

Lecture Slides for

# INTRODUCTION TO MACHINE LEARNING

3RD EDITION

ETHEM ALPAYDIN  
© The MIT Press, 2014

[alpaydin@boun.edu.tr](mailto:alpaydin@boun.edu.tr)  
<http://www.cmpe.boun.edu.tr/~ethem/i2ml3e>

CHAPTER 8:

# NONPARAMETRIC METHODS

# Nonparametric Estimation

3

- Parametric (single global model), semiparametric (small number of local models)
- Nonparametric: Similar inputs have similar outputs
- Functions (pdf, discriminant, regression) change smoothly
- Keep the training data; “let the data speak for itself”
- Given  $x$ , find a small number of **closest** training instances and **interpolate** from these
- Aka lazy/memory-based/case-based/instance-based learning

# Density Estimation

4

- Given the training set  $\mathbf{X}=\{x^t\}_t$  drawn iid from  $p(x)$
- Divide data into bins of size  $h$

- Histogram:

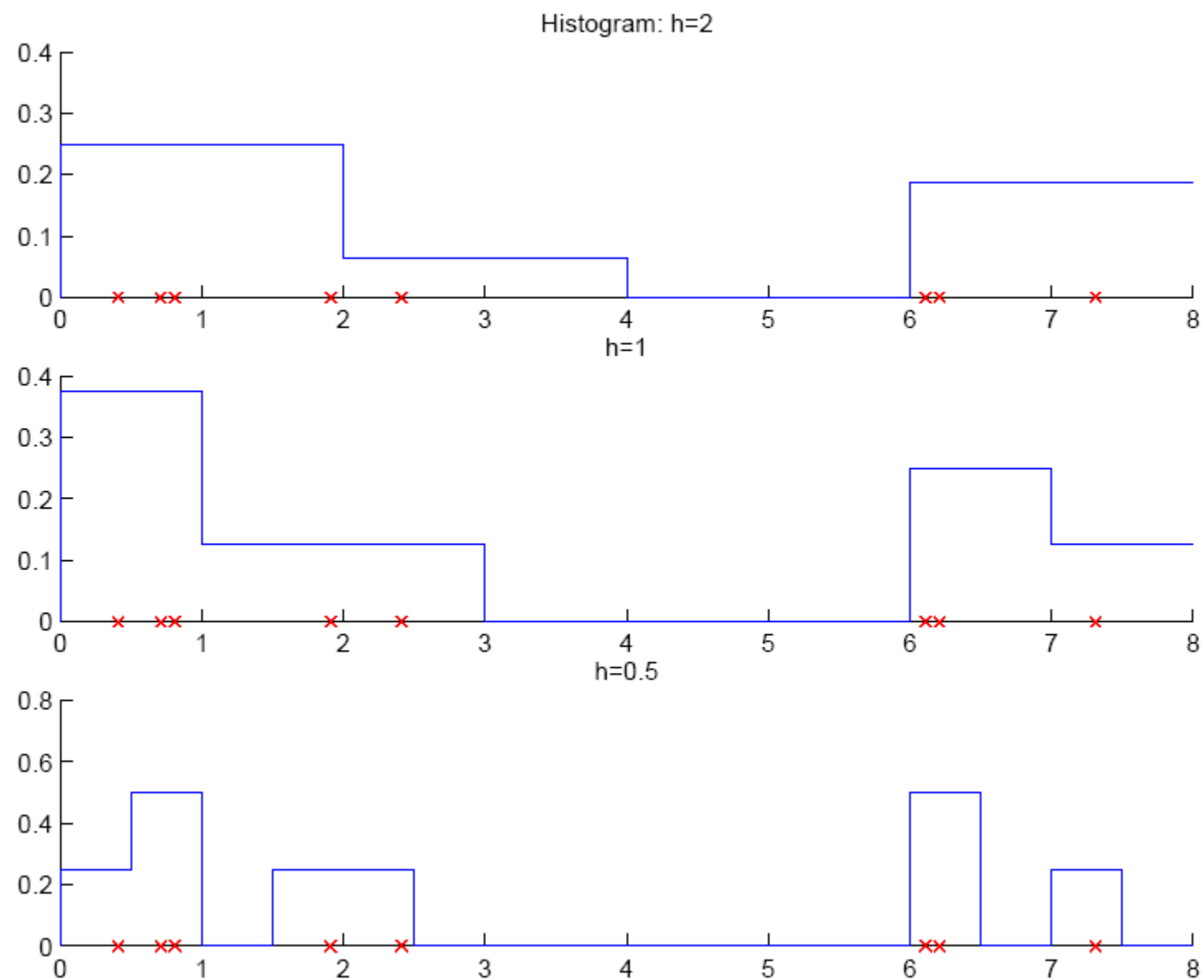
$$\hat{p}(x) = \frac{\#\{x^t \text{ in the same bin as } x\}}{Nh}$$

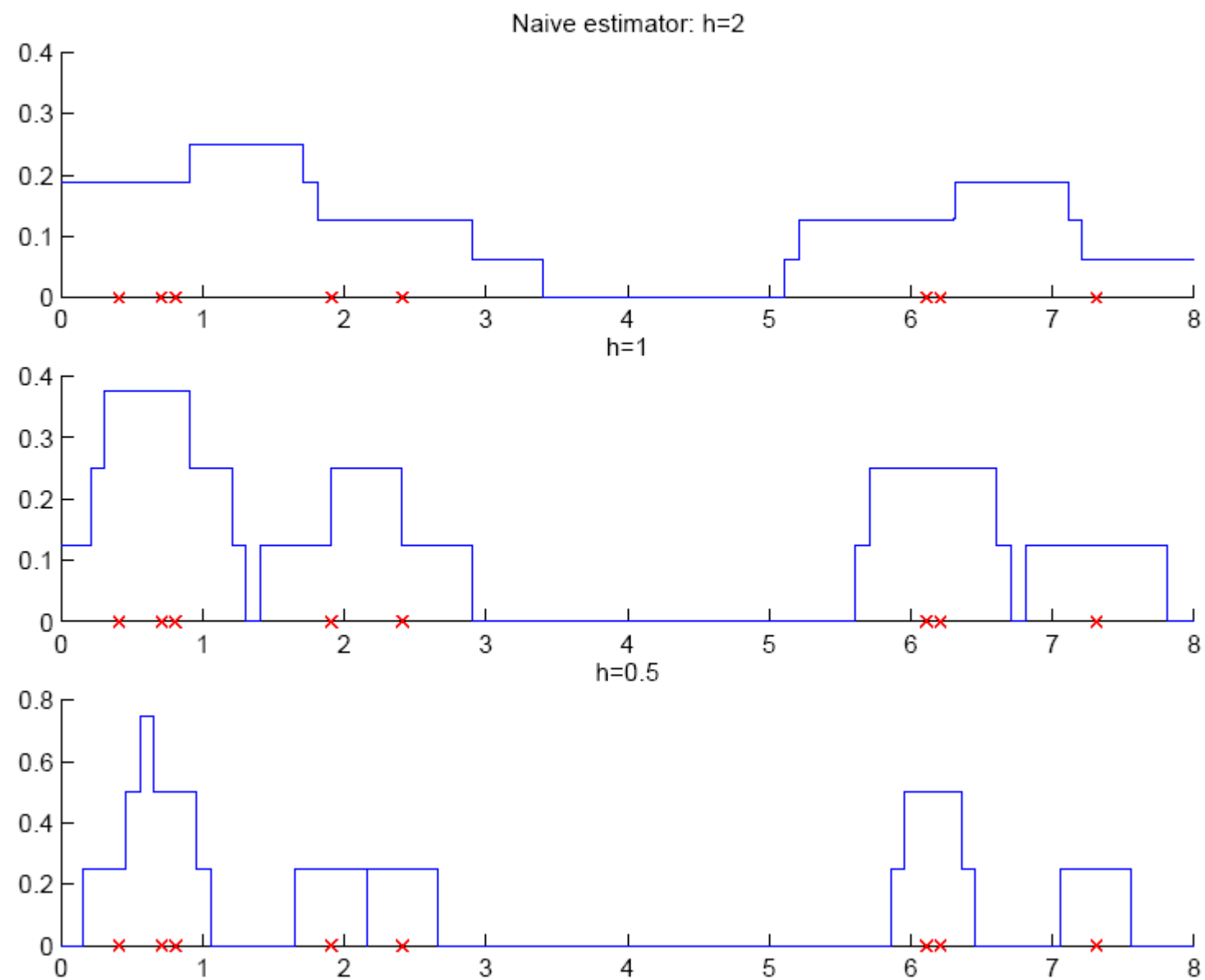
- Naive estimator:

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

or

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





# Kernel Estimator

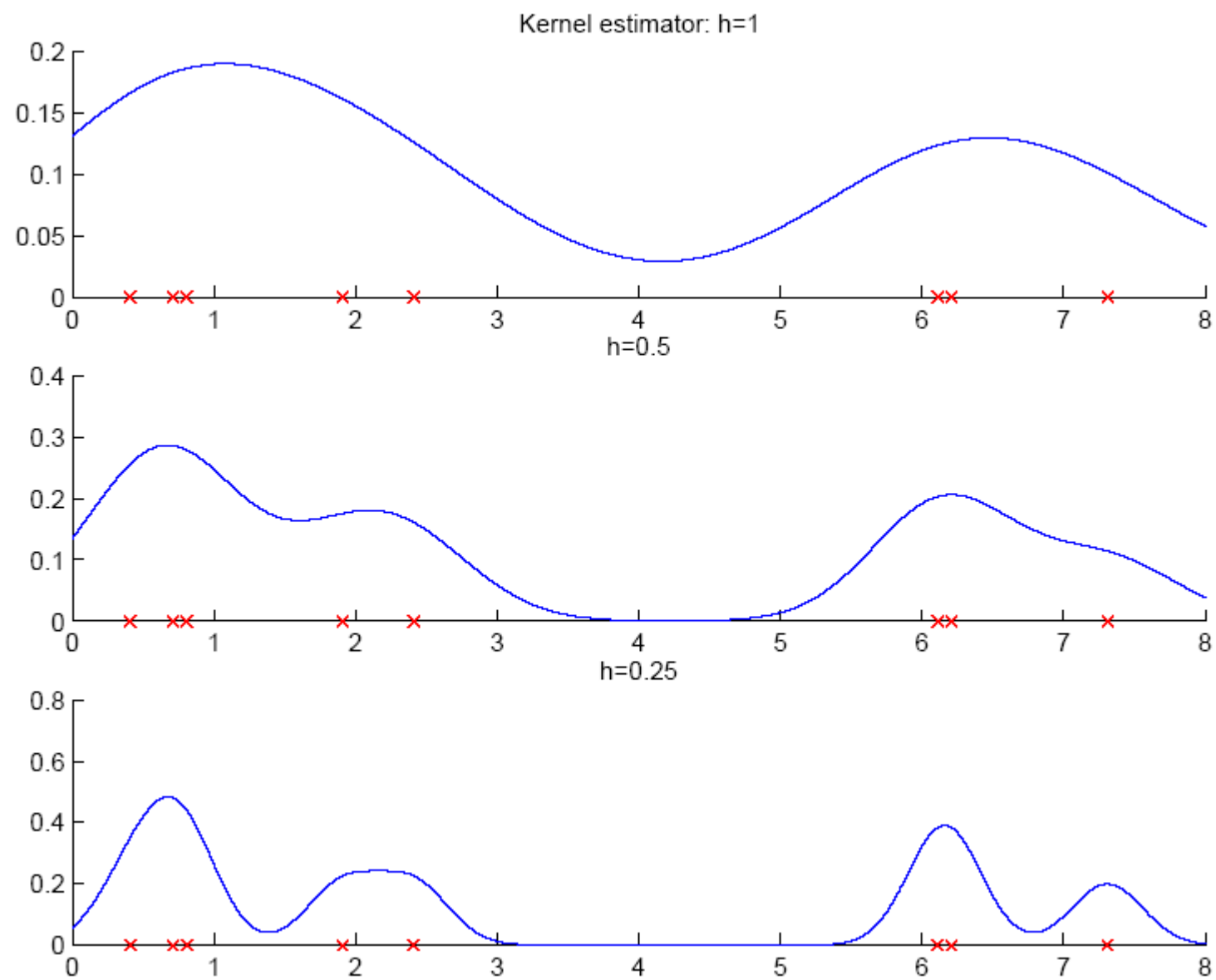
7

- Kernel function, e.g., Gaussian kernel:

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

- Kernel estimator (Parzen windows)

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





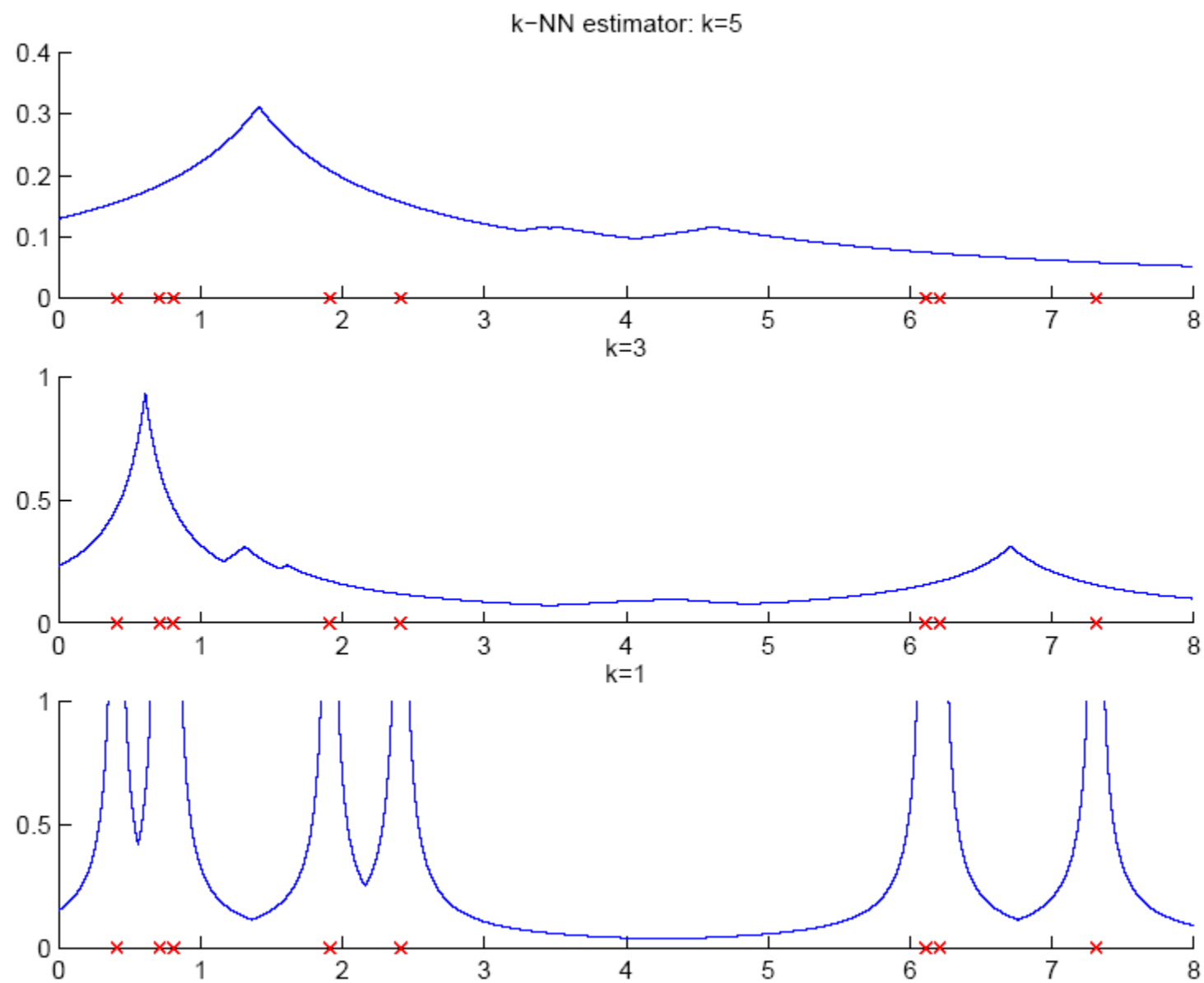
# k-Nearest Neighbor Estimator

9

- Instead of fixing bin width  $h$  and counting the number of instances, fix the instances (neighbors)  $k$  and check bin width

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

$d_k(x)$ , distance to  $k$ th closest instance to  $x$



# Multivariate Data

11

- Kernel density estimator

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

Multivariate Gaussian kernel

spheric

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

ellipsoid

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

# Nonparametric Classification

12

- Estimate  $p(\mathbf{x} | C_i)$  and use Bayes' rule
- Kernel estimator

$$\hat{p}(\mathbf{x} | C_i) = \frac{1}{N_i h^d} \sum_{t=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}^t}{h}\right) r_i^t \quad \hat{P}(C_i) = \frac{N_i}{N}$$

$$g_i(\mathbf{x}) = \hat{p}(\mathbf{x} | C_i) \hat{P}(C_i) = \frac{1}{N h^d} \sum_{t=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}^t}{h}\right) r_i^t$$

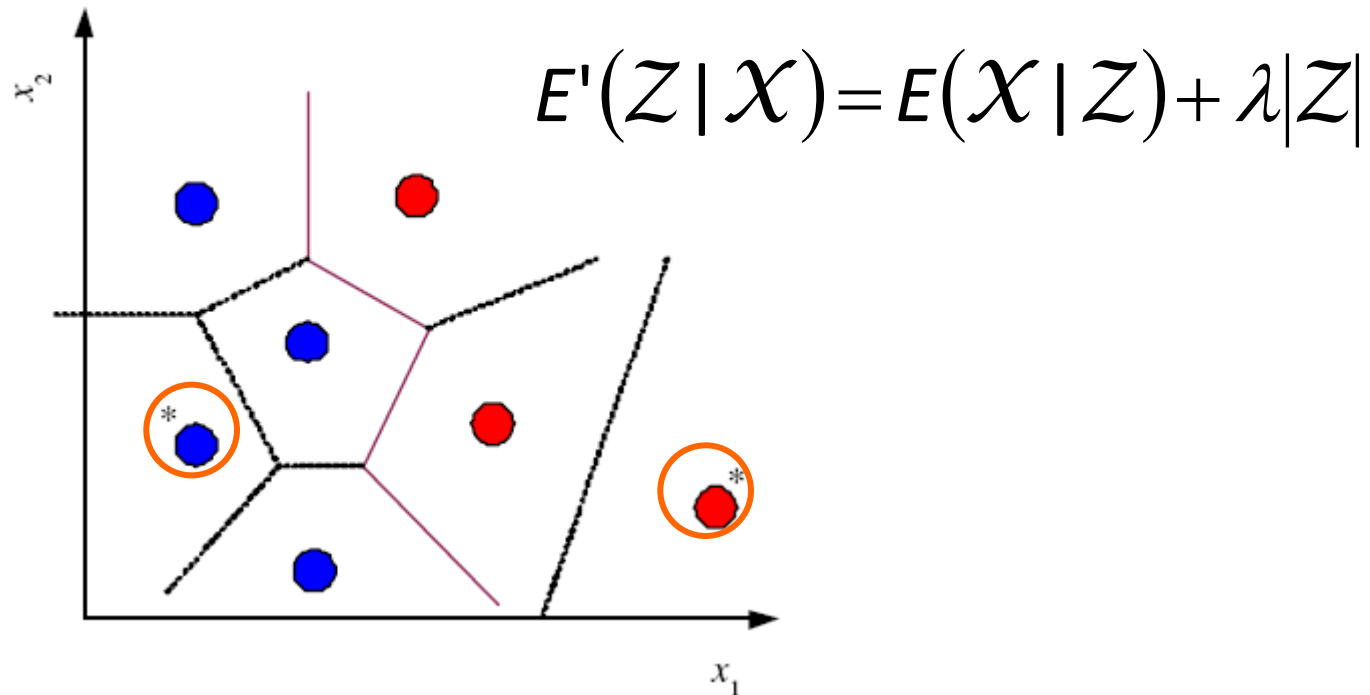
- $k$ -NN estimator

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

# Condensed Nearest Neighbor

13

- Time/space complexity of  $k$ -NN is  $O(N)$
- Find a subset  $Z$  of  $X$  that is small and is accurate in classifying  $X$  (Hart, 1968)



# Condensed Nearest Neighbor

14

- Incremental algorithm: Add instance if needed

$\mathcal{Z} \leftarrow \emptyset$

Repeat

For all  $\mathbf{x} \in \mathcal{X}$  (in random order)

Find  $\mathbf{x}' \in \mathcal{Z}$  s.t.  $\|\mathbf{x} - \mathbf{x}'\| = \min_{\mathbf{x}^j \in \mathcal{Z}} \|\mathbf{x} - \mathbf{x}^j\|$

If  $\text{class}(\mathbf{x}) \neq \text{class}(\mathbf{x}')$  add  $\mathbf{x}$  to  $\mathcal{Z}$

Until  $\mathcal{Z}$  does not change

# Distance-based Classification

15

- Find a distance function  $\mathcal{D}(\mathbf{x}^r, \mathbf{x}^s)$  such that  
if  $\mathbf{x}^r$  and  $\mathbf{x}^s$  belong to the same class, distance is small and if they belong to different classes, distance is large
- Assume a parametric model and learn its parameters using data, e.g.,

$$\mathcal{D}(\mathbf{x}, \mathbf{x}^t | \mathbf{M}) = (\mathbf{x} - \mathbf{x}^t)^T \mathbf{M} (\mathbf{x} - \mathbf{x}^t)$$

# Learning a Distance Function

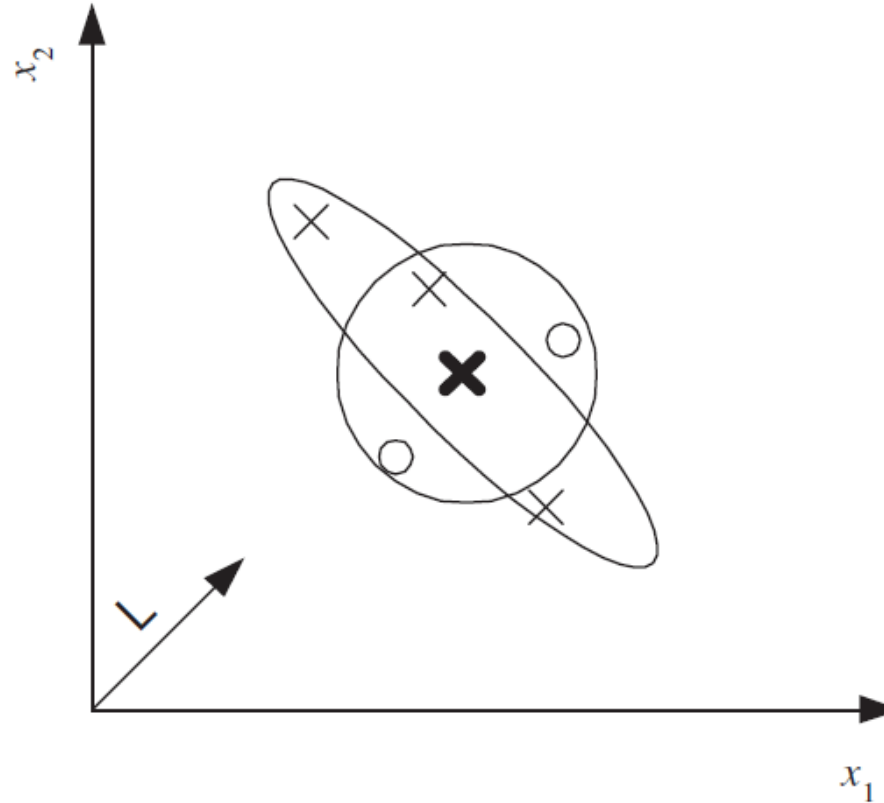
16

- The three-way relationship between distances, dimensionality reduction, and feature extraction.
- $\mathbf{M} = \mathbf{L}^T \mathbf{L}$  is  $d \times d$  and  $\mathbf{L}$  is  $k \times d$

$$\begin{aligned}\mathcal{D}(\mathbf{x}, \mathbf{x}^t | \mathbf{M}) &= (\mathbf{x} - \mathbf{x}^t)^T \mathbf{M} (\mathbf{x} - \mathbf{x}^t) = (\mathbf{x} - \mathbf{x}^t)^T \mathbf{L}^T \mathbf{L} (\mathbf{x} - \mathbf{x}^t) \\ &= (\mathbf{L}(\mathbf{x} - \mathbf{x}^t))^T (\mathbf{L}(\mathbf{x} - \mathbf{x}^t)) = (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}^t)^T (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}^t) \\ &= (\mathbf{z} - \mathbf{z}^t)^T (\mathbf{z} - \mathbf{z}^t) = \|\mathbf{z} - \mathbf{z}^t\|^2\end{aligned}$$

- Similarity-based representation using similarity scores
- Large-margin nearest neighbor (chapter 13)





Euclidean distance (circle) is not suitable,  
Mahalanobis distance using an  $\mathbf{M}$  (ellipse) is suitable.  
After the data is projected along  $\mathbf{L}$ , Euclidean distance can be used.

# Outlier Detection

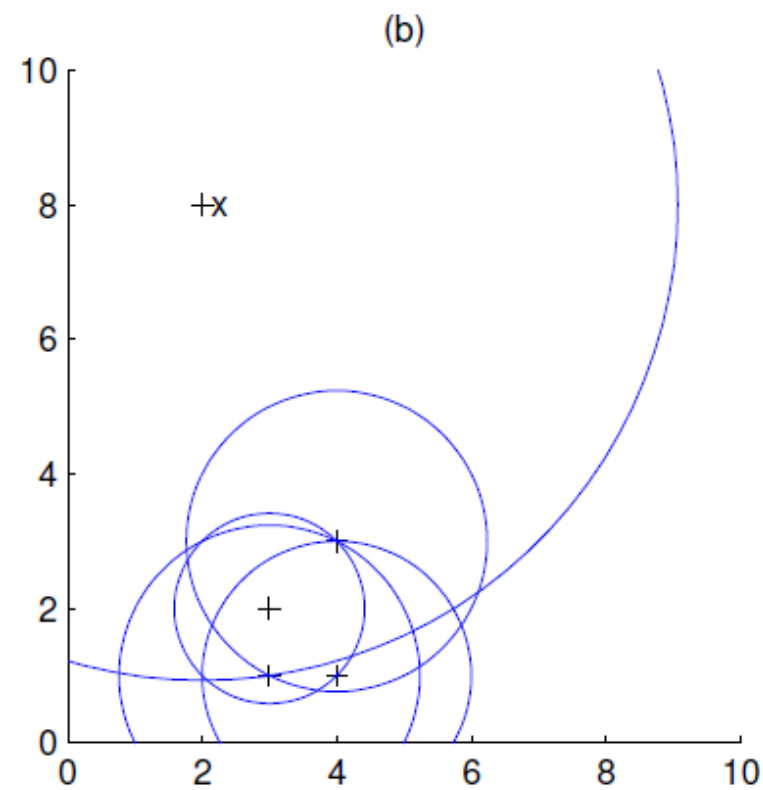
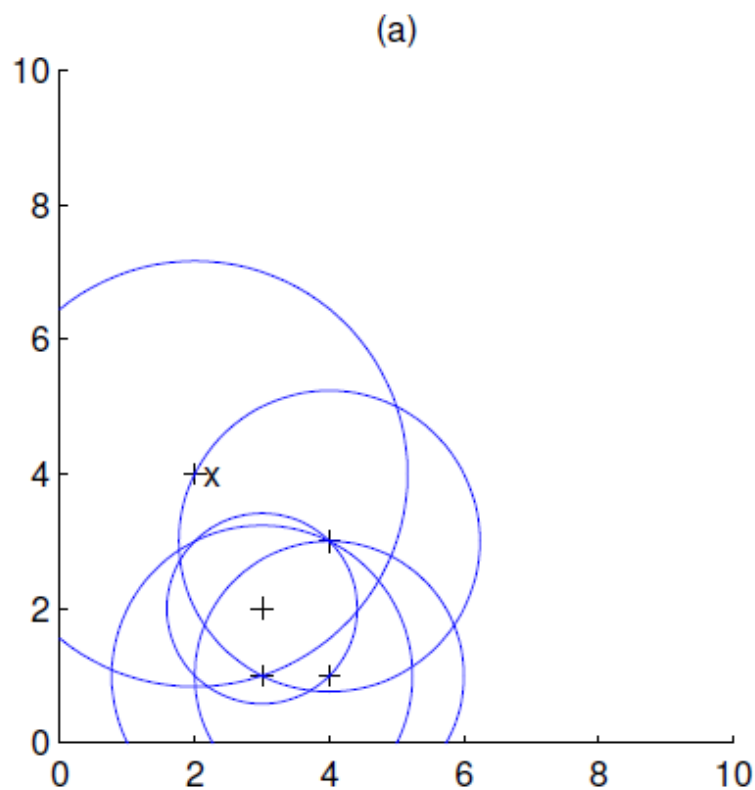
18

- Find outlier/novelty points
- Not a two-class problem because outliers are very few, of many types, and seldom labeled
- Instead, one-class classification problem: Find instances that have low probability
- In nonparametric case: Find instances far away from other instances

# Local Outlier Factor

19

$$\text{LOF}(\mathbf{x}) = \frac{d_k(\mathbf{x})}{\sum_{\mathbf{s} \in \mathcal{N}(\mathbf{x})} d_k(\mathbf{s}) / |\mathcal{N}(\mathbf{x})|}$$



# Nonparametric Regression

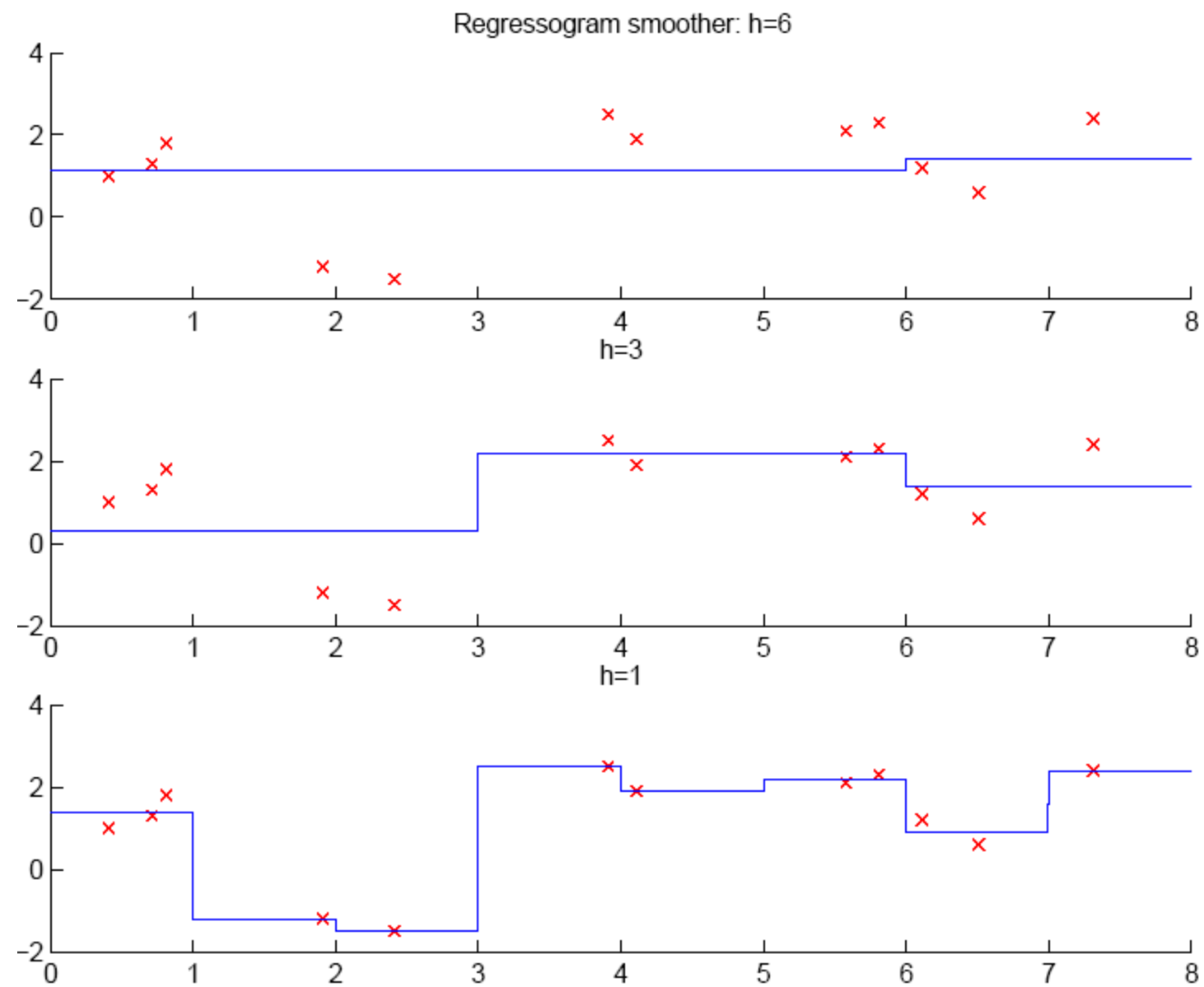
20

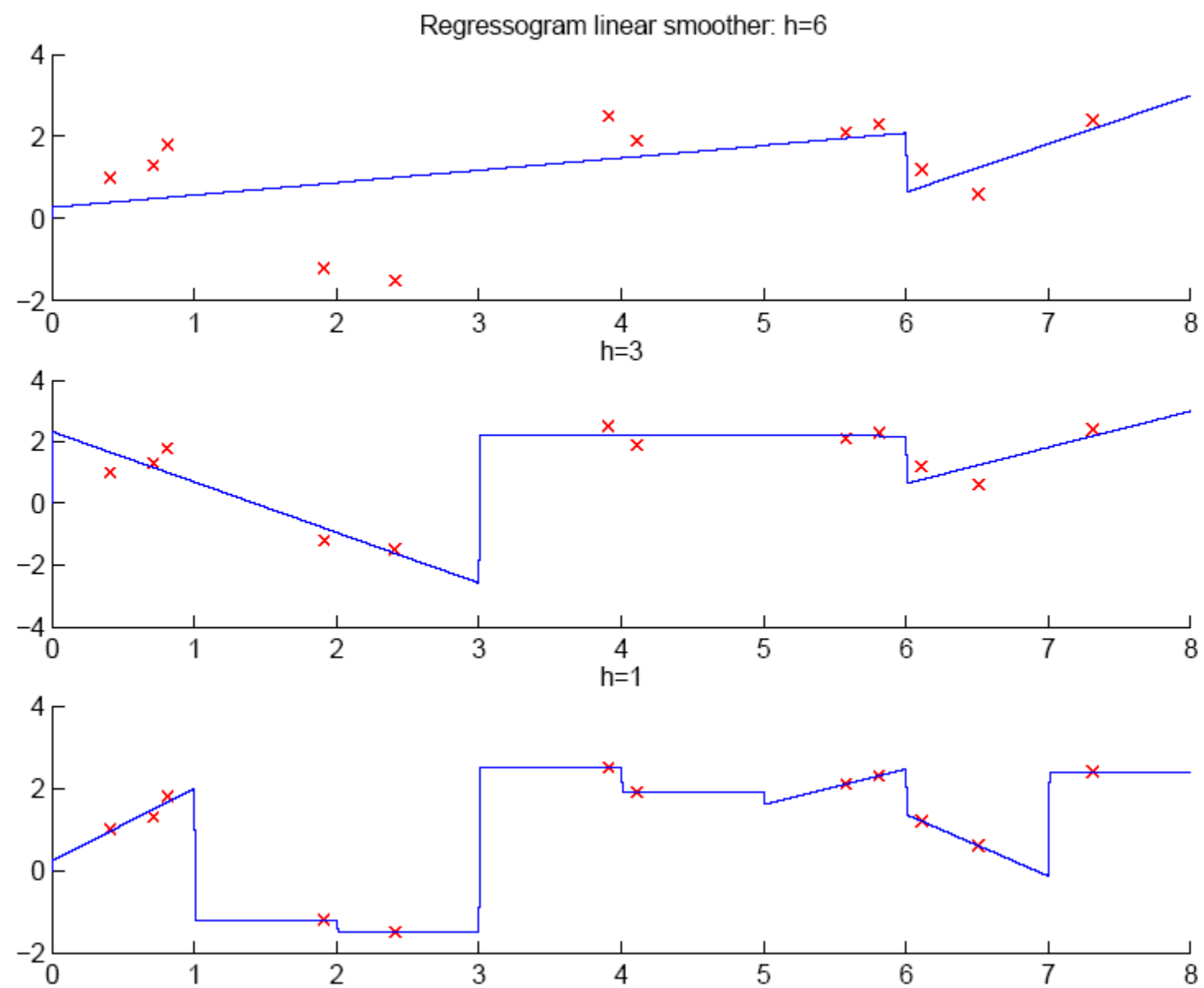
- AKA smoothing models
- Regressogram

$$\hat{g}(x) = \frac{\sum_{t=1}^N b(x, x^t) r^t}{\sum_{t=1}^N b(x, x^t)}$$

where

$$b(x, x^t) = \begin{cases} 1 & \text{if } x^t \text{ is in the same bin with } x \\ 0 & \text{otherwise} \end{cases}$$





# Running Mean/Kernel Smoother

23

- Running mean smoother

$$\hat{g}(x) = \frac{\sum_{t=1}^N w\left(\frac{x - x^t}{h}\right) r^t}{\sum_{t=1}^N w\left(\frac{x - x^t}{h}\right)}$$

where

$$w(u) = \begin{cases} 1 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$

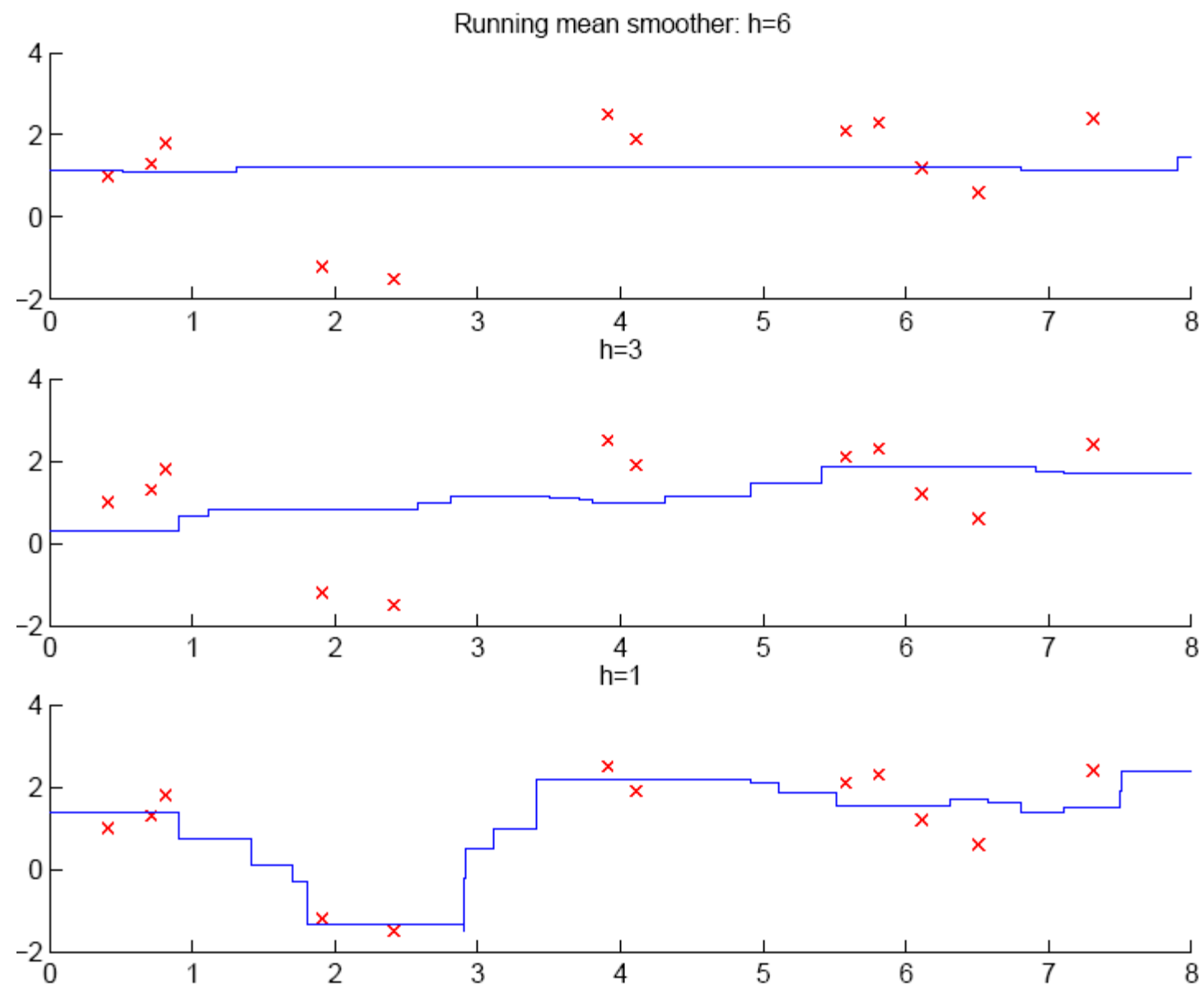
- Running line smoother

- Kernel smoother

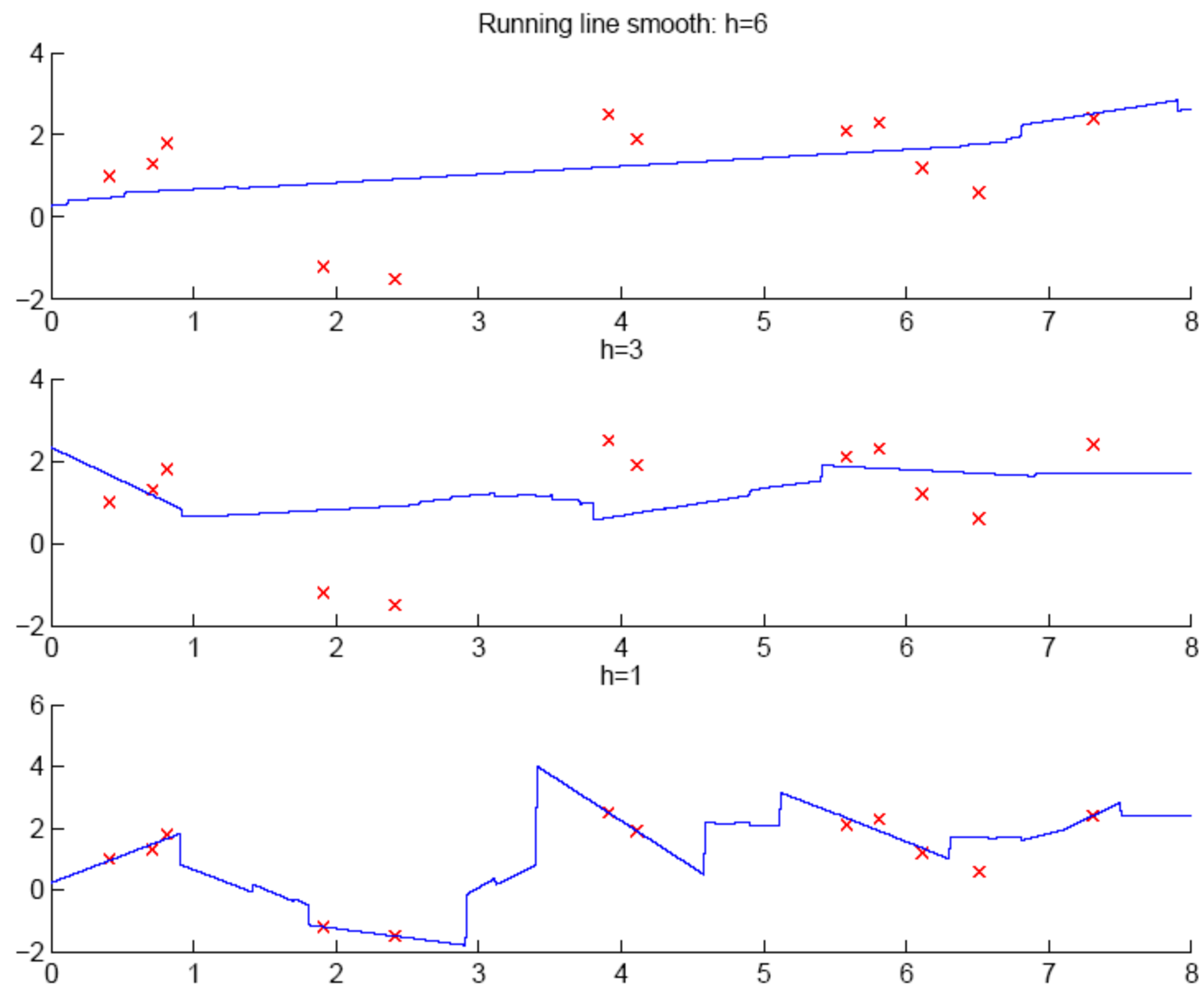
$$\hat{g}(x) = \frac{\sum_{t=1}^N K\left(\frac{x - x^t}{h}\right) r^t}{\sum_{t=1}^N K\left(\frac{x - x^t}{h}\right)}$$

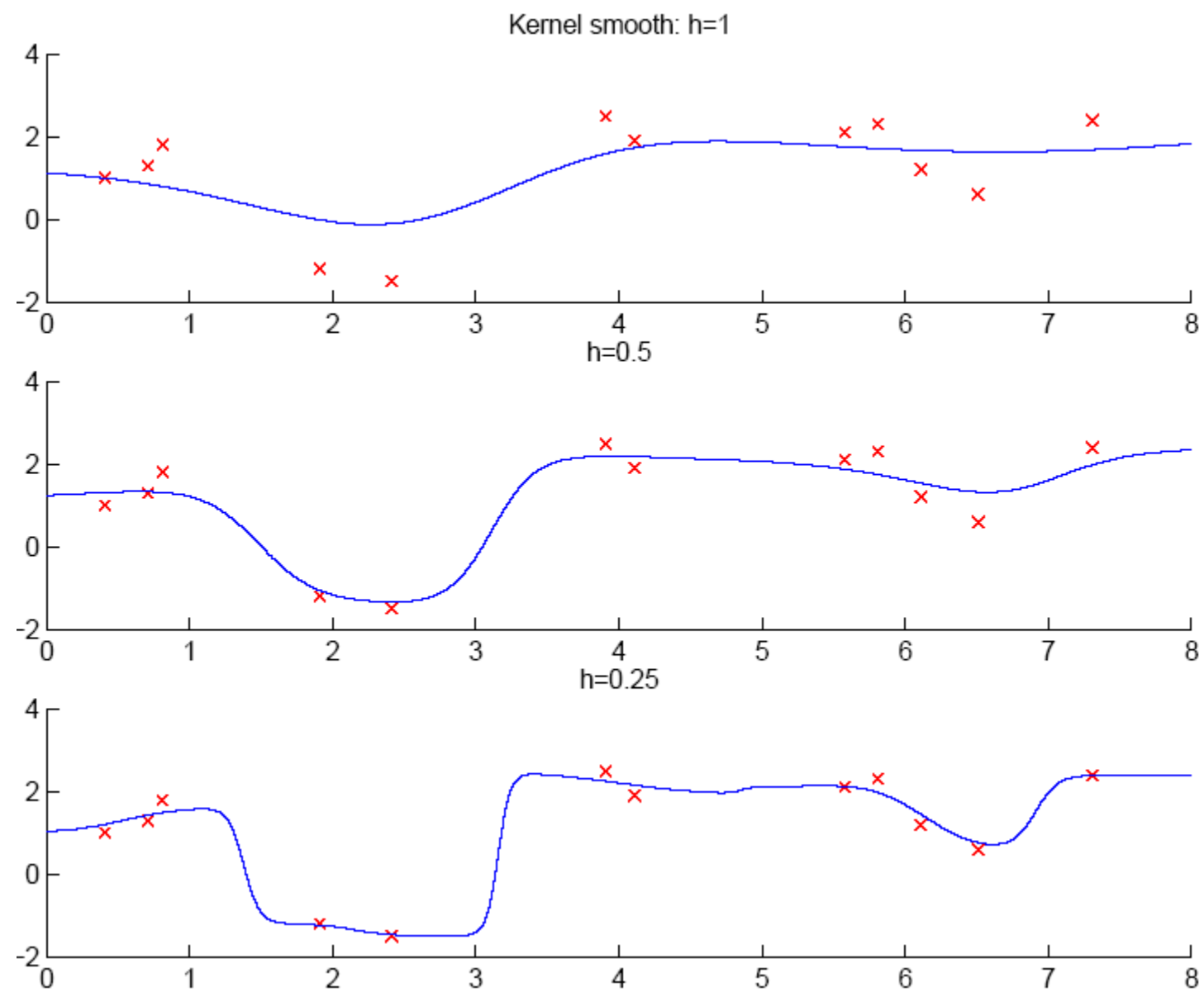
where  $K(\cdot)$  is Gaussian

- Additive models (Hastie and Tibshirani, 1990)





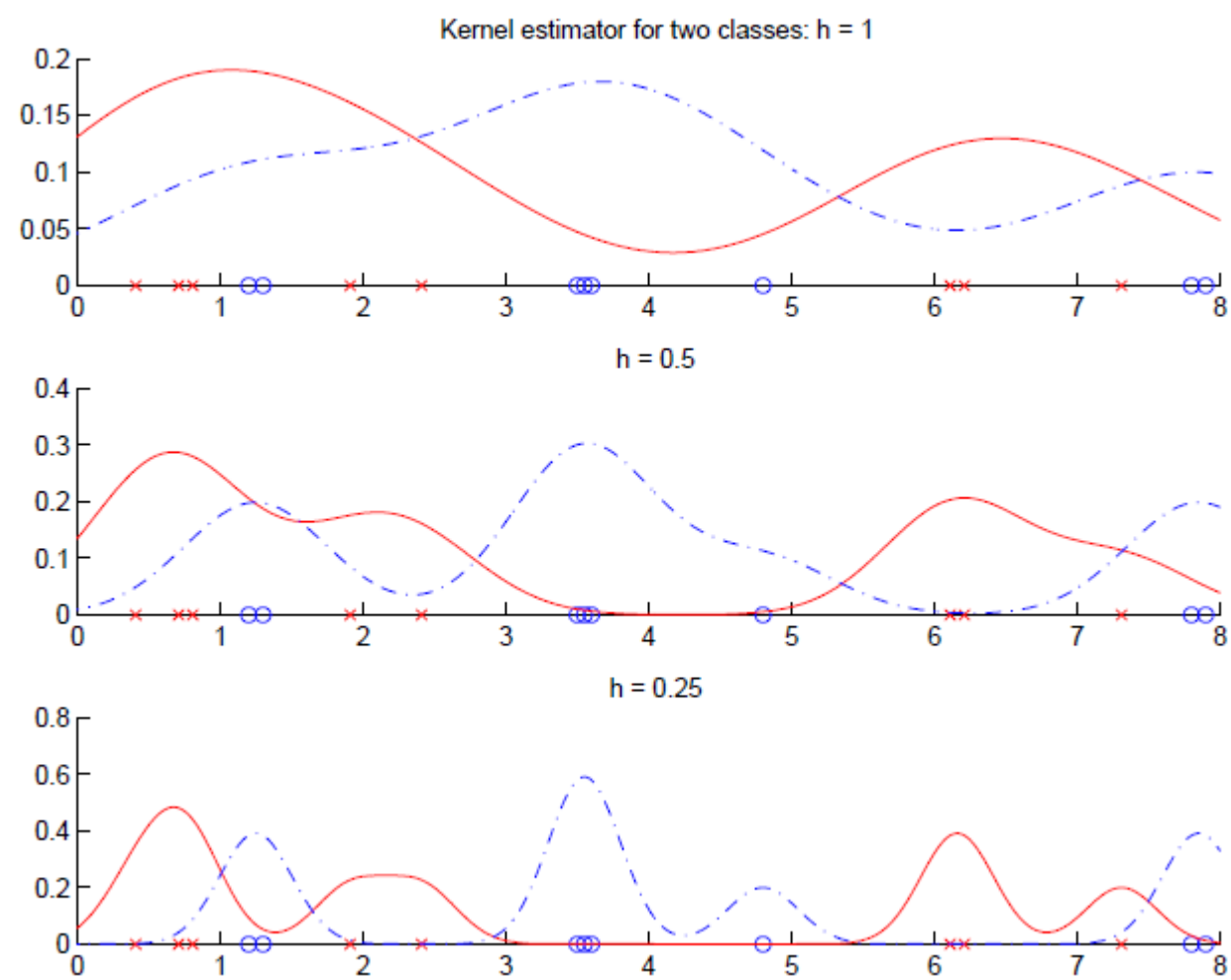




# How to Choose $k$ or $h$ ?

27

- When  $k$  or  $h$  is small, single instances matter; bias is small, variance is large (undersmoothing): High complexity
- As  $k$  or  $h$  increases, we average over more instances and variance decreases but bias increases (oversmoothing): Low complexity
- Cross-validation is used to finetune  $k$  or  $h$ .



READ ALPAYDIN CH. 9

