Nonparametrics and Perceptron etc

STAT261: Introduction to Machine Learning

May 9, 2016

Prof. Allie Fletcher

Outline

- Introduction to nonparametric techniques
- Basic Issues in Density Estimation

Introduction

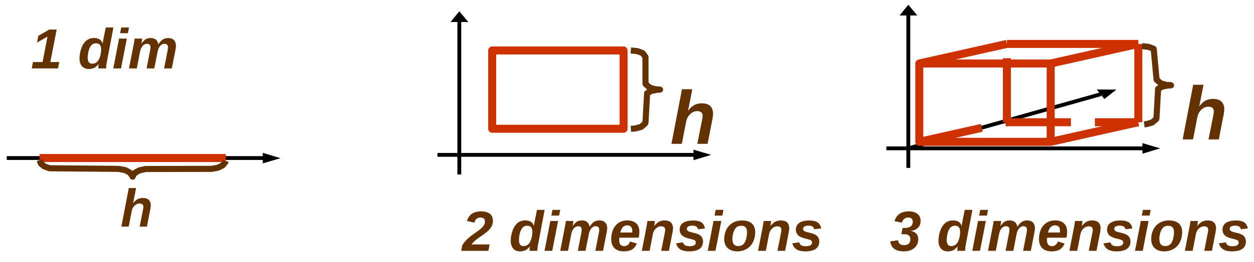
- Nonparametric methods
  - Density Estimation
  - Classification
  - Regression
- Nonparametric procedures: workwith arbitrary distributions
  - No assumptions on the forms of the underlying densities
  - Allow non unimodal distributions

Parzen Windows: Basics

- Parzen-window approach:we fix the size and shape of region
- Basic Model: histogram

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

- Assume region of $\boldsymbol{d}$-dimensional hypercube side length $\boldsymbol{h}$: volume is $\boldsymbol{h^d}$



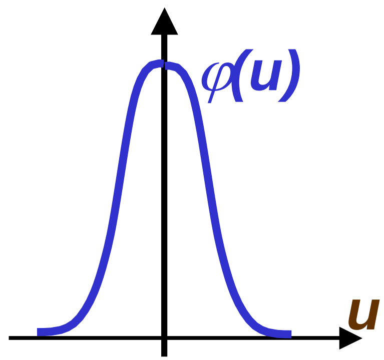
$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right) \qquad w(u) = \begin{cases} 1/2 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$

Parzen Windows: Smoothing kernels

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

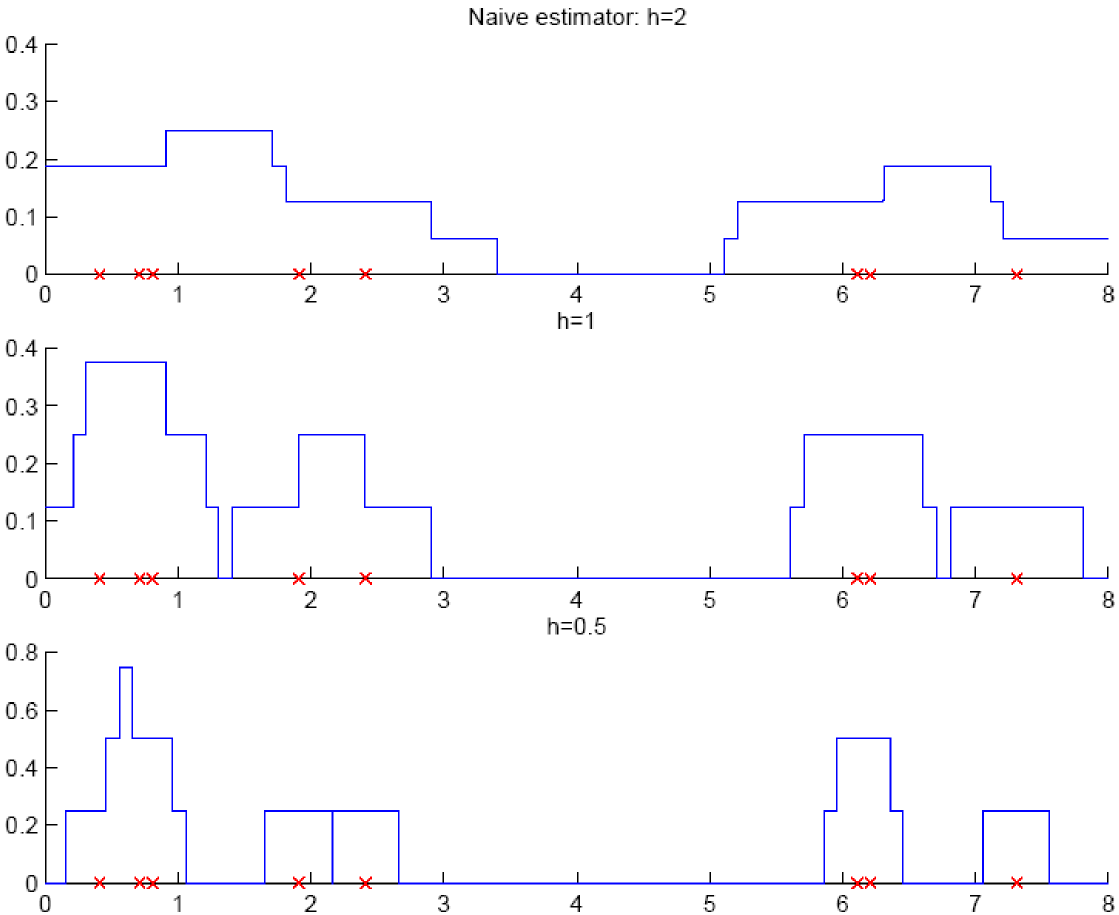
- A popular choice for φ is $\boldsymbol{N(0,1)}$ density

$$K(\boldsymbol{u}) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\pi}} \boldsymbol{e}^{-\boldsymbol{u}^{\mathbf{2}}/\mathbf{2}}$$



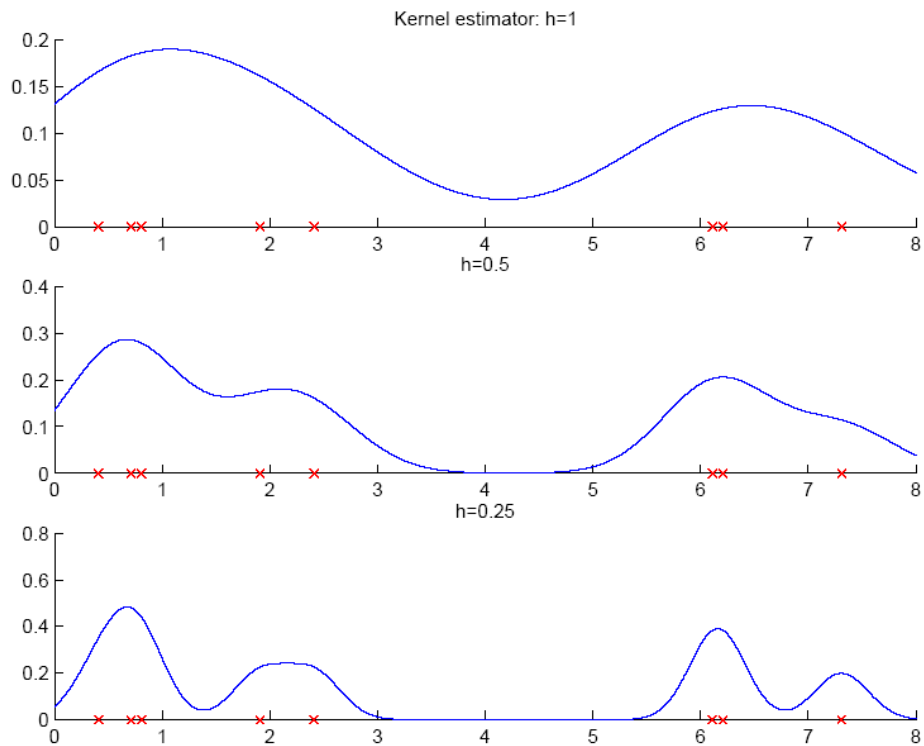
- Solves both drawbacks of the “box” window
  - Points $\boldsymbol{x}$ which are close to the sample point $\boldsymbol{x_i}$ receive higher weight
  - Resulting density $\boldsymbol{p}_K(\boldsymbol{x})$ is smooth

Parzen Windows: Naïve "boxes"



- $\hat{p}(x) \approx$ #points / bin
- Tradeoff in bin size
- Large bin size is better:
  - More points per bin
  - Less variance in density estimate
- But, larger bin size also:
  - "Smooths out" density
  - Cannot capture changes within a bin
  - Bias error

Parzen Windows: Smoothing kernels



- Using boxes results in discontinuous $\hat{p}(x)$
- Often we know $p(x)$ is smooth
- Smooth Kernel $\Rightarrow$ smooth $\hat{p}(x)$
- Bandwidth h controls smoothing
- Large h
  - More smoothing
  - Average over more samples
  - Lower variance in estimate
- Small h
  - Capture faster changes in $p(x)$
  - Lower bias

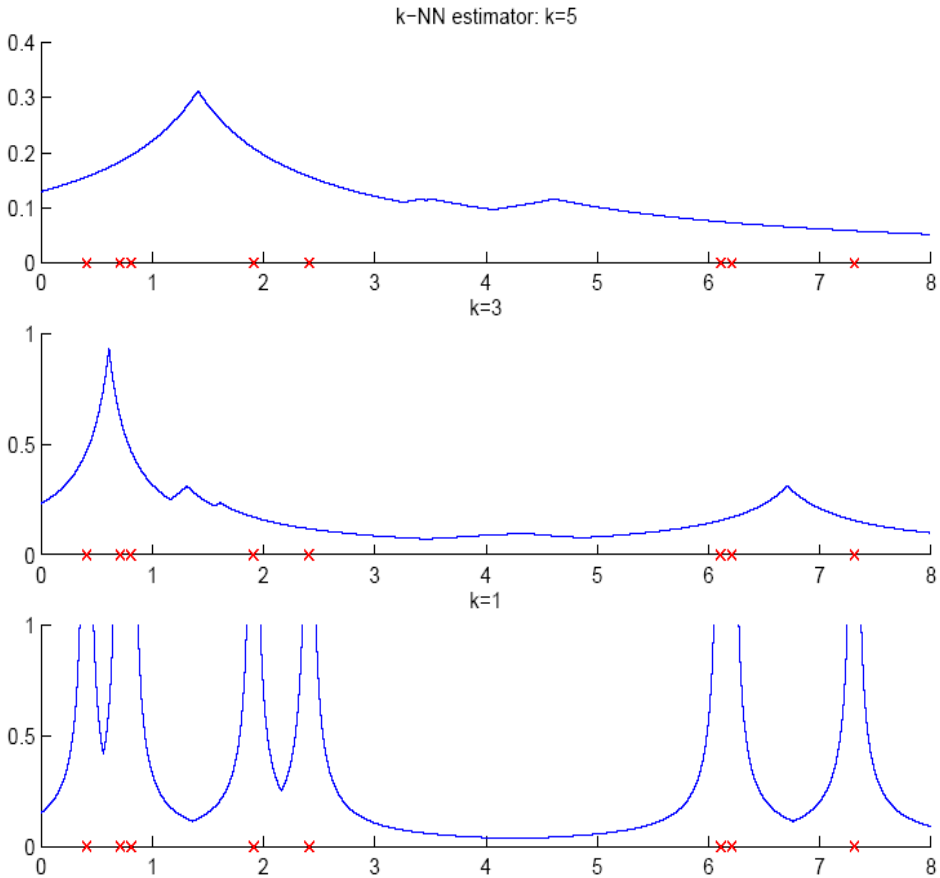
k-Nearest Neighbor Estimator

- Goal $$p_n(x) \underset{n \to \infty}{\rightarrow} p(x)$$
- 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*, *i.e.*
$d_1(x) <= d_2(x) <= d_3(x)$,..

k-Nearest Neighbor Estimator



- Adaptive level of smoothing
  - $h \approx d_k(x)$
- Ensures $\approx k$ samples in each average
- Uses larger h where data samples are sparse
  - Get lower variance estimate
- Uses smaller h where data samples are frequent
  - Get finer resolution estimate

k-Nearest Neighbor Estimator

- Problems with k nearest neighbor:
  - Not smooth
  - It is not a pdf as it integrates to infinity, not 1

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

- Kernel k-nearest neighbor

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

- Advantages:
  - Kernel k-nn is a proper density (integrates to one) and is smooth
  - Provides adaptive level of smoothing $h \approx d_k(x)$
  - Smaller bandwidth where data is frequent
  - Larger bandwidth where data is sparse

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

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

- Spheric implies kernel scaled equally on all dimensions! Inputs should be normalized to have same variance
- Better results if kernel has same form as underlying distributions: correlations should be taken into account

Nonparametric Classification

- Classification example

In classifiers based on Parzen-window estimation:

- We estimate the densities for each category and classify a test point by the label corresponding to the maximum posterior
- The decision region for a Parzen-window classifier depends upon the choice of window function

Nonparametric Classification

- Estimate $p(\boldsymbol{x}|C_i)$ and use Bayes' rule
- Nonparametric Kernel estimator for $p(\boldsymbol{x}|C_i)$
- MLE estimate for the prior $p(C_i)$

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

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

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

- xis just assigned to class where discriminant is maximized
- Ignore common $1/(Nh^d)$ term
- Each training instance votes for its class,
  has no effect on other classes
- Weight of vote is given by K(.), more weight to closer instances

Nonparametric Classification

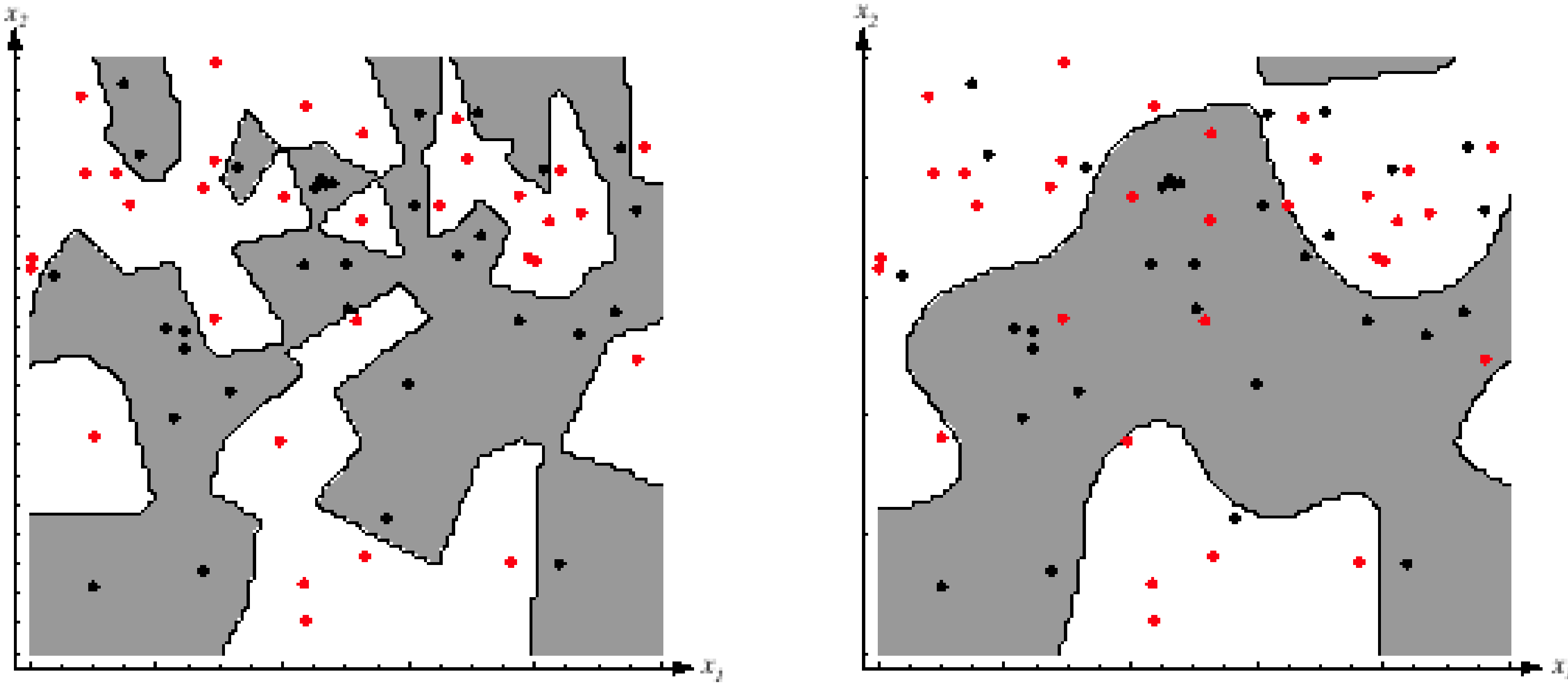


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the upper region, while a large h would be appropriate for the lower region; no single window width is ideal overall. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Nonparametric kNN Classification

- Estimate $p(\mathbf{x} \mid C_i)$ with k NN and use Bayes' rule
- *k*-NN estimator
  - $\hat{p}(x|C_i) = \frac{k_i}{N_i V^k(x)}$
  - $\hat{p}(x) = \frac{k}{N V^k(x)}$
  - $\hat{p}(C_i|x) = \frac{\hat{p}(x|C_i)}{\hat{p}(x)} \hat{P}(C_i) = \frac{k_i}{k}$
- Assigns class with most examples amongst k neighbors
- All neighbors = one vote
- Voronoi region

k_i is the number of neighbors out of nearest k that belong to C_i

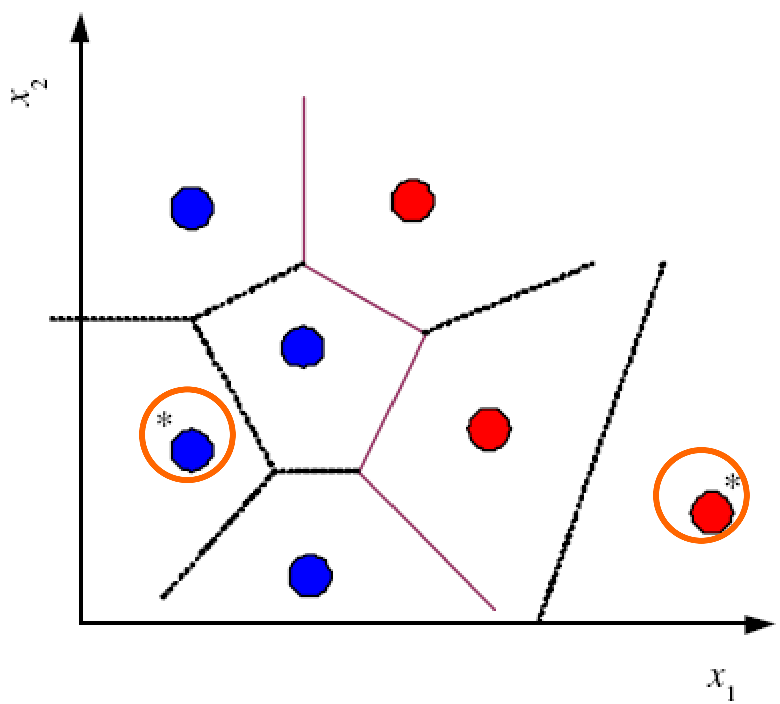
$V^k(\boldsymbol{x})$ is the volume of the d-dimensional hypercube centered at $\boldsymbol{x}$

$V^k = r^d\, c_d$ with c_d as volume of unit sphere in d dimensions…

$$c_1 = 1, c_2 = \pi, c_3 = \frac{4\pi}{3}, \dots$$

Voronoi neighborhood/tesselation

- 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

- Incremental algorithm: Add instance if needed
  - If instances nearest neighbors are same class, need not keep it
  - Consistent subset

$\mathcal{Z} \leftarrow \emptyset$

Repeat

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

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

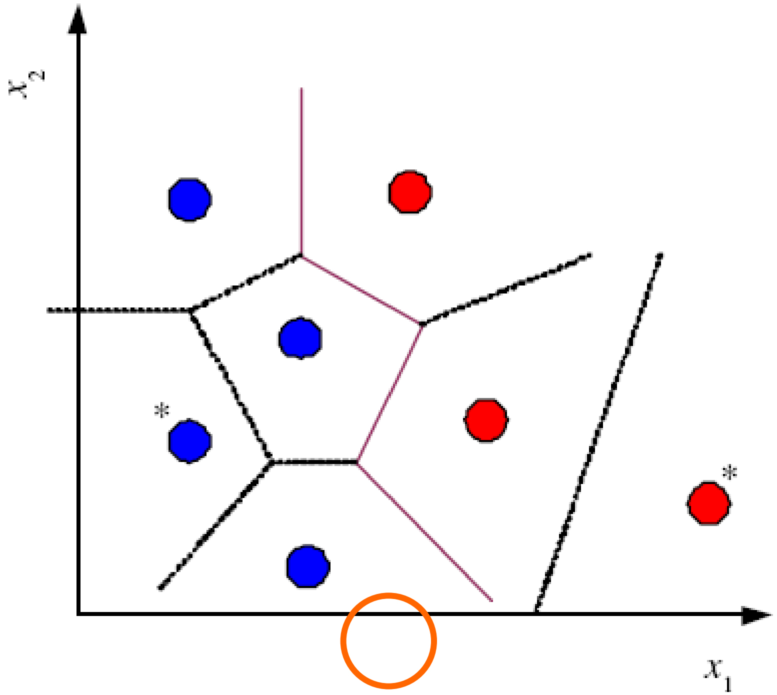
    $\boxed{\text{If class}(\boldsymbol{x}) \neq \text{class}(\boldsymbol{x}') \text{ add } \boldsymbol{x} \text{ to } \mathcal{Z}}$

Until $\mathcal{Z}$ does not change

  - Does not guarantee finding minimal consistent subset

Condensed Nearest Neighbor

- 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)



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

- Minimize training error plus complexity second term regularizes complexity

Nonparametric Regression

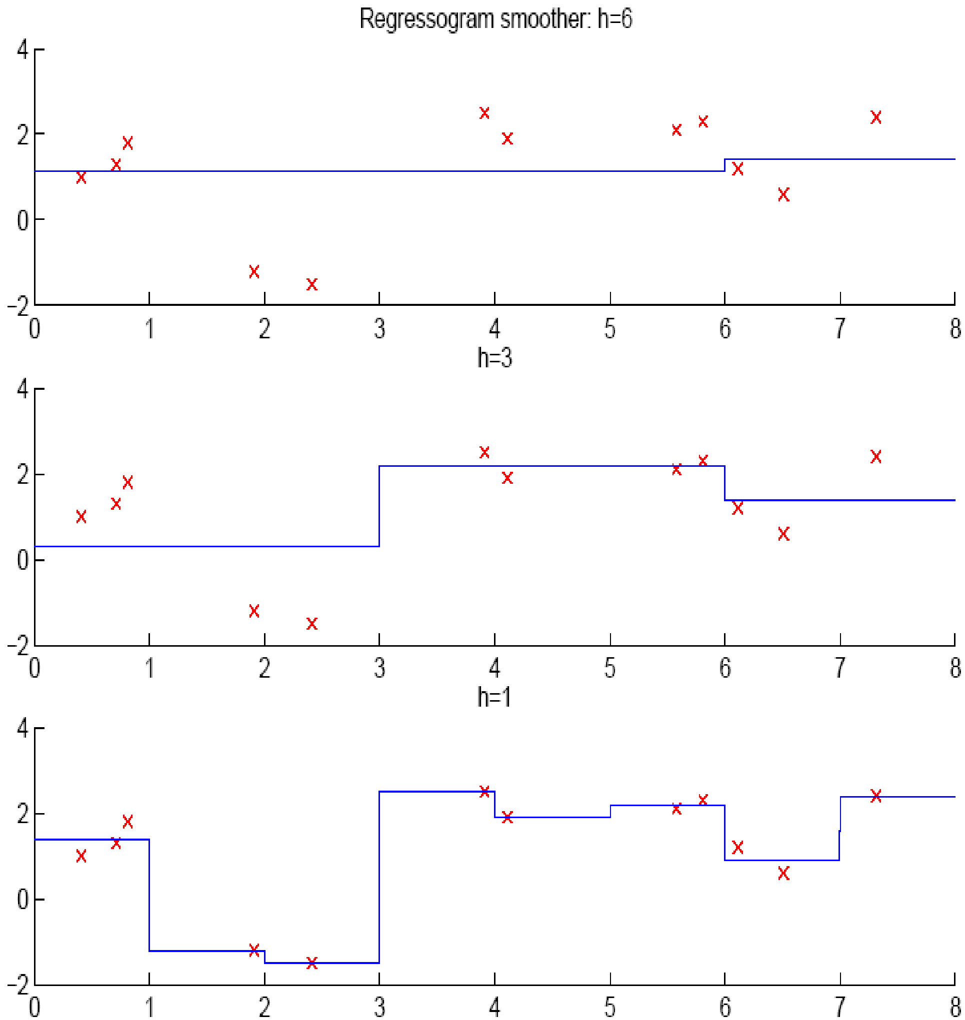
- Want to estimate a function $r = g(x)$
  - Up to now, we assumed a model for $g(x)$, e.g. linear
  - What if we don't have a model?
  - Do not want to assume a parametric structure for $g(x)$
- Simple non-parametric estimate: Regressogram
  - Divide x space into "bins"
  - $\hat{g}(x)$ = Average observed response in each bin.

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

where

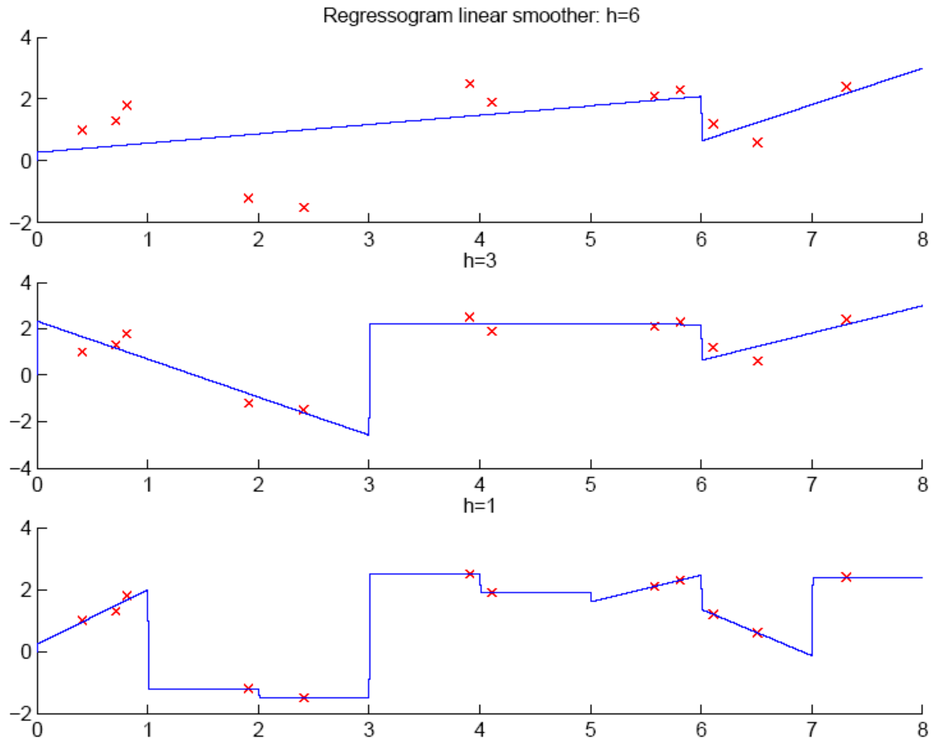
$$b(x, x^t) = \begin{cases} 1 & \text{if } x^t \text{ is in the same bin with } x \\ 0 & \text{otherwise} \end{cases}$$

Nonparametric Regression



- Figure shows bins at $[kh, (k+1)h]$
- h determines level of smoothing
- Large h smooths $\hat{g}(x)$
  - More samples in each bin
  - Lower variance
- But, large h assumes $\hat{g}(x)$ is constant over larger region
  - Larger error if actual $g(x)$ changes

Regressogram with Piecewise Linear Functions



- Previous example uses a piecewise constant $\hat{g}(x)$
- Fit a linear model in each bin
  - When only one sample is available, use a constant model
- Enables a richer class
- But, needs more samples per bin
- Again, h determines level of smoothing

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

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

- Running line smoother

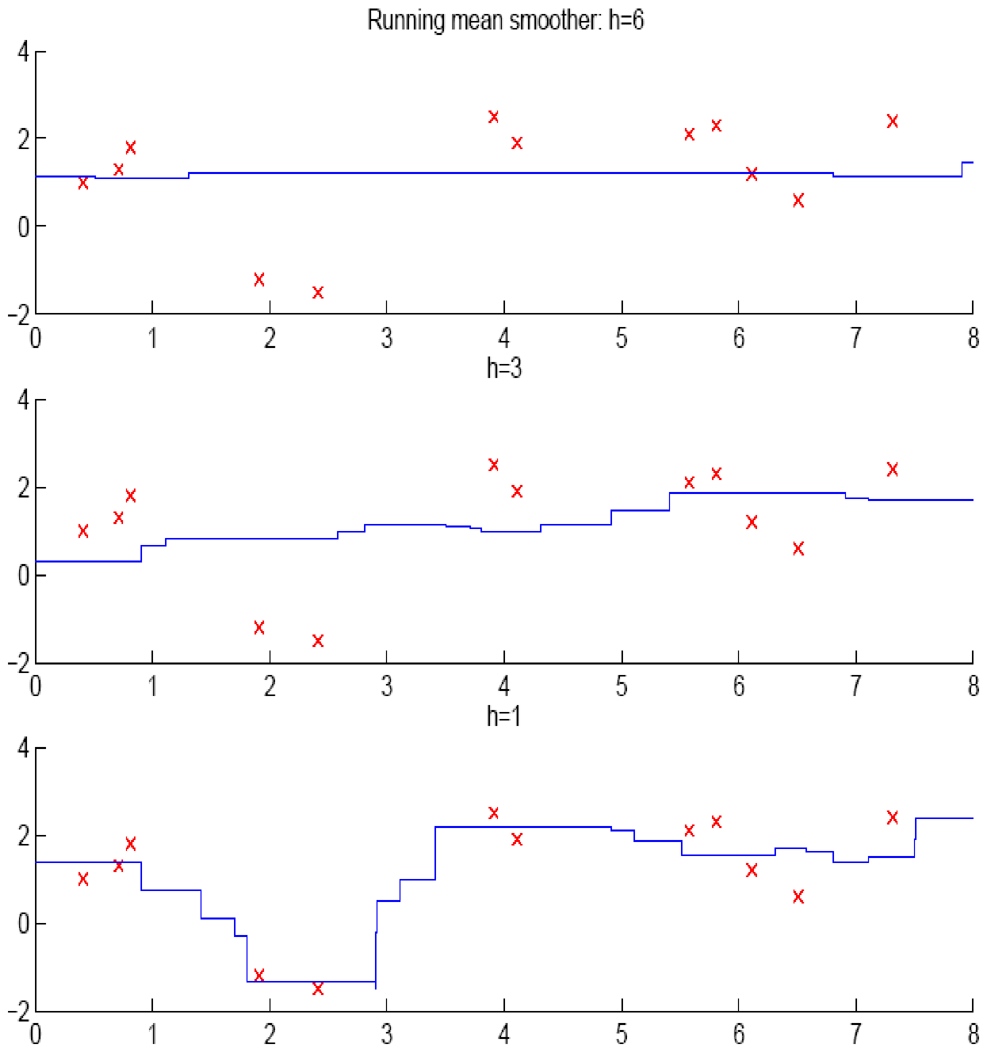
- Kernel smoother

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

where *K*( ) is Gaussian

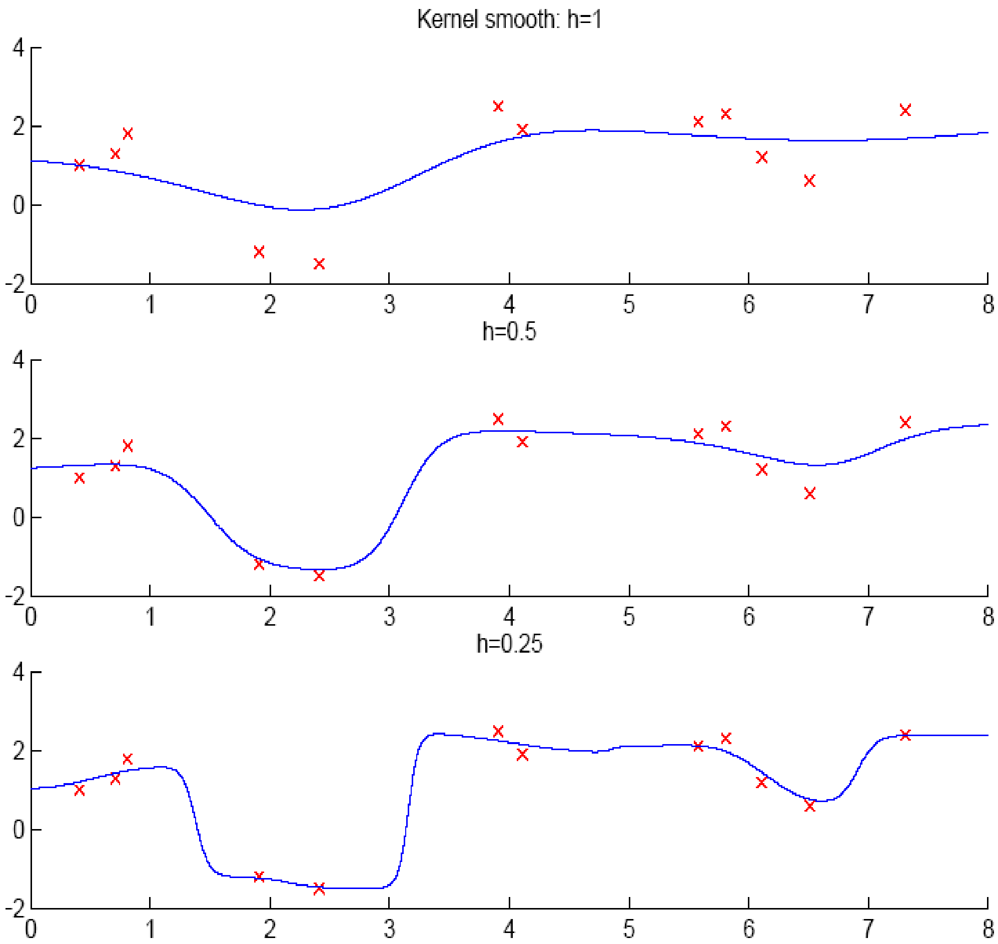
- Additive models (Hastie and Tibshirani, 1990)

Kernel Smoother



- Boundaries of the bin intervals are not fixed
- Based on locations of the data points
- Provides less abrupt changes
- Again, h determines level of smoothing

Nonparametric Regression

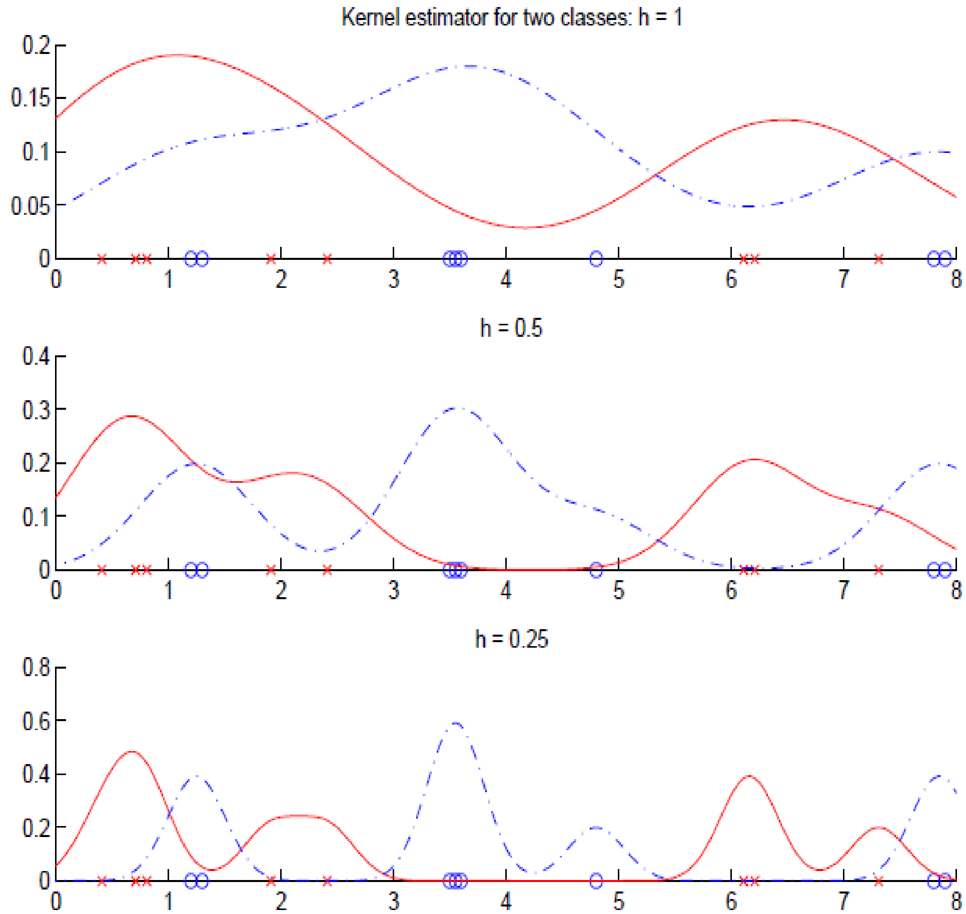


- Using naïve window, results in discontinuous $\hat{g}(x)$
- Kernel smoothing provides smooth $\hat{g}(x)$
- More realistic in most applications
  - Real response is often smooth
- Can take derivative of $\hat{g}(x)$ if necessary

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*.

How to Choose k or h?



- Figure shows density estimate for two classes under different h
- Smaller h / k enables more finer resolution density estimate
  - $\hat{p}(x|C_i)$ can change rapidly over x
  - Enables more complex classifier
  - But, requires more data