CSCI 5521: Introduction to Machine Learning (Spring 2020)

Nonparametric Methods

Rui Kuang

Department of Computer Science and Engineering

University of Minnesota

University of Minnesota
Twin Cities • Duluth • Morris • Crookston • Rochester • Other Locations

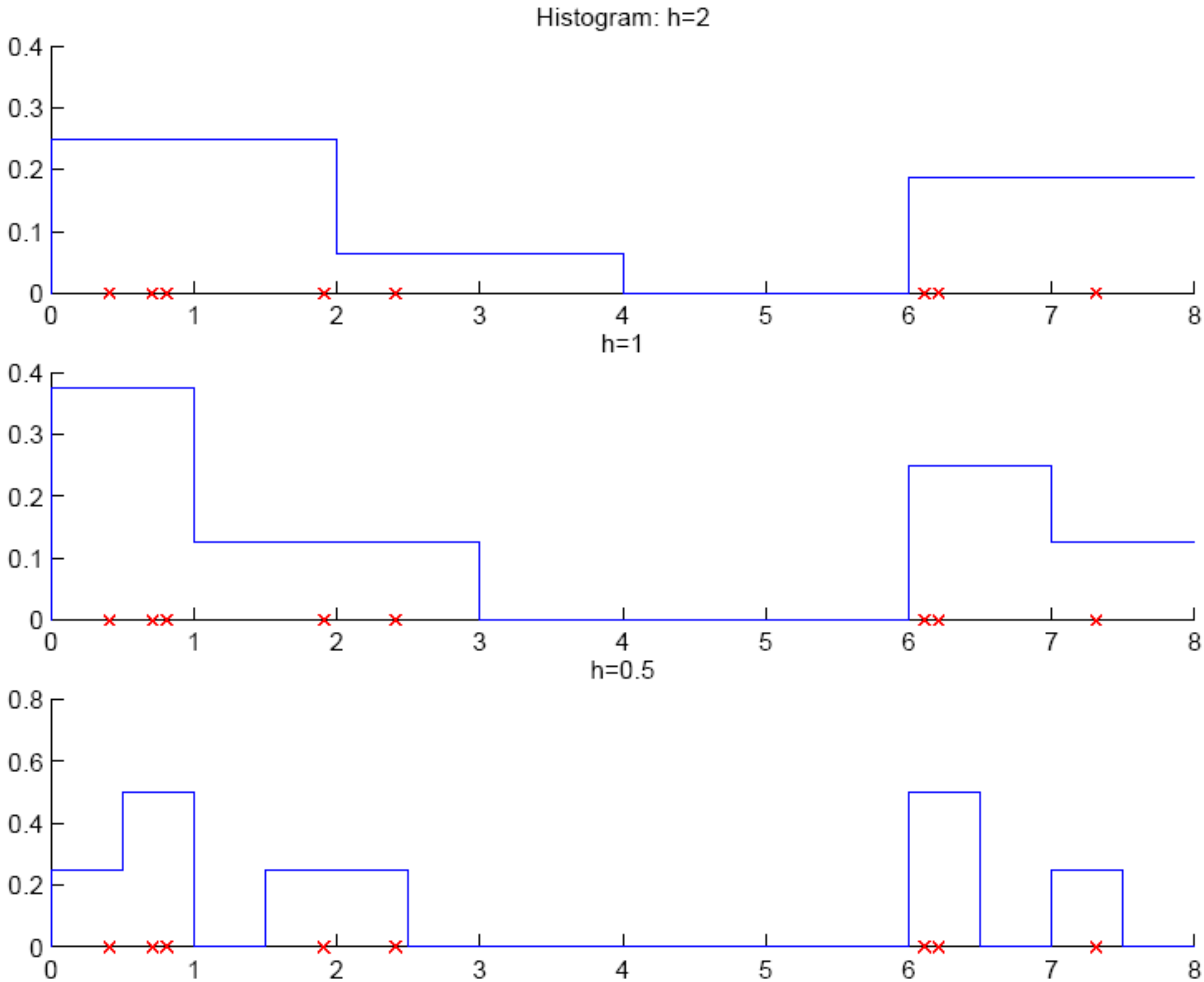
Nonparametric Estimation

- Parametric (single global model), semiparametric (small number of local models)
- Nonparametric:
  - Keep the training data;“let the data speak for itself”
  - Similar inputs have similar outputs
  - Functions (pdf, discriminant, regression) change smoothly
  - 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

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

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



Histogram: h=2
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=1
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
h=0.5
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8

Density Estimation

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

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

- Naive estimator:

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

or

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

Density Estimation

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

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

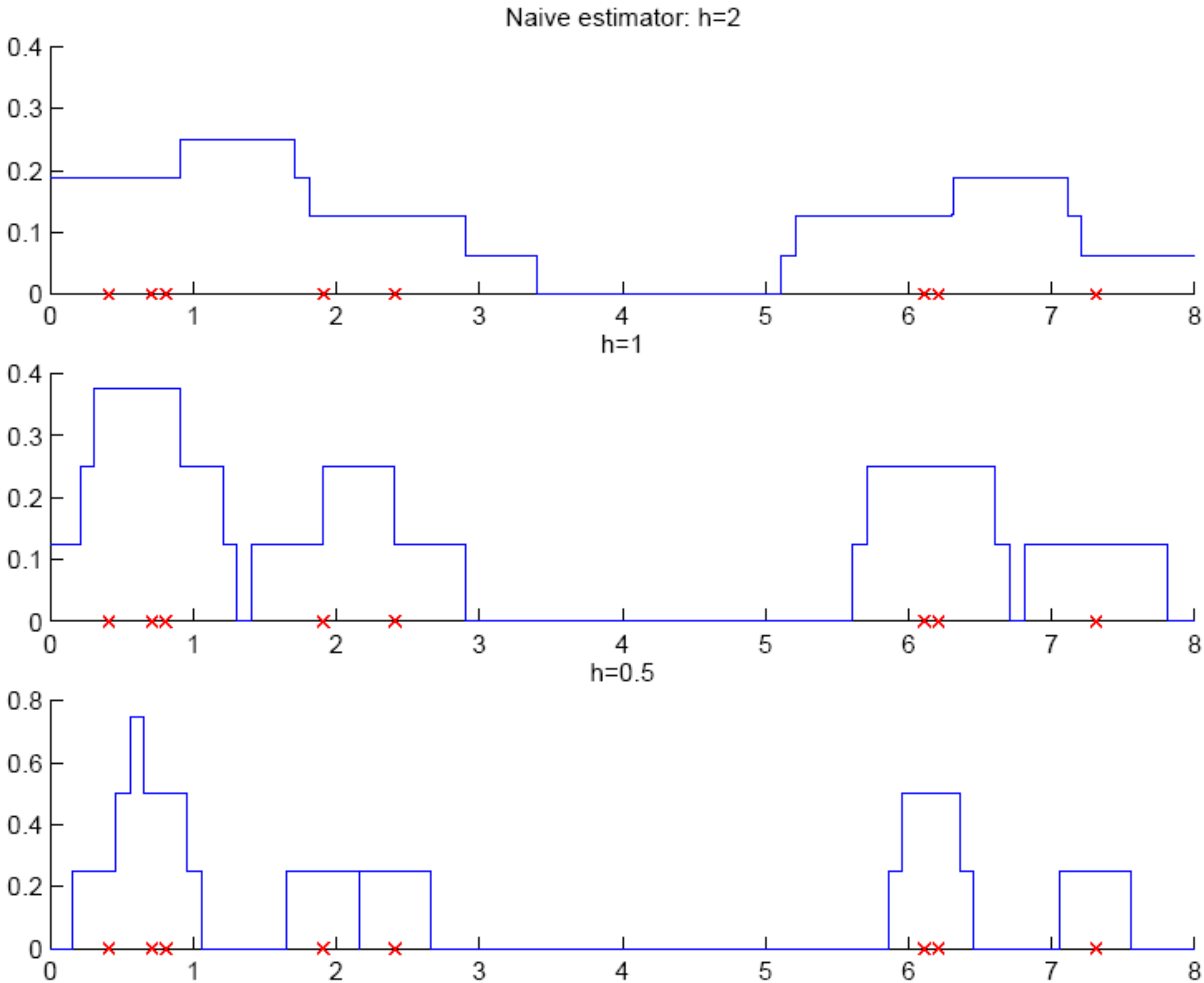
- Naive estimator:

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

or

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

of x^t in the same h-span as x



Naive estimator: h=2
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
h=1
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
h=0.5
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
8

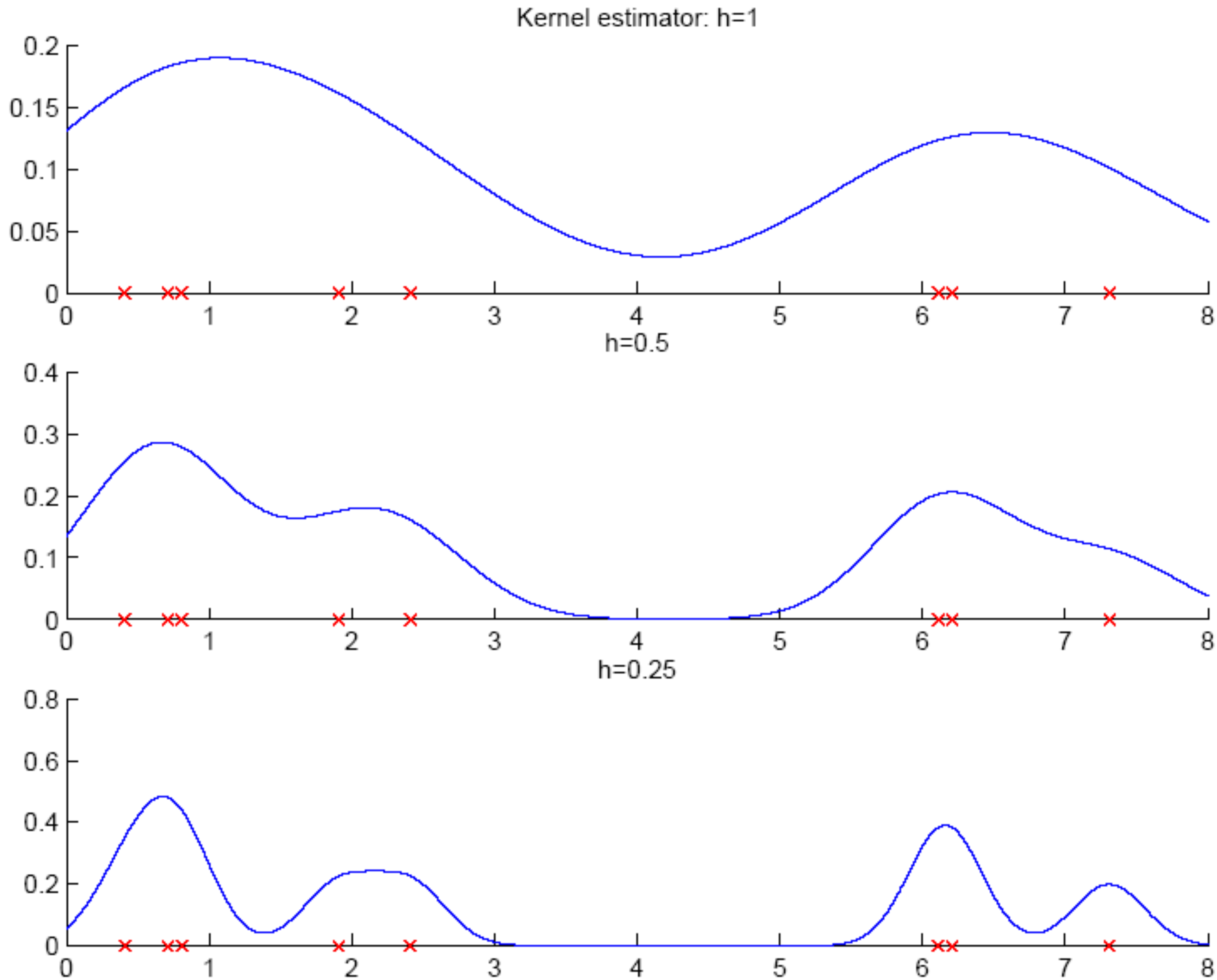
Kernel Estimator

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

$$K(u) = N(0,1) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{u^2}{2}\right]$$

- Kernel estimator (Parzen windows)

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


Kernel estimator: h=1
h=0.5
h=0.25
0.2
0.15
0.1
0.05
0
0.4
0.3
0.8
0.6
0 1 2 3 4 5 6 7 8

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

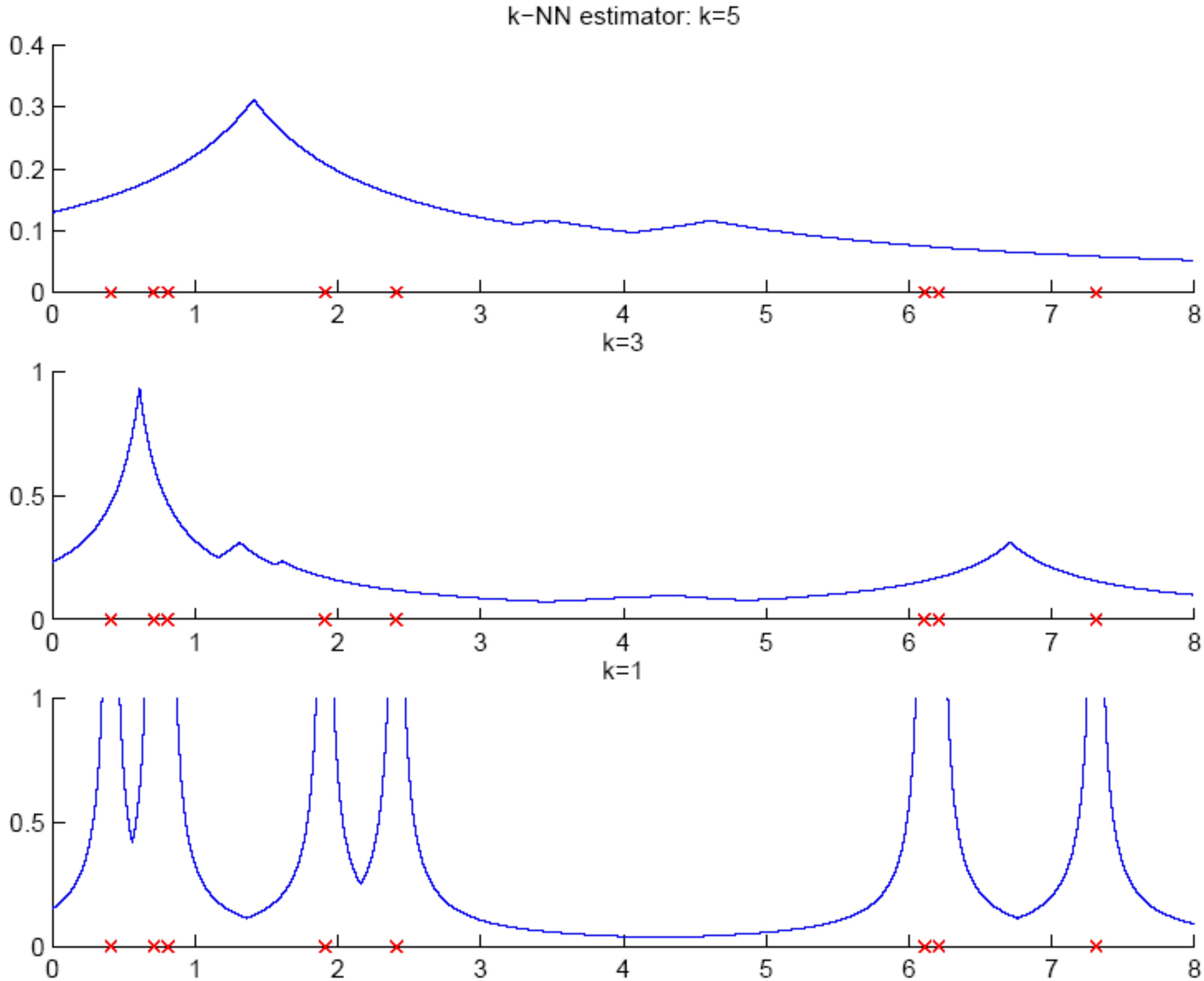
k-Nearest Neighbor Estimator

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

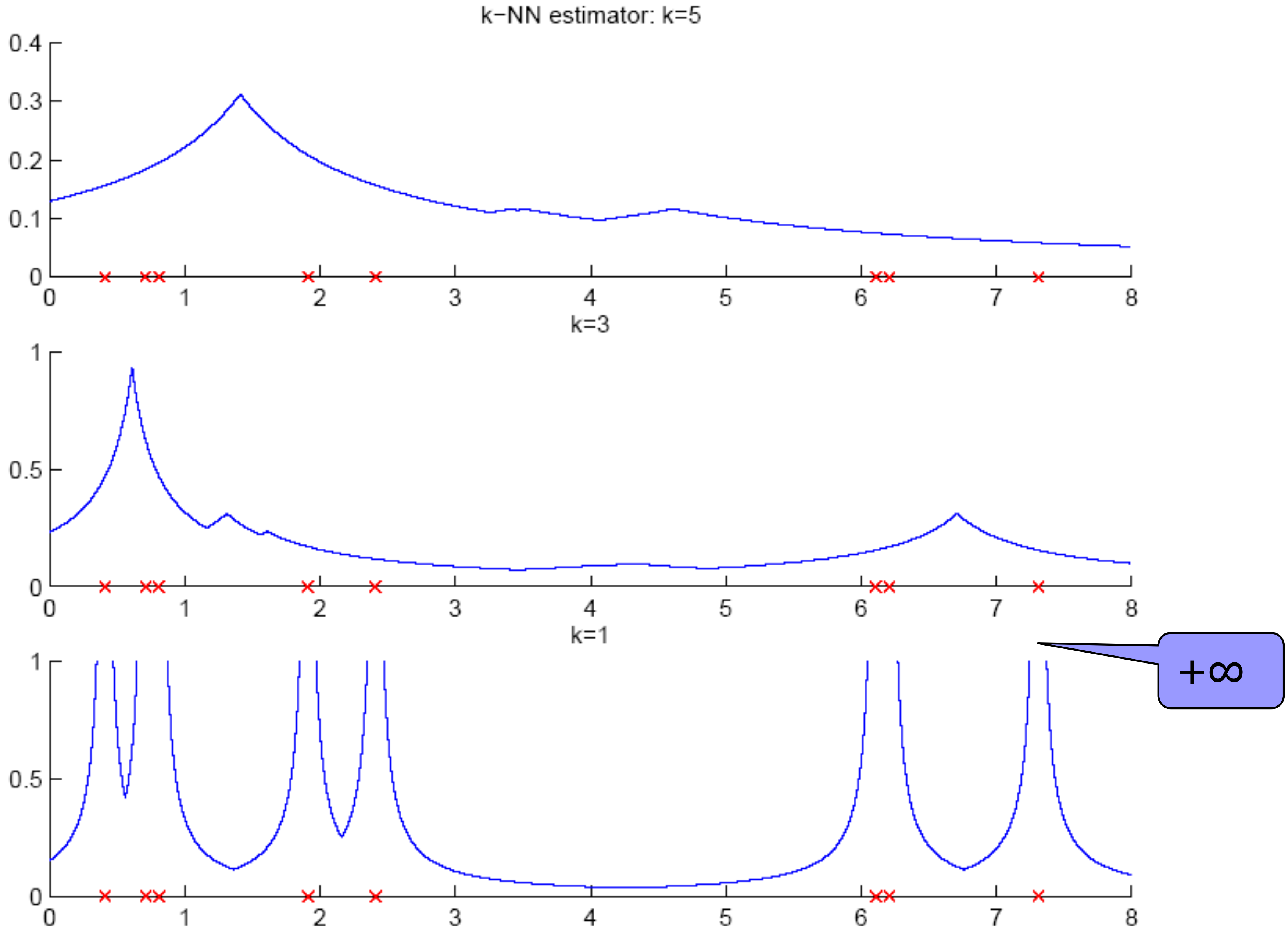
$$\hat{p}(x) = \frac{k}{2Nd_k(x)}$$

$d_k(x)$, distance to kth closest instance to x

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



k−NN estimator: k=5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
k=3
1
0.5
0
0 1 2 3 4 5 6 7 8
k=1
1
0.5
0
0 1 2 3 4 5 6 7 8


k−NN estimator: k=5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8
k=3
1
0.5
0
0 1 2 3 4 5 6 7 8
k=1
1
0.5
0
0 1 2 3 4 5 6 7 8
+∞

Density Estimation

- Parametric:

$$p\left(\mathbf{x} \mid C_i\right) = p(x \mid \Phi_i) = N(\mu_i, \Sigma_i)$$

- Nonparametric (Parzen Windows):

$$p\left(\mathbf{x} \mid C_i\right) = p(x \mid \mathcal{X}_i) = \frac{1}{N_i h} \sum_{t=1}^{N_i} K\left(\frac{x - x_{(i)}^t}{h}\right)$$

Nonparametric Classification

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

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

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

- *k*-NN estimator

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

Nonparametric Classification

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

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

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

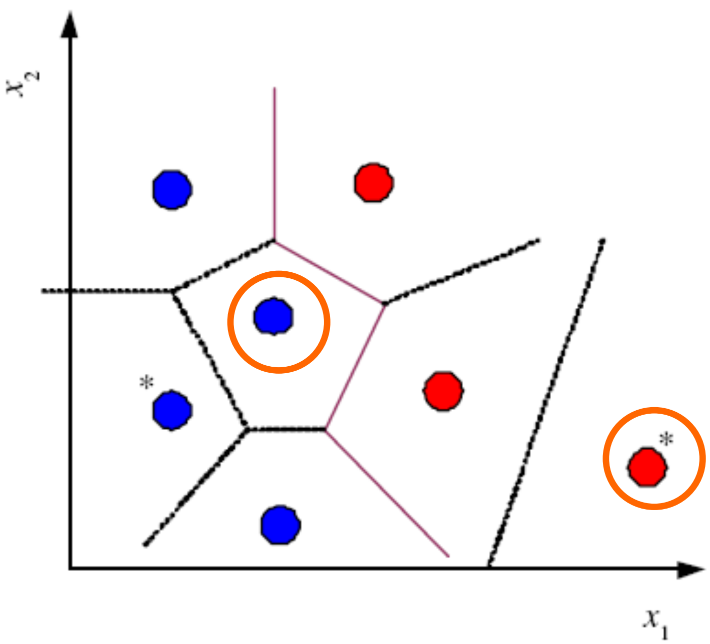
- k-NN estimator

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

KNN classifier

Condensed Nearest Neighbor

- Time/space complexity of *k*-NN is O (*N*) to find top-k neighbors.
- Find a subset Z of X that is small and is accurate in classifying X (Hart, 1968)



$$E'(\mathcal{Z} \mid \mathcal{X}) = E(\mathcal{X} \mid \mathcal{Z}) + \lambda |\mathcal{Z}|$$

E. Alpaydin, Introduction to Machine Learning

Condensed Nearest Neighbor

- Incremental algorithm: Add instance if needed

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

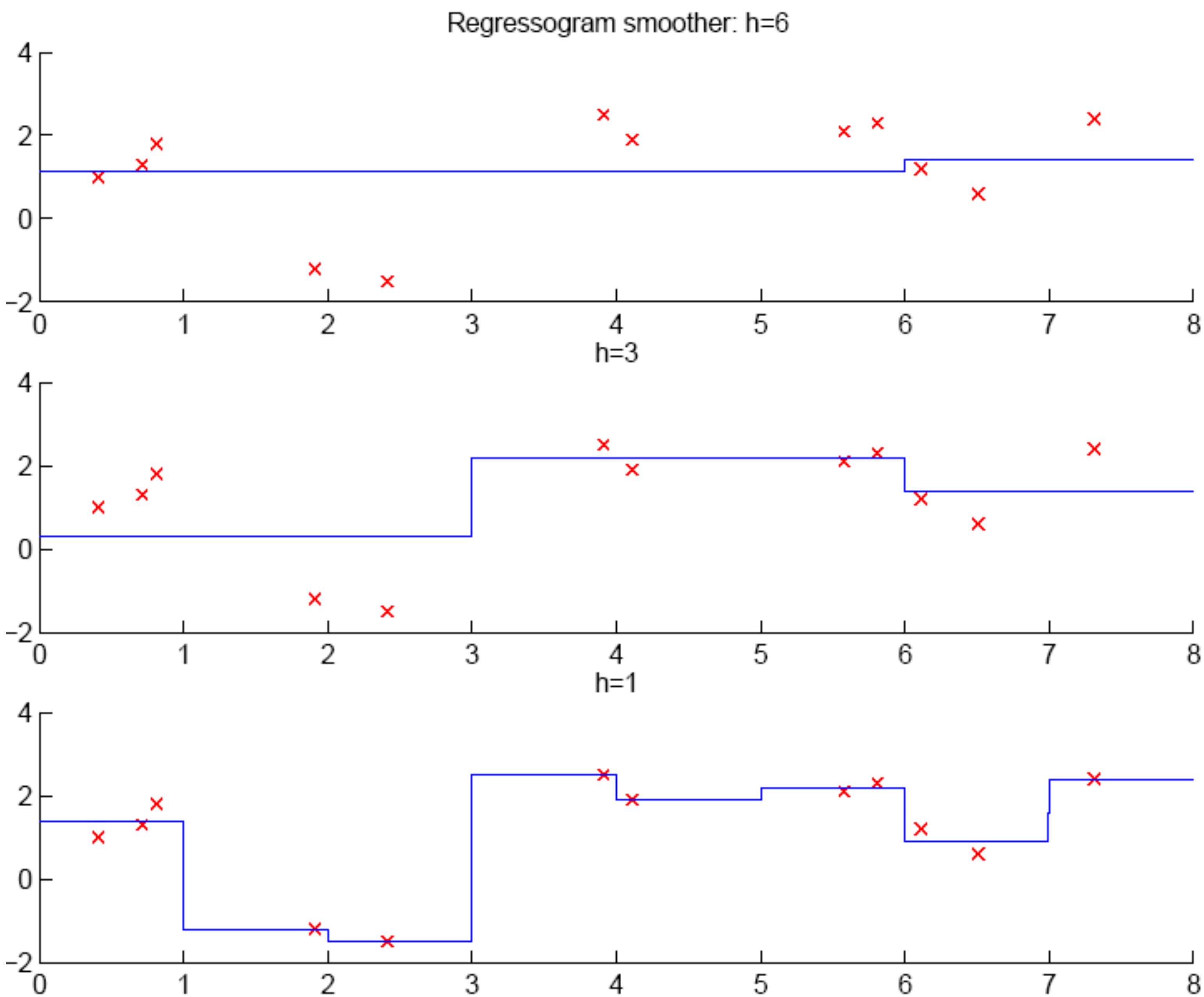
Nonparametric Regression

- Aka smoothing models: Regressogram
- Mean smoother

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

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
h=3
h=1
4
2
0
−2
0
1
2
3
4
5
6
7
8

Running Mean/Kernel Smoother

- Running mean smoother

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

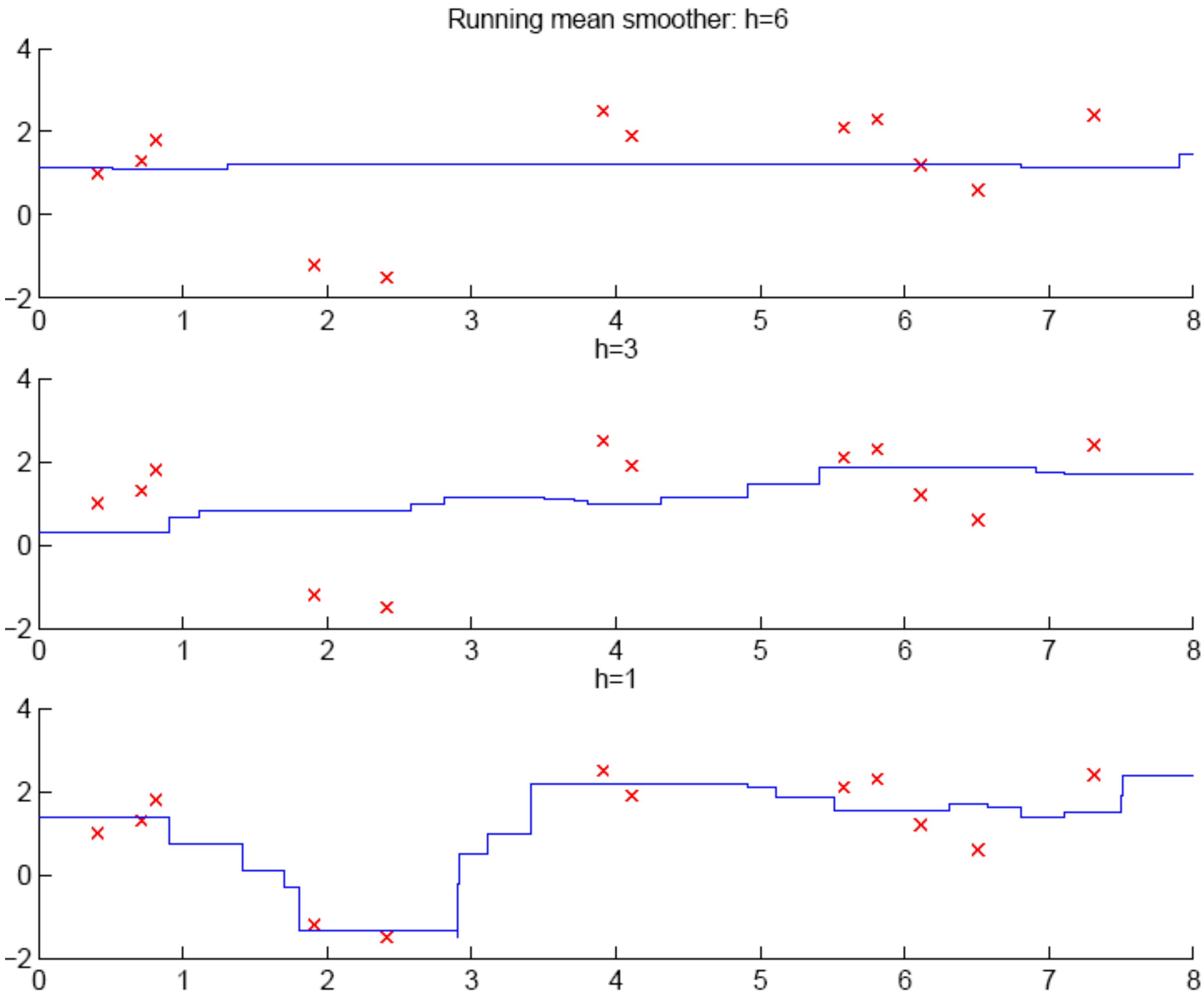
where

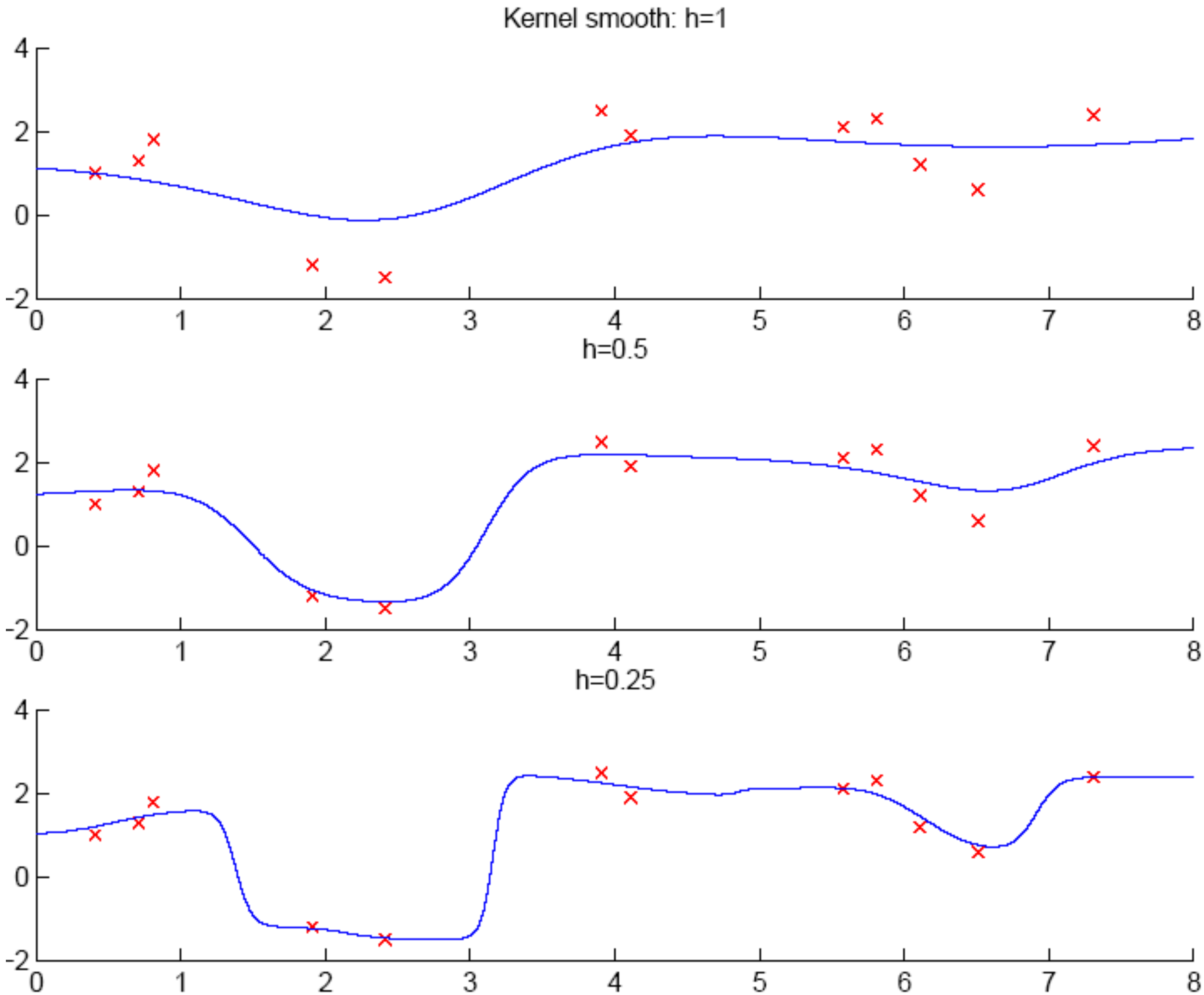
$$w(u) = \begin{cases} 1 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$

- Kernel smoother

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

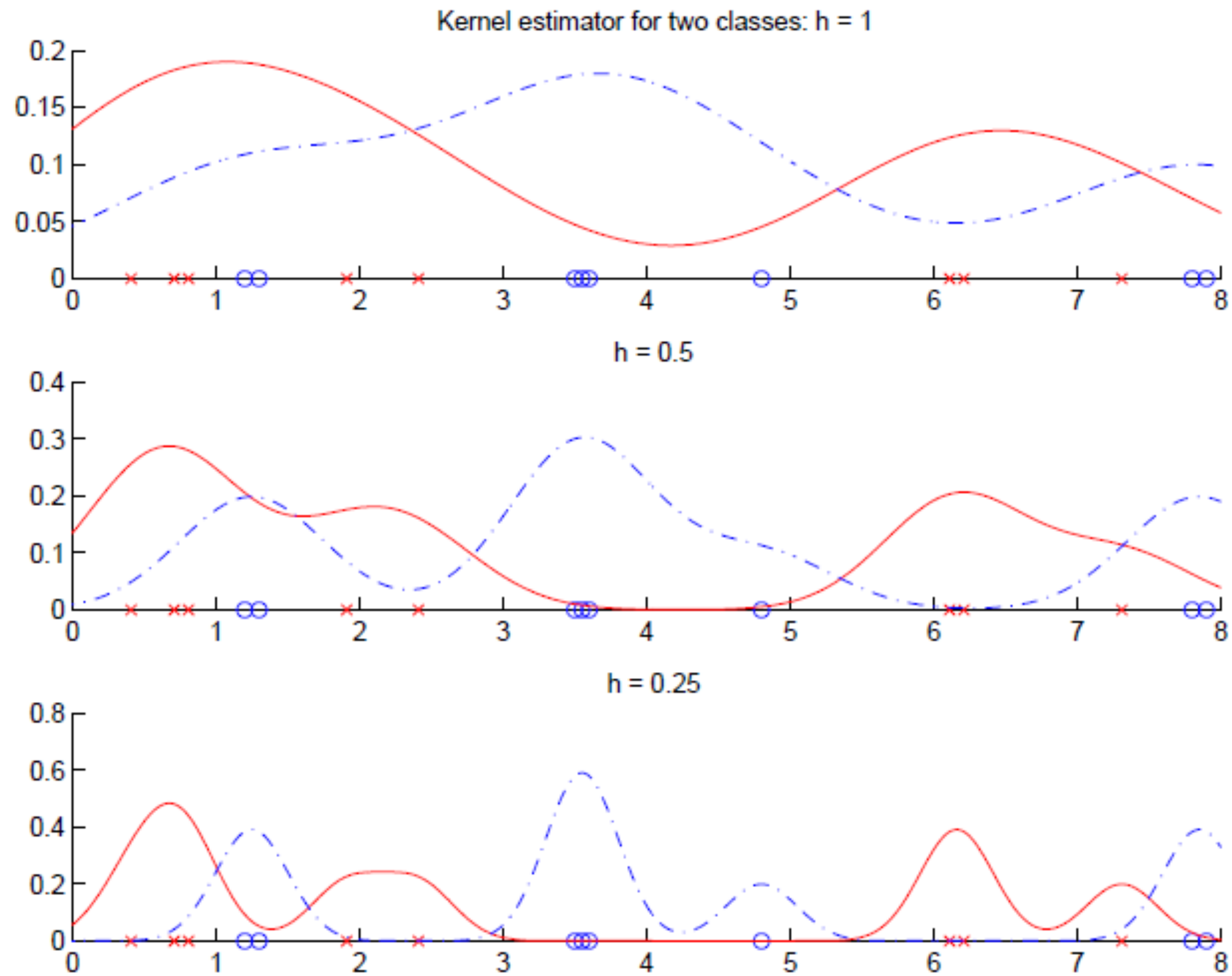
where *K*( ) is Gaussian


Running mean smoother: h=6
h=3
h=1
4
2
0
−2
0
1
2
3
4
5
6
7
8


Kernel smooth: h=1
h=0.5
h=0.25
4
2
0
-2
0
1
2
3
4
5
6
7
8

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*.



E. Alpaydin, Introduction to Machine Learning

Parametric vs Nonparametric

	Assumption	Efficiency	Sensitivity to outliers	Decision Power
Parametric				
Nonparametric				

Parametric vs Nonparametric

	Assumption	Efficiency	Sensitivity to outliers	Decision Power
Parametric	Assumption on distribution; maybe false; few choices of distributions with limited flexibility.			
Nonparametric	Smoothness assumption; Few assumptions on distribution.			

Parametric vs Nonparametric

	Assumption	Efficiency	Sensitivity to outliers	Decision Power
Parametric	Assumption on distribution; maybe false; few choices of distributions with limited flexibility.	Only need to keep the parameters of the distribution; Easy to make prediction;		
Nonparametric	Smoothness assumption; Few assumptions on distribution.	Expensive computation; Need to keep training data points for making a prediction.		

Parametric vs Nonparametric

	Assumption	Efficiency	Sensitivity to outliers	Decision Power
Parametric	Assumption on distribution; maybe false; few choices of distributions with limited flexibility.	Only need to keep the parameters of the distribution; Easy to make prediction;	Sensitive to outliers; An outlier might change parameters significantly.	
Nonparametric	Smoothness assumption; Few assumptions on distribution.	Expensive computation; Need to keep training data points for making a prediction.	Outlier might have a local impact.	

Parametric vs Nonparametric

	Assumption	Efficiency	Sensitivity to outliers	Decision Power
Parametric	Assumption on distribution; maybe false; few choices of distributions with limited flexibility.	Only need to keep the parameters of the distribution; Easy to make prediction;	Sensitive to outliers; An outlier might change parameters significantly.	Generalize better when the assumed distribution is correct.
Nonparametric	Smoothness assumption; Few assumptions on distribution.	Expensive computation; Need to keep training data points for making a prediction.	Outlier might have a local impact.	Poor power with small sample size; Tied values are expected in many regions.