



## *Lecture Slides for*

INTRODUCTION TO

# *Machine Learning*

ETHEM ALPAYDIN

© The MIT Press, 2004

*alpaydin@boun.edu.tr*

*<http://www.cmpe.boun.edu.tr/~ethem/i2ml>*

CHAPTER 8:

# *Nonparametric Methods*



## *Nonparametric Estimation*

- Parametric (single global model, learning problem becomes finding values for a small number of model parameters), semiparametric (small number of local models)
- Nonparametric: Similar inputs have similar outputs (this is the main general assumption of these techniques)
- 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



- **Lazy?**
  - No ‘training’: computation is deferred until testing.
- **Memory/instance based?**
  - Because they store training instances in a lookup table and interpolate...
- **In terms of computation time/memory requirements:**
  - If a parametric model has  $d$  parameters:
    - Storage is  $O(d)$  or  $O(d^2)$
    - Typically  $N \gg d$ .
  - Nonparametric implies  $O(N)$  memory requirement and  $O(N)$  computation for testing (e.g. find similar instances).



# Density Estimation

- Given the training set  $X = \{x^t\}_t$  drawn iid from  $p(x)$
- Divide data into bins of size  $h$
- Histogram:
$$\hat{p}(x) = \frac{\#\{x^t \text{ in the same bin as } x\}}{Nh}$$
  - Needs an origin  $x_0$  and bin width  $h$ . Bins are then intervals  $[x_0 + mh, x_0 + (m+1)h]$  ( $m$  positive and negative integers).
- Naive estimator: 
$$\hat{p}(x) = \frac{\#\{x - h < x^t \leq x + h\}}{2Nh}$$

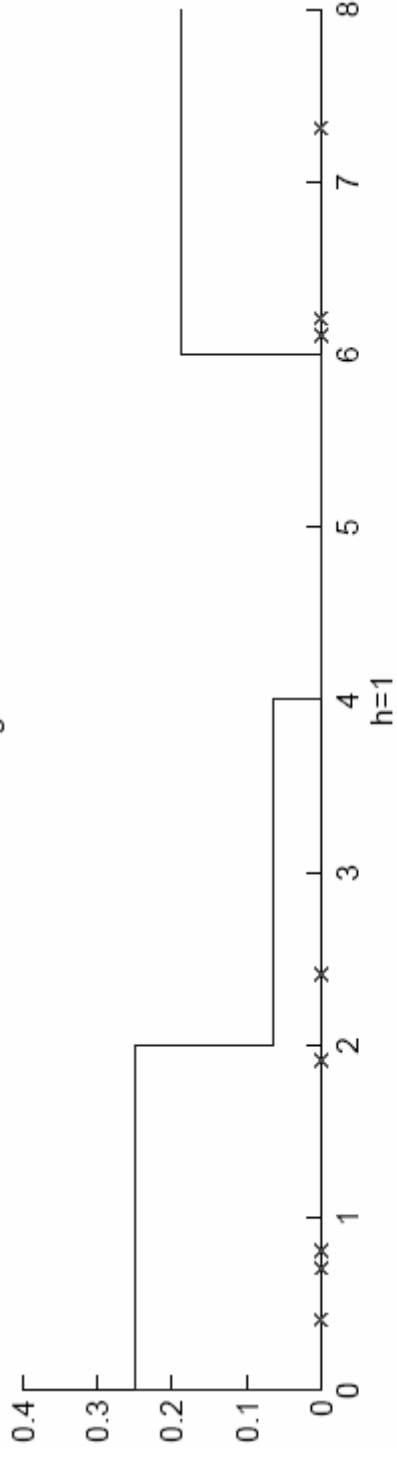
$$\text{or } \hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^N w \left( \frac{x - x^t}{h} \right) \quad w(u) = \begin{cases} 1/2 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$



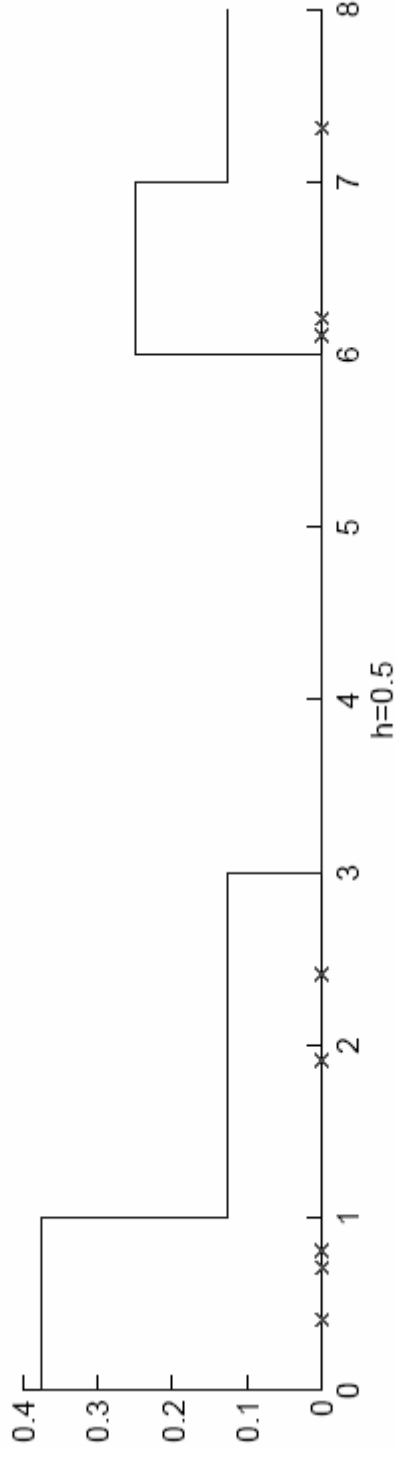
- For naïve estimator:
  - No need to set origin
  - $w(u)$  is a weight function
  - $\hat{p}(x)$  is equal to histogram where  $x$  is always at the center of a bin of size  $2h$  (see figs...).



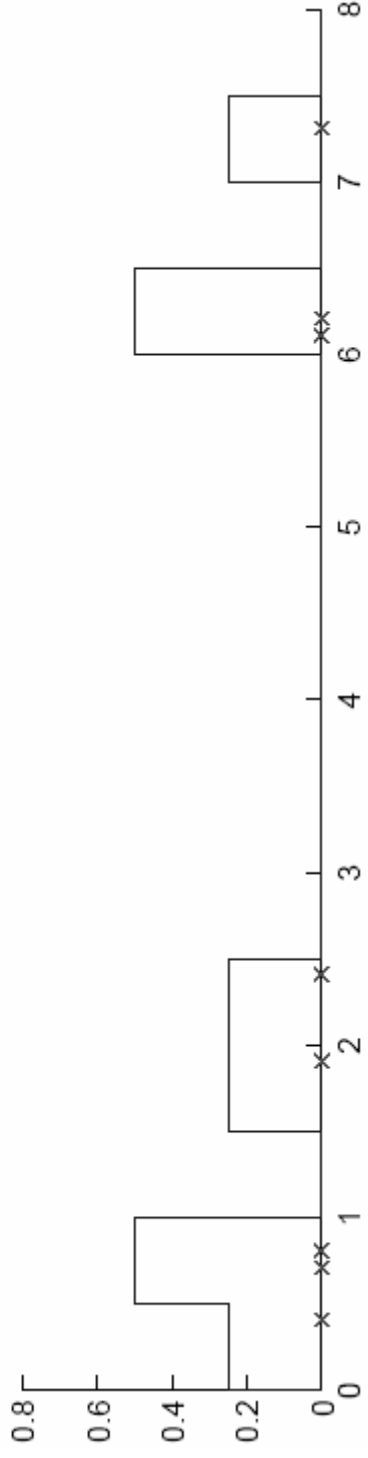
Histogram:  $h=2$



$h=1$

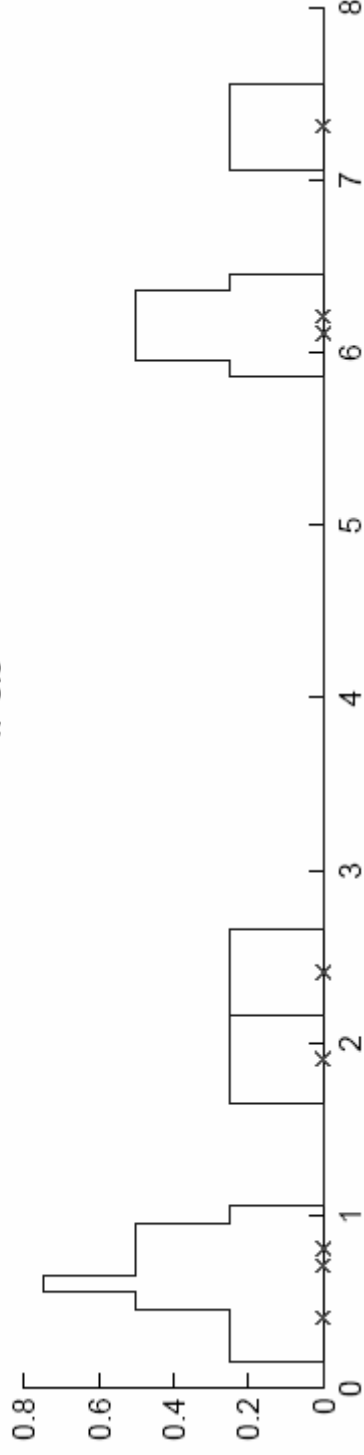
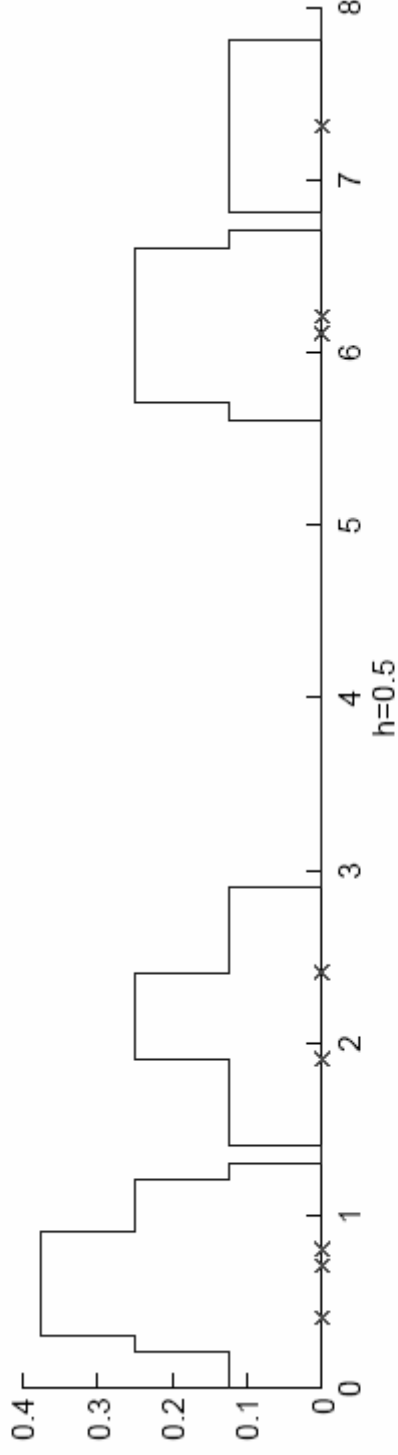
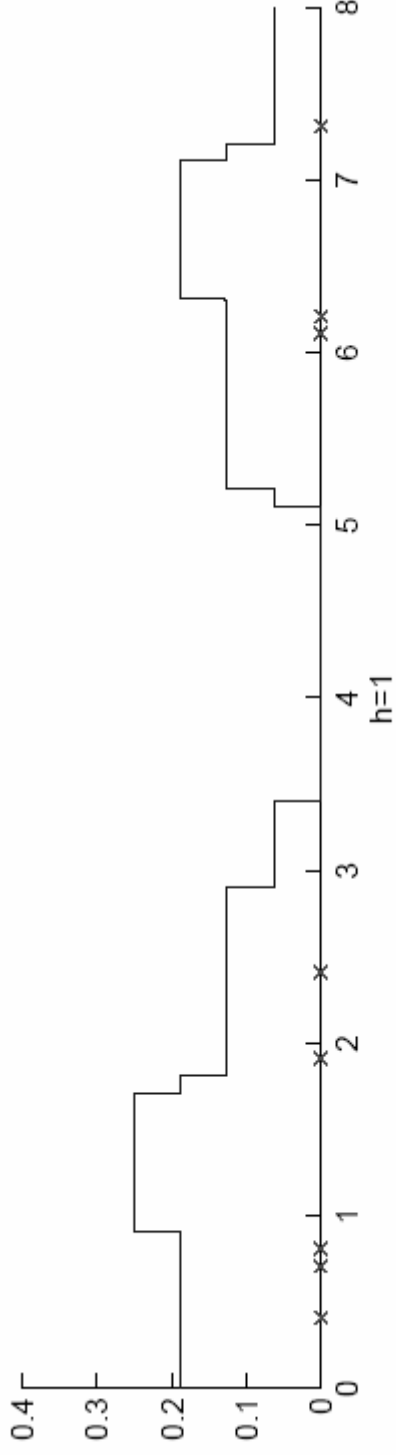


$h=0.5$





Naive estimator:  $h=2$





# Kernel Estimator

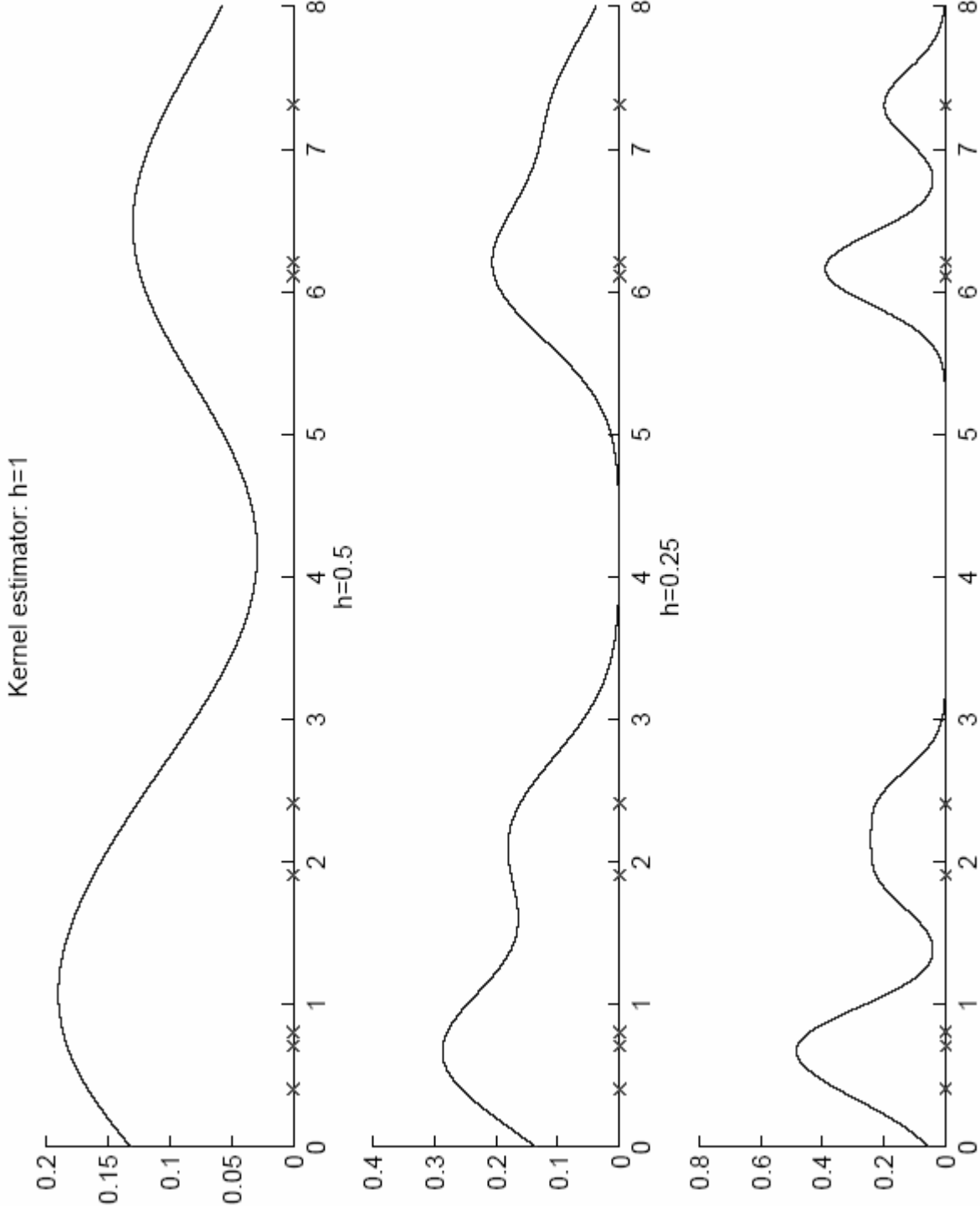
- Kernel function, e.g., Gaussian kernel:

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

- Idea: use a smooth weight function (cf naïve estimator)

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

- $h$  is window width (parameter to be chosen)
- Estimator is a sum of bumps





## *k-Nearest Neighbor Estimator*

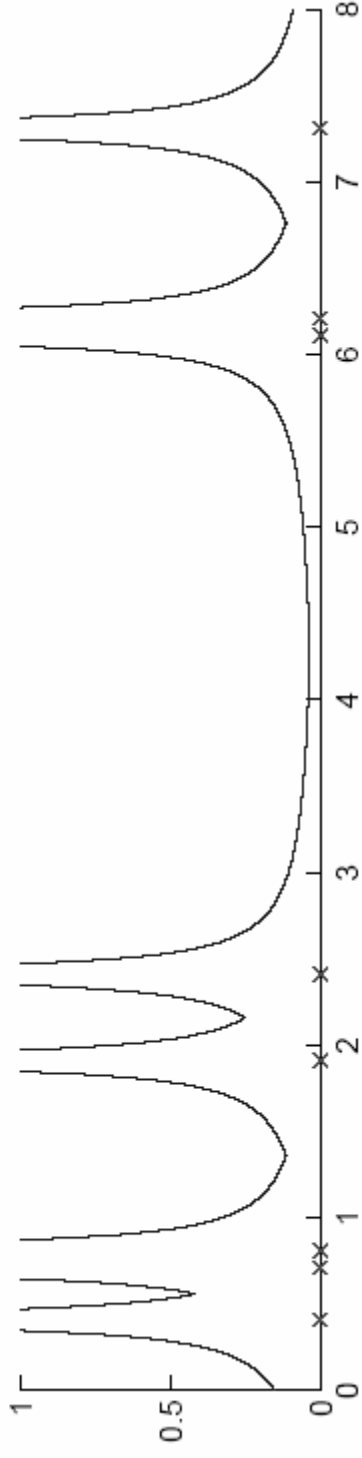
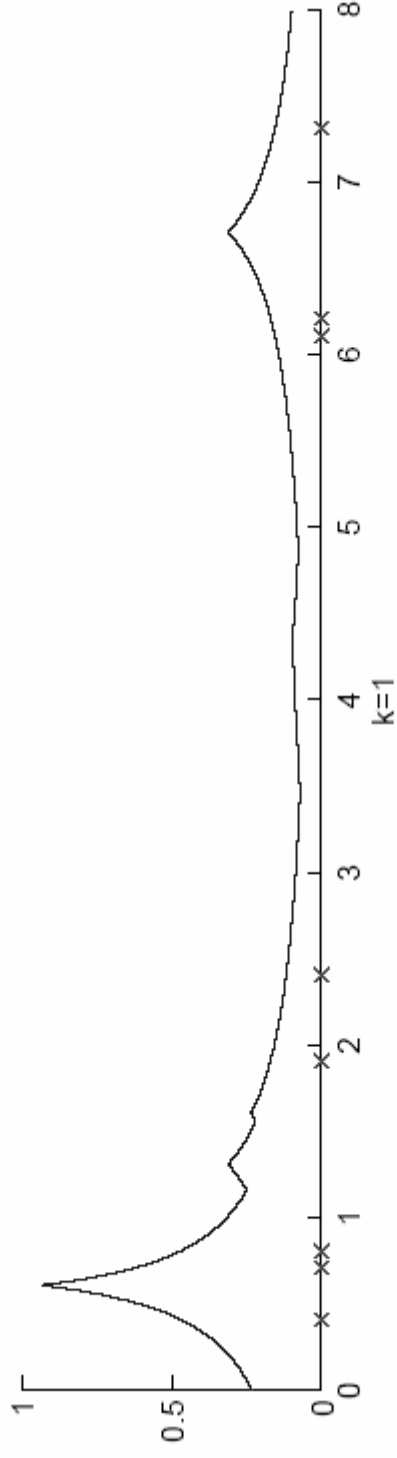
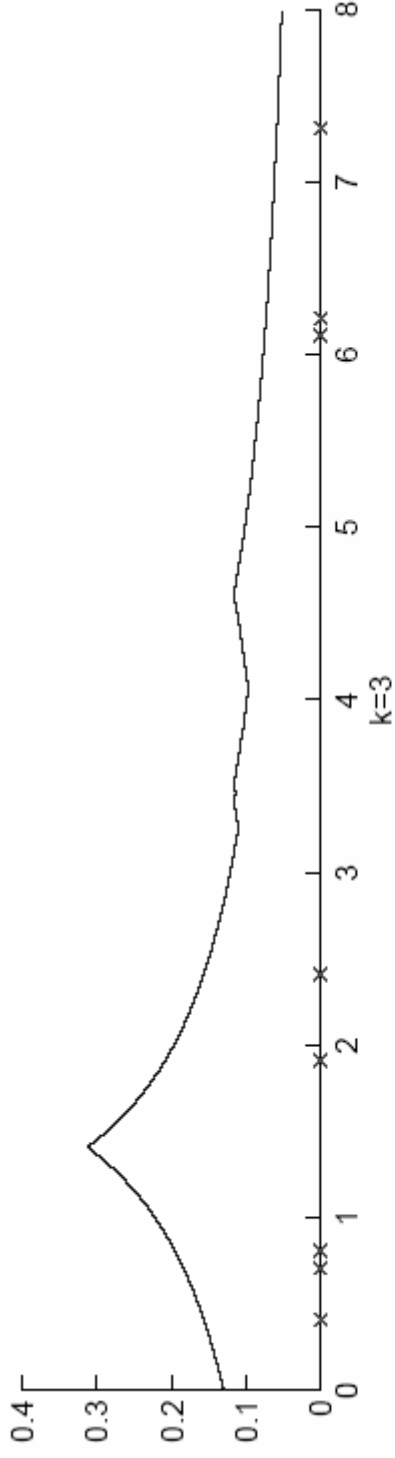
- Adapts amount of smoothing to the local density of data.
- Instead of fixing bin width  $h$  and counting the number of instances, fix the instances (neighbors)  $k$  and check bin width

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

- $d_k(x)$ , distance to  $k$ th closest instance to  $x$
- Where density is high, bins are small and vice versa.



k-NN estimator: k=5





# Multivariate Data

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



- Have to be careful in high dimensions:
  - E.g.  $x$  is 8D, 10 bins per dimension
  - Means  $10^8$  bins, most of which are empty!
  - (Curse of dimensionality)



# Nonparametric Classification

- 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

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



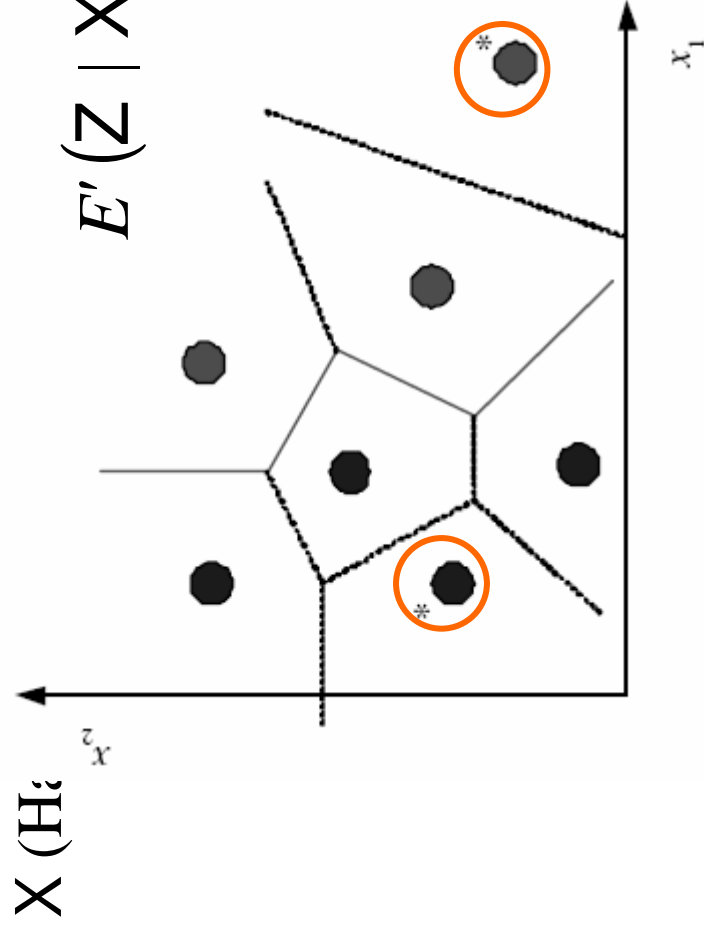
## *K-nearest neighbor classification*

- $k_i$  is number of neighbors out of the  $k$  nearest that belong to class  $C_i$ .
- $V^k(x)$  is volume of  $d$ -dimensional hypersphere centered at  $x$ .
- For  $k$ -nn:
- Ties are broken arbitrarily or weighted vote
- $k = 1$  is nearest neighbor
  - Corresponds to Voronoi tessellation of input space
    - Matlab: `voronoi()`



# Condensed Nearest Neighbor

- Time/space complexity of  $k$ -NN is  $O(N)$ 
  - Condensed nn tries to improve this.
- Find a subset  $Z$  of  $X$  that is small and is accurate in classifying





# Condensed Nearest Neighbor

- Basis is 1-nn:
  - Only need to keep instances that define the discriminant (others don't cause any change as their nn is of the same class).
- Incremental algorithm: Add instance if needed

$Z \leftarrow \emptyset$   
Repeat  
For all  $\mathbf{x} \in \mathcal{X}$  (in random order)  
Find  $\mathbf{x}' \in Z$  s.t.  $\|\mathbf{x} - \mathbf{x}'\| = \min_{\mathbf{x}^j \in Z} \|\mathbf{x} - \mathbf{x}^j\|$   
If  $\text{class}(\mathbf{x}) \neq \text{class}(\mathbf{x}')$  add  $\mathbf{x}$  to  $Z$   
Until  $Z$  does not change

Check if the instances in  $X$  can be classified correctly using the instances in  $Z$

No? -> add  $x$  to  $Z$

Yes? -> no change in  $Z$ .



# Nonparametric Regression

- Aka smoothing models
- Regressogram

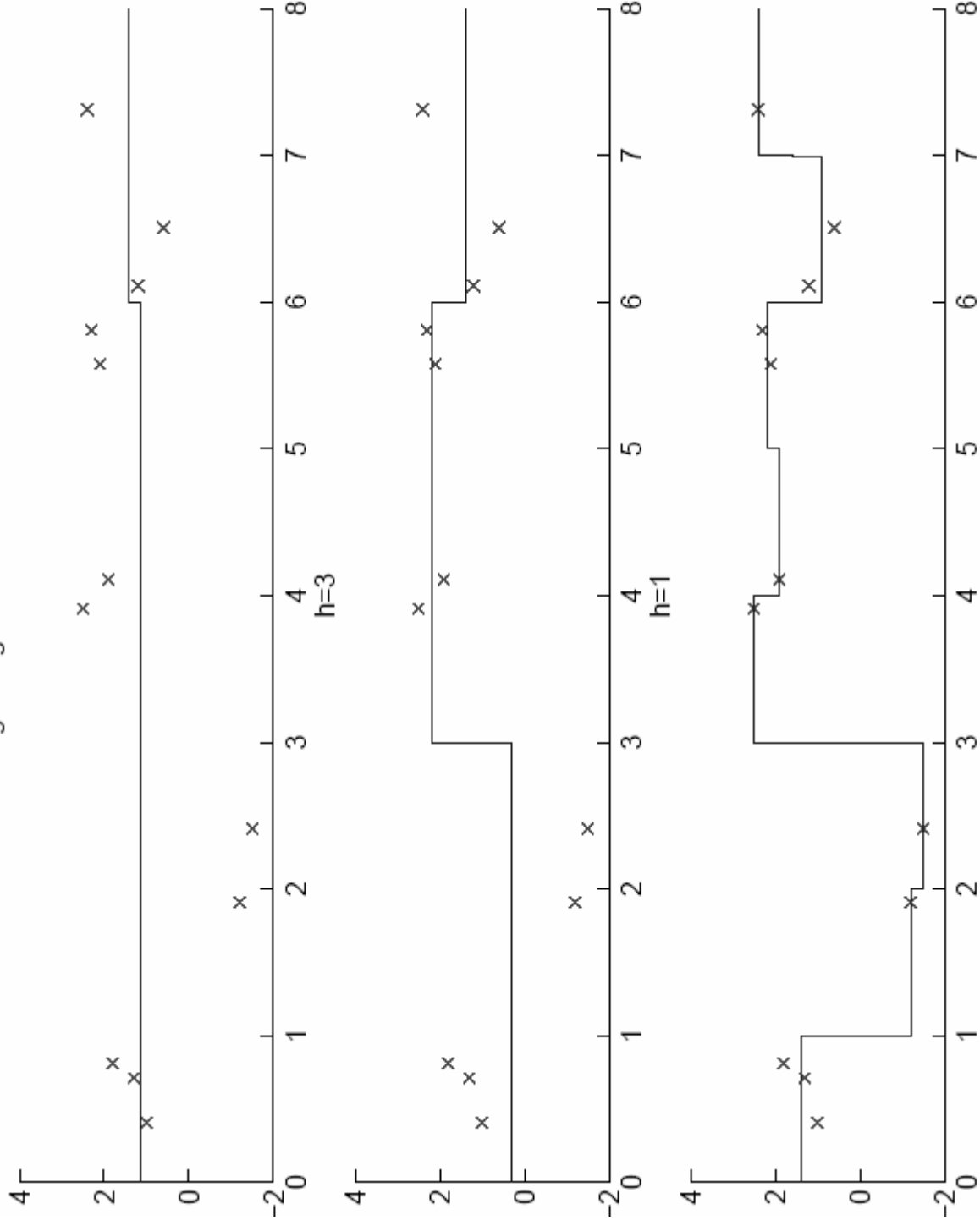
$$\hat{g}(x) = \frac{\sum_{t=1}^N b(x, x^t) r^t}{\sum_{t=1}^N b(x, x^t)}$$

where

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

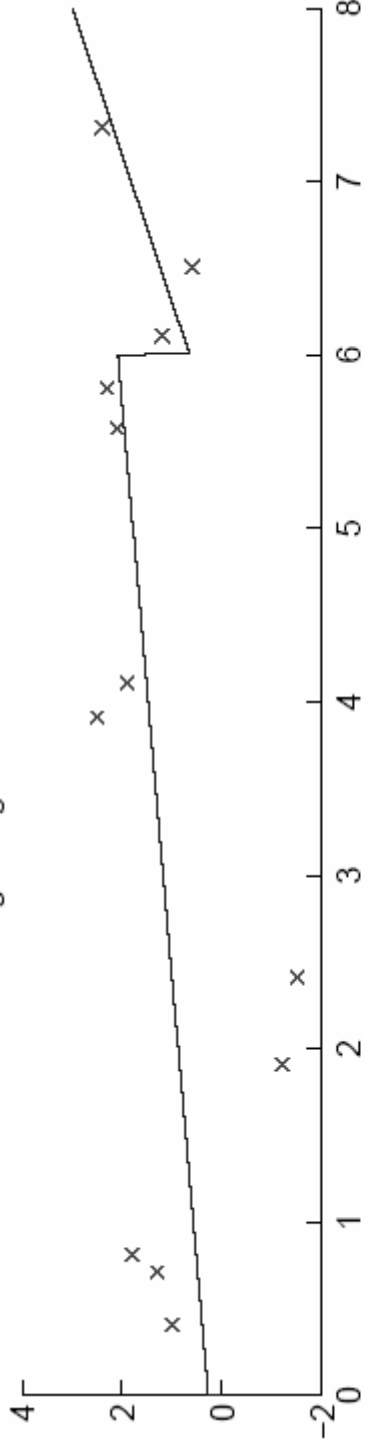


Regressogram smoother:  $h=6$

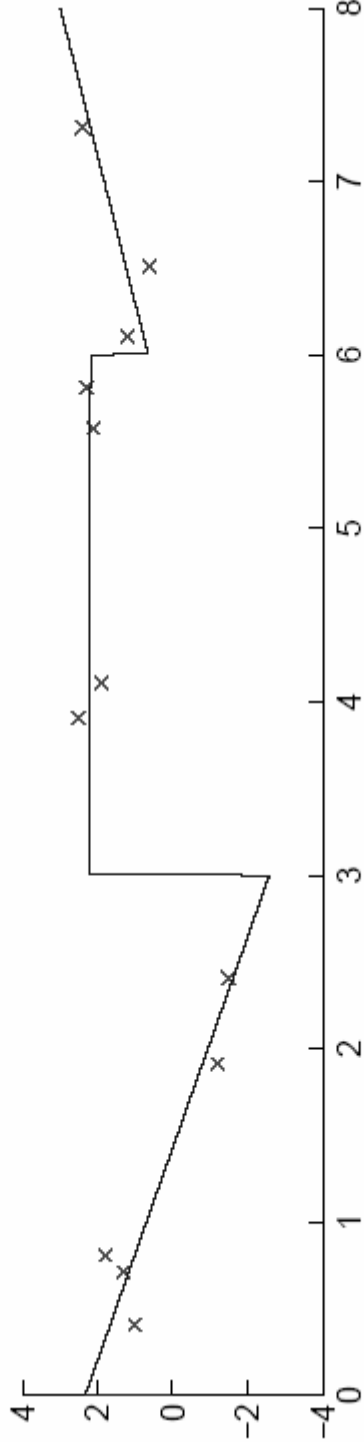




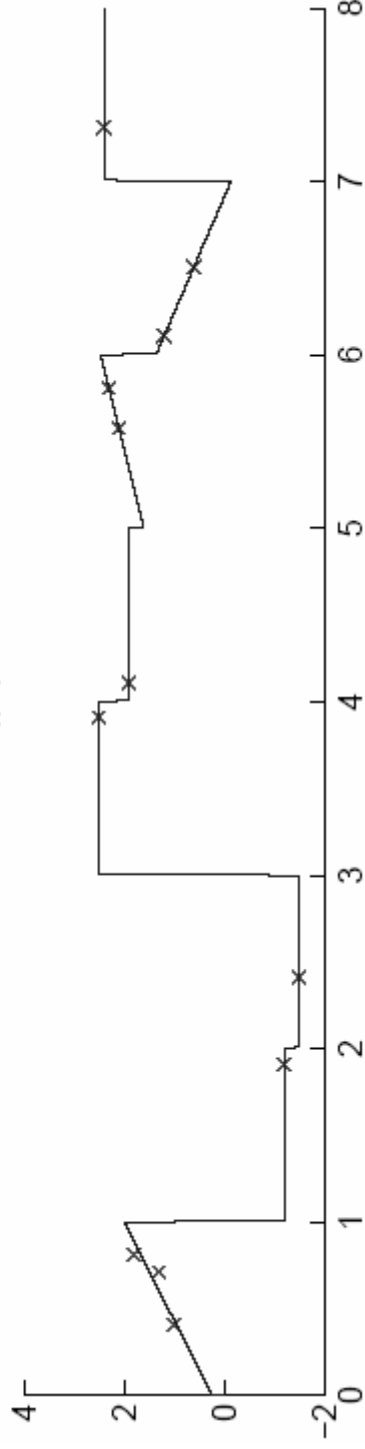
Regressogram linear smoother:  $h=6$



$h=3$



$h=1$





# Running Mean/Kernel Smoother

- Running mean smoother

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

where

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

- Running line smoother

- Kernel smoother

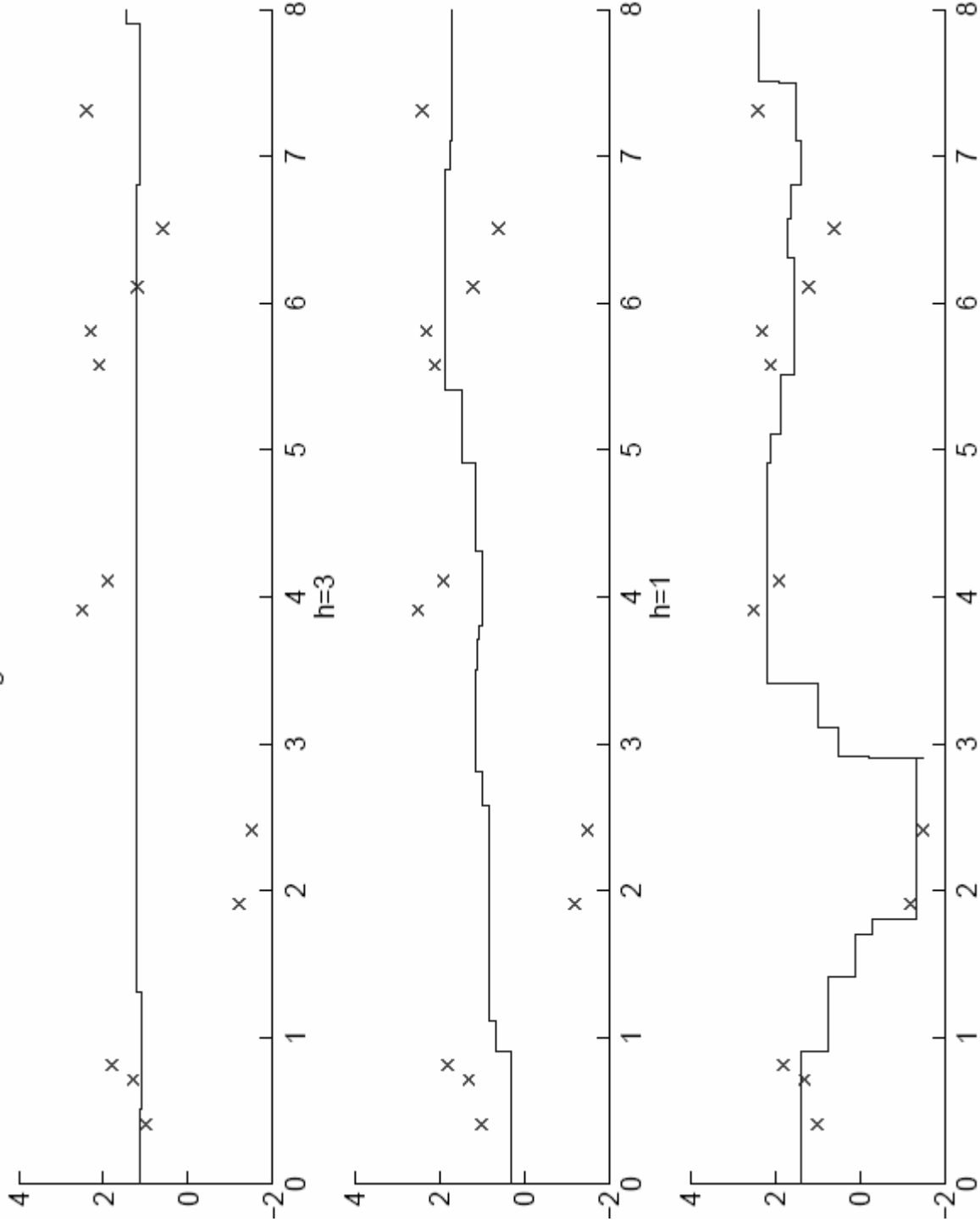
$$\hat{g}(x) = \frac{\sum_{t=1}^N K \left( \frac{x - x^t}{h} \right) r^t}{\sum_{t=1}^N K \left( \frac{x - x^t}{h} \right)}$$

where  $K(\cdot)$  is Gaussian

- Additive models (Hastie and Tibshirani, 1990)

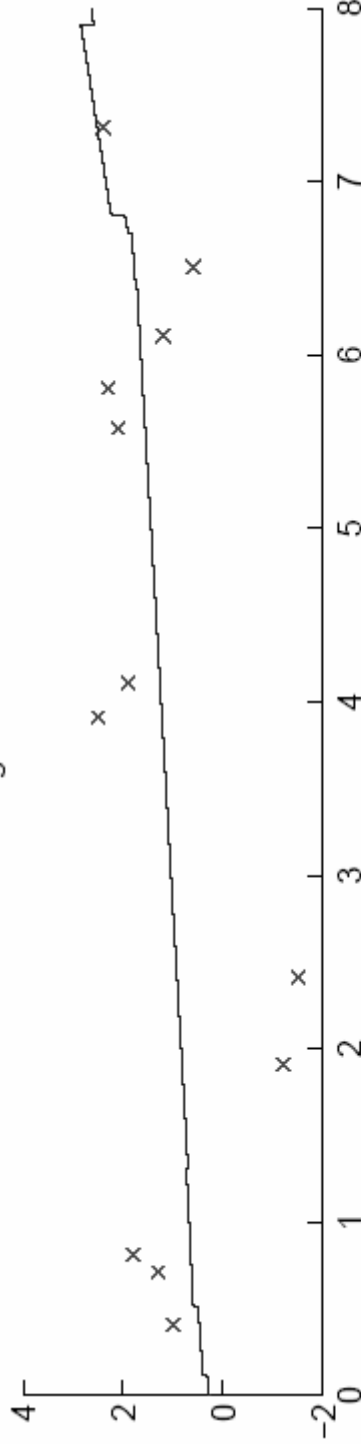


Running mean smoother:  $h=6$





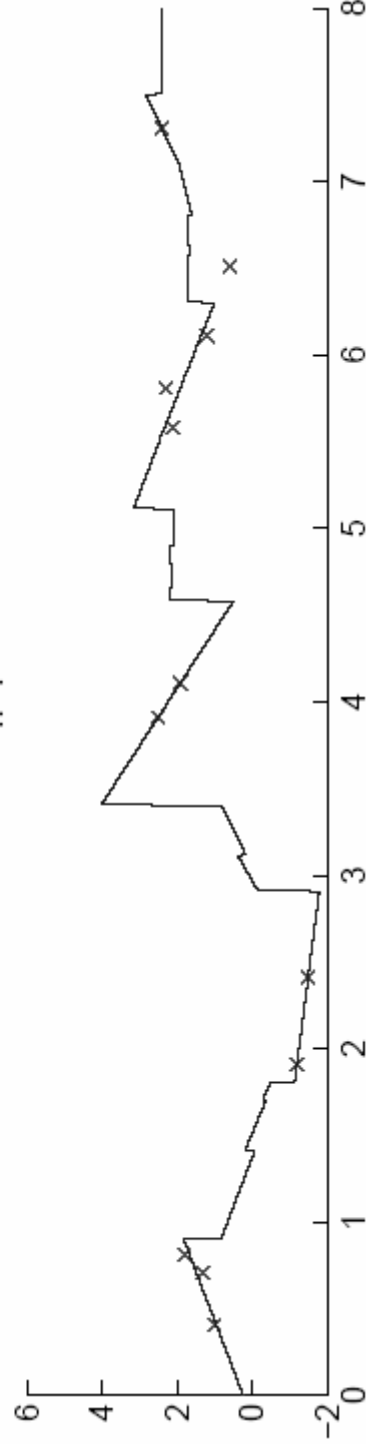
Running line smooth:  $h=6$



$h=3$

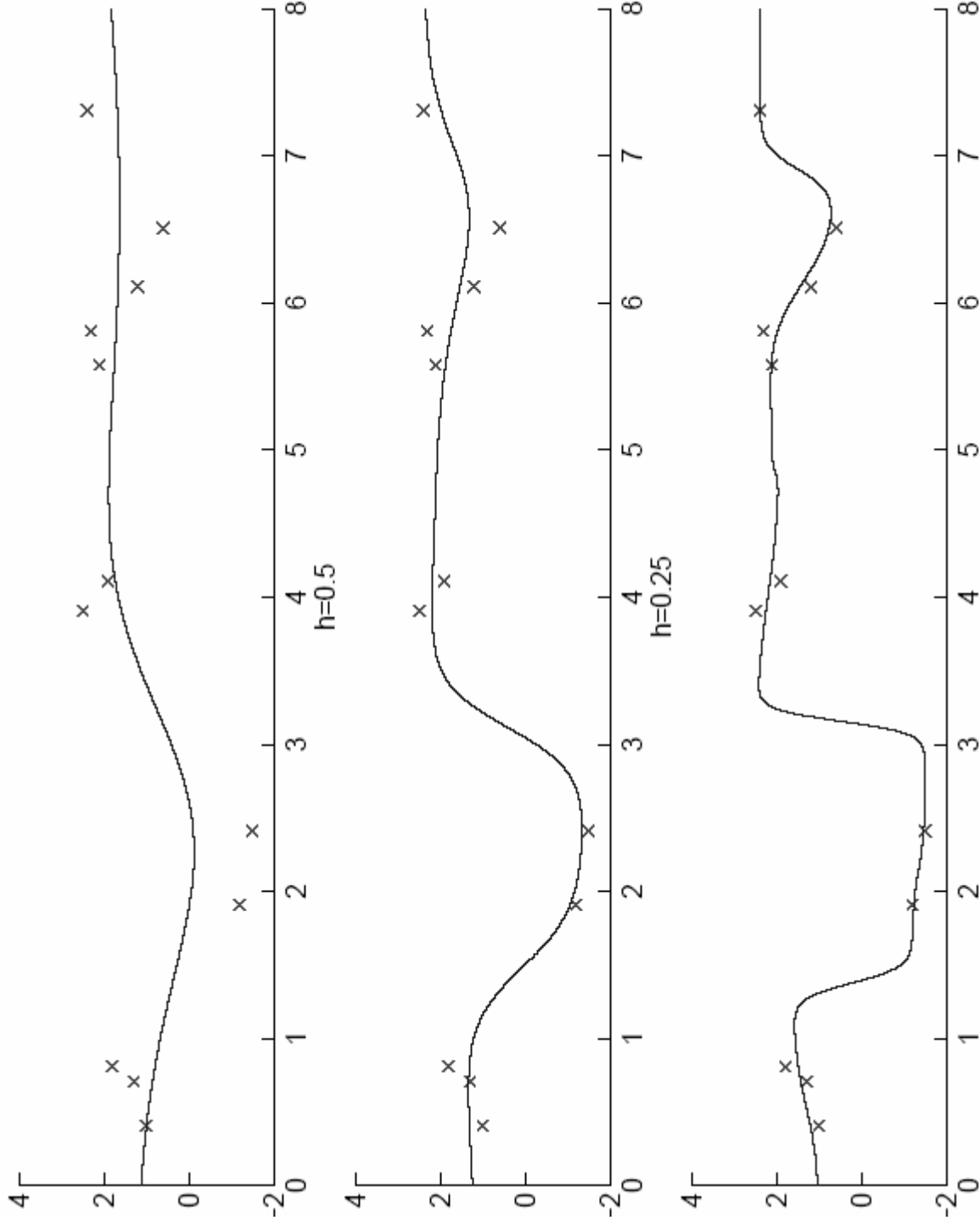


$h=1$





Kernel smooth:  $h=1$





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

- 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  $h$ .