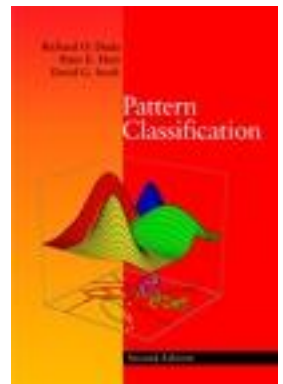


Chapter 4 (Part 1): Non-Parametric Classification

- Introduction
- Density Estimation
- Parzen Windows



- Introduction

- All Parametric densities are unimodal (have a single local maximum), whereas many practical problems involve multi-modal densities.

Nonparametric procedures can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known.

- There are two types of nonparametric methods:
 - Estimating $P(\mathbf{x}|\omega_j)$
 - Bypass probability and go directly to *a*-posteriori probability estimation

Density Estimation

- Basic idea:

Probability that a vector \mathbf{x} will fall in region \mathbf{R} is:

$$P = \int_{\mathfrak{R}} p(\mathbf{x}') d\mathbf{x}' \quad (1)$$

P is a smoothed (or averaged) version of the density function $p(\mathbf{x})$ if we have a sample of size n ; therefore, the probability that k points fall in \mathbf{R} is then:

$$P_k = \text{BIN}(k | n, P) = \binom{n}{k} P^k (1-P)^{n-k} = \frac{n!}{k!(n-k)!} P^k (1-P)^{n-k} \quad (2)$$

Prob. that the rest are not

No. of unique splits k vs $n-k$

Prob. that k of particular x -es are in R

and the expected and variance value for k is:

$$E(k) = nP, \text{Var}(k) = nP(1-P) \quad (3)$$

What is ML estimation of $P = \theta$?

$$\nabla_P \ln(P_k) = \nabla_P (\ln \binom{n}{k} + k \ln(P) + (n-k) \ln(1-P)) = \frac{k}{P} - \frac{n-k}{1-P} = 0$$

→ $Max_{\theta}(P_k | \theta)$ is reached for $\hat{\theta} = \frac{k}{n} \cong P$ (4)

- Therefore, the ratio k/n is a good estimate for the probability P and hence for the density function p .
- If $p(\mathbf{x})$ is continuous and that the region R is so small that p does not vary significantly within it, we can write:

$$\int_{\mathfrak{R}} p(\mathbf{x}') d\mathbf{x}' \cong p(\mathbf{x})V \quad (5)$$

where \mathbf{x} is a point within R and V the volume enclosed by R .

Combining equation (1), (4) and (5) yields:

$$p(\mathbf{x}) \cong \frac{k/n}{V} \quad 4$$

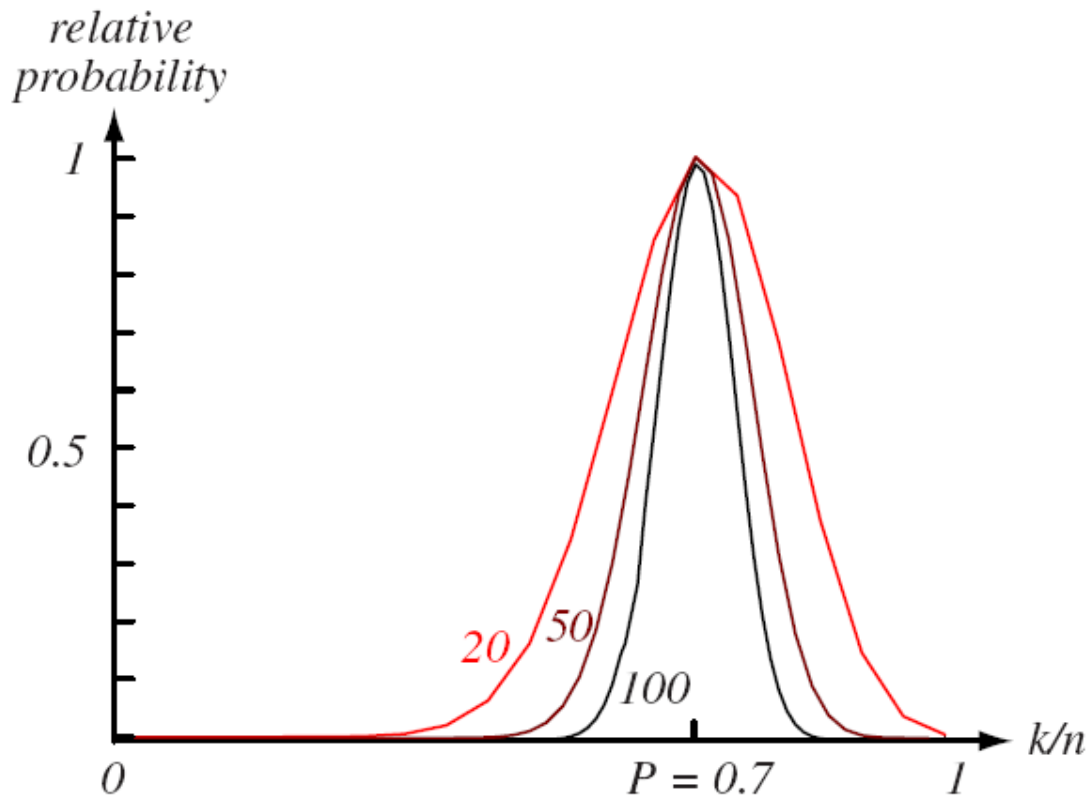
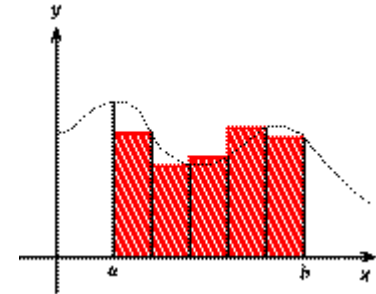


FIGURE 4.1. The relative probability an estimate given by Eq. 4 will yield a particular value for the probability density, here where the true probability was chosen to be 0.7. Each curve is labeled by the total number of patterns n sampled, and is scaled to give the same maximum (at the true probability). The form of each curve is binomial, as given by Eq. 2. For large n , such binomials peak strongly at the true probability. In the limit $n \rightarrow \infty$, the curve approaches a delta function, and we are guaranteed that our estimate will give the true probability.

- Density Estimation (cont'd)

- Justification of equation (5)

$$\int_{\mathcal{R}} p(\mathbf{x}') d\mathbf{x}' \cong p(\mathbf{x})V \quad (5)$$



We assume that $p(\mathbf{x})$ is continuous and that region R is so small that p does not vary significantly within R . Since $p(\mathbf{x}) = \text{constant}$, it is not a part of the sum.

$$\int_{\mathfrak{R}} p(\mathbf{x}') d\mathbf{x}' = p(\mathbf{x}') \int_{\mathfrak{R}} d\mathbf{x}' = p(\mathbf{x}') \mu(\mathfrak{R})$$

Where: $\mu(R)$ is: a length in R
a surface in R^2
a volume in R^3
a hypervolume in R^n

Since $p(\mathbf{x}) \cong p(\mathbf{x}') = \text{constant}$, therefore in R^3 :

$$\int_{\mathfrak{R}} p(\mathbf{x}') d\mathbf{x}' \cong p(\mathbf{x}) \cdot V$$

$$\text{and } p(\mathbf{x}) \cong \frac{k}{nV}$$

- Condition for convergence

The fraction $k/(nV)$ is a space averaged value of $p(\mathbf{x})$.
 $p(\mathbf{x})$ is obtained only if V approaches zero.

$$\lim_{V \rightarrow 0, k=0} p(\mathbf{x}) = 0 \quad (\text{if } n = \text{fixed})$$

This is the case where no samples are included in R : it is an uninteresting case!

$$\lim_{V \rightarrow 0, k \neq 0} p(\mathbf{x}) = \infty$$

In this case, the estimate diverges: it is an uninteresting case!

The volume V needs to approach 0 anyway if we want to use this estimation

- Practically, V cannot be allowed to become small since the number of samples is always limited
- One will have to accept a certain amount of variance in the ratio k/n
- Theoretically, if an unlimited number of samples is available, we can circumvent this difficulty

To estimate the density of \mathbf{x} , we form a sequence of regions

R_1, R_2, \dots containing \mathbf{x} : the first region contains one sample, the second two samples and so on.

Let V_n be the volume of R_n , k_n the number of samples falling in R_n and $p_n(\mathbf{x})$ be the n^{th} estimate for $p(\mathbf{x})$:

$$p_n(\mathbf{x}) \cong \frac{k_n / n}{V_n} \quad (7)$$

Three necessary conditions should apply if we want $p_n(\mathbf{x})$ to converge to $p(\mathbf{x})$:

1) $\lim_{n \rightarrow \infty} V_n = 0$ smaller and smaller regions

2) $\lim_{n \rightarrow \infty} k_n = \infty$ a huge # of observations should be in R

3) $\lim_{n \rightarrow \infty} k_n / n = 0$ # of observations should be a very small fraction of total

There are two different ways of obtaining sequences of regions that satisfy these conditions:

(a) Fix the volume V_n and determine k_n from the data. Shrink an initial region where $V_n = 1/\sqrt[n]{n}$ and show that

$$p_n(x) \xrightarrow{n \rightarrow \infty} p(x)$$

This is called “the Parzen-window estimation method”

(b) Fix the value of k_n and determine the corresponding volume V_n from the data. Specify k_n as some function of n , such as $k_n = \sqrt[n]{n}$; the volume V_n is grown until it encloses k_n neighbors of \mathbf{x} . This is called “the k_n -nearest neighbor estimation method”

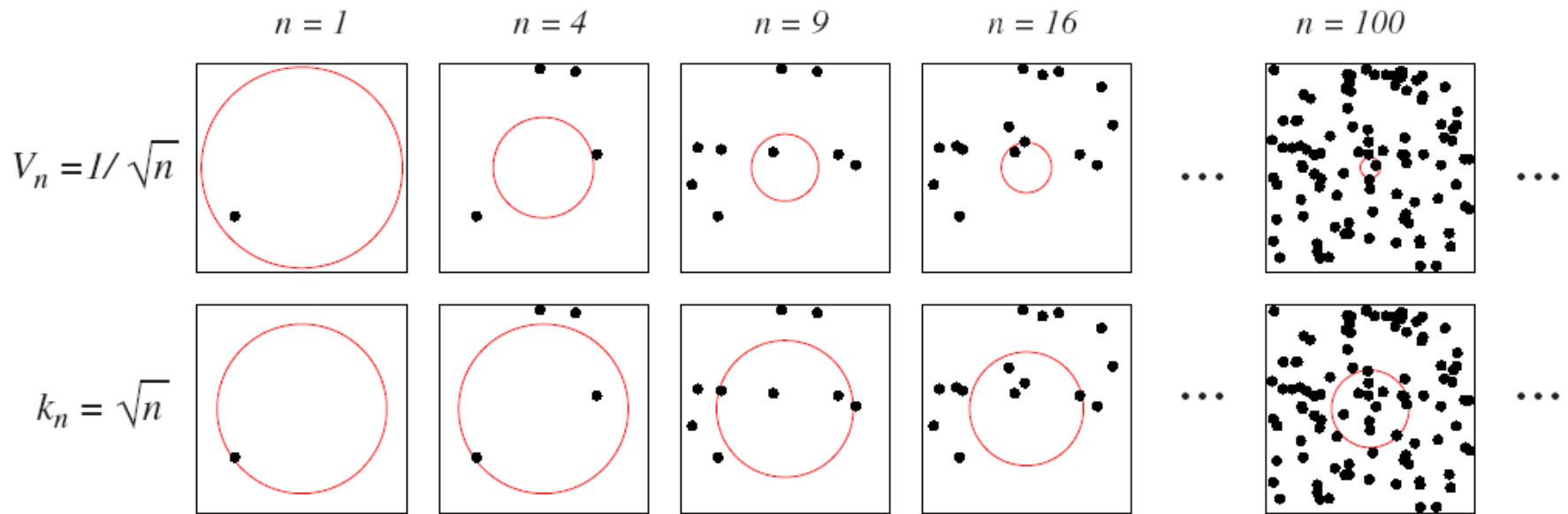


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the bottom row, is to decrease the volume in a data-dependent way, for instance letting the volume enclose some number $k_n = \sqrt{n}$ of sample points. The sequences in both cases represent random variables that generally converge and allow the true density at the test point to be calculated.

- Parzen Windows (Kernel Density Estimation)

- Parzen-window approach to estimate densities
assume that the region R_n is a d -dimensional hypercube

$$V_n = h_n^d \quad (h_n : \text{length of the edge of } \mathfrak{R}_n)$$

Let $\phi(\mathbf{u})$ be the following window function (top hat):

$$\phi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq \frac{1}{2} \quad j = 1, \dots, d \\ 0 & \text{otherwise} \end{cases}$$

- $\phi((\mathbf{x}-\mathbf{x}_i)/h_n)$ is equal to unity if \mathbf{x}_i falls within the hypercube of volume V_n centered at \mathbf{x} and equal to zero otherwise.

- The number of samples in this hypercube is:

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

By substituting k_n in equation (7), we obtain the following estimate:

$$p_n(\mathbf{x}) \cong \frac{k_n / n}{V_n} \quad \Rightarrow \quad p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{i=n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

Parzen windows estimate

$p_n(\mathbf{x})$ estimates $p(\mathbf{x})$ as an average of functions of \mathbf{x} and the samples (\mathbf{x}_i) ($i=1, \dots, n$).

- The density estimate is a superposition of kernel functions and the samples \mathbf{x}_i .

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

- $\varphi(\mathbf{u})$ interpolates the density between samples.
- Each sample \mathbf{x}_i contributes to the estimate based on its distance from \mathbf{x} .

Properties of $\varphi(\mathbf{u})$

- The kernel function $\varphi(\mathbf{u})$ can have a more general form (i.e., not just hypercube).
- In order for $p_n(\mathbf{x})$ to be a legitimate estimate (nonnegative and integrate to one) $\varphi(\mathbf{u})$, must be a valid density itself:

$$\varphi(\mathbf{u}) \geq 0$$

$$\int \varphi(\mathbf{u}) d\mathbf{u} = 1$$

Effect of the *window width* h_n on $p_n(\mathbf{x})$

If we define the function $\delta_n(\mathbf{x})$ by

$$\delta_n(\mathbf{x}) = \frac{1}{V_n} \varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

then we can write $p_n(\mathbf{x})$ as the average

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \delta_n(\mathbf{x} - \mathbf{x}_i).$$

Since $V_n = h_n^d$, h_n clearly affects both the amplitude and the width of $\delta_n(\mathbf{x})$ (Fig. 4.3).

Thus, as h_n approaches zero, $\delta_n(\mathbf{x} - \mathbf{x}_i)$ approaches a Dirac delta function centered at \mathbf{x}_i , and $p_n(\mathbf{x})$ approaches a superposition of delta functions centered at the samples.

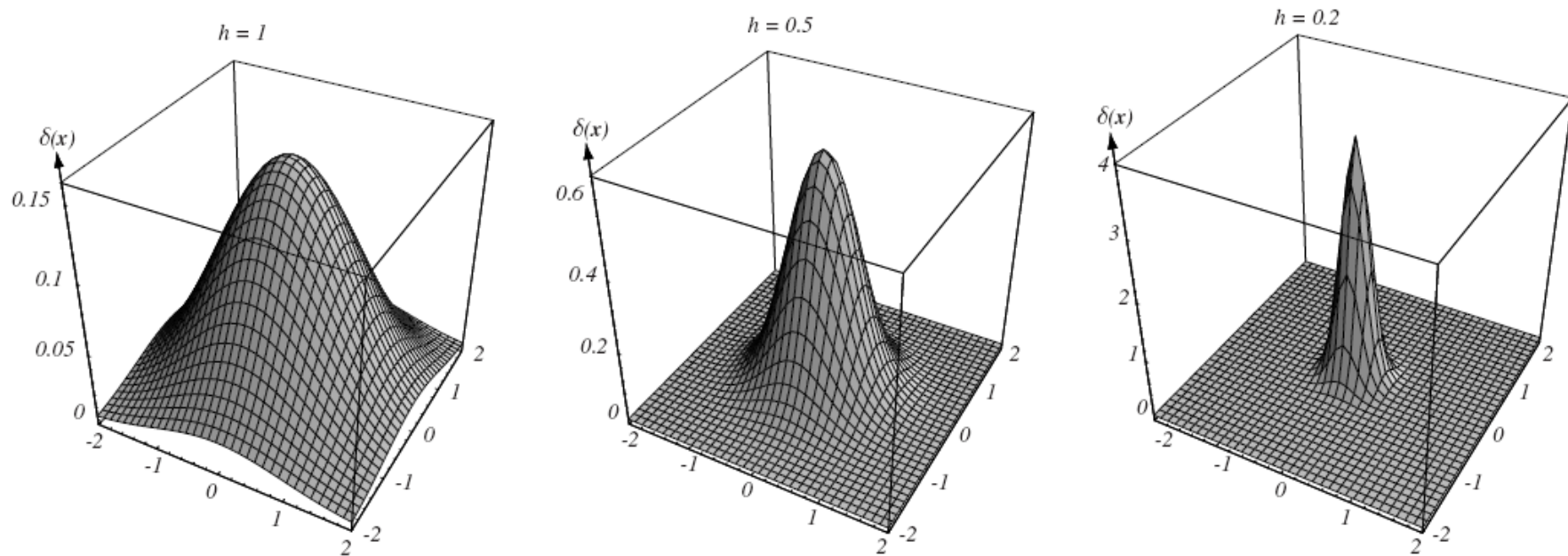


FIGURE 4.3. Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of h . Note that because the $\delta(\mathbf{x})$ are normalized, different vertical scales must be used to show their structure.

If h_n is very large, the amplitude of δ_n is small, and $p_n(\mathbf{x})$ is the superposition of n broad, slowly changing functions and is a very smooth “out-of-focus” estimate of $p(\mathbf{x})$.

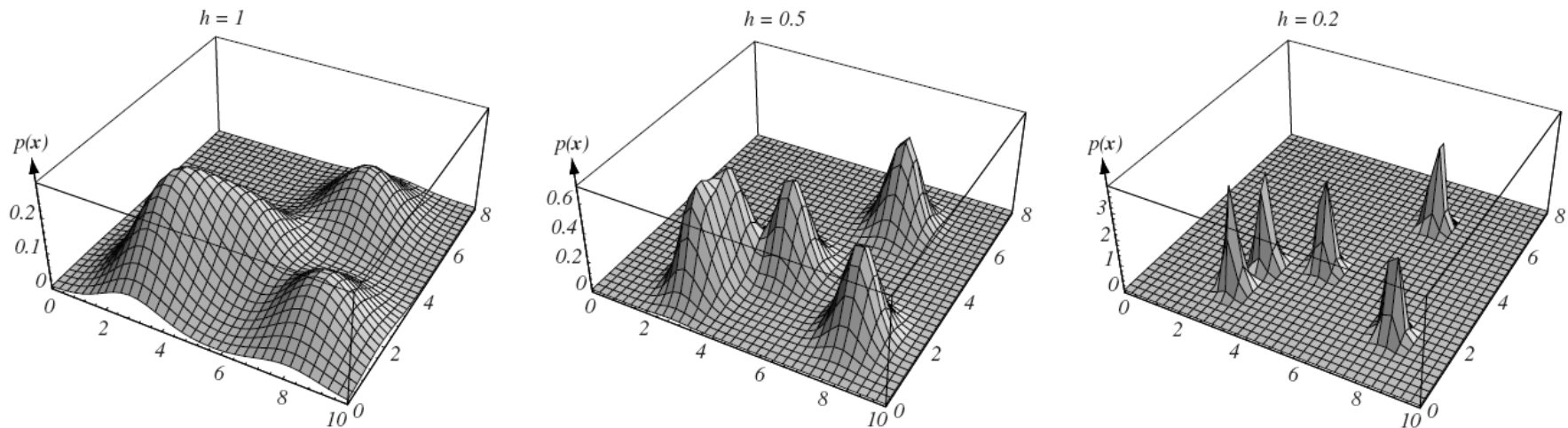


FIGURE 4.4. Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution.

If h_n is very small, the peak value of $\delta_n(\mathbf{x}-\mathbf{x}_i)$ is large and occurs near $\mathbf{x} = \mathbf{x}_i$. In this case $p(\mathbf{x})$ is the superposition of n sharp pulses centered at the samples — an erratic, “noisy” estimate.

If V_n is too large, the estimate will suffer from too little resolution; if V_n is too small, the estimate will suffer from too much statistical variability. With a limited number of samples, the best we can do is to seek some acceptable compromise. However, with an unlimited number of samples, it is possible to let V_n slowly approach zero as n increases and have $p_n(\mathbf{x})$ converge to the unknown density $p(\mathbf{x})$.

$$\lim_{n \rightarrow \infty} \bar{p}_n(\mathbf{x}) = p(\mathbf{x})$$

conditions

$$\lim_{n \rightarrow \infty} \sigma_n^2(\mathbf{x}) = 0.$$

- $\varphi(\mathbf{u})$ must be well-behaved.
- $V_n \rightarrow 0$ at a rate lower than $1/n$


Refer to the textbook for the proof of the convergence of the mean and variance

Expected Value/Variance of estimate $p_n(\mathbf{x})$

- The expected value of the estimates approaches $p(\mathbf{x})$ as $V_n \rightarrow 0$:

$$E[p_n(\mathbf{x})] = \frac{1}{n} \sum_{i=1}^n E\left[\frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)\right] = \int \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{v}}{h_n}\right) p(\mathbf{v}) d\mathbf{v} = \int \delta_n(\mathbf{x} - \mathbf{v}) p(\mathbf{v}) d\mathbf{v}.$$

convolution with true density



- The variance of the estimate is given by:

$$\text{Var}[p_n(\mathbf{x})] \leq \frac{\sup(\varphi(\cdot)) E[p_n(\mathbf{x})]}{nV_n}$$

- The variance can be decreased by allowing

$$nV_n \rightarrow \infty \quad (\text{e.g., } V_n = 1/\sqrt{n})$$

- **Illustration**

- The behavior of the Parzen-window method

- Case where $p(\mathbf{x}) \rightarrow N(0,1)$

Let $\varphi(u) = (1/\sqrt{2\pi}) \exp(-u^2/2)$ and $h_n = h_1/\sqrt{n}$ ($n > 1$)

(h_1 : known parameter)

Thus:
$$p_n(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_n} \phi\left(\frac{x - x_i}{h_n}\right)$$

is an average of normal densities centered at the samples x_i .

Numerical results: For $n = 1$ and $h_1 = 1$

$$p_1(x) = \phi(x - x_1) = \frac{1}{\sqrt{2\pi}} e^{-1/2(x - x_1)^2} \rightarrow N(x_1, 1)$$

For $n = 10$ and $h = 0.1$, the contributions of the individual samples are clearly observable !

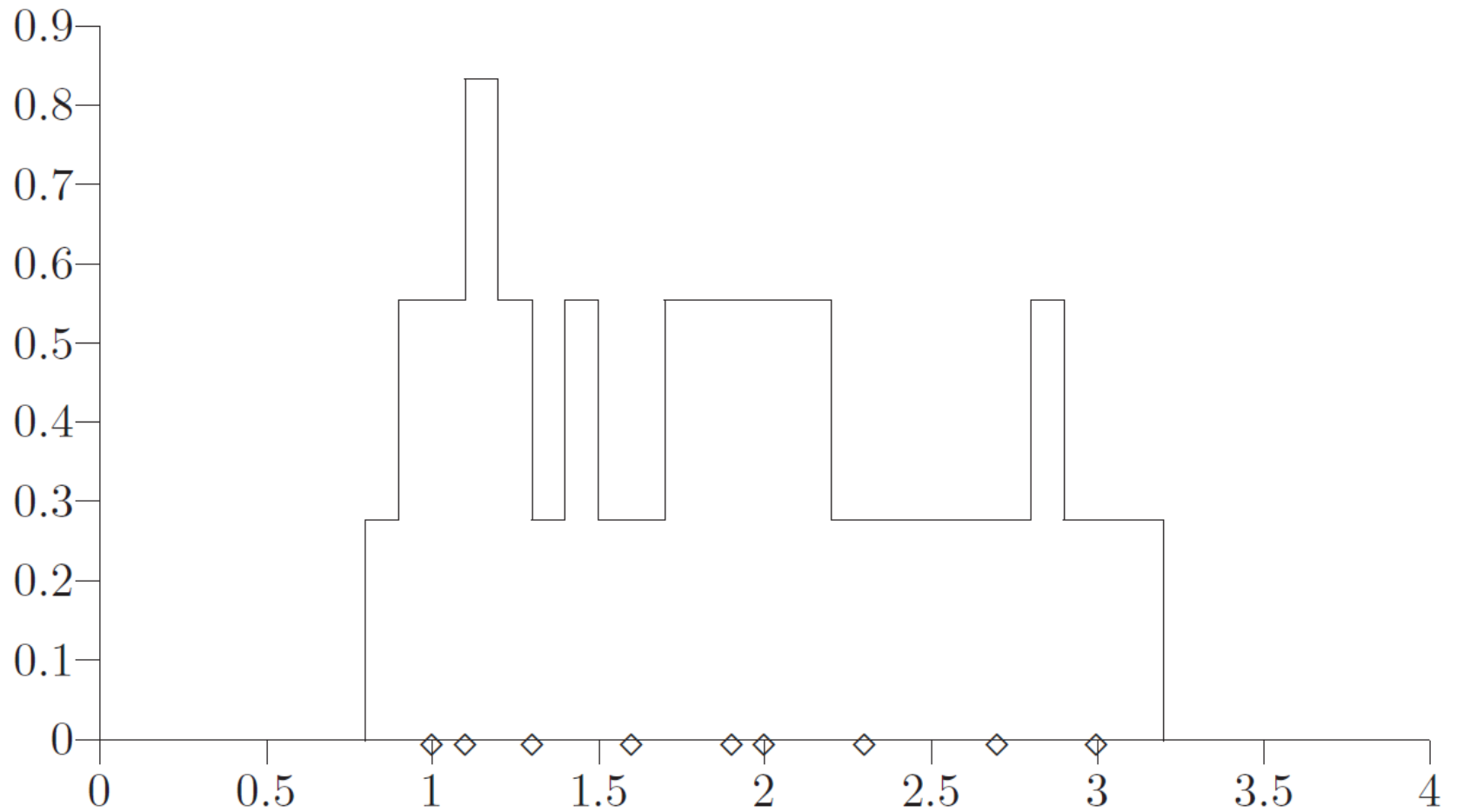


Figure 4.26 Probability density estimate with top hat kernel, $h = 0.2$.

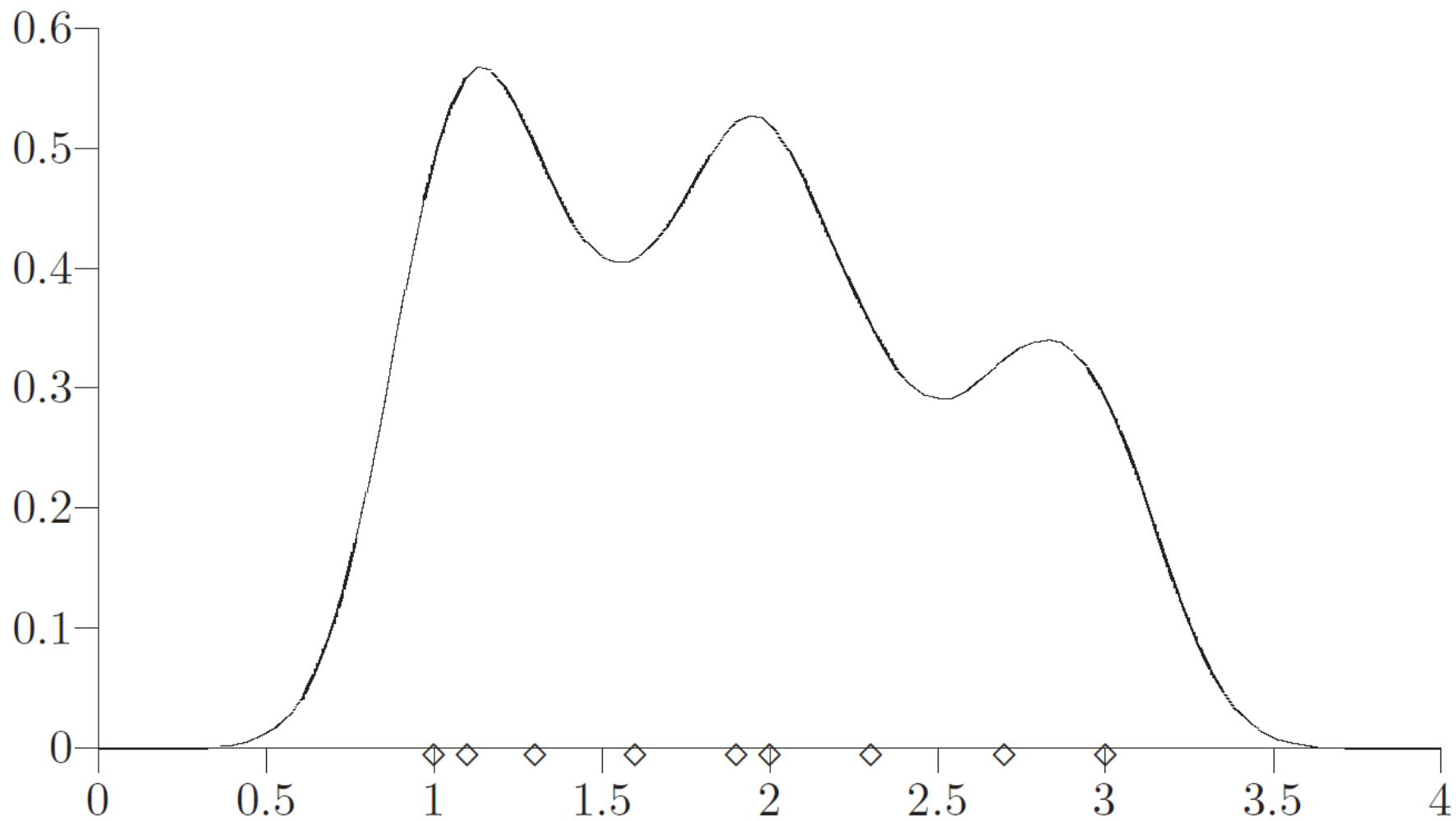
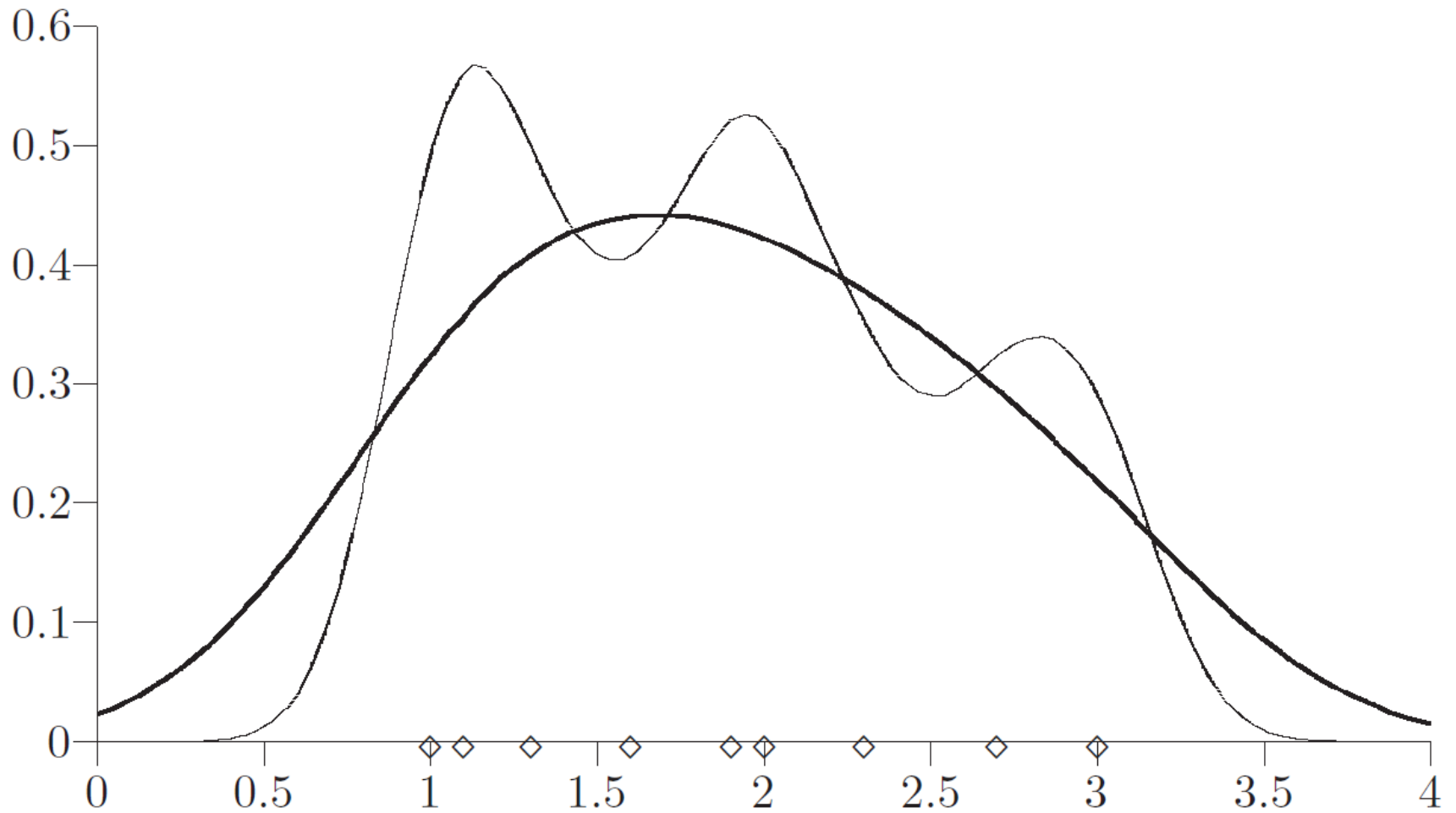


Figure 4.27 Probability density figure with Gaussian kernel and $h = 0.2$.

Table 4.4 Commonly used kernel functions for univariate data.

Kernel function	Analytic form, $K(x)$
Rectangular (or top hat)	$\frac{1}{2}$ for $ x < 1$, 0 otherwise
Triangular	$1 - x $ for $ x < 1$, 0 otherwise
Biweight (or Quartic)	$\frac{15}{16}(1 - x^2)^2$ for $ x < 1$, 0 otherwise
Normal (or Gaussian)	$\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$
Bartlett–Epanechnikov	$\frac{3}{4}(1 - x^2/5)/\sqrt{5}$ for $ x < \sqrt{5}$, 0 otherwise



Probability density with different levels of smoothing ($h = 0.2$ and $h = 0.5$).

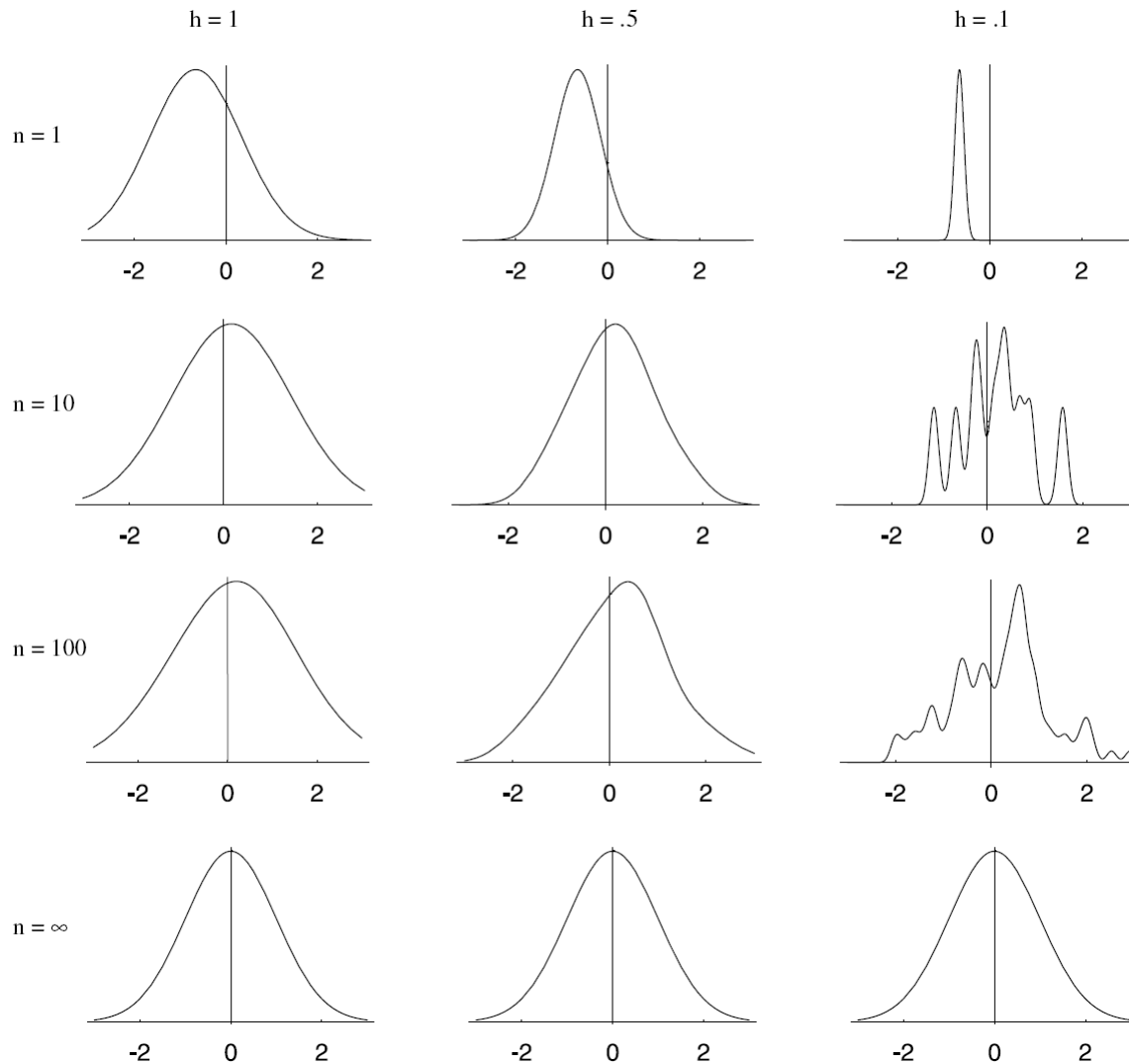


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless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

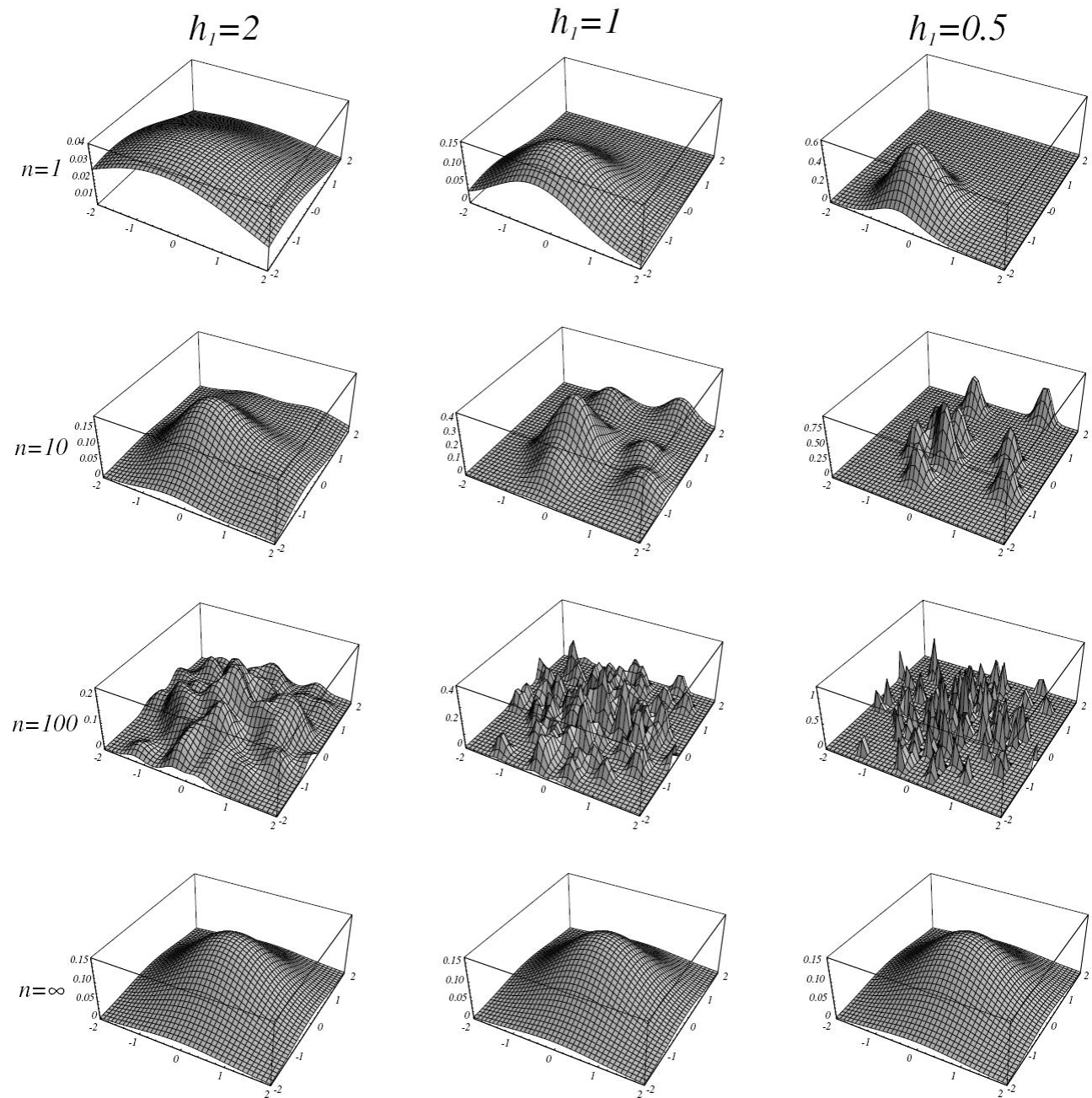


FIGURE 4.6. Parzen-window estimates of a bivariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n=\infty$ estimates are the same (and match the true distribution), regardless of window width.

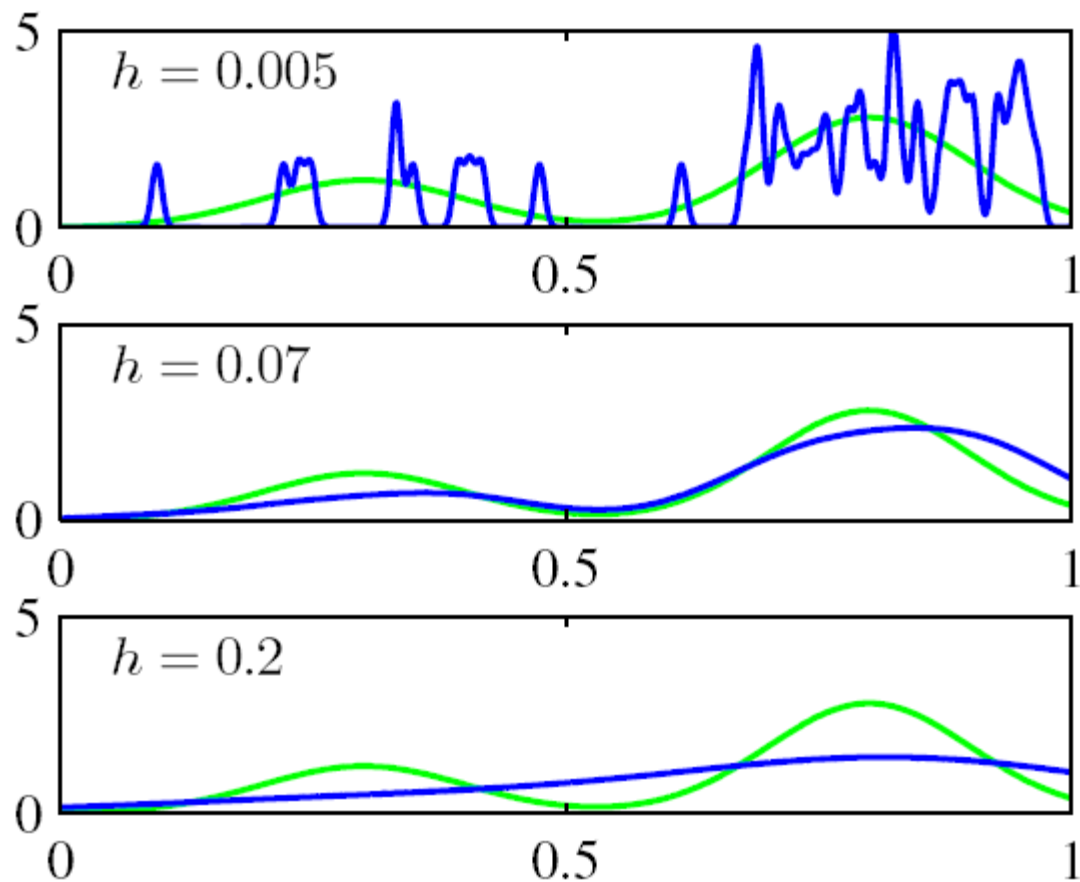


Figure 2.25 Illustration of the kernel density model .We see that h acts as a smoothing parameter and that if it is set too small (top panel), the result is a very noisy density model, whereas if it is set too large (bottom panel), then the bimodal nature of the underlying distribution from which the data is generated (shown by the green curve) is washed out. The best density model is obtained for some intermediate value of h (*middle panel*).

- Case where $p(x) = \lambda_1.U(a,b) + \lambda_2.T(c,d)$ (unknown density) (mixture of a uniform and a triangle density)

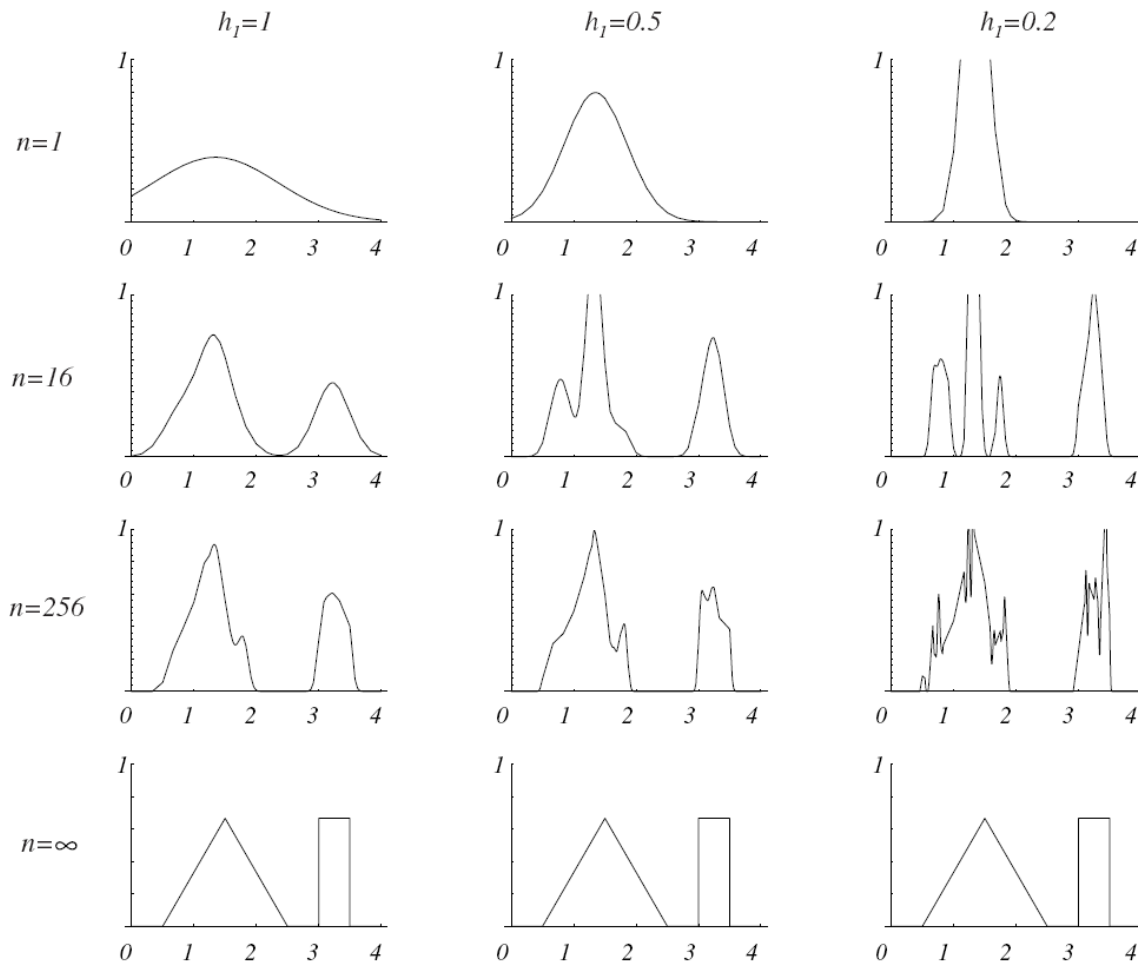
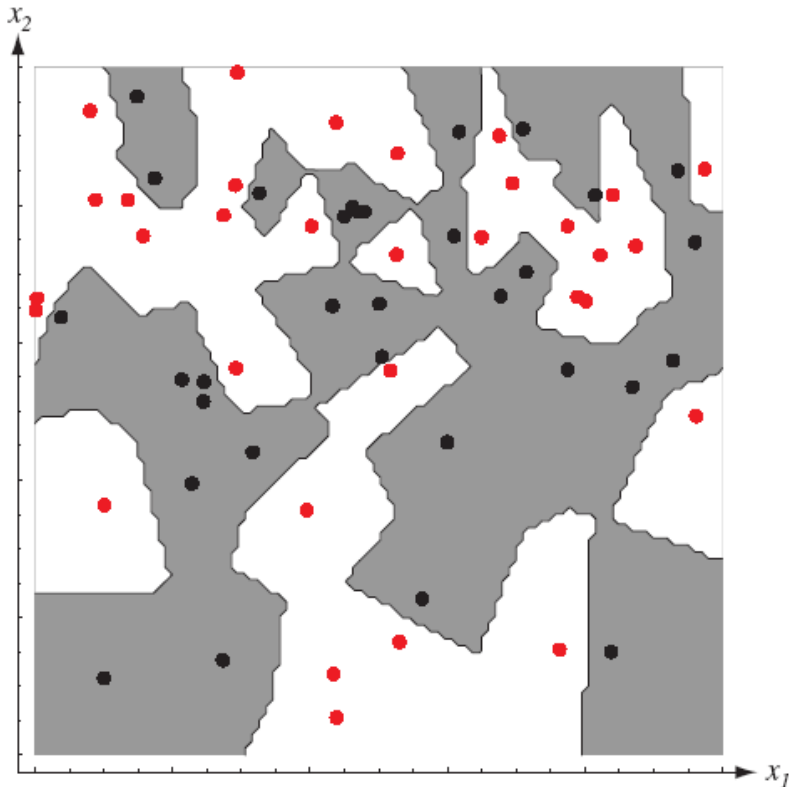


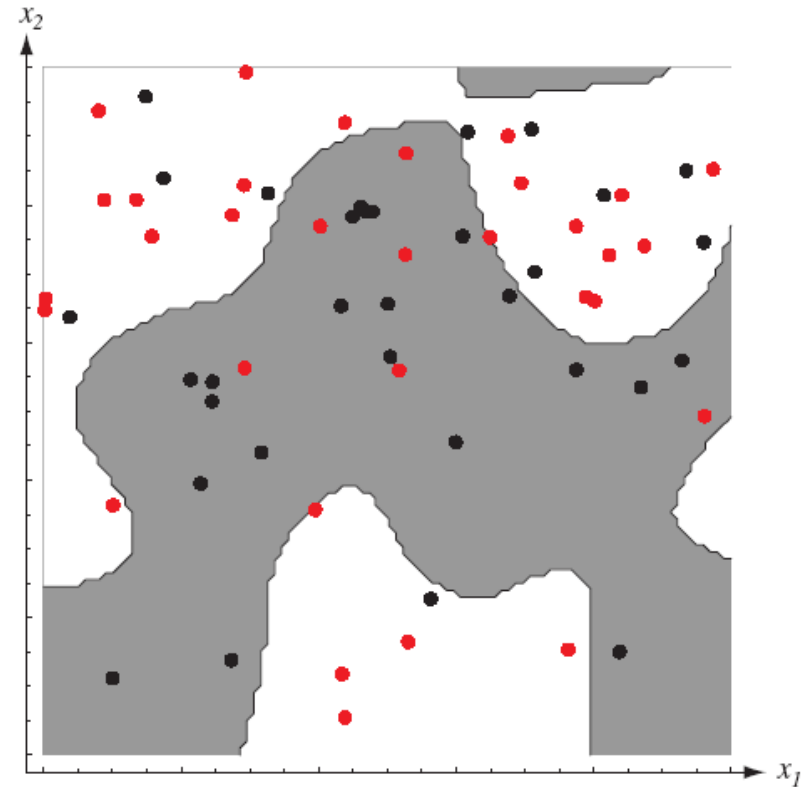
FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width.

Classification using kernel-based density estimation

- 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.



very low error on training examples

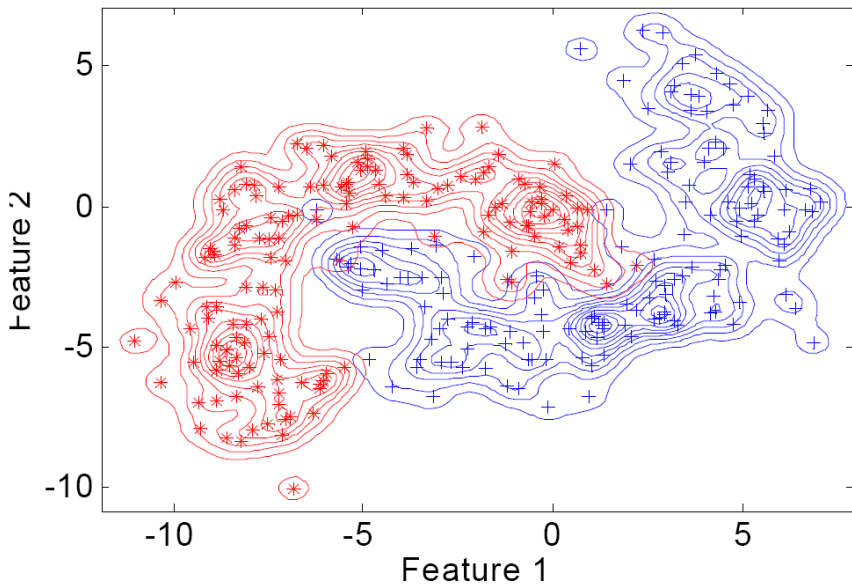


Better generalization

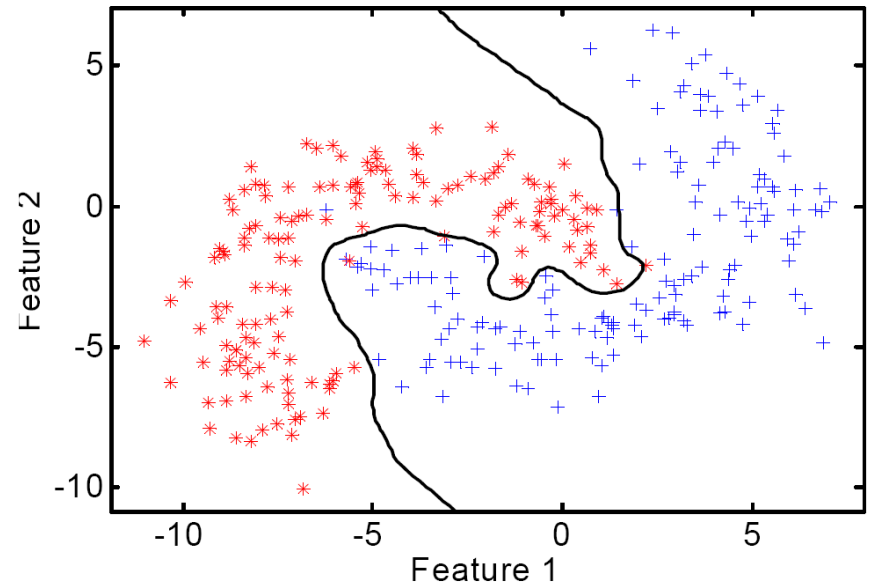
FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h . At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the upper region, while a large h would be appropriate for the lower region; no single window width is ideal overall.

Parzen classifier

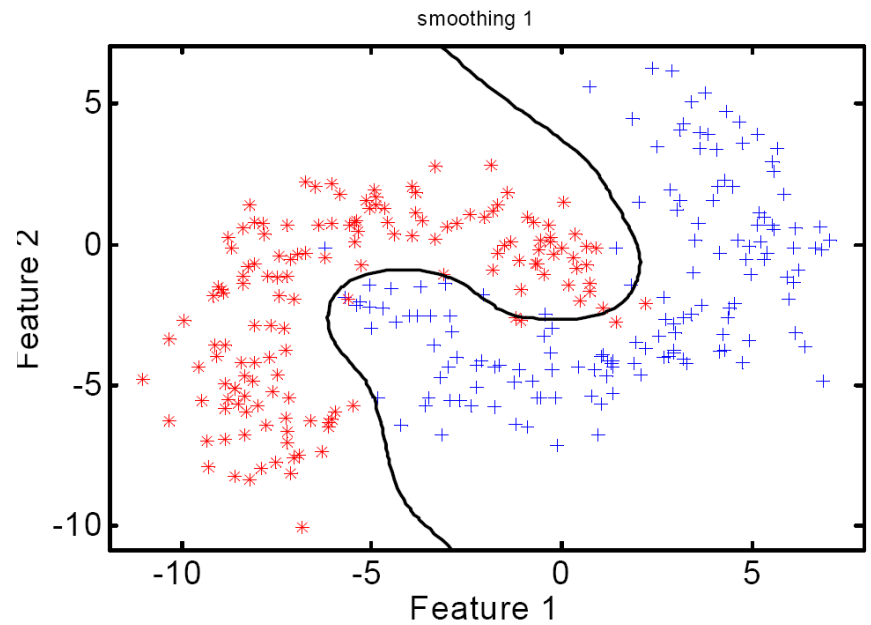
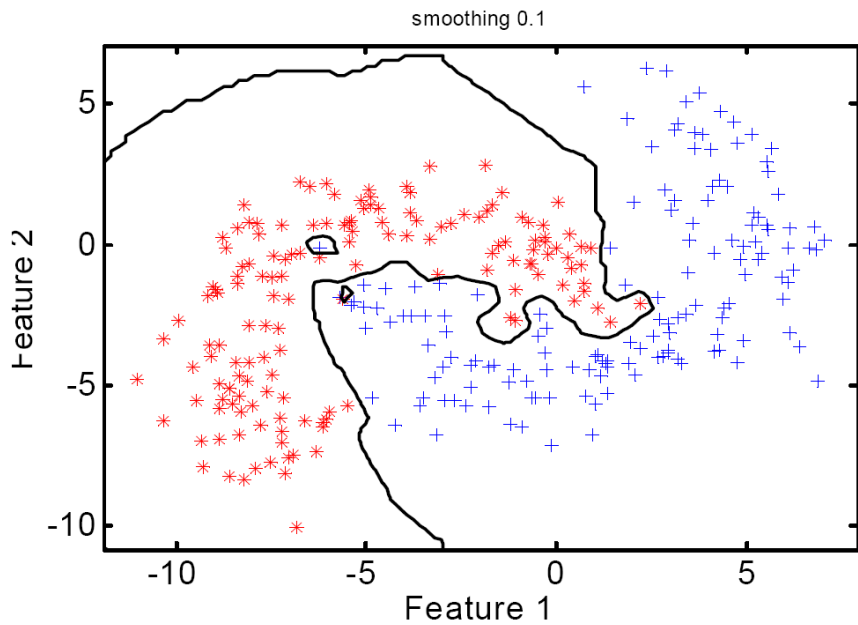
Banana Set



Banana Set



Parzen classifier



Drawbacks of kernel-based methods

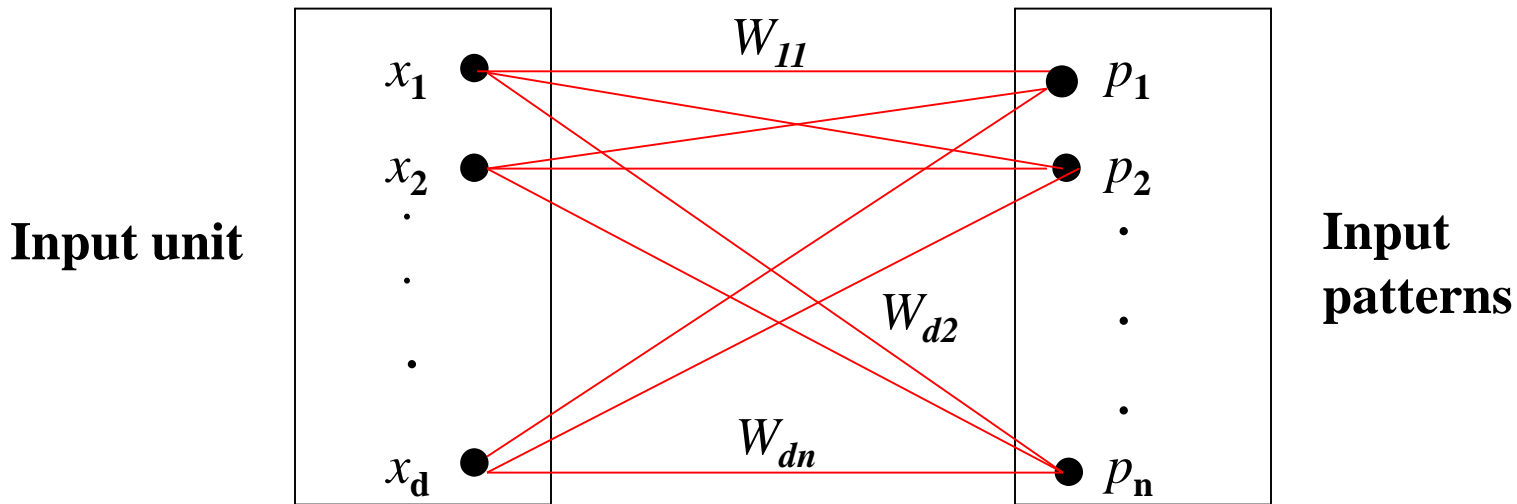
- Require a large number of samples.
- Require all the samples to be stored.
- Evaluation of the density could be very slow if the number of data points is large.
- Possible solution: use fewer kernels and adapt the positions and widths in response to the data (e.g., mixtures of Gaussians!)

Example from Dr Khotanzad's lecture notes (p144)

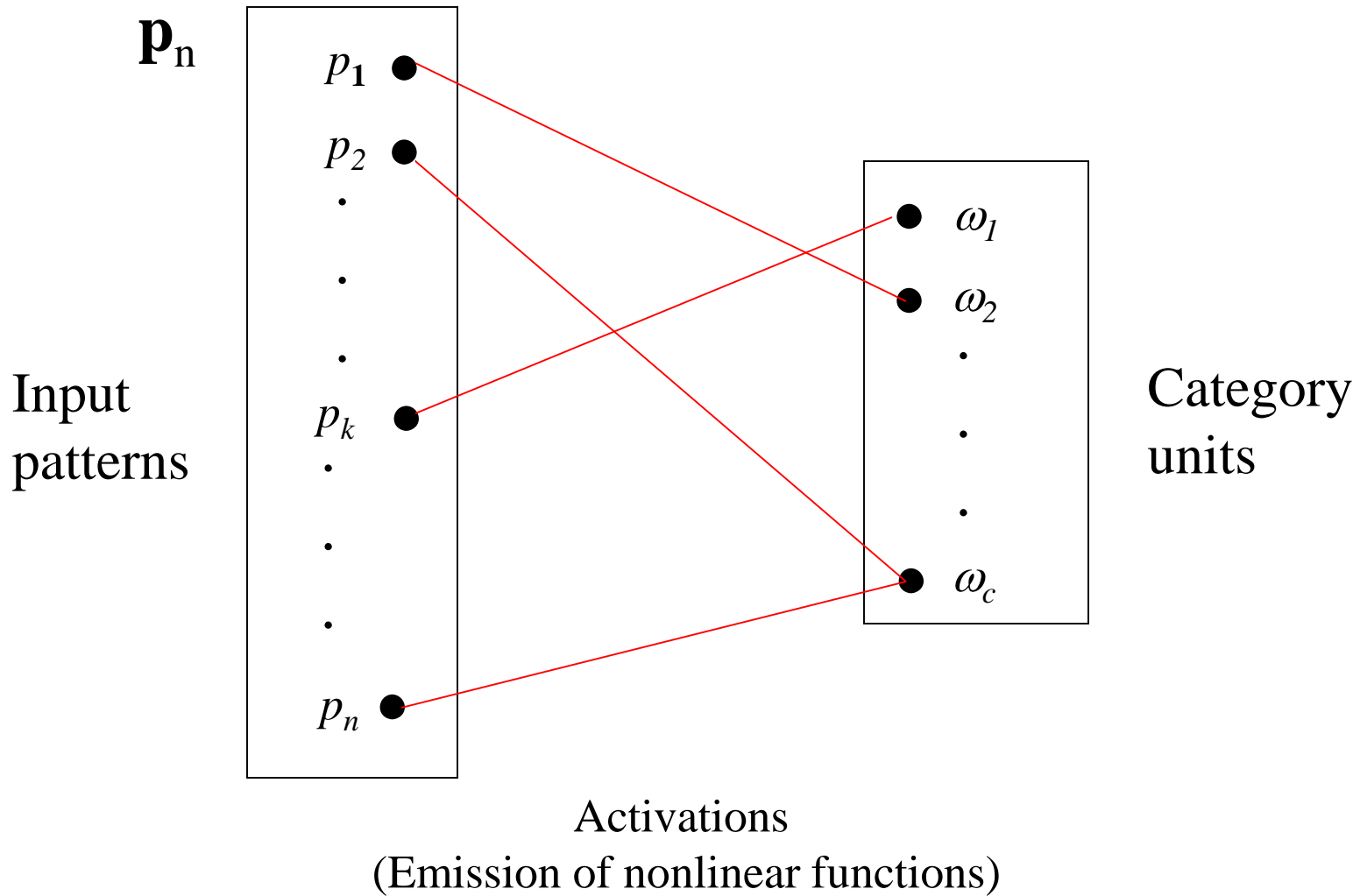
- Parzen Window (cont.)

- Parzen Window – Probabilistic Neural Network

- Compute a Parzen estimate based on n patterns
 - Patterns with d features sampled from c classes
 - The input unit is connected to n patterns



Modifiable weights or θ (trained)



- **Training the network → Algorithm**

1. Normalize each pattern \mathbf{x} of the training set to 1,

$$\sum_{i=1}^d x_i^2 = 1.$$

2. Place the first training pattern on the input units
3. Set the weights linking the input units and the first pattern units such that: $\mathbf{w}_1 = \mathbf{x}_1$
4. Make a single connection from the first pattern unit to the category unit corresponding to the known class of that pattern
5. Repeat the process for all remaining training patterns by setting the weights such that

$$\mathbf{w}_k = \mathbf{x}_k \quad (k = 1, 2, \dots, n)$$

We finally obtain the following network

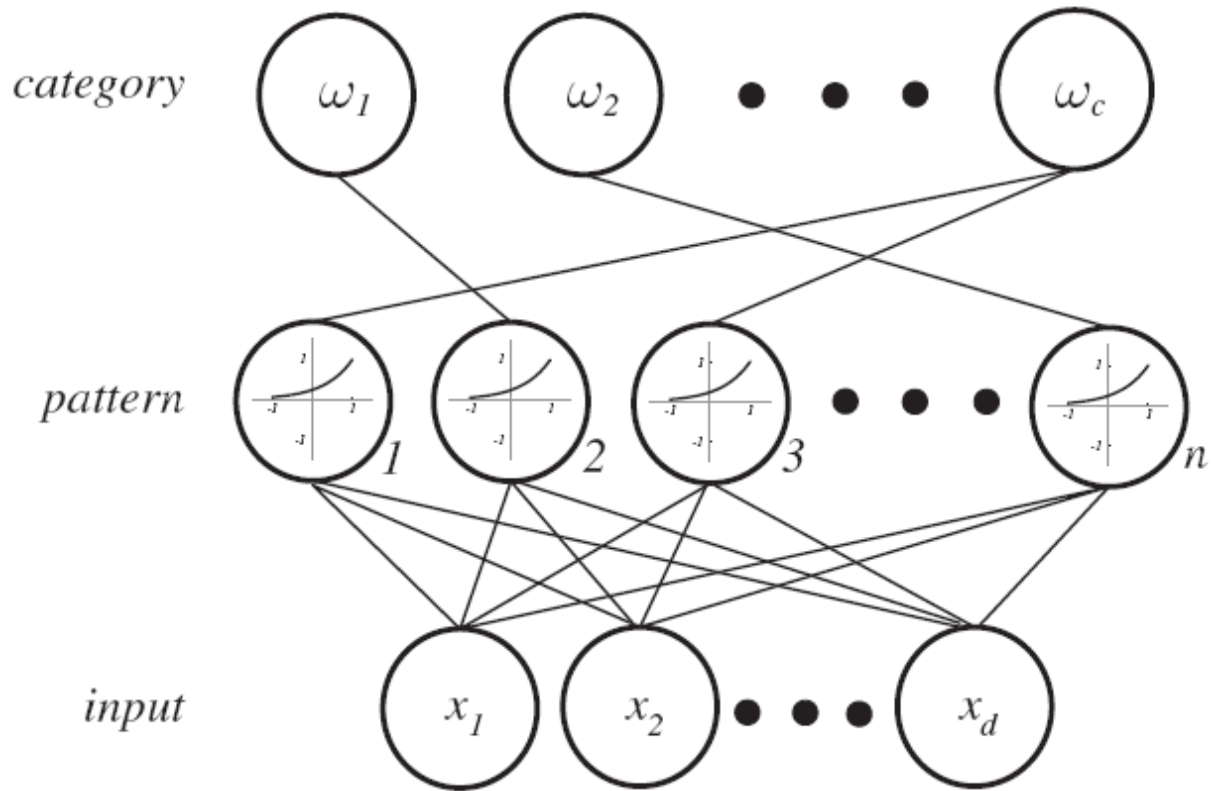


FIGURE 4.9. A probabilistic neural network (PNN) consists of d input units, n pattern units, and c category units. Each pattern unit forms the inner product of its weight vector and the normalized pattern vector \mathbf{x} to form $z = \mathbf{w}^t \mathbf{x}$, and then it emits $\exp[(z - 1)/\sigma^2]$. Each category unit sums such contributions from the pattern unit connected to it. This ensures that the activity in each of the category units represents the Parzen-window density estimate using a circularly symmetric Gaussian window of covariance $\sigma^2 \mathbf{I}$, where \mathbf{I} is the $d \times d$ identity matrix.

- **Testing the network**

- **Algorithm**

1. Normalize the test pattern \mathbf{x} and place it at the input units
2. Each pattern unit computes the inner product in order to yield the net activation

$$z_k = net_k = \mathbf{w}_k^t \cdot \mathbf{x}$$

and emit a nonlinear function $f(z_k) = \exp\left[\frac{z_k - 1}{\sigma^2}\right]$

- That is, if we let our effective width h be a constant, the window function is

$$\begin{aligned} \varphi\left(\frac{\mathbf{x}_k - \mathbf{w}_k}{h_n}\right) &\propto \overbrace{e^{-(\mathbf{x} - \mathbf{w}_k)^t (\mathbf{x} - \mathbf{w}_k) / 2\sigma^2}}^{\text{desired Gaussian}} \\ &= e^{-(\mathbf{x}^t \mathbf{x} + \mathbf{w}_k^t \mathbf{w}_k - 2\mathbf{x}^t \mathbf{w}_k) / 2\sigma^2} = \underbrace{e^{(z_k - 1) / \sigma^2}}_{\text{transfer function}}, \end{aligned}$$

3. Each output unit sums the contributions from all pattern units connected to it

$$p_n(\mathbf{x} | \omega_j) = \sum_{i=1}^n \varphi_i \propto P(\omega_j | \mathbf{x})$$

4. Classify by selecting the maximum value of $p_n(\mathbf{x} | \omega_j)$ ($j = 1, \dots, c$)

Benefits of PNNs

- Speed of learning; requires only a single pass through the training data.
- The space complexity; $O((n+1)d)$. The time complexity for classification by the parallel implementation is $O(1)$.
- New training patterns can be incorporated into a previously trained classifier quite easily;

Choosing the window function

- One of the problems encountered in the Parzen-window/PNN approach concerns the choice of the sequence of cell volumes sizes V_1, V_2, \dots or overall window size (or indeed other window parameters, such as shape or orientation).
- If we take $V_n = V_1/\sqrt[n]{n}$, the results for any finite n will be very sensitive to the choice for the initial vol. V_1 .
- If V_1 is too small, most of the volumes will be empty, and the estimate $p_n(\mathbf{x})$ will be very erratic.
- On the other hand, if V_1 is too large, important spatial variations in $p(\mathbf{x})$ may be lost due to averaging over the cell volume.