# CENG 463
# Machine Learning

## Lecture 08 - Nonparametric Classification

Slides were prepared using Introduction to Machine Learning 2e, E. Alpaydin, 2010

# Why not parametric methods?

- Parametric approaches require knowing the form of the density.
  - E.g. With ML estimation in Lecture 2, we assumed that the underlying function of our data is a Gaussian.
- However, in many cases,
  - Either the form is not known;
  - Or the form does not let you to find a unique solution.
    - In other words, the distribution consists of multi-modal densities (e.g. one Gaussian and one uniform distribution together).
- The solution is to use **nonparametric methods**.

# Nonparametric Methods

- Ideas behind nonparametric methods:
  - “Similar inputs have similar outputs”
  - “Let the training data speak for itself”
- What is done basically:
  - Given x, find a small number of **closest training instances** and interpolate from these.
- Also known as:
  - case-based / instance-based learning
  - memory-based / lazy learning
    - since  no models are trained, all of the training samples is needed to asses a new query.

# Density Estimation

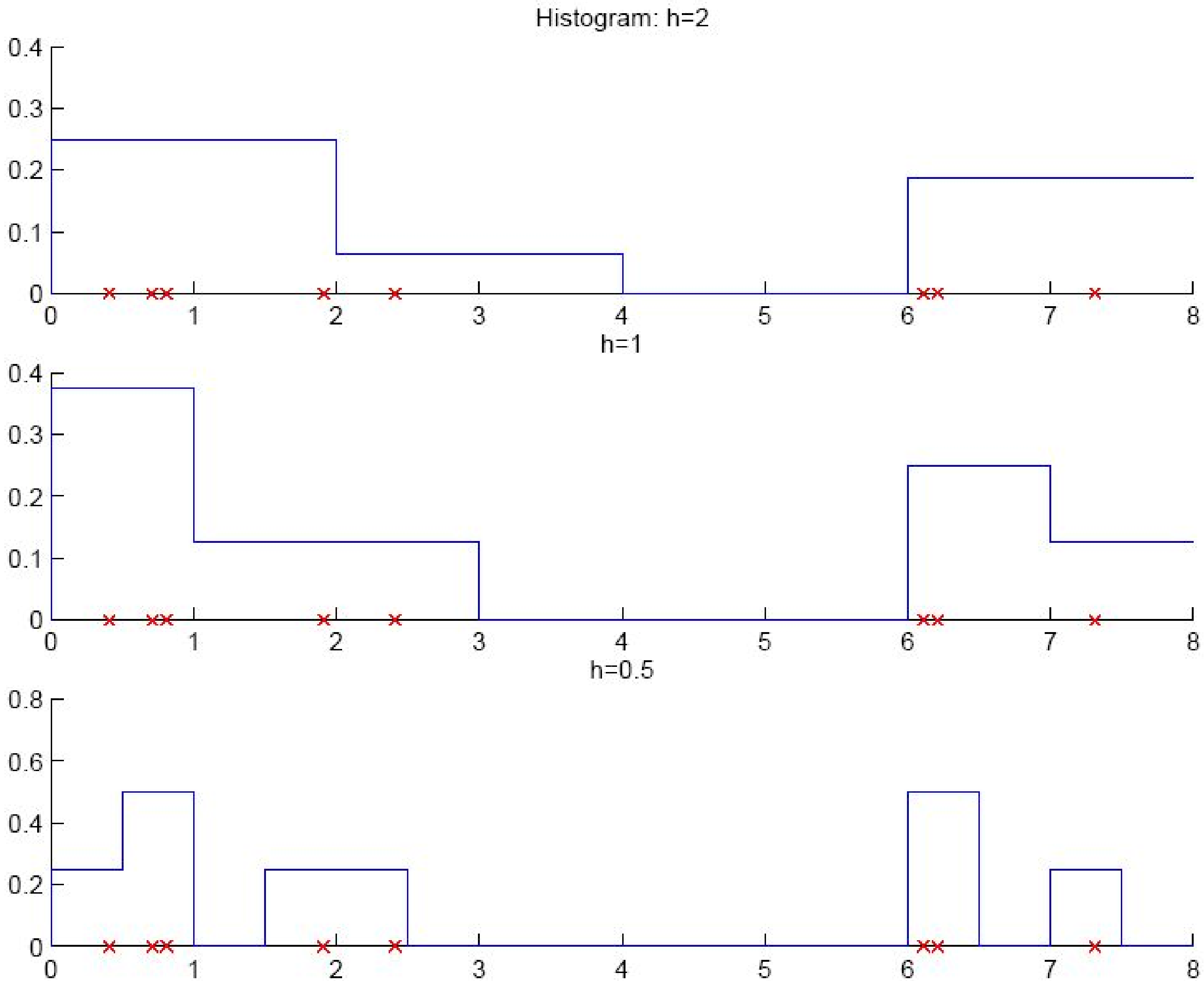
- Given the training set $X = \{x^t\}_{t=1:N}$ with samples from 1D space, let $\hat{p}(x)$ be our probability estimate at point x.
- There are many nonparametric methods to calculate $\hat{p}(x)$.
- We'll learn about the following:
    - Histogram Estimator
    - Naive Estimator
    - Kernel Estimator
    - K Nearest Neighbor Estimator

# Histogram Estimator

- Simplest approach is using a histogram:
    - Divide data into bins of size h
    - Estimator:

$$\hat{p}(x) = \frac{\#\{x^t \text{ in the same bin as } x\}}{Nh}$$

# Histogram Estimator



# Naive Estimator

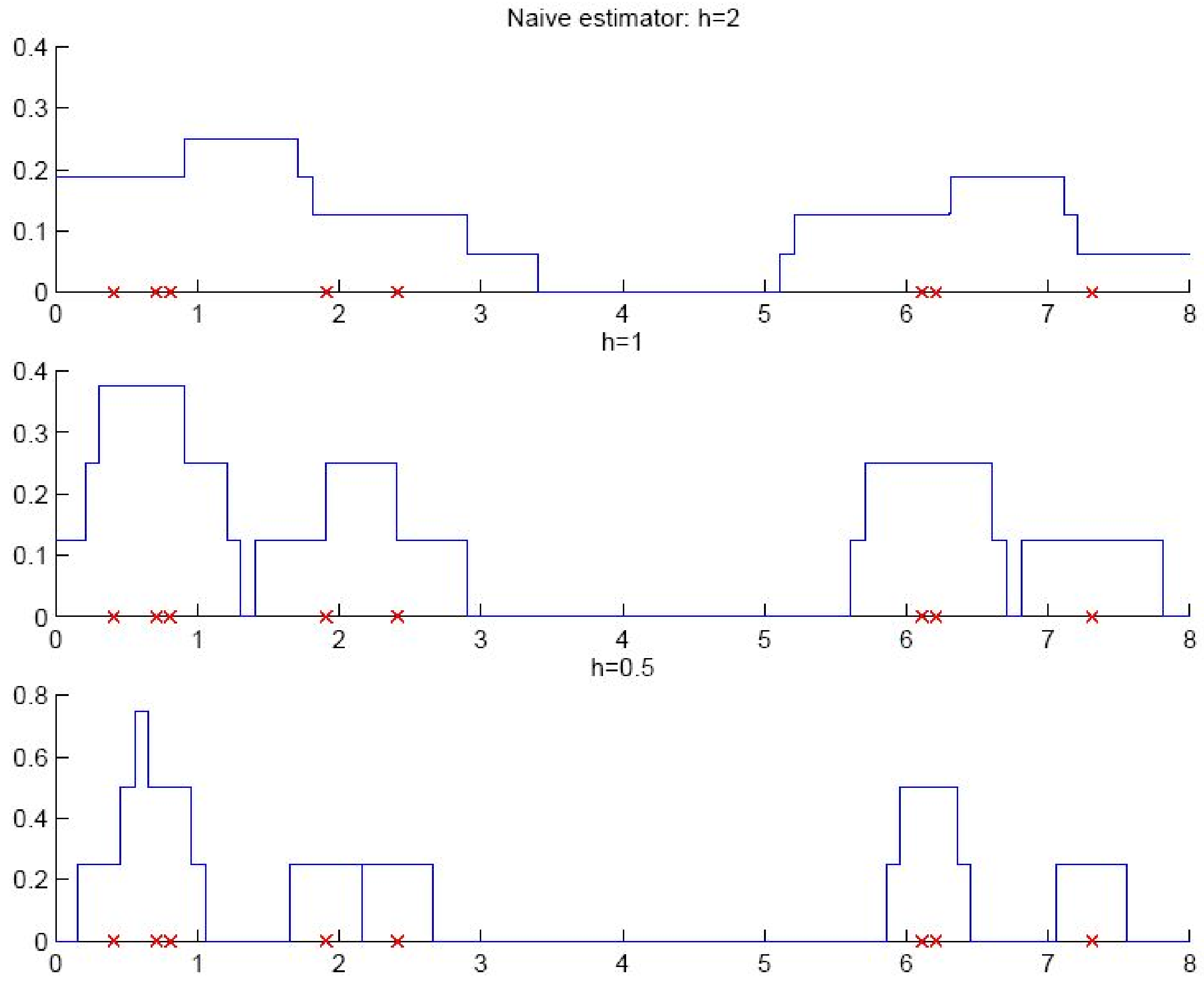
- This time, h does not represent the bin but neighbourhood of the given x point:

$$\hat{p}(x) = \frac{\#\{x - h/2 < x^t \le x + h/2\}}{Nh}$$

- Same thing can be written as a sum of weighted contribution where w is the weight function:

$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right) \qquad w(u) = \begin{cases} 1 & \text{if } |u| < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

# Naive Estimator



# Kernel Estimator

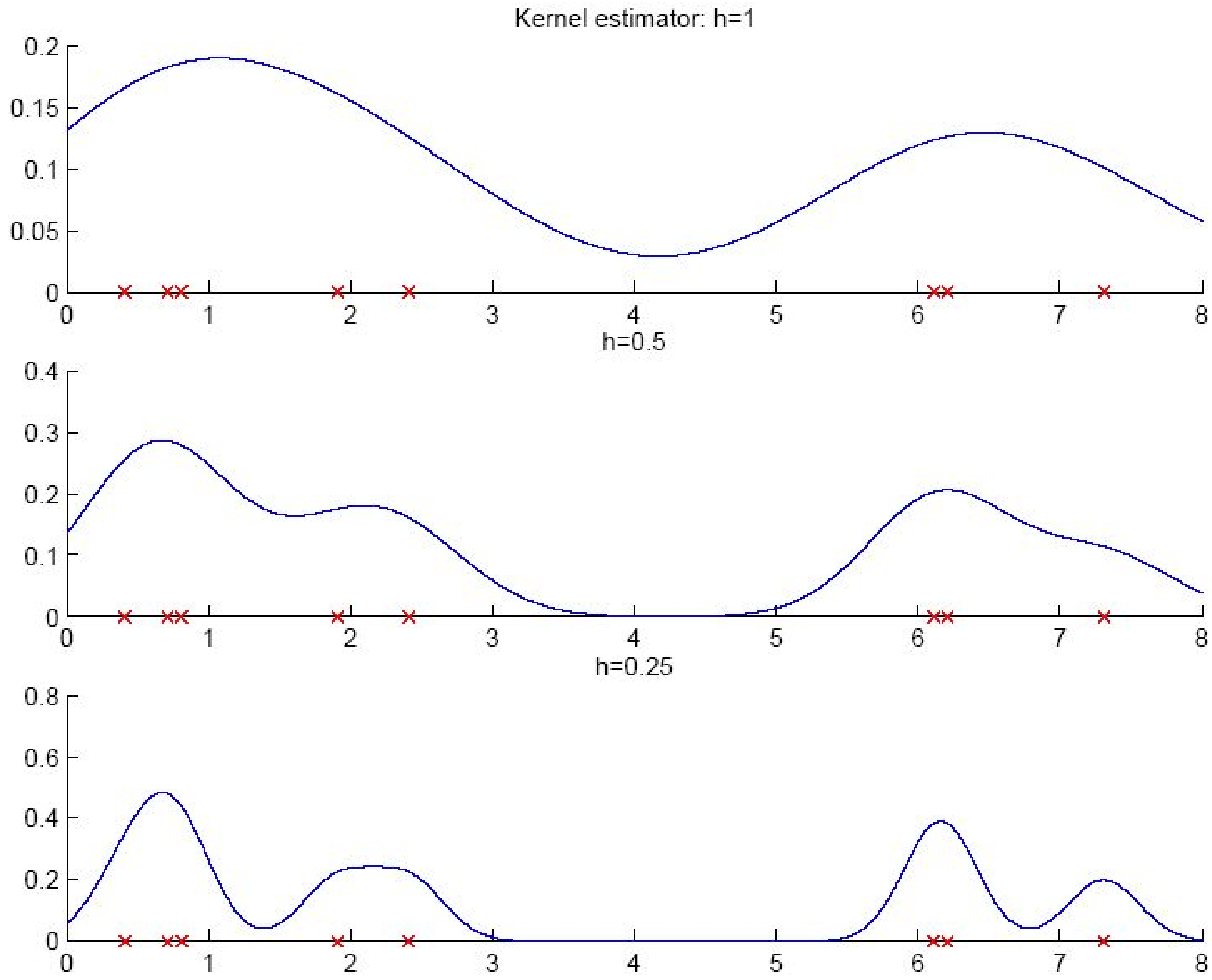
- Since the region of influence in Naive estimator is '**hard**' (0 or 1), the estimate is not a continuous function and has jumps at $x^t \pm h/2$.
- To get a smooth estimate, we use a smooth weight function, called a **kernel**. The most popular is the Gaussian kernel:

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

- The kernel estimator, (a.k.a. Parzen windows) is defined as:

$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)$$

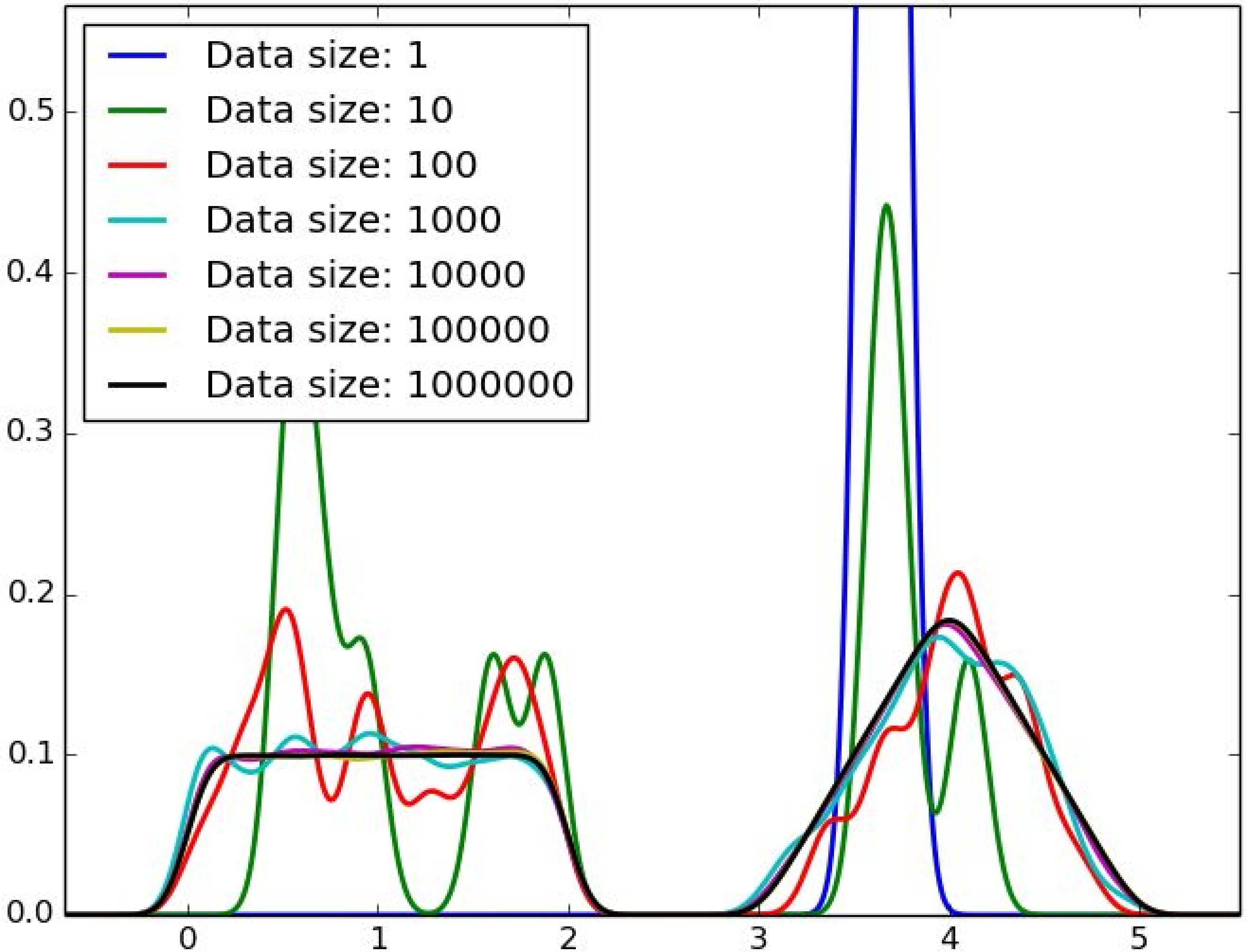
# Kernel Estimator



# Adaptation to multi-modal densities

- As mentioned at the beginning, nonparametric approaches have power to represent multi-modal densities.
  - An example is given with a bimodal density (one pyramid and one uniform distribution).
  - As more samples are provided, the result of the kernel estimator becomes closer to the true density.

# Adaptation to multi-modal densities



# K Nearest Neighbor Estimator

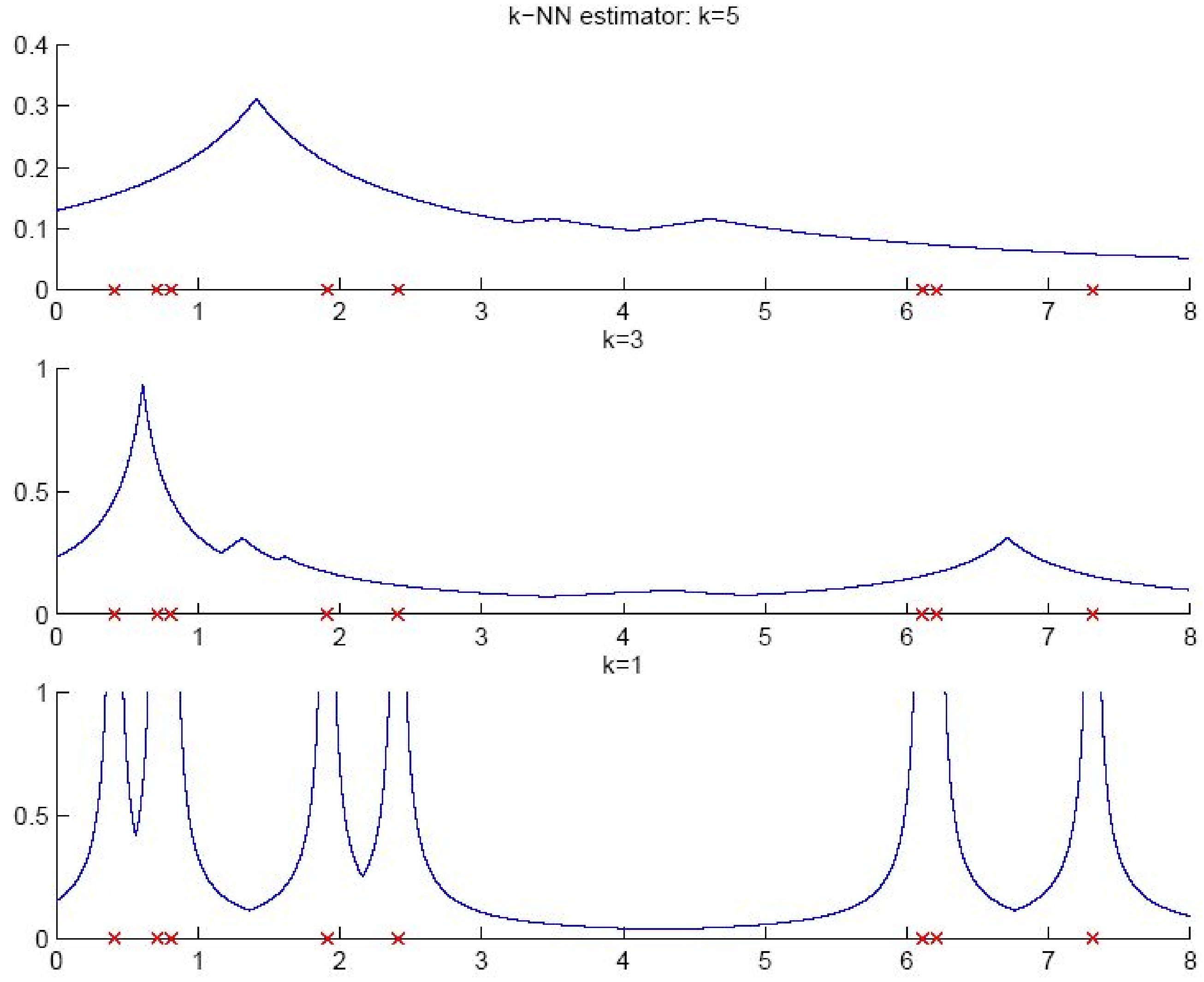
- Instead of fixing bin width h and counting the number of instances, we fix the number of instances (neighbours) k and check bin width:

$$\hat{p}(x) = \frac{k}{2Nd_k(x)}$$

  where $d_k(x)$ is distance of $k^{th}$ closest instance to x

- **<u>Notice</u>**: When $d_k(x)=h/2$, k-NN gives the same output with the Naive estimator.

# K Nearest Neighbor Estimator



# Multivariate Data

- Kernel density estimator: $\hat{p}(x) = \frac{1}{Nh^d} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)$
- Multivariate Gaussian kernel:
  - with fixed variance:

$$K(u) = \left(\frac{1}{\sqrt{2\pi}}\right)^d \exp\left[-\frac{\|u\|^2}{2}\right]$$

  - with different variance in different dimensions (S: covariance matrix):

$$K(u) = \frac{1}{(2\pi)^{d/2} |S|^{1/2}} \exp\left[-\frac{1}{2} u^T S^{-1} u\right]$$

# Nonparametric Classification

- Estimate $p(x|C_i)$ and use Bayes' rule to calculate $P(C_i|x)$ to classify
- Kernel estimator:

$$\hat{p}(x|C_i) = \frac{1}{N_i h^d} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right) r_i^t \qquad r_i^t = \begin{cases} 1 \text{ if } x^t \in C_i \\ 0 \text{ if } x^t \in C_j,\ j \neq i \end{cases}$$

$$\hat{P}(C_i) = \frac{N_i}{N}$$

$$g_i(x) = \underbrace{\hat{p}(x|C_i)\hat{P}(C_i)}_{\text{unnormalized posterior}} = \frac{1}{Nh^d} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right) r_i^t$$

# Nonparametric Classification

- kNN estimator:

$$\hat{p}(x|C_i) = \frac{k_i}{N_i V^k(x)} \qquad \hat{P}(C_i|x) = \frac{\hat{p}(x|C_i)\hat{P}(C_i)}{\hat{p}(x)} = \frac{k_i}{k}$$

  - $k_i$ is the number of neighbours out of k nearest that belong to $C_i$
  - $V_k(x)$ is the volume of the d-dimensional hypersphere centered at x, with radius $r = x - x(k)$.
  - When k=1, k-NN divides the space in the form of a **Voronoi tesselation**.