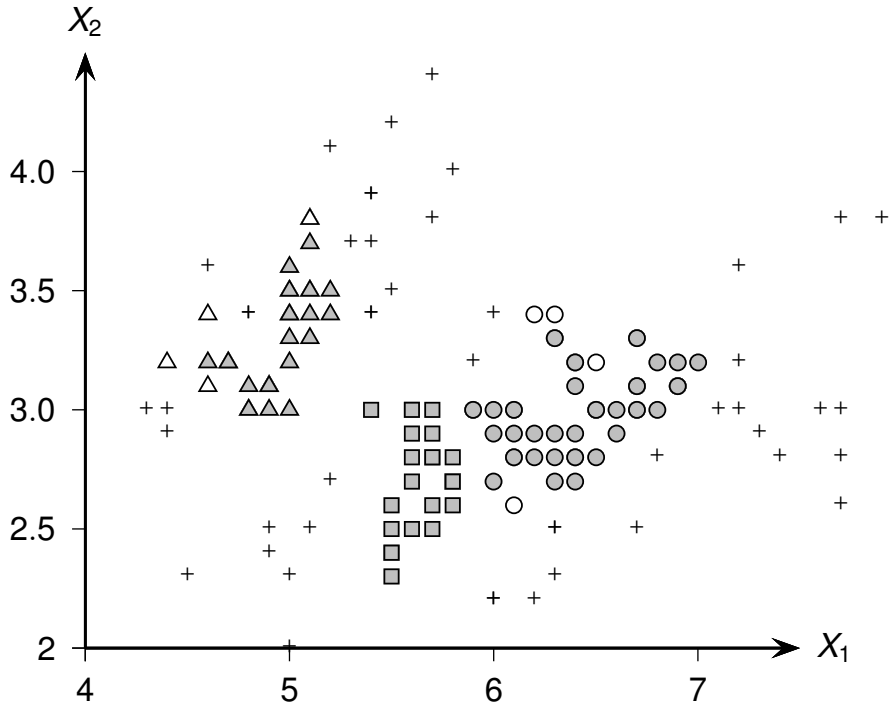
(3) everything else is noise

Density-based Clusters

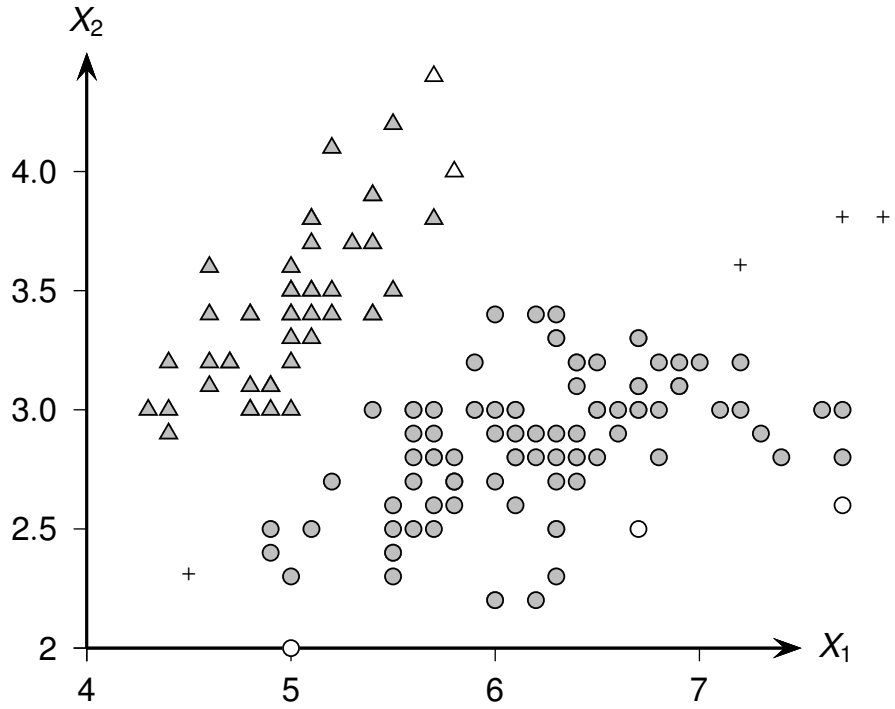
$\epsilon = 15$ and $minpts = 10$



DBSCAN Clustering: Iris Dataset



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



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

Kernel Density Estimation

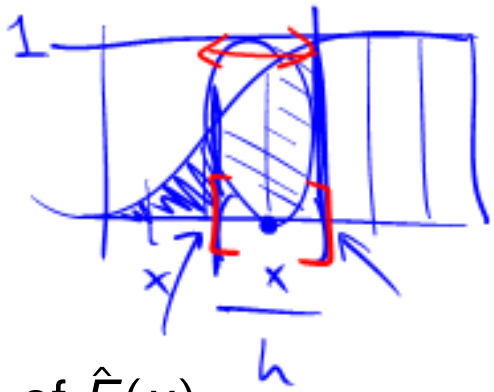
There is a close connection between density-based clustering and density estimation. The goal of density estimation is to determine the unknown probability density function by finding the dense regions of points, which can in turn be used for clustering.

Kernel density estimation is a nonparametric technique that does not assume any fixed probability model of the clusters. Instead, it tries to directly infer the underlying probability density at each point in the dataset.

Univariate Density Estimation

Assume that X is a continuous random variable, and let $x_1, x_2, \ldots, x_n$ be a random sample. We directly estimate the cumulative distribution function from the data by counting how many points are less than or equal to x:

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



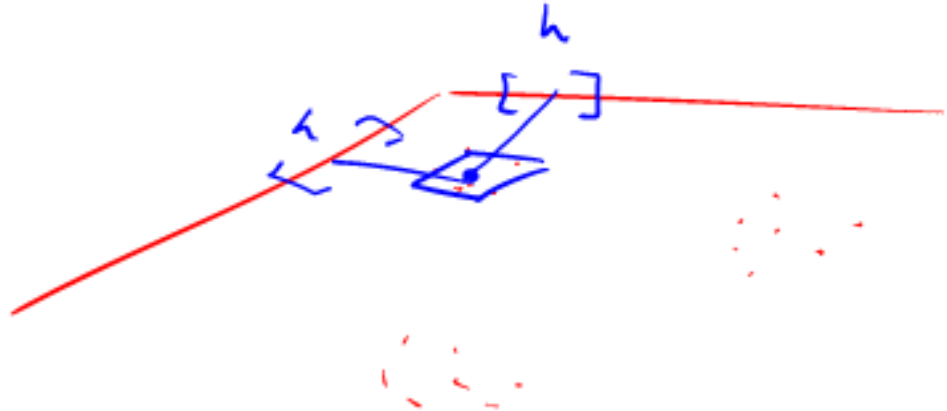
where I is an indicator function.

We estimate the density function by taking the derivative of $\hat{F}(x)$

$h \to 0$

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

where k is the number of points that lie in the window of width h centered at x. The density estimate is the ratio of the fraction of the points in the window (k/n) to the volume of the window (h).



$$\hat{f}(x) = \frac{\text{fraction of points within } \left(\begin{matrix}\text{Parzen}\\ \text{window}\end{matrix}\right) \text{ of } h \times h \text{ around } x}{h^2}$$

subsubsectionKernel Estimator Kernel density estimation relies on a *kernel function* K that is non-negative, symmetric, and integrates to 1, that is, $K(x) \geq 0$, $K(-x) = K(x)$ for all values x, and $\int K(x)dx = 1$.

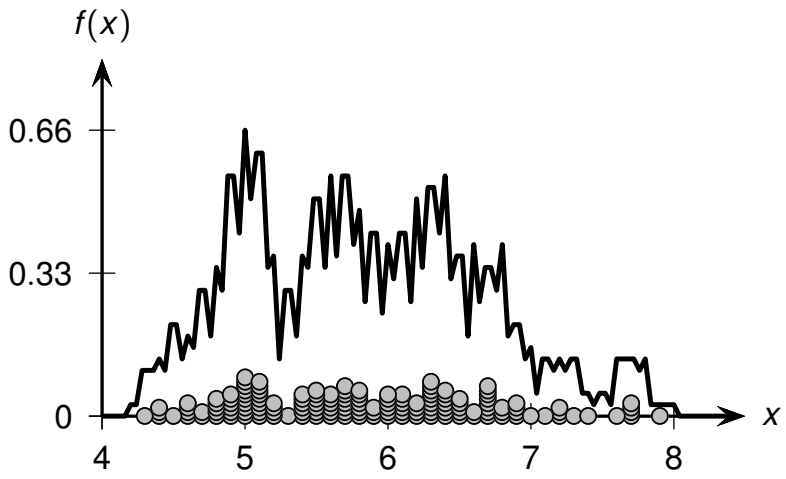
Discrete Kernel Define the **discrete kernel** function K, that computes the number of points in a window of width h

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

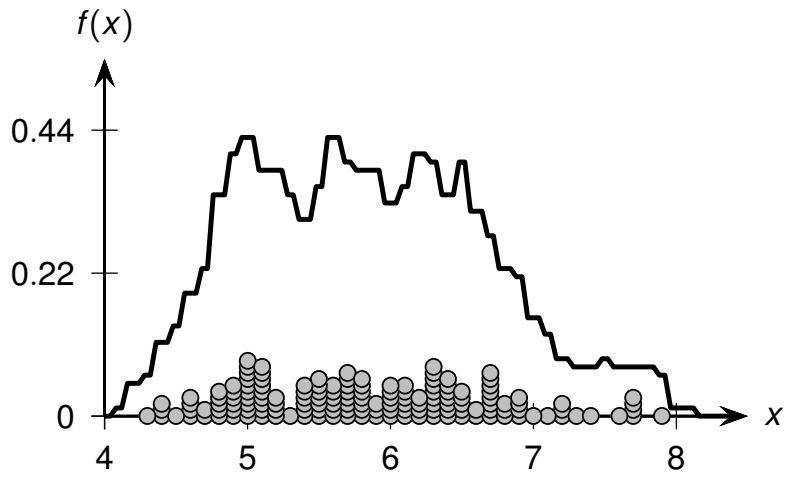
The density estimate $\hat{f}(x)$ can be rewritten in terms of the kernel function as follows:

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

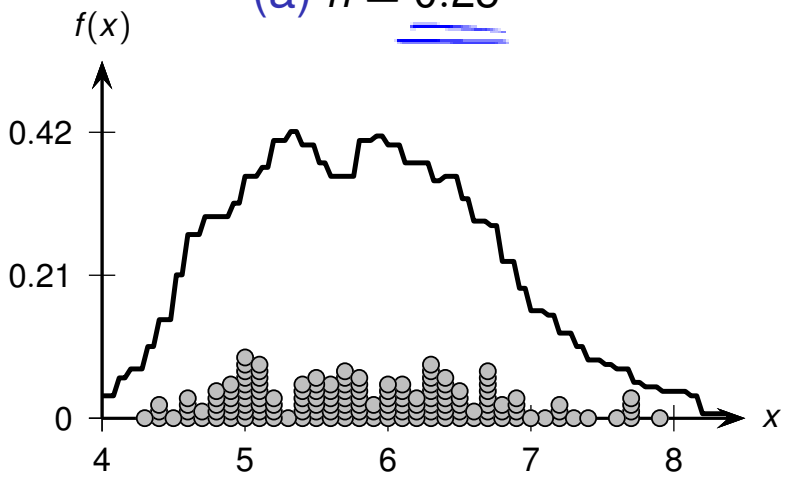
Kernel Density Estimation: Discrete Kernel (Iris 1D)



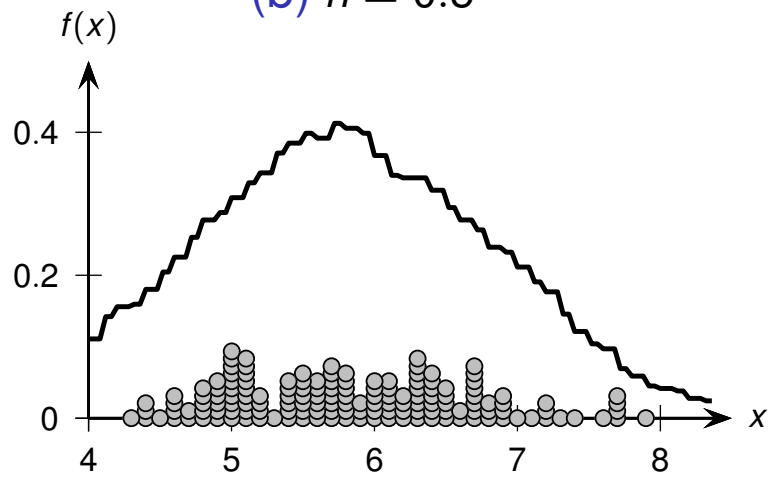
(a) $h = 0.25$



(b) $h = 0.5$



(c) $h = 1.0$



(d) $h = 2.0$

The discrete kernel yields a non-smooth (or jagged) density function.

kernel Density Estimation: Gaussian Kernel

The width h is a parameter that denotes the spread or smoothness of the density estimate. The discrete kernel function has an abrupt influence.

Define a more smooth transition of influence via a Gaussian kernel:

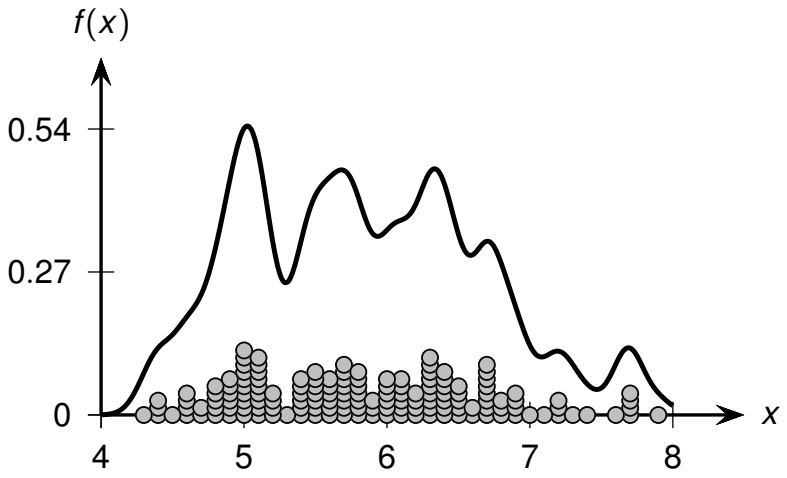
$$K(z) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{z^2}{2}\right\}$$

Thus, we have

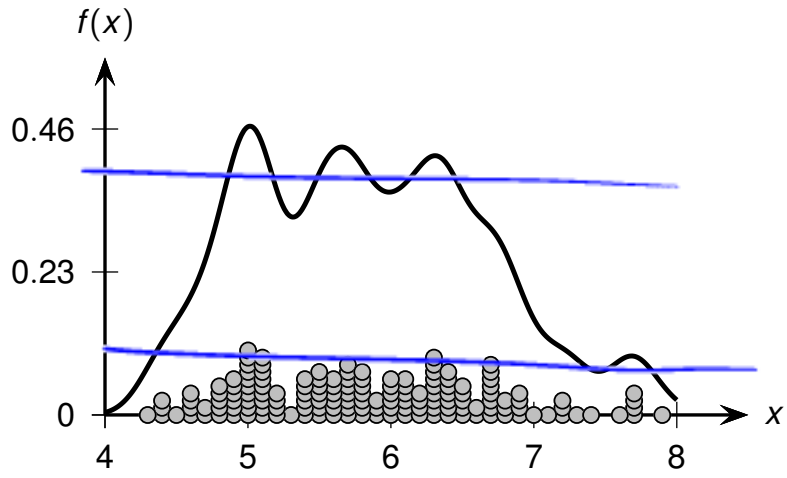
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - x_i)^2}{2h^2}\right\}$$

Here x, which is at the center of the window, plays the role of the mean, and h acts as the standard deviation.

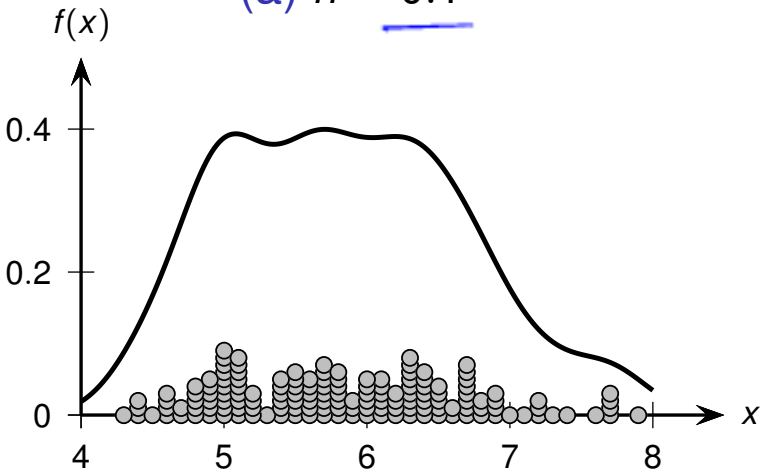
Kernel Density Estimation: Gaussian Kernel (Iris 1D)



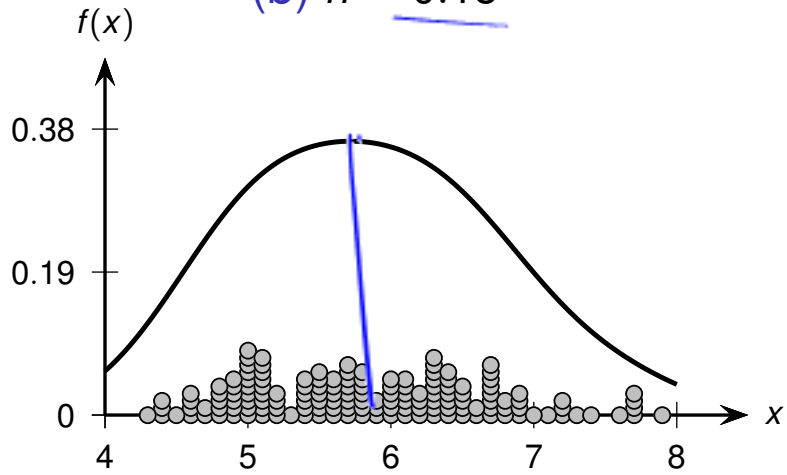
(a) $h = 0.1$



(b) $h = 0.15$



(c) $h = 0.25$



(d) $h = 0.5$

When h is small the density function has many local maxima. A large h results in a unimodal distribution.

Multivariate Density Estimation

To estimate the probability density at a d-dimensional point $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$, we define the d-dimensional "window" as a hypercube in d dimensions, that is, a hypercube centered at $\mathbf{x}$ with edge length h. The volume of such a d-dimensional hypercube is given as

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

The density is estimated as the fraction of the point weight lying within the d-dimensional window centered at $\mathbf{x}$, divided by the volume of the hypercube:

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

where the multivariate kernel function K satisfies the condition $\int K(\mathbf{z}) d\mathbf{z} = 1$.

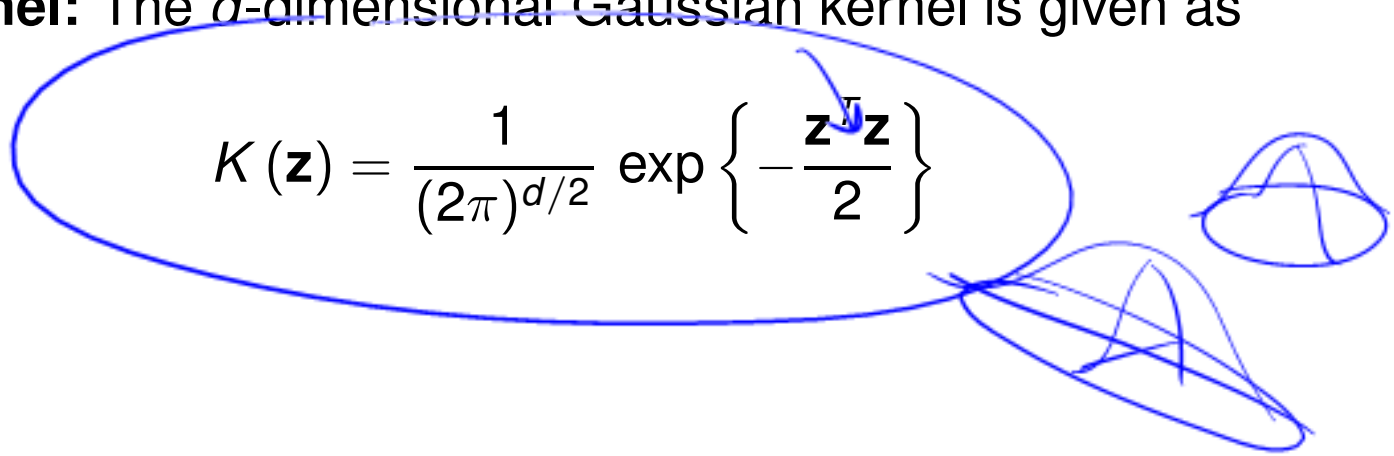
Multivariate Density Estimation: Discrete and Gaussian Kernel

Discrete Kernel: For any d-dimensional vector $\mathbf{z} = (z_1, z_2, \ldots, z_d)^T$, the discrete kernel function in d-dimensions is given as

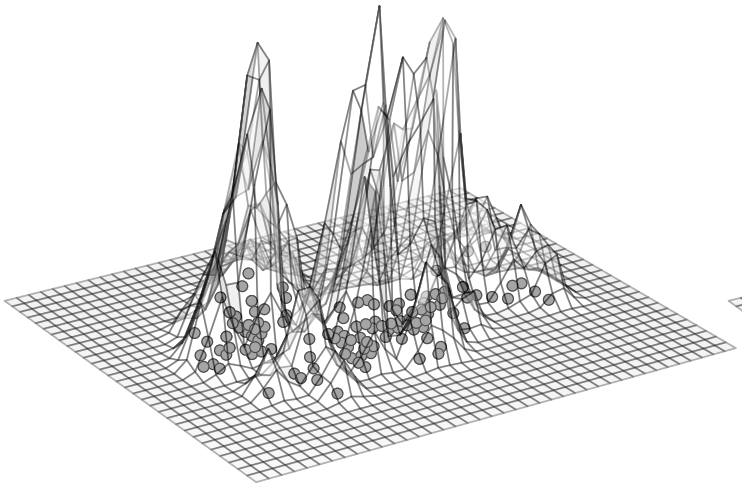
$$K(\mathbf{z}) = \begin{cases} 1 & \text{If } |z_j| \leq \frac{1}{2}, \text{ for all dimensions } j = 1, \ldots, d \\ 0 & \text{Otherwise} \end{cases}$$

Gaussian Kernel: The d-dimensional Gaussian kernel is given as

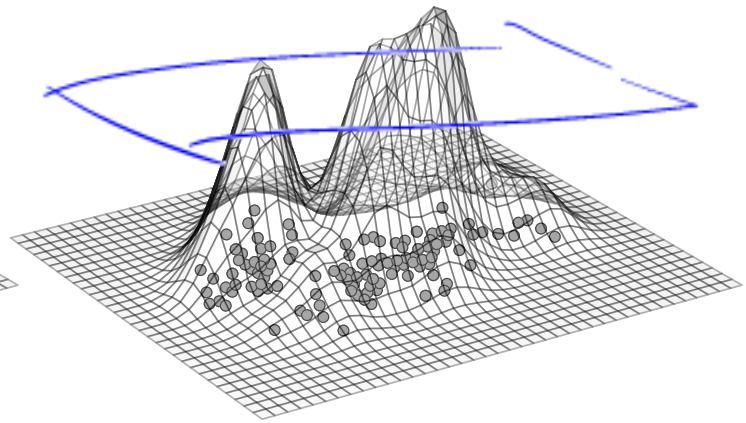
$$K(\mathbf{z}) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{\mathbf{z}^T\mathbf{z}}{2}\right\}$$



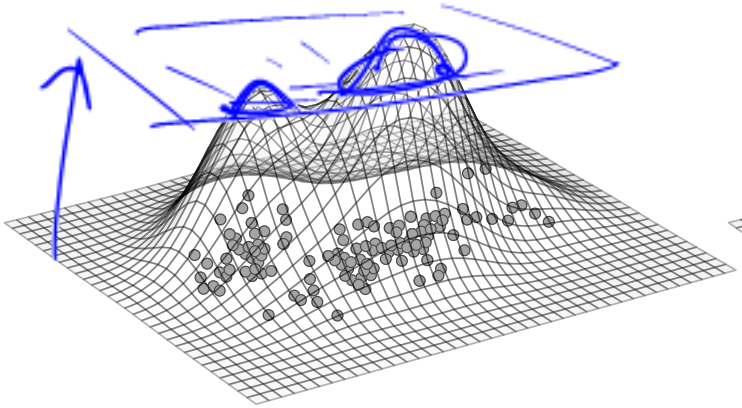
Density Estimation: Iris 2D Data (Gaussian Kernel)



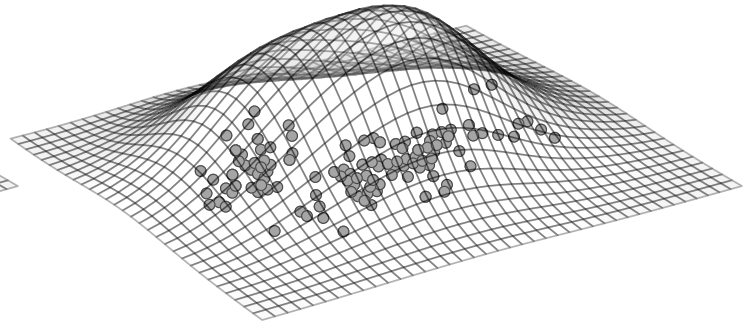
(a) $h = 0.1$



(b) $h = 0.2$



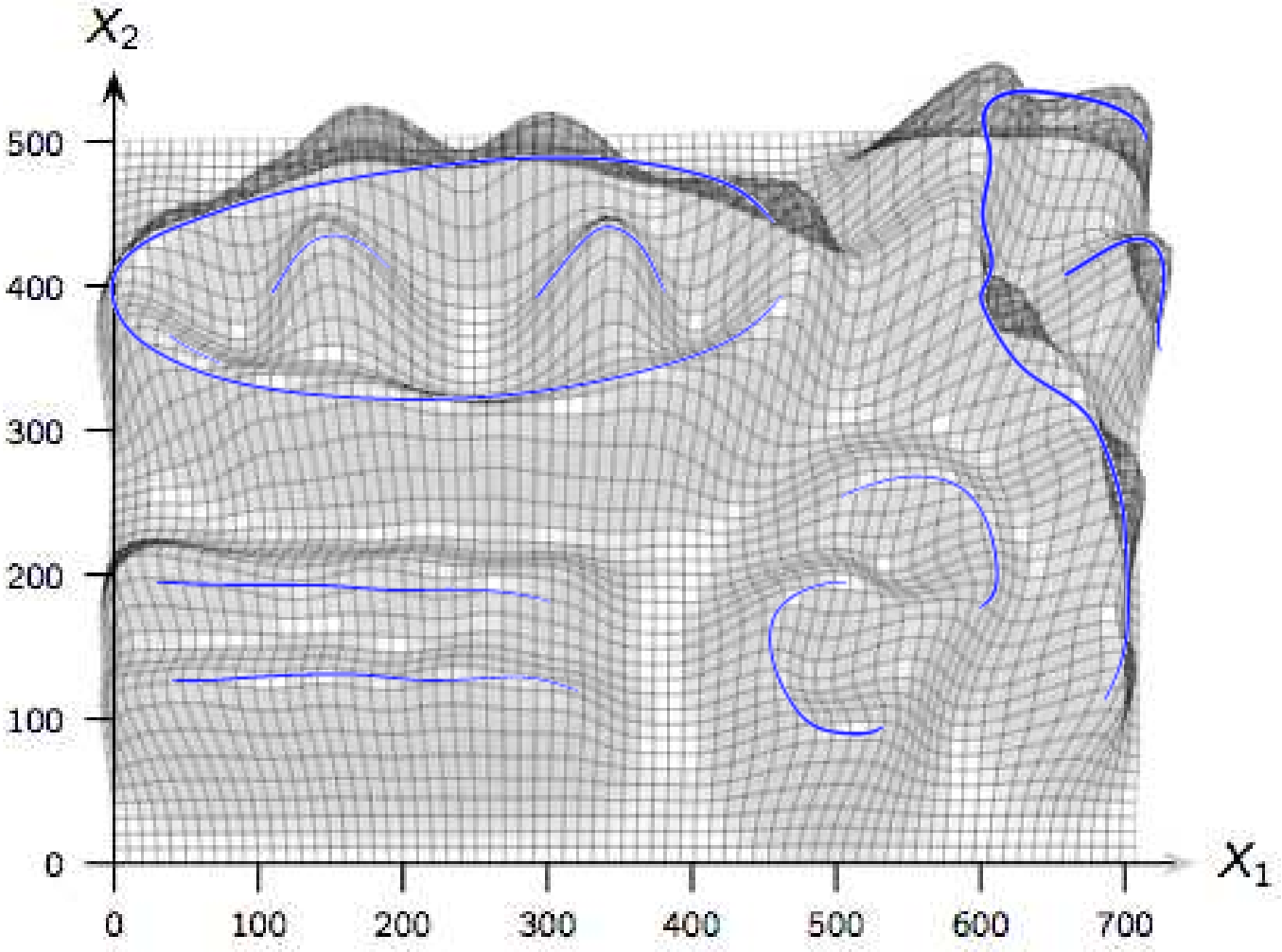
(c) $h = 0.35$



(d) $h = 0.6$

Density Estimation: Density-based Dataset

Gaussian kernel, $h = 20$



Nearest Neighbor Density Estimation

In kernel density estimation we implicitly fixed the volume by fixing the width h, and we used the kernel function to find out the number or weight of points that lie inside the fixed volume region.

An alternative approach to density estimation is to fix k, the number of points required to estimate the density, and allow the volume of the enclosing region to vary to accommodate those k points. This approach is called the k nearest neighbors (KNN) approach to density estimation.

Given k, the number of neighbors, we estimate the density at $\mathbf{x}$ as follows:

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

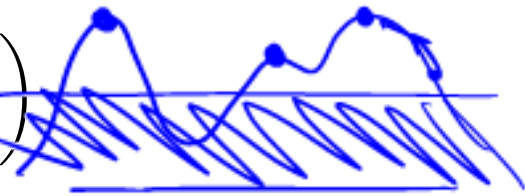
where $h_\mathbf{x}$ is the distance from $\mathbf{x}$ to its kth nearest neighbor, and $\text{vol}(S_d(h_\mathbf{x}))$ is the volume of the d-dimensional hypersphere $S_d(h_\mathbf{x})$ centered at $\mathbf{x}$, with radius $h_\mathbf{x}$.

DENCLUE Density-based Clustering: Attractor and Gradient

A point $\mathbf{x}^*$ is called a *density attractor* if it is a local maxima of the probability density function f.

The density gradient at a point $\mathbf{x}$ is the multivariate derivative of the probability density estimate

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

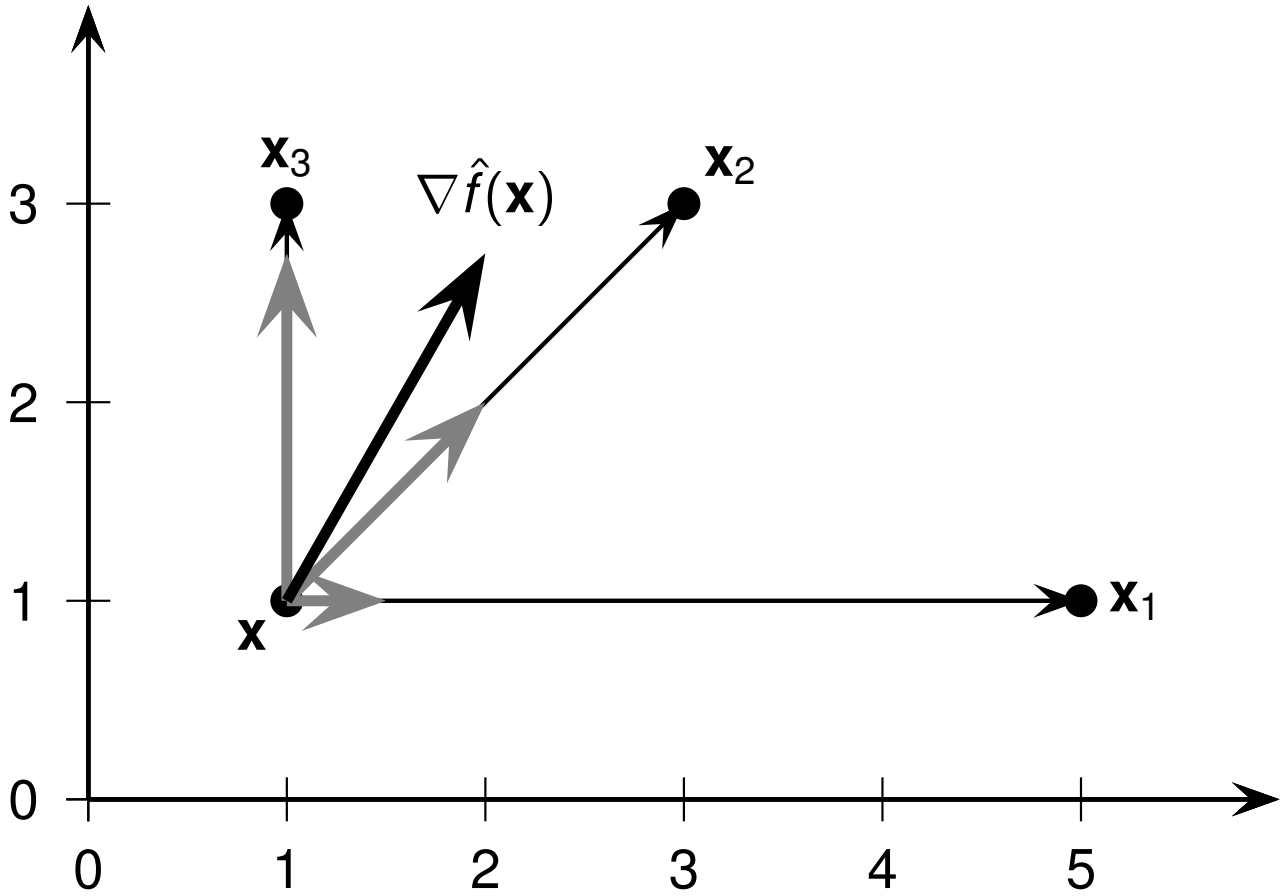


For the Gaussian kernel the gradient at a point $\mathbf{x}$ is given as

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

This equation can be thought of as having two parts for each point: a vector $(\mathbf{x}_i - \mathbf{x})$ and a scalar *influence* value $K(\frac{\mathbf{x}-\mathbf{x}_i}{h})$.

The Gradient Vector



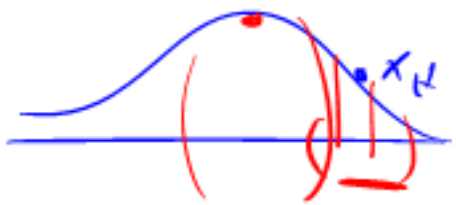
DENCLUE: Density Attractor

We say that $\mathbf{x}^*$ is a *density attractor* for $\mathbf{x}$, or alternatively that $\mathbf{x}$ is *density attracted* to $\mathbf{x}^*$, if a hill climbing process started at $\mathbf{x}$ converges to $\mathbf{x}^*$.

That is, there exists a sequence of points $\mathbf{x} = \mathbf{x}_0 \rightarrow \mathbf{x}_1 \rightarrow \ldots \rightarrow \mathbf{x}_m$, starting from $\mathbf{x}$ and ending at $\mathbf{x}_m$, such that $\|\mathbf{x}_m - \mathbf{x}^*\| \leq \epsilon$, that is, $\mathbf{x}_m$ converges to the attractor $\mathbf{x}^*$.

Setting the gradient to the zero vector leads to the following *mean-shift* update rule:

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



where t denotes the current iteration and $\mathbf{x}_{t+1}$ is the updated value for the current vector $\mathbf{x}_t$.

DENCLUE: Density-based Cluster

A cluster $C \subseteq \mathbf{D}$, is called a *center-defined cluster* if all the points $\mathbf{x} \in C$ are density attracted to a unique density attractor $\mathbf{x}^*$, such that $\hat{f}(\mathbf{x}^*) \geq \xi$, where ξ is a user-defined minimum density threshold.

An arbitrary-shaped cluster $C \subseteq \mathbf{D}$ is called a *density-based cluster* if there exists a set of density attractors $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_m^*$, such that

1. Each point $\mathbf{x} \in C$ is attracted to some attractor $\mathbf{x}_i^*$.
2. Each density attractor has density above ξ.
3. Any two density attractors $\mathbf{x}_i^*$ and $\mathbf{x}_j^*$ are *density reachable*, that is, there exists a path from $\mathbf{x}_i^*$ to $\mathbf{x}_j^*$, such that for all points $\mathbf{y}$ on the path, $\hat{f}(\mathbf{y}) \geq \xi$.

The DENCLUE Algorithm

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

1 $\mathcal{A} \leftarrow \emptyset$

2 **foreach** $\mathbf{x} \in \mathbf{D}$ **do** // find density attractors

4 $\quad \mathbf{x}^* \leftarrow \text{FINDATTRACTOR}(\mathbf{x}, \mathbf{D}, h, \epsilon)$

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** $\mathcal{C}$

The DENCLUE Algorithm: Find Attractor

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

$t \leftarrow 0$

$\mathbf{x}_t \leftarrow \mathbf{x}$

repeat

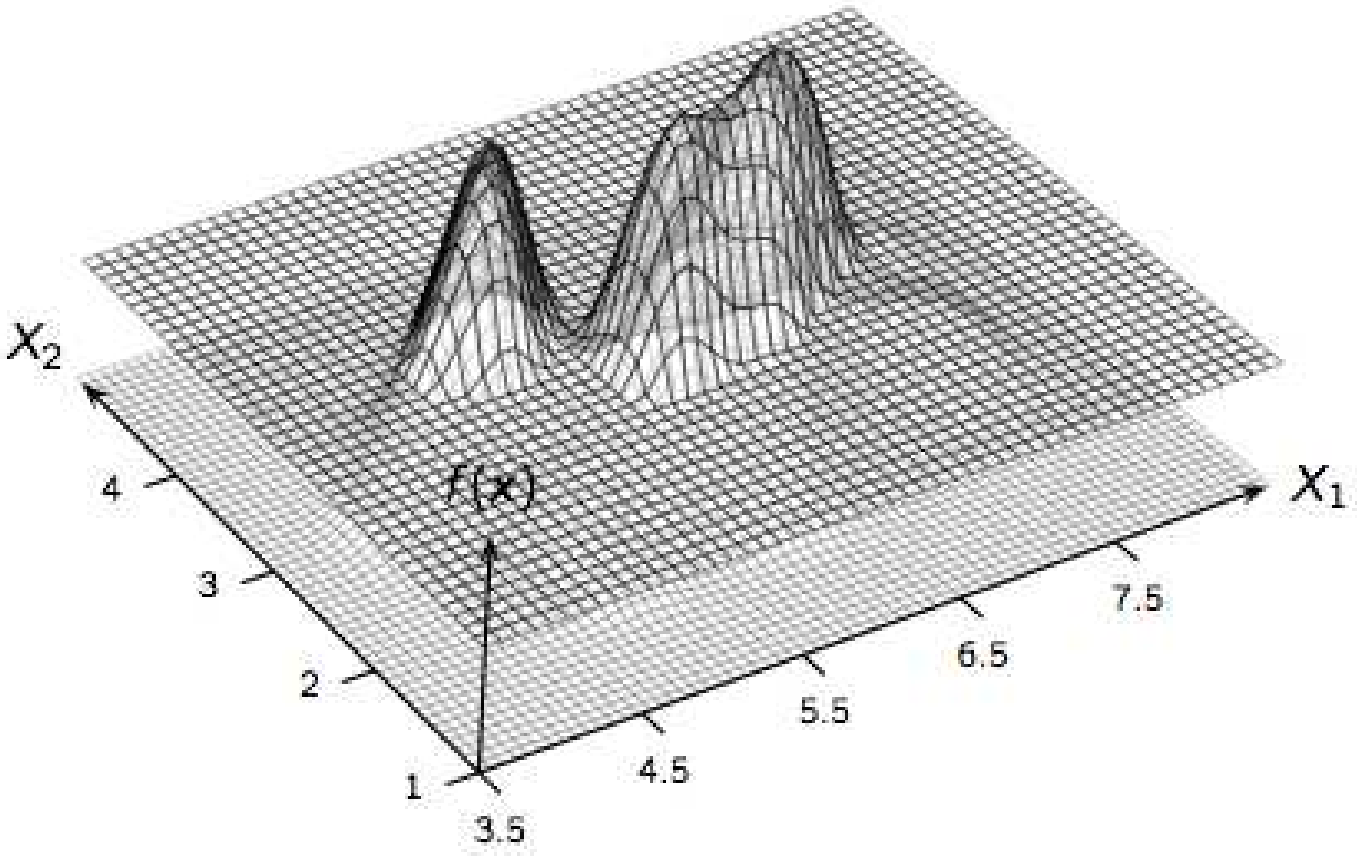
$$\mathbf{x}_{t+1} \leftarrow \frac{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right) \cdot \mathbf{x}_t}{\sum_{i=1}^{n} K\left(\frac{\mathbf{x}_t - \mathbf{x}_i}{h}\right)}$$

$t \leftarrow t + 1$

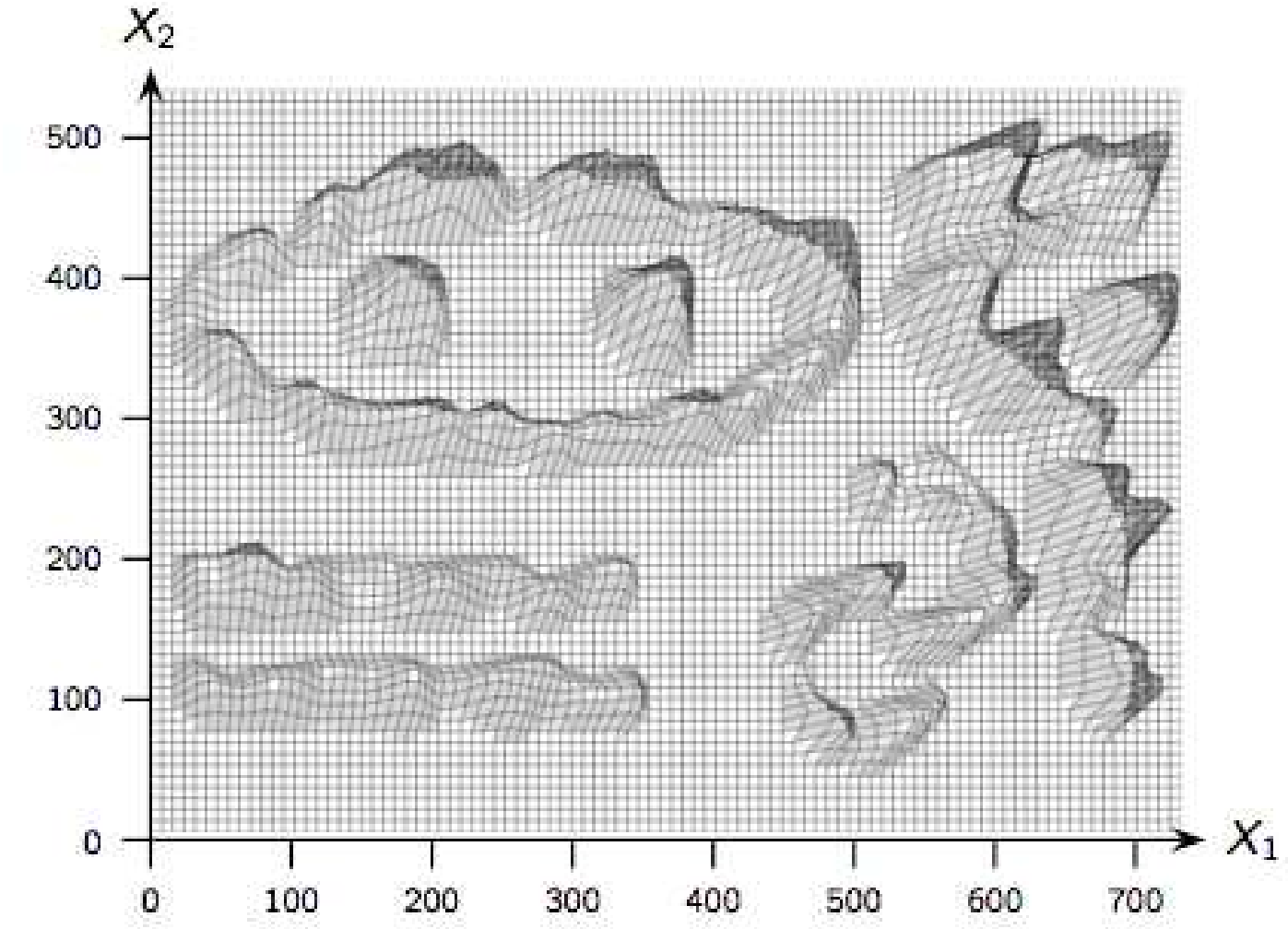
until $\|\mathbf{x}_t - \mathbf{x}_{t-1}\| \leq \epsilon$

return $\mathbf{x}_t$

DENCLUE: Iris 2D Data



DENCLUE: Density-based Dataset



Exam II Syllabus

1. Bayes classifier
   - → full
   - → Naive
2. Decision Trees
3. Neural Networks
   - → feed forward
   - → back propagation
   - → decision boundaries
4. Classifier Evaluation
   - → measures
   - → Roc/Auc
   - → cross-validation / bootstrapping
   - → t-test, confidence
   - → Ensembles
5. Regression
   - → linear
   - → logistic
   - → shrinkage
6. Clustering
   - → kmeans
   - → EM
   - → hierarchical