Nonparametric Density Estimation
Intro
Parzen Windows

Non-Parametric Methods

- Neither probability distribution nor discriminant function is known
  - Happens quite often
- All we have is labeled data

- Estimate the probability distribution from the labeled data

a lot is known "easier"

little is known "harder"

NonParametric Techniques: Introduction

- In previous lectures we assumed that either
  1. someone gives us the density $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{c_j})$
     - In pattern recognition applications this never happens
  2. someone gives us $\boldsymbol{p}(\boldsymbol{x}|\theta_{\boldsymbol{cj}})$
     - Does happen sometimes, **but**
       - we are likely to suspect whether the given $\boldsymbol{p}(\boldsymbol{x}|\theta)$ models the data well
       - Most parametric densities are unimodal (have a single local maximum), whereas many practical problems involve multi-modal densities

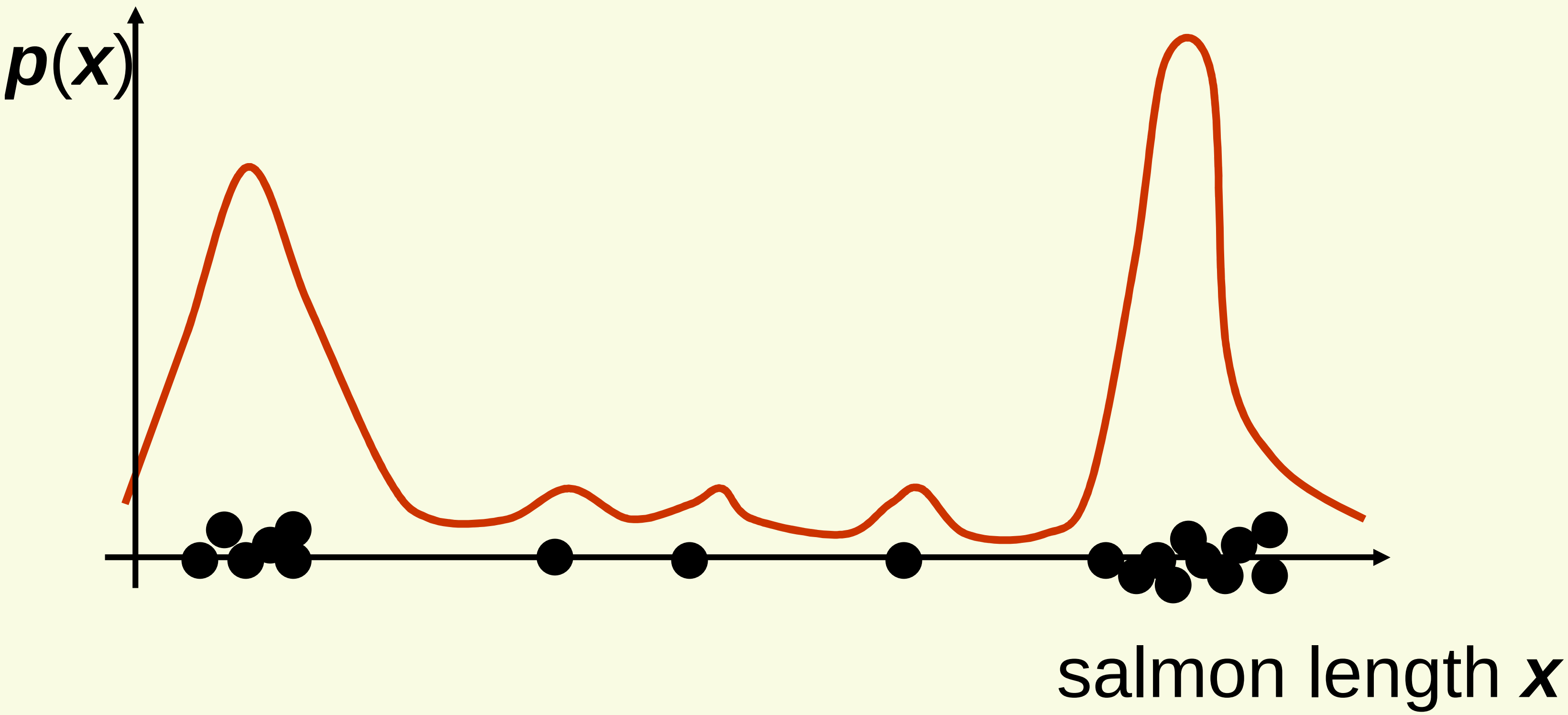
NonParametric Techniques: Introduction

- Nonparametric procedures can be used with arbitrary distributions and without any assumption about the forms of the underlying densities
- There are two types of nonparametric methods:
  - Parzen windows
    - Estimate likelihood $p(x \mid c_j)$
  - Nearest Neighbors
    - Bypass likelihood and go directly to posterior estimation $P(c_j \mid x)$

NonParametric Techniques: Introduction

- Nonparametric techniques attempt to estimate the underlying density functions from the training data
  - Idea: the more data in a region, the larger is the density function

$$\boldsymbol{Pr}\left[\boldsymbol{X} \in \Re\right] = \int_{\Re} \boldsymbol{f}\left(\boldsymbol{x}\right) \boldsymbol{dx}$$



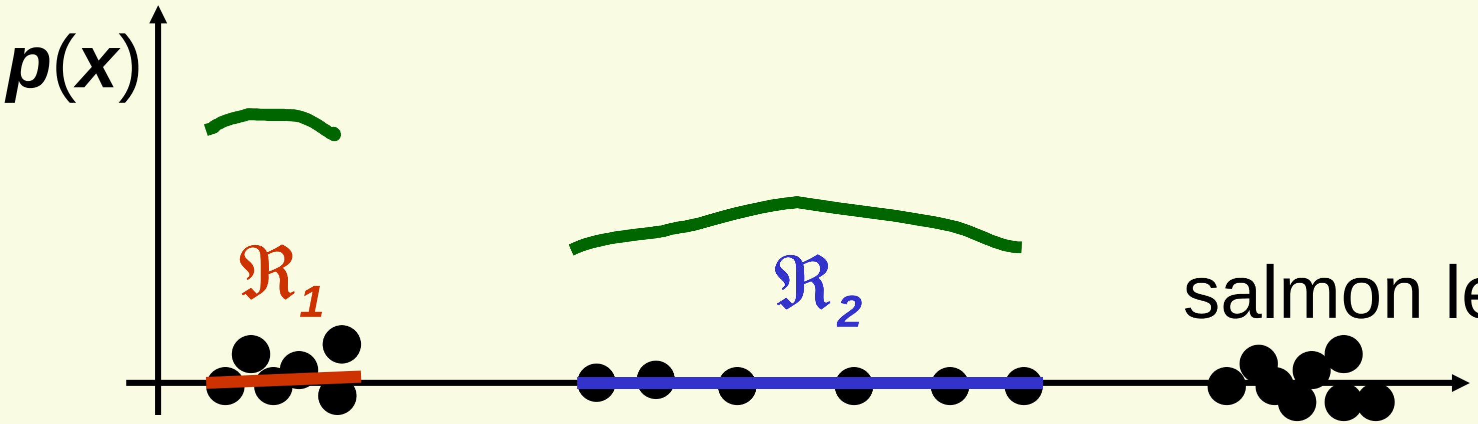
NonParametric Techniques: Introduction

$$Pr[X \in \Re] = \int_{\Re} f(x)dx$$

- How can we approximate $Pr[X \in \Re_1]$ and $Pr[X \in \Re_2]$?
  - $Pr[X \in \Re_1] \approx \frac{6}{20}$ and $Pr[X \in \Re_2] \approx \frac{6}{20}$
- Should the density curves above $\mathcal{R}_1$ and $\mathcal{R}_2$ be equally high?
  - No, since $\mathcal{R}_1$ is smaller than $\mathcal{R}_2$

$$Pr[X \in \Re_1] = \int_{\Re_1} f(x)dx \approx \int_{\Re_2} f(x)dx = Pr[X \in \Re_2]$$

  - To get density, normalize by region size



NonParametric Techniques: Introduction

- Assuming ***f***(***x***) is basically flat inside $\mathcal{R}$,

$$\frac{\#\,of\ samples\ in\,\Re}{total\,\#\,of\ samples} \approx Pr\left[X \in \Re\right] = \int_{\Re} f\left(y\right)dy \approx f\left(x\right) * Volume\left(\Re\right)$$

- Thus, density at a point ***x*** inside $\mathcal{R}$, can be approximated

$$f\left(x\right) \approx \frac{\#\,of\ samples\ in\,\Re}{total\,\#\,of\ samples}\ \frac{1}{Volume\left(\Re\right)}$$

- Now let's derive this formula more formally

Binomial Random Variable

- Let us flip a coin ***n*** times (each one is called "trial")
  - Probability of head ρ, probability of tail is 1-ρ
- Binomial random variable ***K*** counts the number of heads in ***n*** trials

$$P(K=k)=\binom{n}{k}\rho^k(1-\rho)^{n-k}$$

where $$\binom{n}{k}=\frac{n!}{k!(n-k)!}$$

- Mean is $E(K)=n\rho$
- Variance is $\mathrm{var}(K)=n\rho(1-\rho)$

Density Estimation: Basic Issues

- From the definition of a density function, probability ρ that a vector $\boldsymbol{x}$ will fall in region $\mathcal{R}$ is:

$$\rho = \boldsymbol{Pr}\left[\boldsymbol{x} \in \Re\right] = \int_{\Re} \boldsymbol{p}(\boldsymbol{x}')d\boldsymbol{x}'$$

- Suppose we have samples $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$ drawn from the distribution $\boldsymbol{p}(\boldsymbol{x})$. The probability that $\boldsymbol{k}$ points fall in $\mathcal{R}$ is then given by binomial distribution:

$$\mathbf{Pr}\left[\boldsymbol{K} = \boldsymbol{k}\right] = \binom{\boldsymbol{n}}{\boldsymbol{k}} \rho^{k} (\mathbf{1} - \rho)^{\boldsymbol{n-k}}$$

- Suppose that $\boldsymbol{k}$ points fall in $\mathcal{R}$, we can use MLE to estimate the value of ρ. The likelihood function is

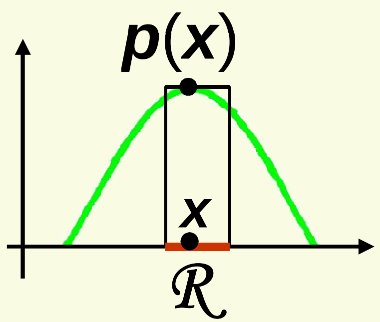
$$\boldsymbol{p}\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \mid \rho\right) = \binom{\boldsymbol{n}}{\boldsymbol{k}} \rho^{k} (\mathbf{1} - \rho)^{\boldsymbol{n-k}}$$

Density Estimation: Basic Issues

$$p(x_1,\dots,x_n \mid \rho) = \binom{n}{k} \rho^k (1-\rho)^{n-k}$$

- This likelihood function is maximized at $\rho = \frac{k}{n}$
- Thus the MLE is $\hat{\rho} = \frac{k}{n}$
- Assume that ***p(x)*** is continuous and that the region $\mathcal{R}$ is so small that ***p(x)*** is approximately constant in $\mathcal{R}$

$$\int_{\Re} p(x')dx' \cong p(x)V$$

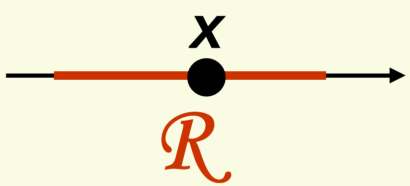


  - x is in $\mathcal{R}$ and V is the volume of $\mathcal{R}$
- Recall from the previous slide: $\rho = \int_{\Re} p(x')dx'$
- Thus ***p(x)*** can be approximated: $\boxed{p(x) \approx \frac{k/n}{V}}$

Density Estimation: Basic Issues

- This is exactly what we had before:

$$p(x) \approx \frac{k/n}{V}$$



***x* is inside some region $\mathcal{R}$**
***k* = number of samples inside $\mathcal{R}$**
***n*=total number of samples**
***V* = volume of $\mathcal{R}$**

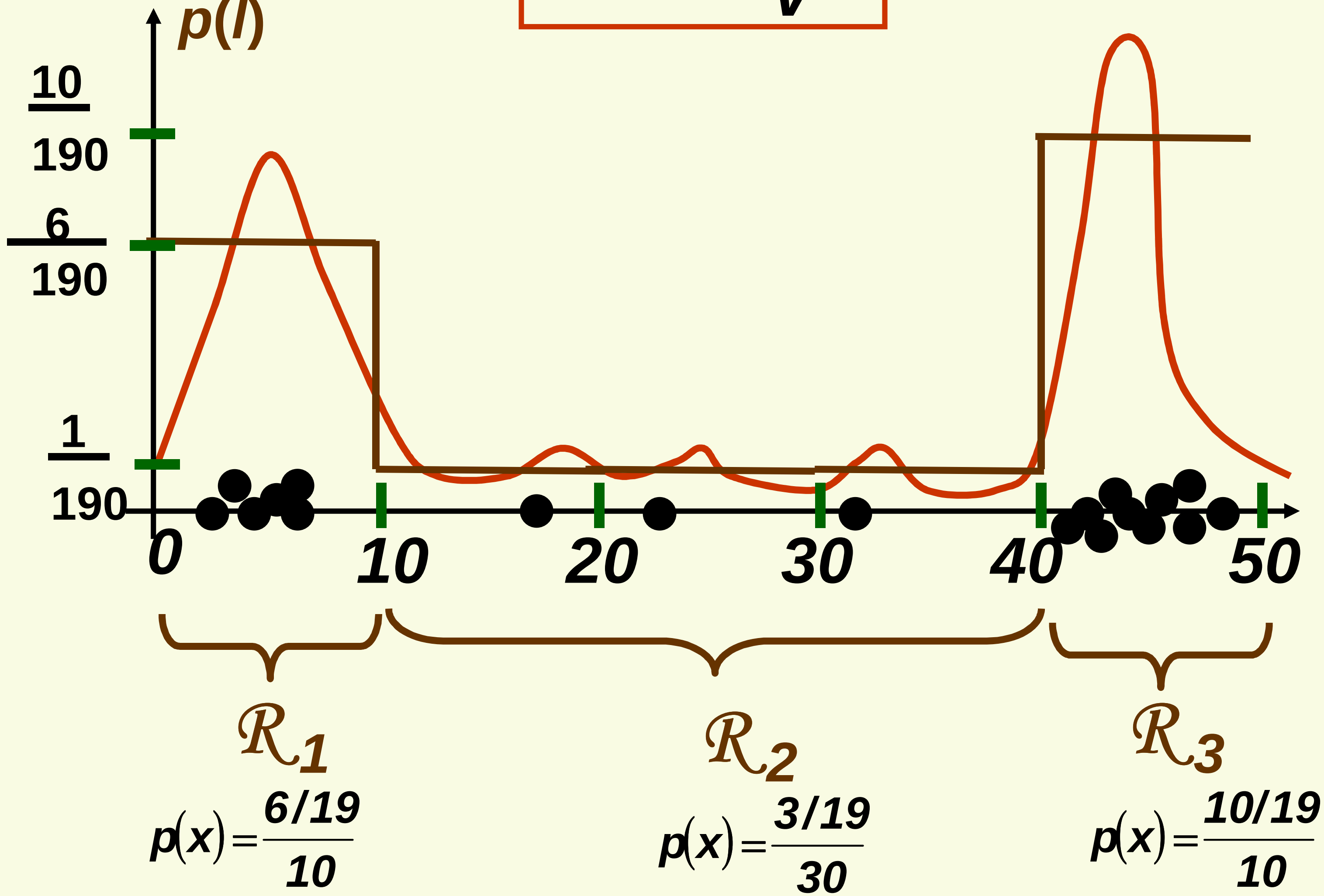
- Our estimate will always be the average of true density over $\mathcal{R}$

$$p(x) \approx \frac{k/n}{V} = \frac{\hat{\rho}}{V} \approx \frac{\int_{\Re} p(x')dx'}{V}$$

- Ideally, $p(x)$ should be constant inside $\mathcal{R}$

Density Estimation: Histogram

$$p(x) \approx \frac{k/n}{V}$$

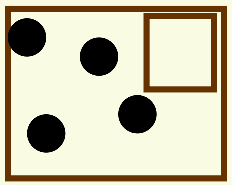


- If regions $\mathcal{R}_i$'s do not overlap, we have a histogram

Density Estimation: Accuracy

- How accurate is density approximation $p(x) \approx \frac{k/n}{V}$?
- We have made two approximations

1. $\hat{\rho} = \frac{k}{n}$
   - as n increases, this estimate becomes more accurate
2. $\int_{\Re} p(x')dx' \cong p(x)V$
   - as $\mathcal{R}$ grows smaller, the estimate becomes more accurate



- As we shrink $\mathcal{R}$ we have to make sure it contains samples, otherwise our estimated $p(x) = 0$ for all x in $\mathcal{R}$

- Thus in theory, if we have an unlimited number of samples, we get convergence as we simultaneously increase the number of samples ***n***, and shrink region $\mathcal{R}$, but not too much so that $\mathcal{R}$ still contains a lot of samples

Density Estimation: Accuracy

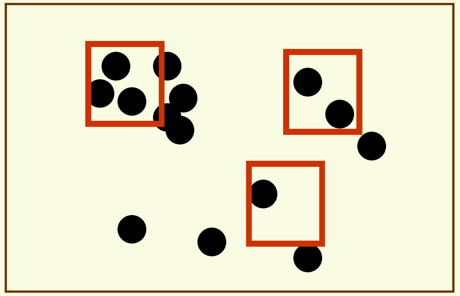
$$\boldsymbol{p}(\boldsymbol{x}) \approx \frac{\boldsymbol{k} / \boldsymbol{n}}{\boldsymbol{V}}$$

- In practice, the number of samples is always fixed
- Thus the only available option to increase the accuracy is by decreasing the size of $\mathcal{R}$ (***V*** gets smaller)
  - If ***V*** is too small, ***p***(***x***)=0 for most **x**, because most regions will have no samples
  - Thus have to find a compromise for ***V***
    - not too small so that it has enough samples
    - but also not too large so that ***p***(***x***) is approximately constant inside ***V***

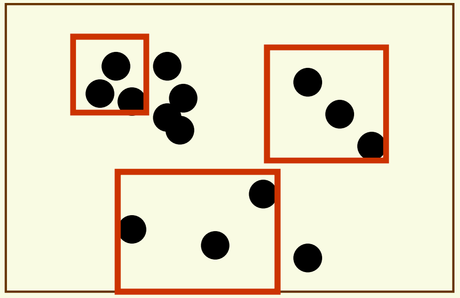
Density Estimation: Two Approaches

$$p(x) \approx \frac{k/n}{V}$$

1. Parzen Windows:
   - Choose a fixed value for volume ***V*** and determine the corresponding ***k*** from the data



2. k-Nearest Neighbors
   - Choose a fixed value for ***k*** and determine the corresponding volume ***V*** from the data



- Under appropriate conditions and as number of samples goes to infinity, both methods can be shown to converge to the true $p(x)$

Parzen Windows

$$p(x) \approx \frac{k/n}{V}$$

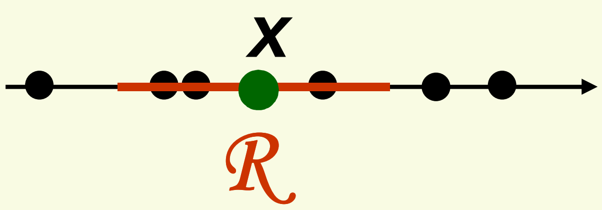
***x* is inside some region $\mathcal{R}$**

***k* = number of samples inside $\mathcal{R}$**

***n*=total number of samples**

***V* = volume of $\mathcal{R}$**

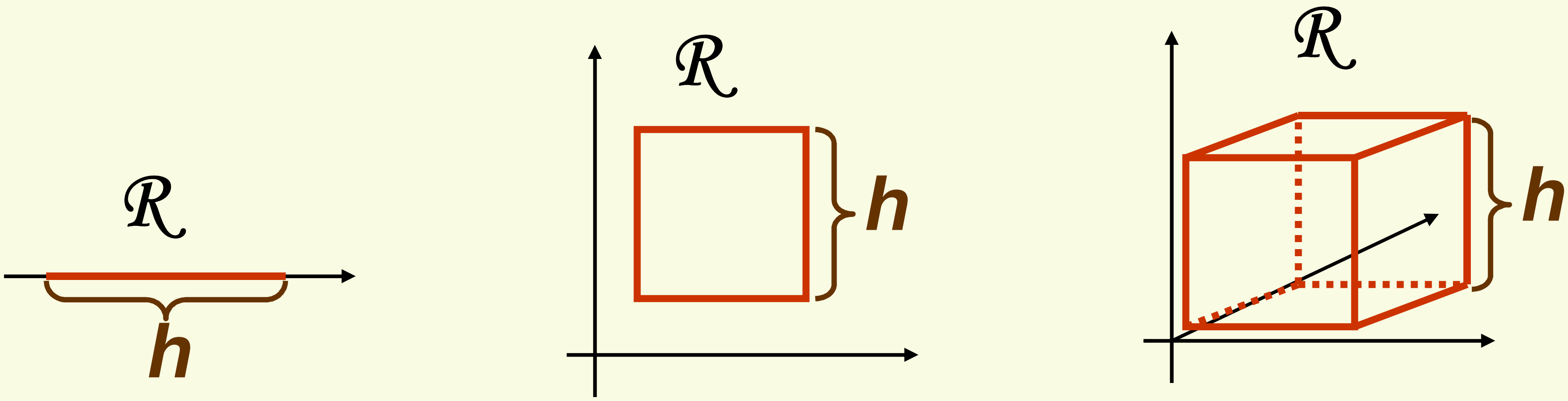
- To estimate the density at point ***x***, simply center the region $\mathcal{R}$ at ***x***, count the number of samples in $\mathcal{R}$, and substitute everything in our formula



$$p(x) \approx \frac{3/6}{10}$$

Parzen Windows

- In Parzen-window approach to estimate densities we fix the size and shape of region $\mathcal{R}$
- Let us assume that the region $\mathcal{R}$ is a ***d***-dimensional hypercube with side length ***h*** thus it's volume is h^d



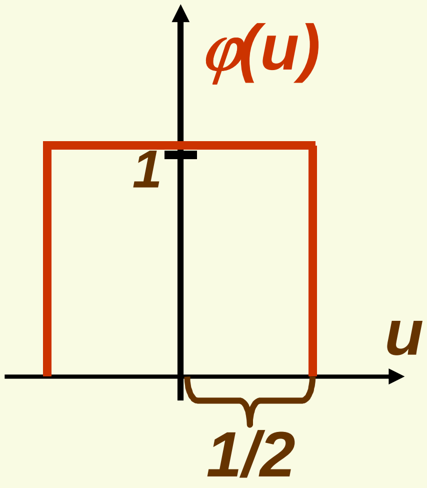
1 dimension ***2 dimensions*** ***3 dimensions***

Parzen Windows

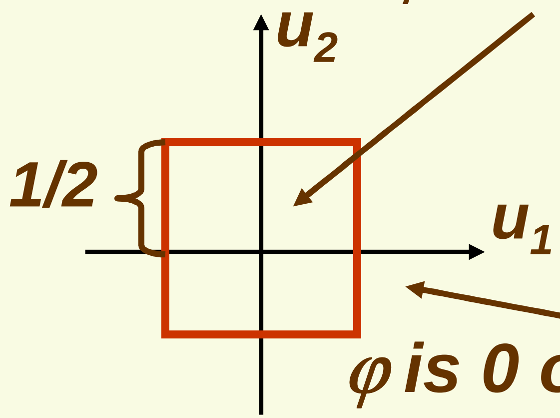
- Let u=[u_1, u_2,…, u_d] and define a window function

$$\varphi(u) = \begin{cases} 1 & \left|u_j\right| \le \frac{1}{2} \quad j = 1,\dots,d \\ 0 & otherwise \end{cases}$$

1 dimension



φ is 1 inside



φ is 0 outside

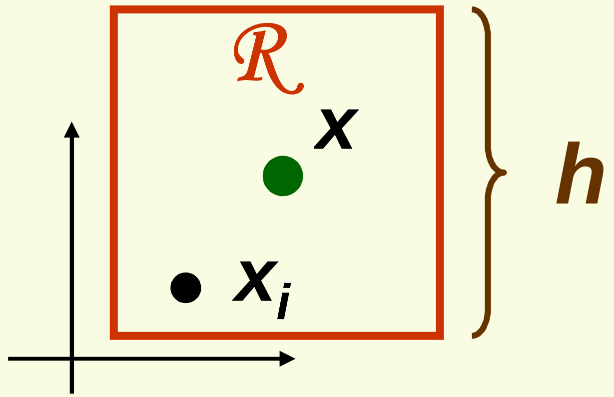
2 dimensions

Parzen Windows

- Recall we have d-dimensional samples $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$. Let x_{ij} be the jth coordinate of sample x_i. Then

$$\varphi\Big(\underbrace{\frac{x - x_i}{h}}_{u}\Big) = \begin{cases} 1 & |x_j - x_{ij}| \le \dfrac{h}{2} \quad j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

$$|u_j| \le \frac{1}{2}$$



$$\varphi\Big(\frac{x - x_i}{h}\Big) = \begin{cases} \textbf{1} & \textit{if } x_i \textit{ is inside the hypercube with width } h \textit{ and centered at } x \\ \textbf{0} & \textit{otherwise} \end{cases}$$

Parzen Windows

- How do we count the total number of sample points $\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}$ which are inside the hypercube with side h and centered at $\boldsymbol{x}$?

$$k = \sum_{i=1}^{i=n} \varphi\left(\frac{x - x_i}{h}\right)$$

- Recall $p(x) \approx \frac{k/n}{V}$, $V = h^d$

- Thus we get the desired analytical expression for the estimate of density $p_\varphi(\boldsymbol{x})$

$$p_\varphi(x) = \frac{\sum_{i=1}^{i=n} \varphi\left(\frac{x - x_i}{h}\right) / n}{h^d} = \frac{1}{n} \sum_{i=1}^{i=n} \frac{1}{h^d} \varphi\left(\frac{x - x_i}{h}\right)$$

Parzen Windows

$$p_\varphi(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

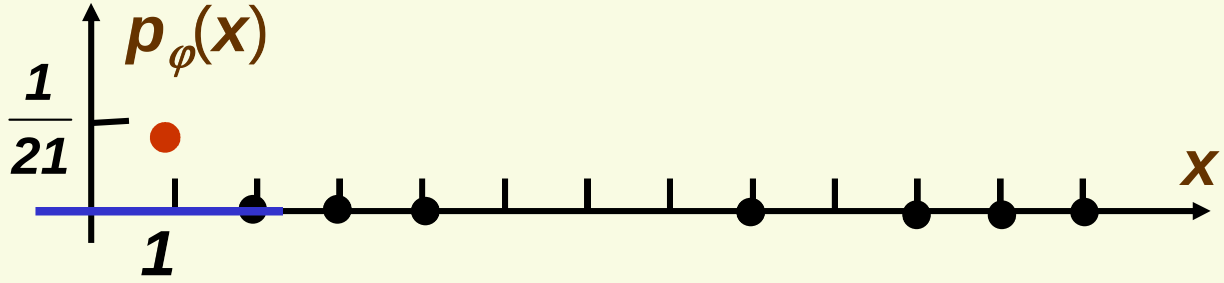
- Let's make sure $p_\varphi(\mathbf{x})$ is in fact a density
- $p_\varphi(\mathbf{x}) \geq 0 \qquad \forall \mathbf{x}$
- $$\int p_\varphi(\mathbf{x})d\mathbf{x} = \int \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x} = \frac{1}{h^d n}\sum_{i=1}^{i=n}\overbrace{\int \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)d\mathbf{x}}^{\text{volume of hypercube}}$$

$$= \frac{1}{n}\frac{1}{h^d}\sum_{i=1}^{i=n}h^d = 1$$

Parzen Windows: Example in 1D

$$p_\varphi(x)=\frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x-x_i}{h}\right)$$

- Suppose we have 7 samples D={2,3,4,8,10,11,12}



- Let window width h=3, estimate density at x=1

$$p_\varphi(1)=\frac{1}{7}\sum_{i=1}^{i=7}\frac{1}{3}\varphi\left(\frac{1-x_i}{3}\right)=\frac{1}{21}\left[\varphi\left(\frac{1-2}{3}\right)+\varphi\left(\frac{1-3}{3}\right)+\varphi\left(\frac{1-4}{3}\right)+\dots+\varphi\left(\frac{1-12}{3}\right)\right]$$

$$\left|-\frac{1}{3}\right|\le 1/2 \qquad \left|-\frac{2}{3}\right|>1/2 \qquad \left|-1\right|>1/2 \qquad \left|-\frac{11}{3}\right|>1/2$$

$$p_\varphi(1)=\frac{1}{7}\sum_{i=1}^{i=7}\frac{1}{3}\varphi\left(\frac{1-x_i}{3}\right)=\frac{1}{21}\left[1+0+0+\dots+0\right]=\frac{1}{21}$$

Parzen Windows: Sum of Functions

- Now let's look at our density estimate $\boldsymbol{p}_\varphi(\boldsymbol{x})$ again:

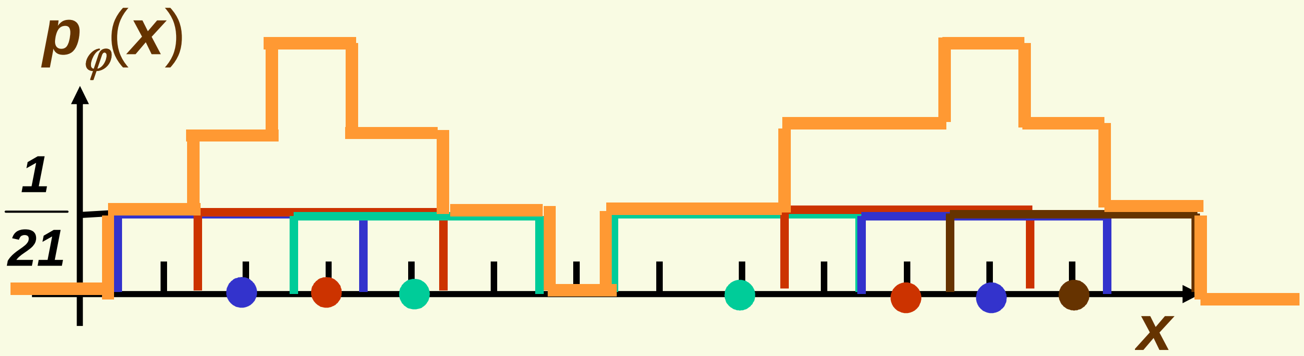
$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right) = \sum_{i=1}^{i=n}\frac{1}{nh^d}\underbrace{\varphi\left(\frac{x - x_i}{h}\right)}$$

1 inside square centered at x_i
0 otherwise

- Thus $\boldsymbol{p}_\varphi(\boldsymbol{x})$ is just a sum of $\boldsymbol{n}$ "box like" functions each of height $\dfrac{1}{nh^d}$

Parzen Windows: Example in 1D

- Let's come back to our example
  - 7 samples ***D***={2,3,4,8,10,11,12}, ***h***=3

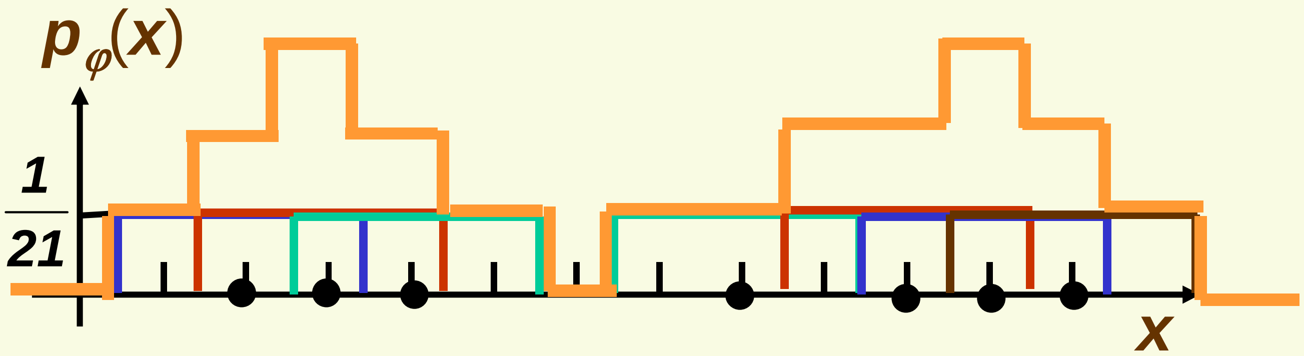


- To see what the function looks like, we need to generate 7 boxes and add them up
- The width is ***h***=3 and the height is

$$\frac{1}{nh^d} = \frac{1}{21}$$

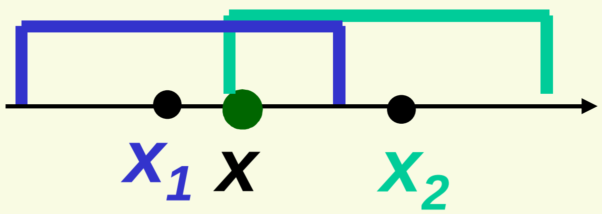
Parzen Windows: Interpolation

- In essence, window function φ is used for interpolation: each sample $\boldsymbol{x_i}$ contributes to the resulting density at $\boldsymbol{x}$ if $\boldsymbol{x}$ is close enough to $\boldsymbol{x_i}$



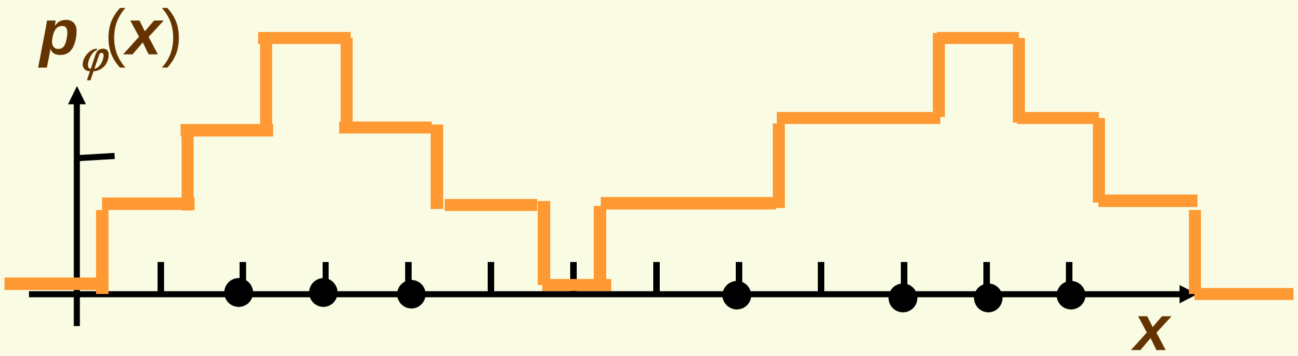
Parzen Windows: Drawbacks of Hypercube φ

- As long as sample point $\mathbf{x}_i$ and $\mathbf{x}$ are in the same hypercube, the contribution of $\mathbf{x}_i$ to the density at $\mathbf{x}$ is constant, regardless of how close $\mathbf{x}_i$ is to $\mathbf{x}$



$$\varphi\left(\frac{\mathbf{x}-\mathbf{x}_1}{h}\right)=\varphi\left(\frac{\mathbf{x}-\mathbf{x}_2}{h}\right)=\mathbf{1}$$

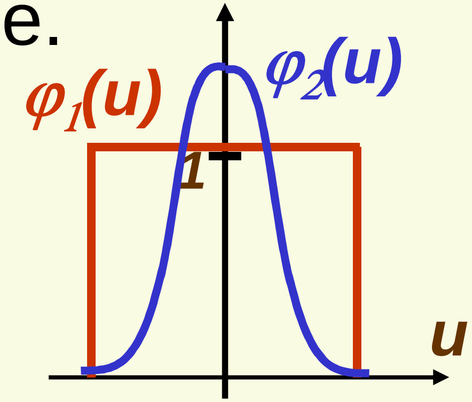
- The resulting density $p_\varphi(\mathbf{x})$ is not smooth, it has discontinuities



Parzen Windows: general φ

$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

- We can use a general window φ as long as the resulting $p_\varphi(x)$ is a legitimate density, i.e.



1. $p_\varphi(u) \geq 0$
   - satisfied if $\varphi(u) \geq 0$
2. $\int p_\varphi(x)dx = 1$
   - satisfied if $\int \varphi(u)du = 1$

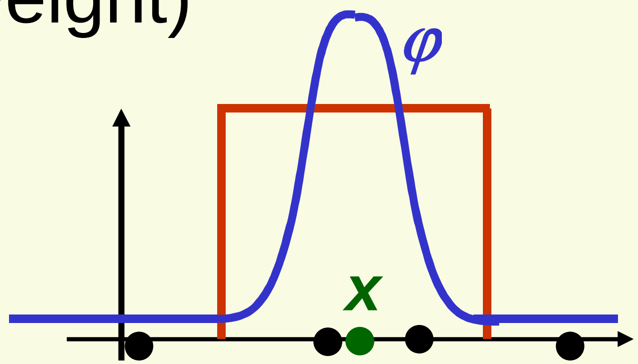
$$\int p_\varphi(x)dx = \frac{1}{nh^d}\sum_{i=1}^{i=n}\int \varphi\left(\frac{x - x_i}{h}\right)dx = \frac{1}{nh^d}\sum_{i=1}^{n}\int h^d\varphi(u)du = 1$$

change coordinates to $u = \dfrac{x - x_i}{h}$, thus $du = \dfrac{dx}{h}$

Parzen Windows: general φ

$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

- Notice that with the general window φ we are no longer counting the number of samples inside $\mathcal{R}$.
- We are counting the weighted average of potentially every single sample point (although only those within distance ***h*** have any significant weight)



- With infinite number of samples, and appropriate conditions, it can still be shown that

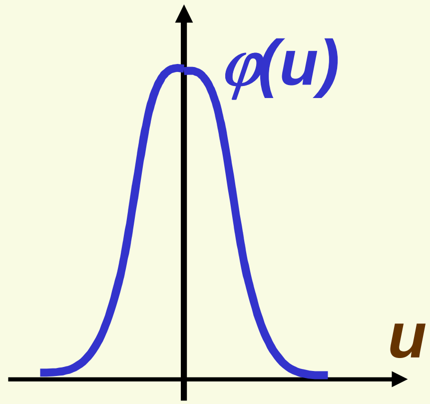
$$p_\varphi^n(x) \to p(x)$$

Parzen Windows: Gaussian φ

$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

- A popular choice for φ is $N(0,1)$ density

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

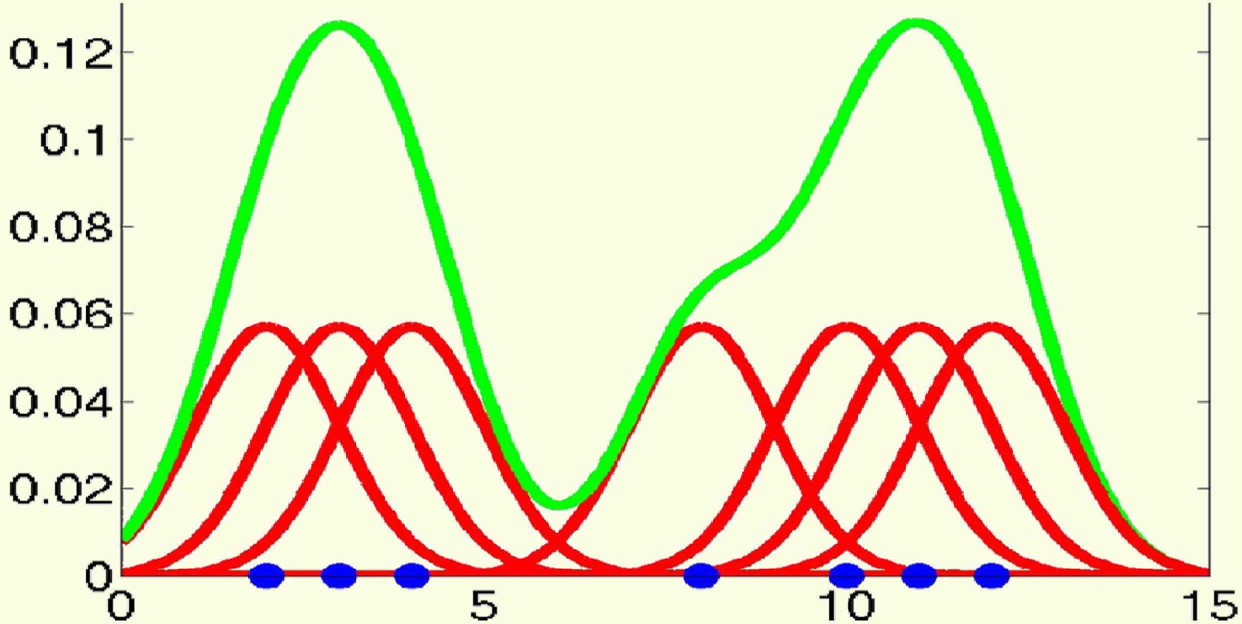


- Solves both drawbacks of the "box" window
  - Points x which are close to the sample point x_i receive higher weight
  - Resulting density $p_\varphi(x)$ is smooth

Parzen Windows: Example with General φ

- Let's come back to our example
  - 7 samples ***D***={2,3,4,8,10,11,12}, ***h***=1

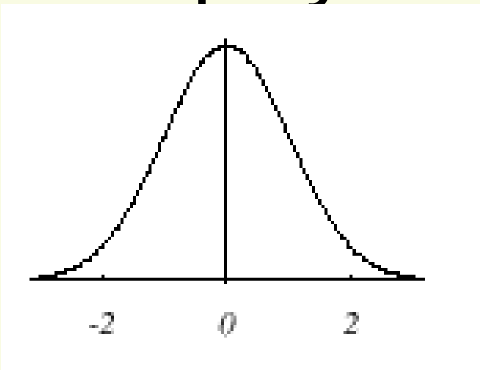
$$p_\varphi(x) = \frac{1}{7}\sum_{i=1}^{i=7}\varphi(x - x_i)$$



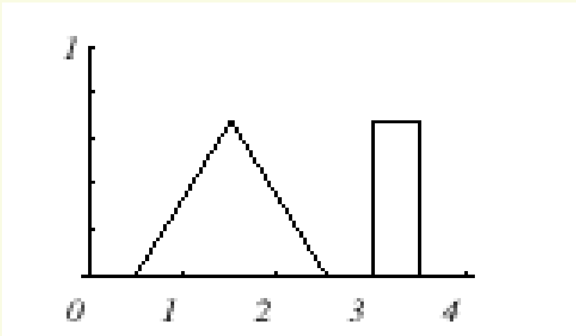
- $p_\varphi(x)$ is the sum of of 7 Gaussians, each centered at one of the sample points, and each scaled by 1/7

Parzen Windows: Did We Solve the Problem?

- Let's test if we solved the problem
  1. Draw samples from a known distribution
  2. Use our density approximation method and compare with the true density
- We will vary the number of samples ***n*** and the window size ***h***
- We will play with 2 distributions

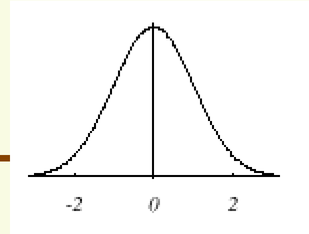


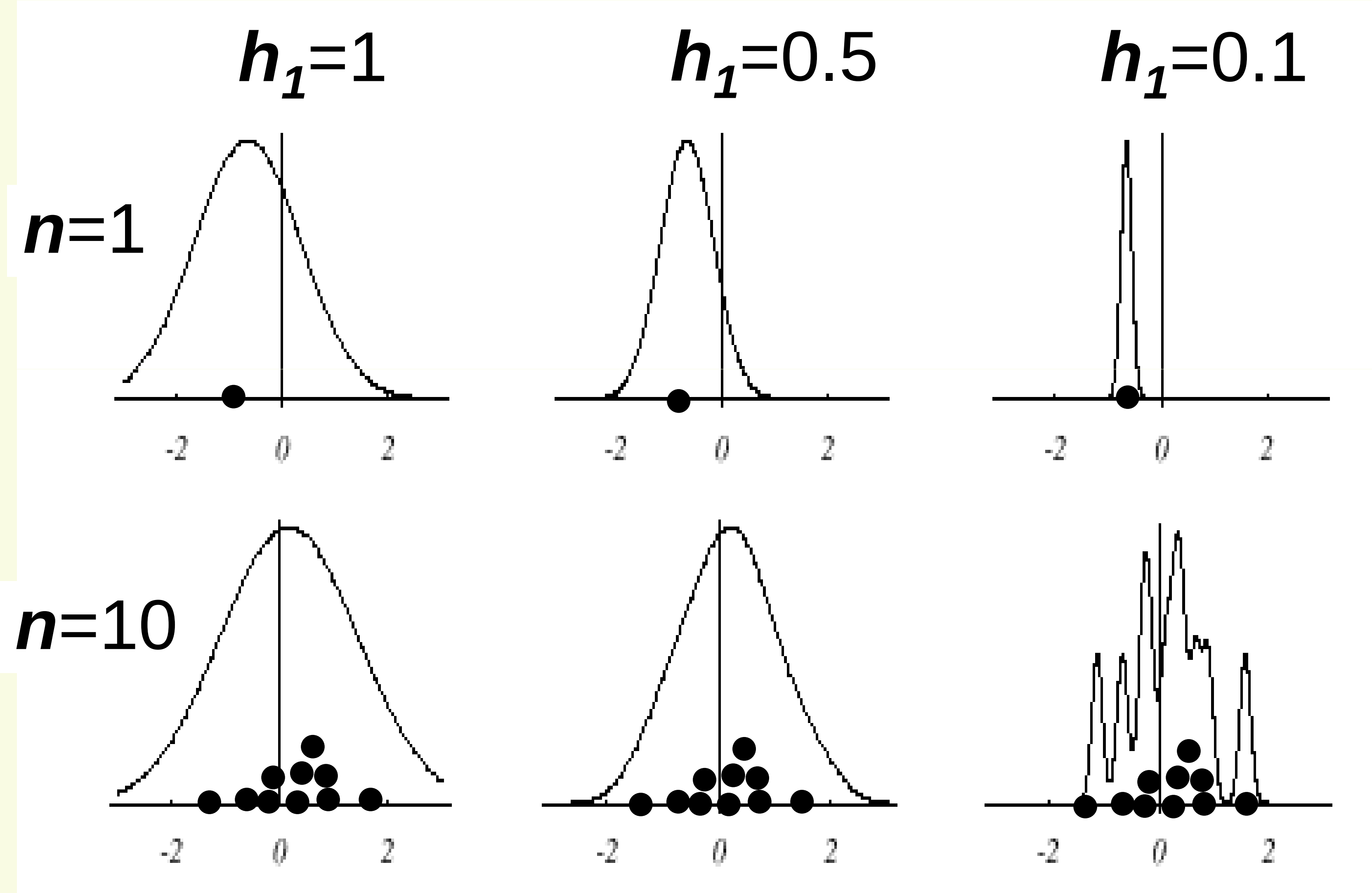
***N*(*0*,*1*)**



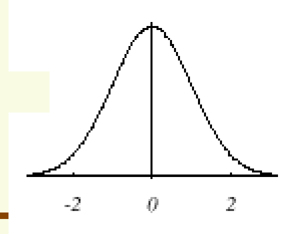
triangle and uniform mixture

Parzen Windows: True Density N(0,1)





Parzen Windows: True Density N(0,1)



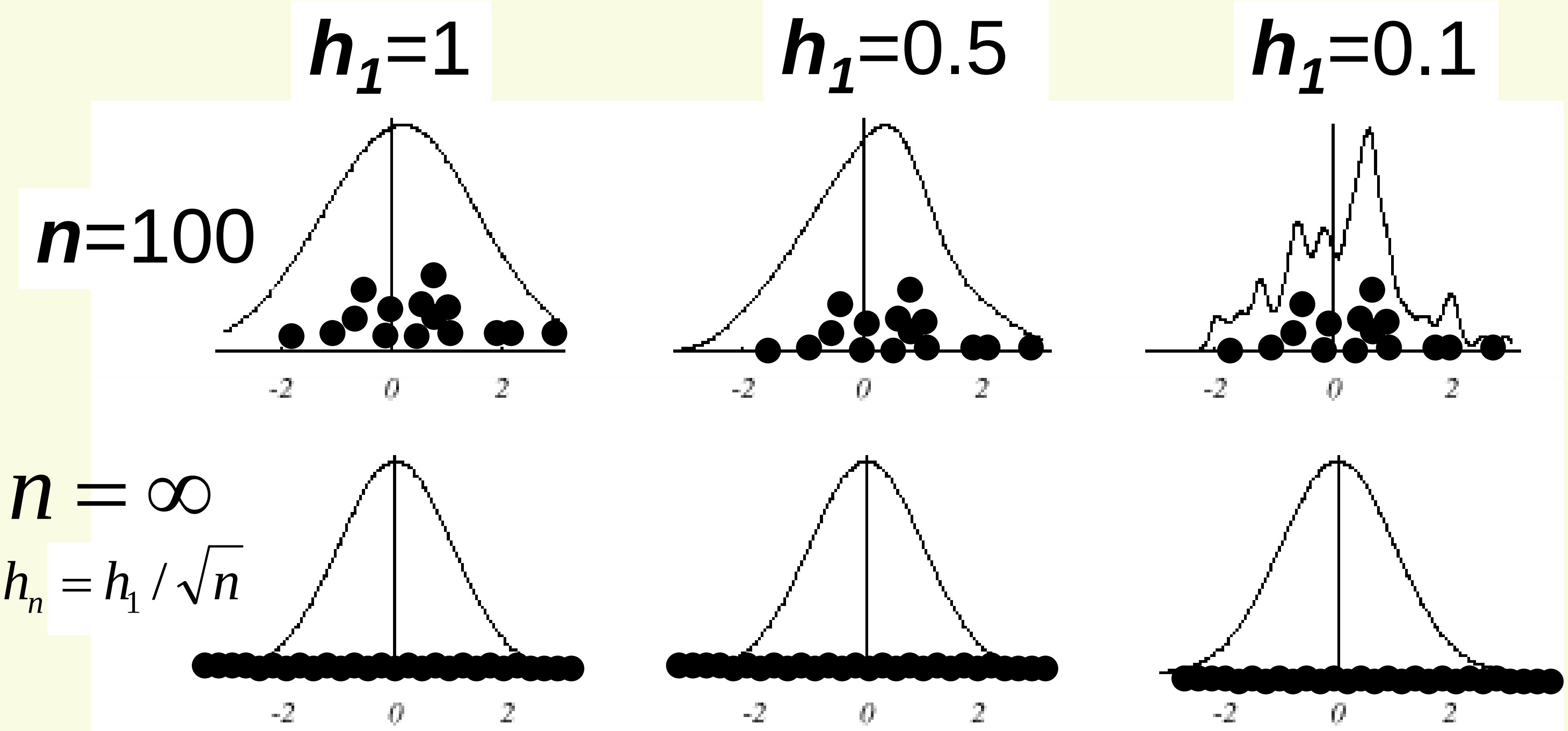
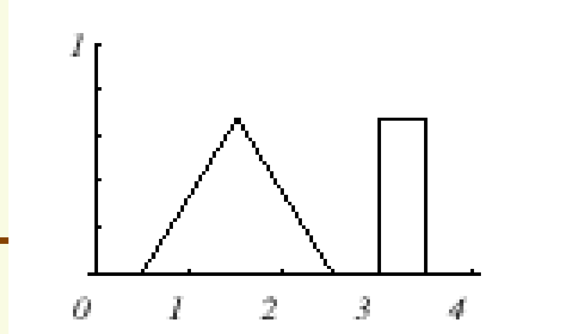
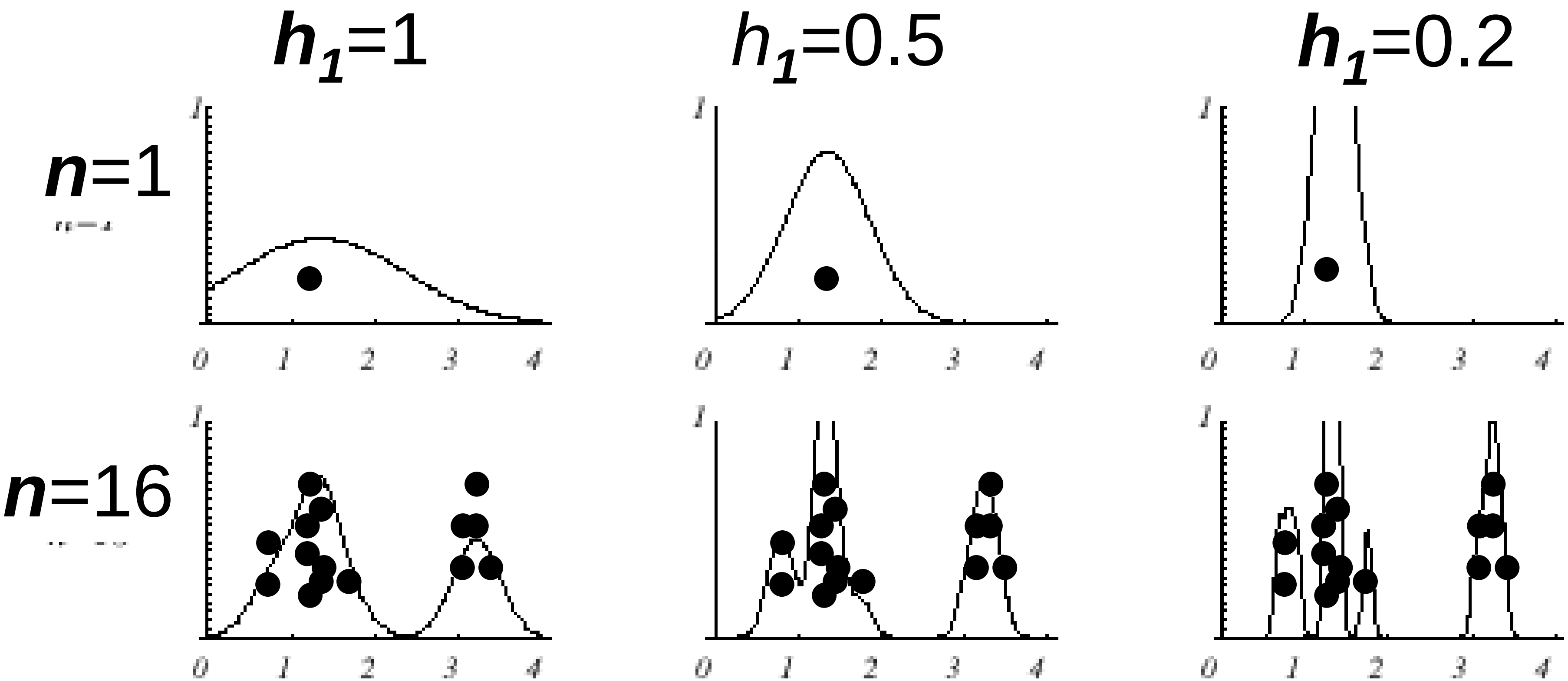


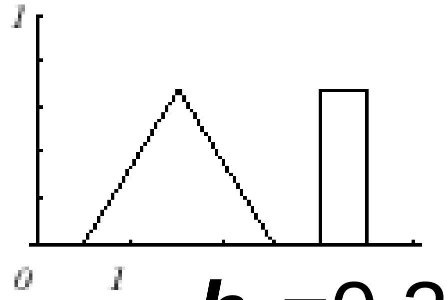
FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard

Parzen Windows: True density is Mixture of Uniform and Triangle





Parzen Windows: True density is Mixture of Uniform and Triangle



h_1=1 **h_1**=0.5 **h_1**=0.2

n=256

$n = \infty$

$h_n = h_1 / \sqrt{n}$

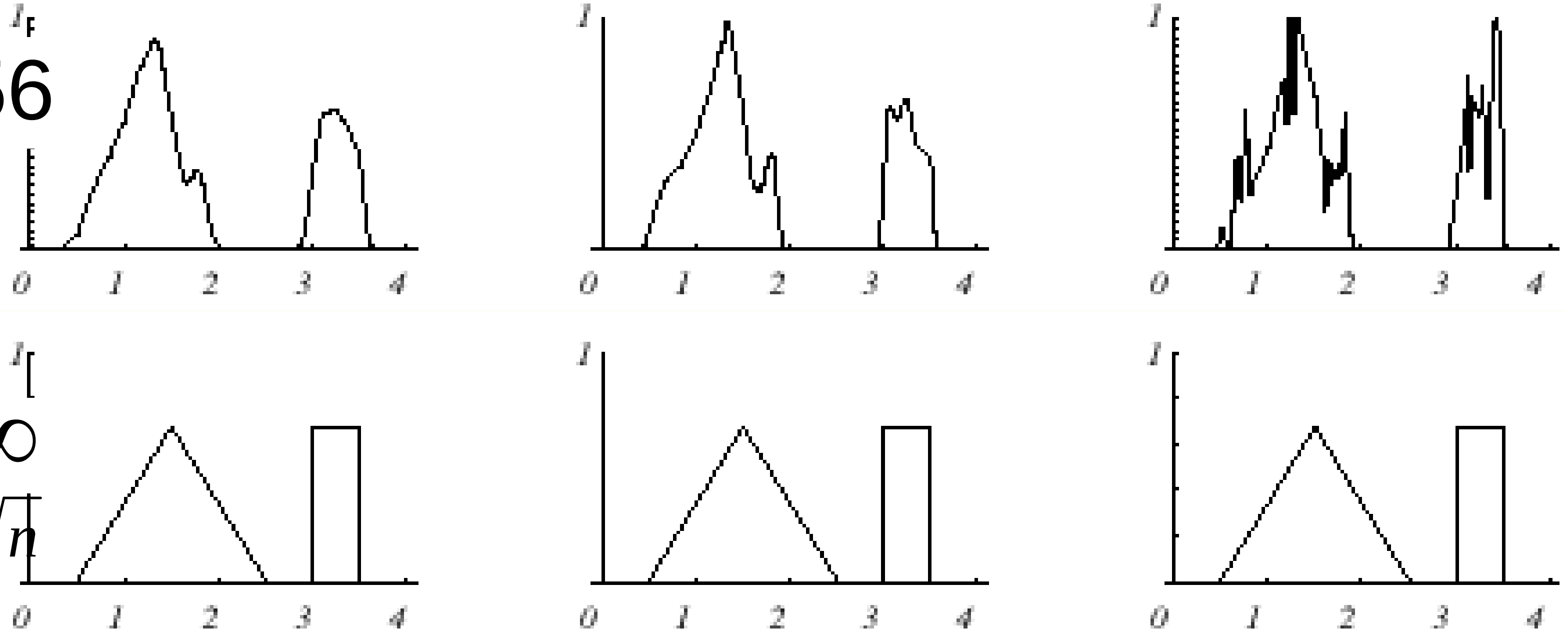
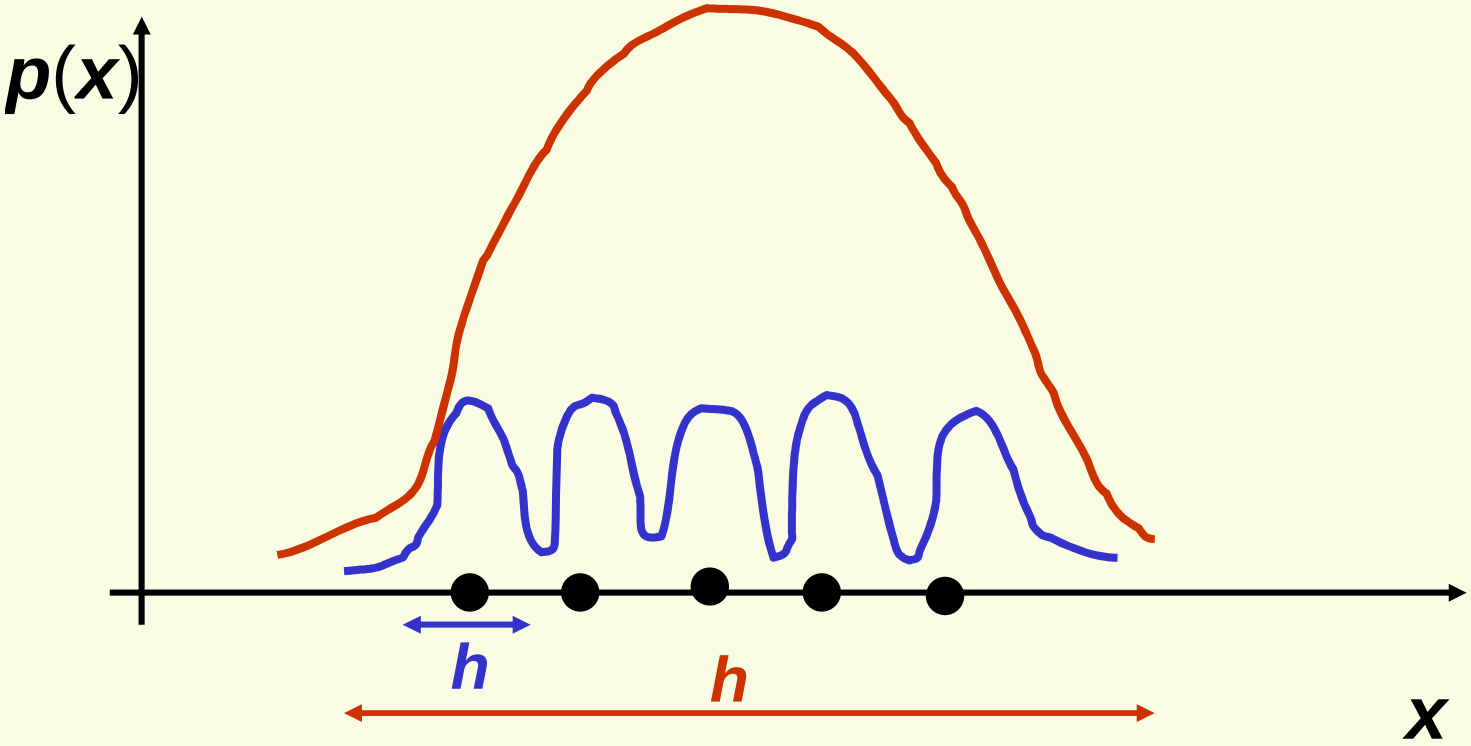


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows: Effect of Window Width h

- By choosing ***h*** we are guessing the region where density is approximately constant
- Without knowing anything about the distribution, it is really hard to guess were the density is approximately constant



Parzen Windows: Effect of Window Width h

- If ***h*** is small, we superimpose ***n*** sharp pulses centered at the data
  - Each sample point x_i influences too small range of x
  - Smoothed too little: the result will look noisy and not smooth enough
- If ***h*** is large, we superimpose broad slowly changing functions,
  - Each sample point x_i influences too large range of x
  - Smoothed too much: the result looks oversmoothed or “out-of-focus”
- Finding the best ***h*** is challenging, and indeed no single ***h*** may work well
  - May need to adapt ***h*** for different sample points
- However we can try to learn the best ***h*** to use from our labeled data

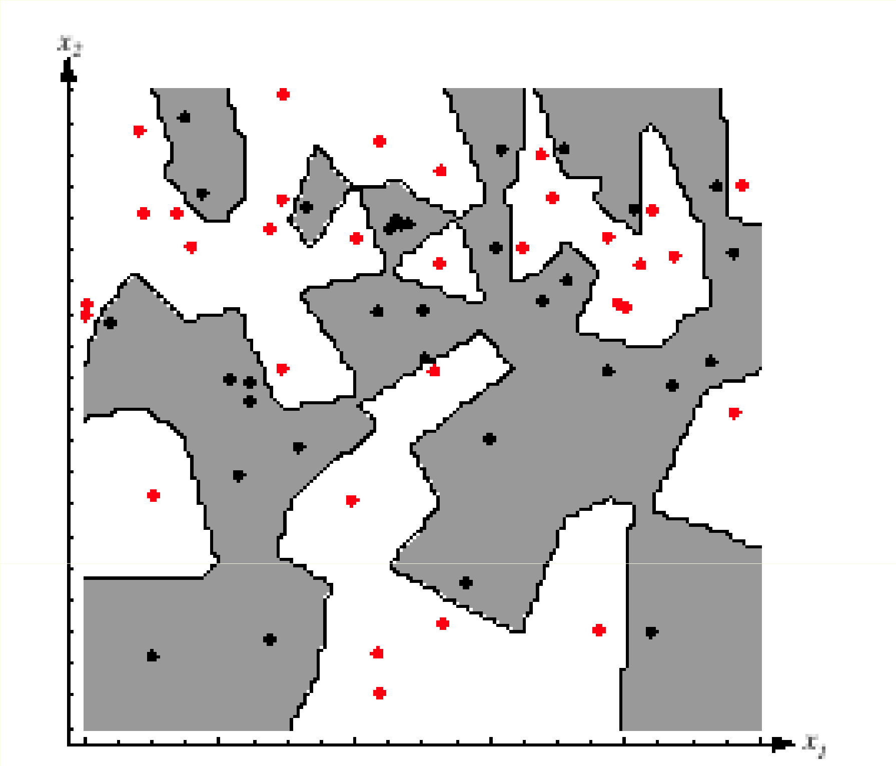
Learning window width *h* From Labeled Data

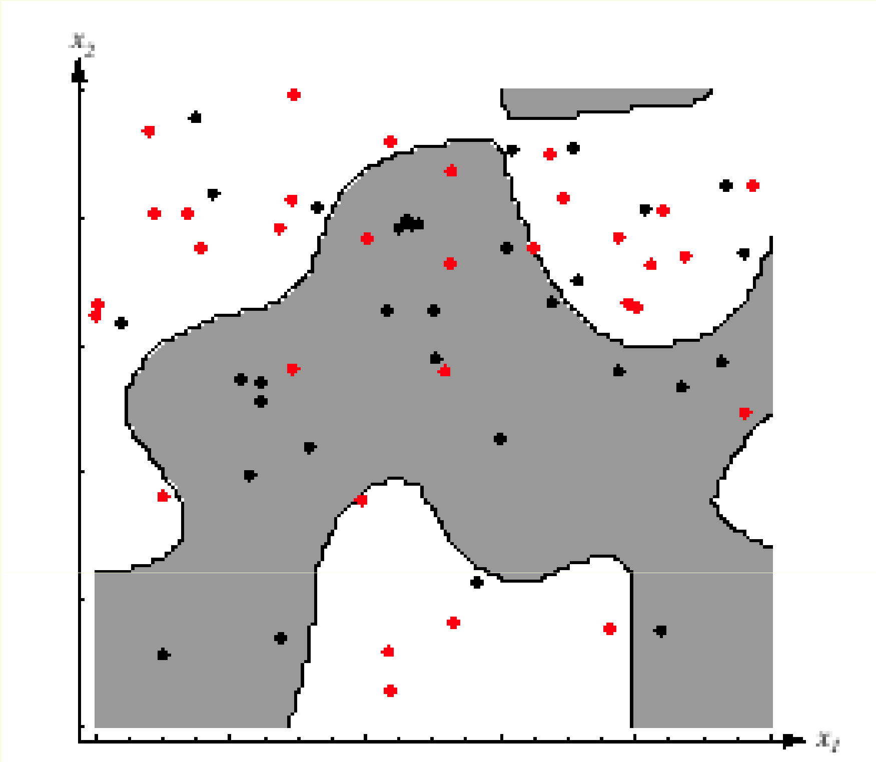
- Divide labeled data into *training* set, *validation* set, *test* set
- For a range of different values of *h* (possibly using binary search), construct density estimate p(x) using Parzen windows
- Test the classification performance on the ***validation*** set for each value of *h* you tried
- For the final density estimate, choose *h* giving the smallest error on the ***validation*** set
- Now you can test the performance of the classifier on the test set
  - Notice we need validation set to find best parameter h, we can't use test set for this because test set cannot be used for training
  - In general, need validation set if our classifier has some tunable parameters

Parzen Windows: 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 as illustrated in the following figure

Parzen Windows: Classification Example





- **For small enough window size *h* the classification on training data is perfect**
- **However decision boundaries are complex and this solution is not likely to generalize well to novel data**

- **For larger window size *h*, classification on training data is not perfect**
- **However decision boundaries are simpler and this solution is more likely to generalize well to novel data**

Parzen Windows: Summary

- Advantages
  - Can be applied to the data from any distribution
  - In theory can be shown to converge as the number of samples goes to infinity
- Disadvantages
  - Number of training data is limited in practice, and so choosing the appropriate window size ***h*** is difficult
  - May need large number of samples for accurate estimates
  - Computationally heavy, to classify one point we have to compute a function which potentially depends on all samples

$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

  - **But we need a lot of samples for accurate density estimation!**