CpE 646 Pattern Recognition and Classification

Prof. Hong Man

Department of Electrical and Computer Engineering
Stevens Institute of Technology

STEVENS Institute of Technology

Visual Information Environment Laboratory

Non-Parametric Classification -- I

Chapter 4 (Section 4.1 -- 4.3):

- Introduction
- Density Estimation
- Parzen Windows

Introduction

- In Chapter 3, we consider supervised learning under several assumptions that are usually not true in practice.
  - The underlying density functions are known. In practice common parametric forms rarely fit the true densities.
  - All parametric densities are unimodal (have a single local maximum), whereas many practical problems involve multi-modal densities.
  - High-dimensional density functions are represented as the product of one-dimensional functions.
- Nonparametric procedures can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known.

Introduction

- There are two types of nonparametric methods:
  - Estimating the density function $p(x|\omega_j)$ from the sample pattern.
  - Bypass probability density and go directly to *a posteriori* probability estimation $P(\omega_j|x)$, i.e. decision functions.

Density Estimation

- Basic idea of estimating unknown probability density function is simple
  - Probability that a vector $\boldsymbol{x}$ will fall in region R is

$$P = \int_R p(x')dx' \qquad (1)$$

  - P is a smoothed (or averaged) version of the density function $p(\boldsymbol{x})$.
    - Instead of estimating $p(\boldsymbol{x})$ at arbitrary position $\boldsymbol{x}$, we measure P for certain R that contains $\boldsymbol{x}$.

Density Estimation

– If we have a sample of size n, $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, drawn from $p(\boldsymbol{x})$, the probability that k out of these n fall in R follows the binomial law

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \qquad (2)$$

where the expected value for k is

$$E(k) = nP \qquad (3)$$

Density Estimation

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

$$\int_R p(x')dx' \cong p(x)V \qquad (4)$$

where $\boldsymbol{x}$ is a point (vector) within R and V is the volume enclosed by R. $V = \int_R dx'$

Density Estimation

- Combining equation (1) , (3) and (4) yields:

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

- If we fix the volume V and obtain more and more training samples, the ratio k/n will converge to true P (Figure 4.1)
- In order to obtain $p(x)$ we have to let V approach zero.
- In practice, if V becomes very small, it will contain no sample, and the $p(x)$ estimation will be useless. Therefore V can not be arbitrarily small, and one will have to accept a certain amount of variance in k/n

Density Estimation

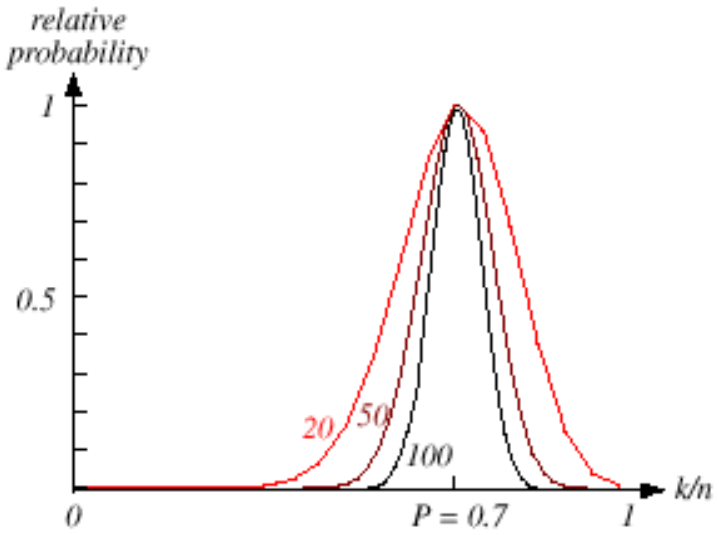


FIGURE 4.1. The relative probability an estimate given by Eq. 4 will yield a particular value for the probability density, here where the true probability was chosen to be 0.7. Each curve is labeled by the total number of patterns n sampled, and is scaled to give the same maximum (at the true probability). 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 $n \to \infty$, the curve approaches a delta function, and we are guaranteed that our estimate will give the true probability. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Density Estimation

- Theoretically, if an unlimited number of samples is available, we can circumvent this difficulty
  - To estimate the density at position $\boldsymbol{x}$, i.e. $p(\boldsymbol{x})$, we form a sequence of regions R_1, R_2,…containing $\boldsymbol{x}$: 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 $p_n(\boldsymbol{x})$ be the n^{th} estimate for $p(\boldsymbol{x})$:

$$p_n(x) = \frac{k_n / n}{V_n} \qquad (7)$$

Density Estimation

– The necessary conditions for $p_n(x)$ converge to $p(x)$:

1) $\lim_{n\to\infty} V_n = 0$

2) $\lim_{n\to\infty} k_n = \infty$

3) $\lim_{n\to\infty} k_n / n = 0$

- Condition 1) will let P/V converge to $p(x)$
- Condition 2) will let k/n converge to P
- Condition 3) will ensure $p_n(x)$ in equation (7) converge (because denominator of equation (7) goes to 0 as n goes to ∞)

Density Estimation

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

  • Shrink an initial region where $V_n = 1/\sqrt{n}$ and show that

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

  This is called the Parzen-window estimation method

  • Specify k_n as some function of n, such as $k_n = \sqrt{n}$

  (note that k_n has to increase slower than n because of condition 3), the volume V_n is adjusted accordingly until it encloses k_n neighbors of $\mathbf{x}$. The volume V_n is recorded. This is called the k_n-nearest neighbor estimation method

Density Estimation

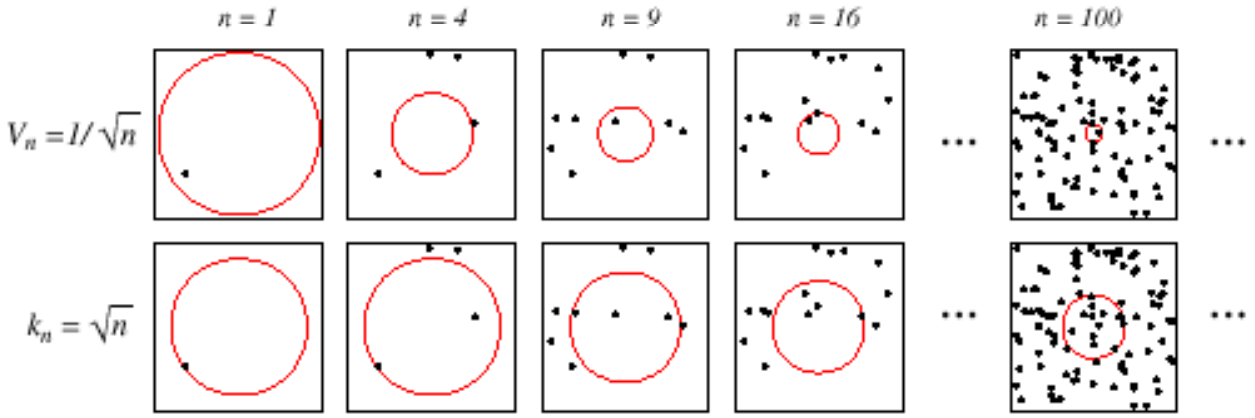


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the 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. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows

- Parzen-window approach to estimate densities assumes that the region R_n is a d-dimensional hypercube $V_n = h_n^d$, where h_n is the length of the edge of R_n.
- Define a window function

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

  - $\varphi\left(\frac{x - x_i}{h_n}\right)$ is equal to unity if $\boldsymbol{x_i}$ falls within the hypercube of volume V_n centered at x and is equal to zero otherwise.

Parzen Windows

– The number of samples out of total n samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ in this hypercube is:

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

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

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n}\ \varphi\left(\frac{x - x_i}{h_n}\right) \qquad (11)$$

$p_n(\boldsymbol{x})$ approximates $p(\boldsymbol{x})$ as an average of some distance functions of $\boldsymbol{x}$ and the samples $\boldsymbol{x}_i$ (i= 1, … ,n).

Parzen Windows

- These functions φ can be general, it represents an interpolation operator.
- To ensure $p_n(x)$ be a legitimate density function, the window function has also to be a density function

$$\varphi(x) \geq 0$$

$$\int \varphi(u)du = 1$$

Parzen Windows

– The Effects of window width h_n

- Define a normalized window function

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

- We have

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

so $p_n(x)$ is the average of all δ_n

- The window width h_n specifies the smoothness of the window function and density estimate $p_n(x)$

Parzen Windows

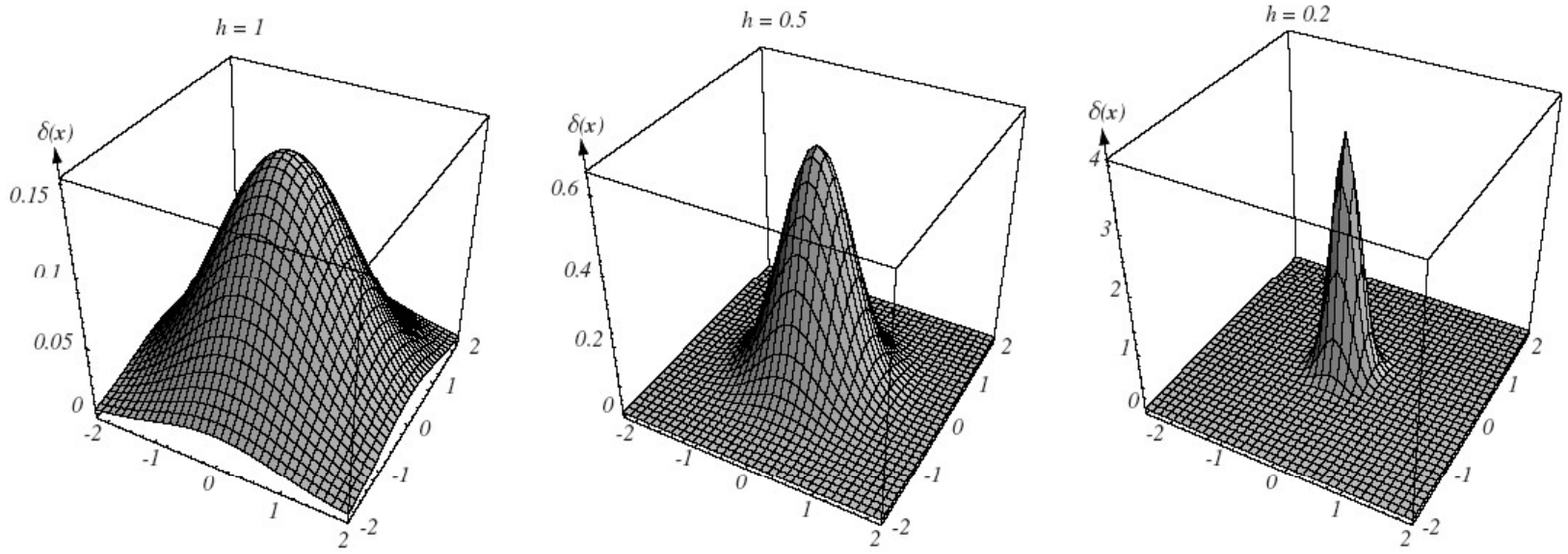


FIGURE 4.3. Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of h. Note that because the $\delta(\mathbf{x})$ are normalized, different vertical scales must be used to show their structure. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows

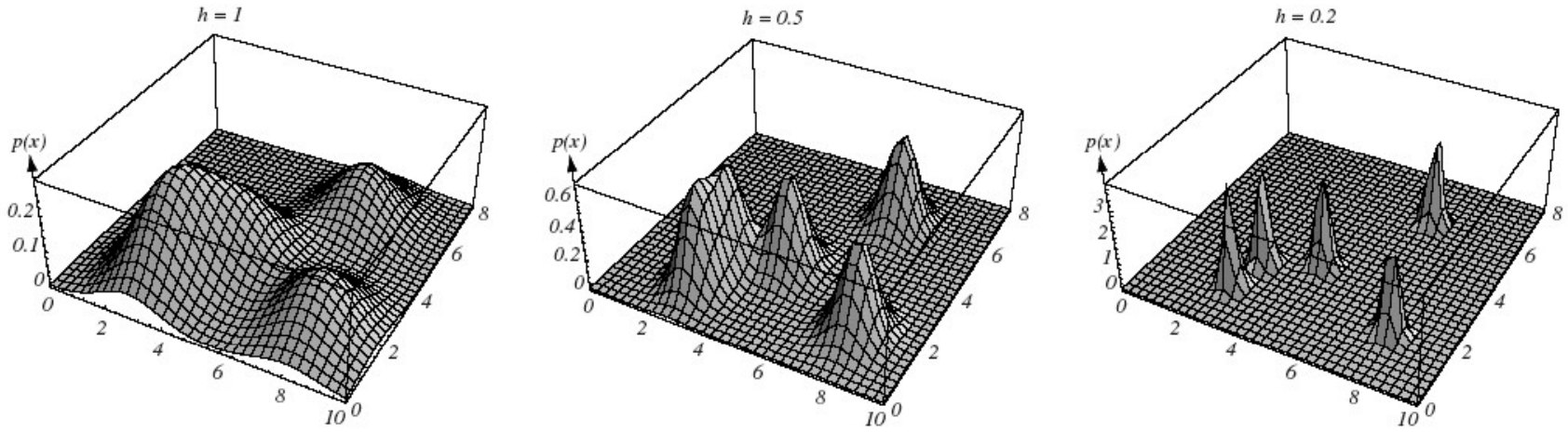


FIGURE 4.4. Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows

– Convergence of the estimate $p_n(x)$ is defined as

$$\lim_{n\to\infty} \bar{p}_n(x) = p(x) \quad \text{and} \quad \lim_{n\to\infty} \sigma_n^2(x) = 0$$

where $\bar{p}_n(x)$ is the mean of the random variables $p_n(x)$ and $\sigma_n^2(x)$ is the variance

Parzen Windows

– The mean can be written as

$$\bar{p}_n(x) = E[p_n(x)] = \frac{1}{n}\sum_{i=1}^{n} E\left[\frac{1}{V_n}\varphi\left(\frac{x - x_i}{h_n}\right)\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\int \frac{1}{V_n}\varphi\left(\frac{x - x_i}{h_n}\right)p(x_i)dx_i$$

$$= \int \frac{1}{V_n}\varphi\left(\frac{x - v}{h_n}\right)p(v)dv \quad (\because x_i \text{ are } i.i.d. \text{ from } p(x))$$

$$= \int \delta_n(x - v)\,p(v)dv$$

which indicates that the expected value of the estimate $p_n(x)$ is a convolution of the unknown density $p(x)$ and the window function, i.e. a blurred version of $p(x)$

Parzen Windows

– The variance of the estimates is

$$\sigma_n^2(x) = \sum_{i=1}^{n} E\left[\left(\frac{1}{nV_n}\varphi\left(\frac{x-x_i}{h_n}\right) - \frac{1}{n}\bar{p}_n(x)\right)^2\right]$$

$$= \frac{1}{nV_n}\int \frac{1}{V_n}\varphi^2\left(\frac{x-v}{h_n}\right)p(v)dv - \frac{1}{n}\bar{p}_n^2(x)$$

$$\le \frac{1}{nV_n}\int \varphi\left(\frac{x-v}{h_n}\right)\frac{1}{V_n}\varphi\left(\frac{x-v}{h_n}\right)p(v)dv$$

$$\le \frac{\sup(\varphi(\cdot))\bar{p}_n(x)}{nV_n}$$

drop $\frac{1}{n}\bar{p}_n^2(x)$ and

assume $\varphi\left(\frac{x-v}{h_n}\right)$ is

bounded by a constant

$\sup(\varphi(\cdot))$

- A small variance requires large nV_n

Parzen Windows

- Illustration the behavior of the Parzen window method
  - Case 1 $p(x) \sim N(0, 1)$,
    - let $\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$ and $h_n = h_1 / \sqrt{n}$

      where h_1 is a user-defined parameter and $V_n = h_n^d$

      Thus (from eq.11):

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

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

Parzen Windows

- Numerical results:

  For $n = 1$ and $h_1 = 1$

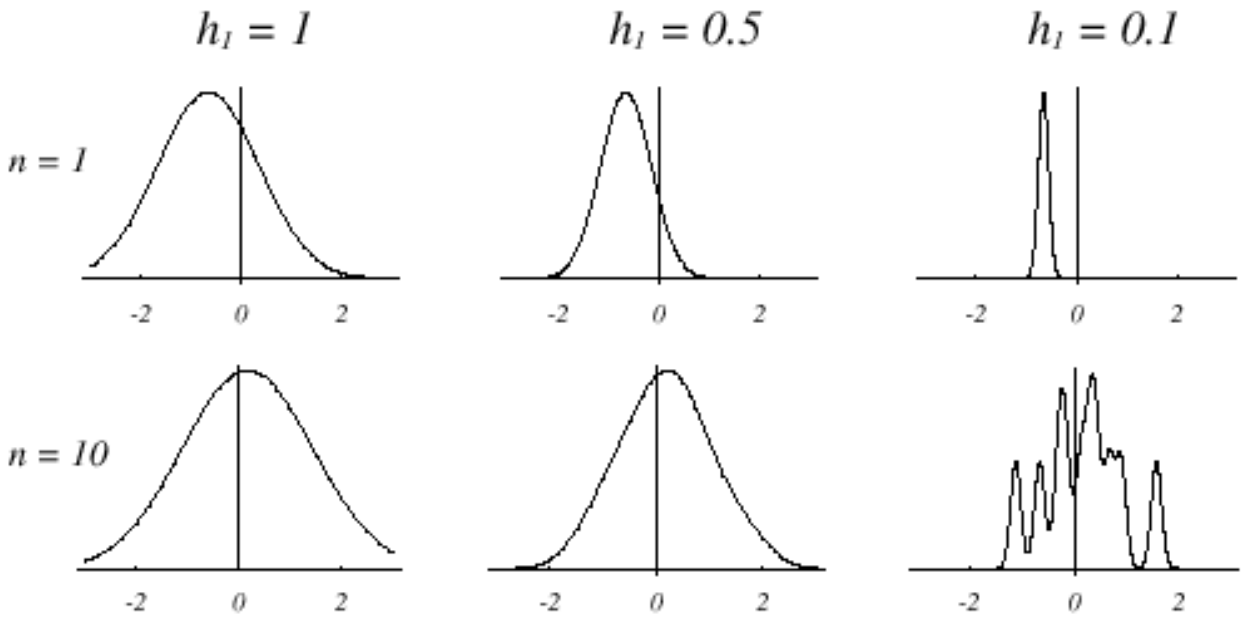
$$p_1(x) = \varphi(x - x_1)$$

$$= \frac{1}{\sqrt{2\pi}} e^{-(x-x_1)^2/2}$$

$$\rightarrow N(x_1, 1)$$

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

Parzen Windows



Parzen Windows

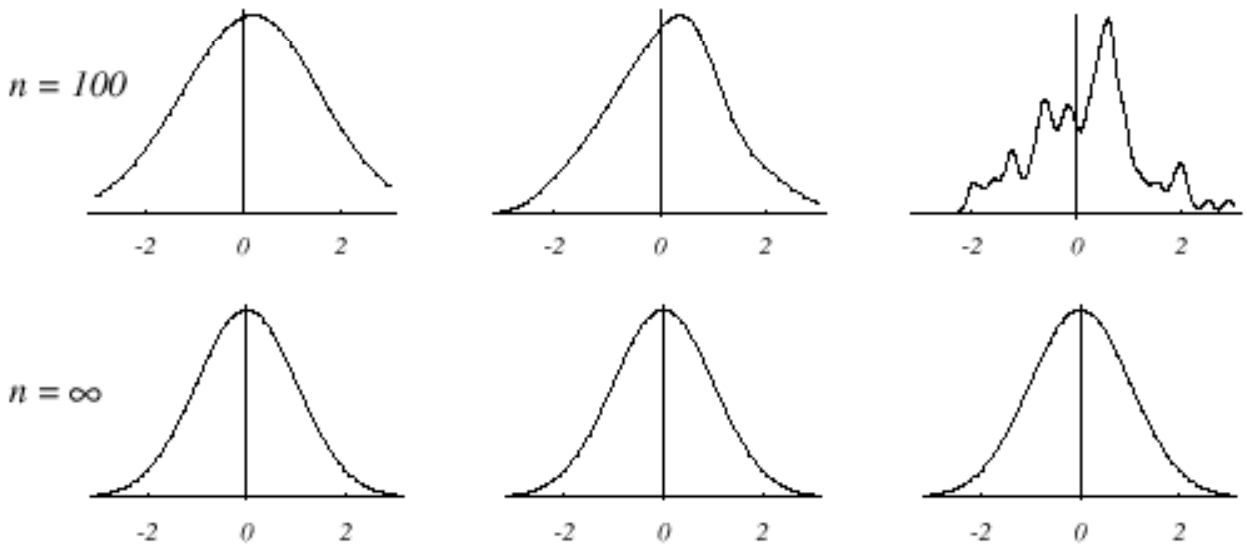
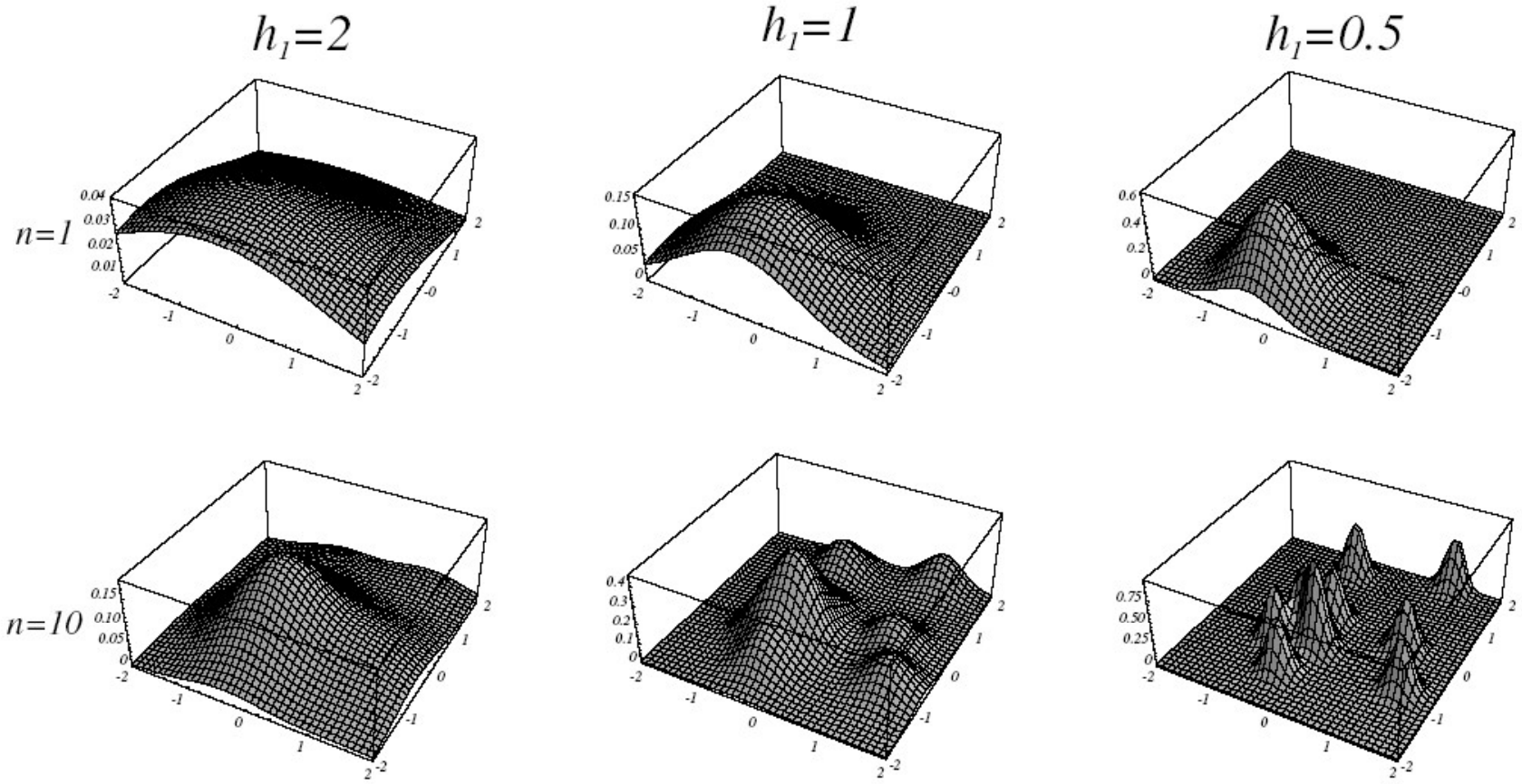


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows



Parzen Windows

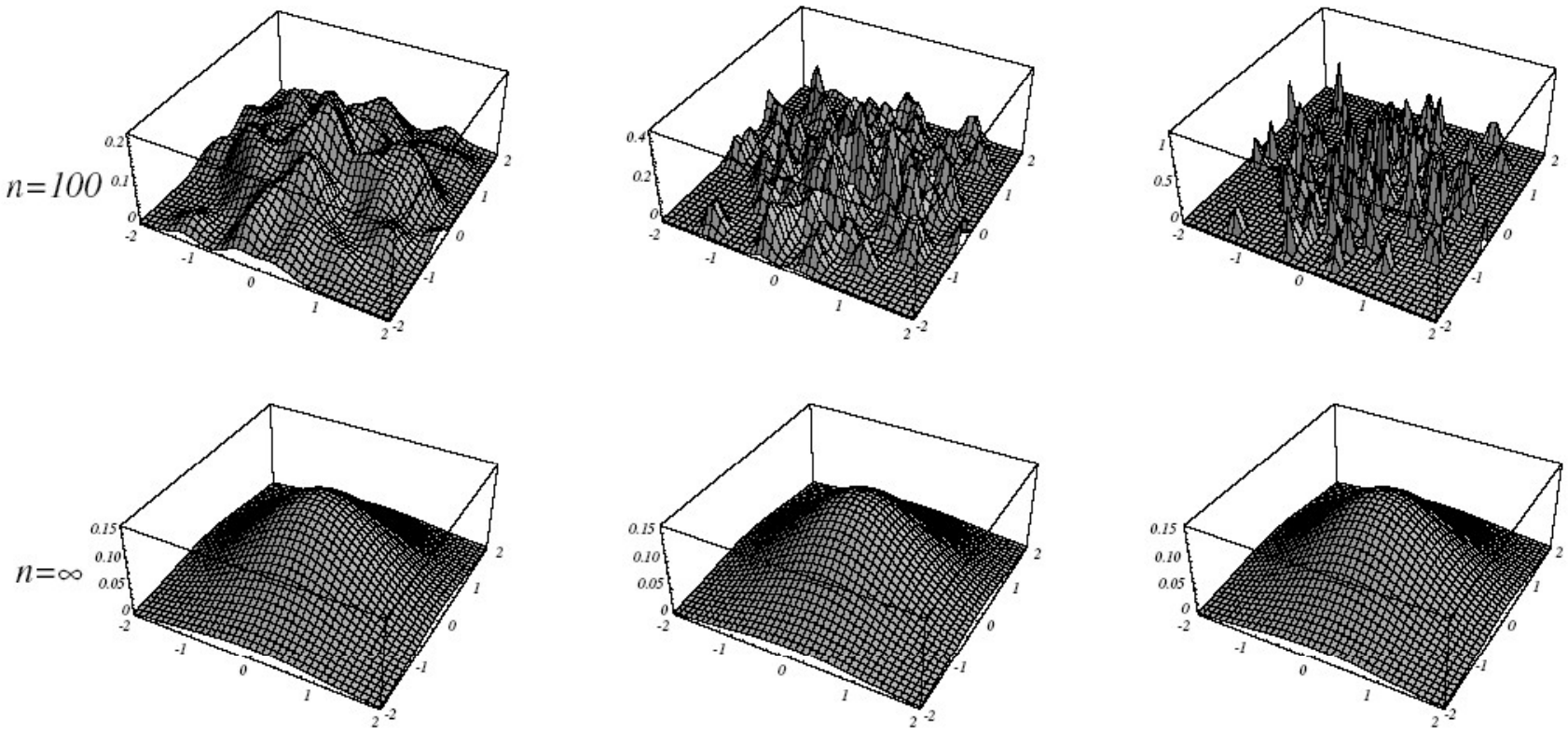
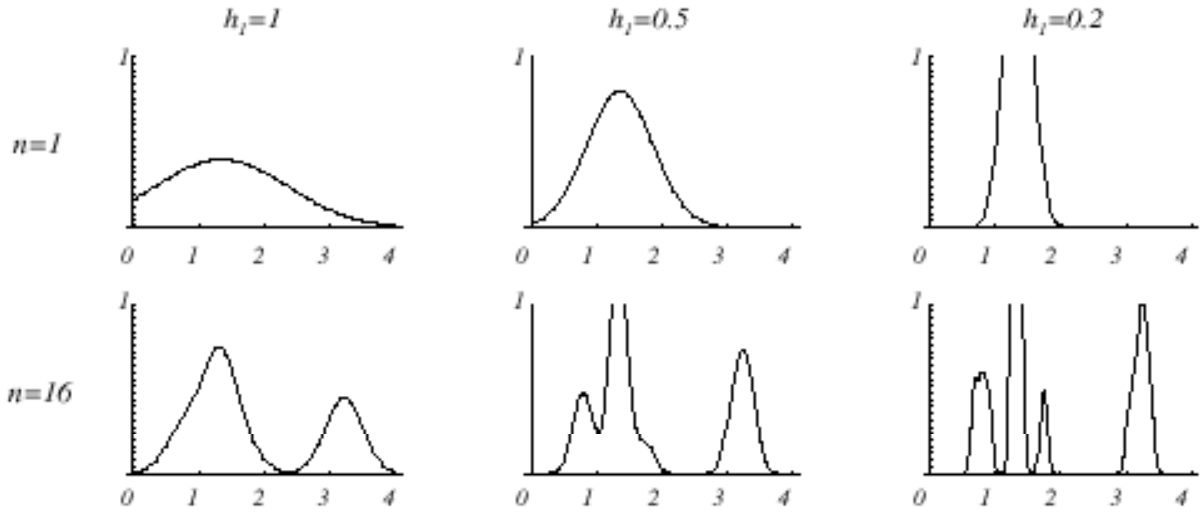


FIGURE 4.6. Parzen-window estimates of a bivariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows

– Case 2 $p(x) = \lambda_1 U(a,b) + \lambda_2 T(c,d)$ (mixture of a uniform and a triangle density)



Parzen Windows

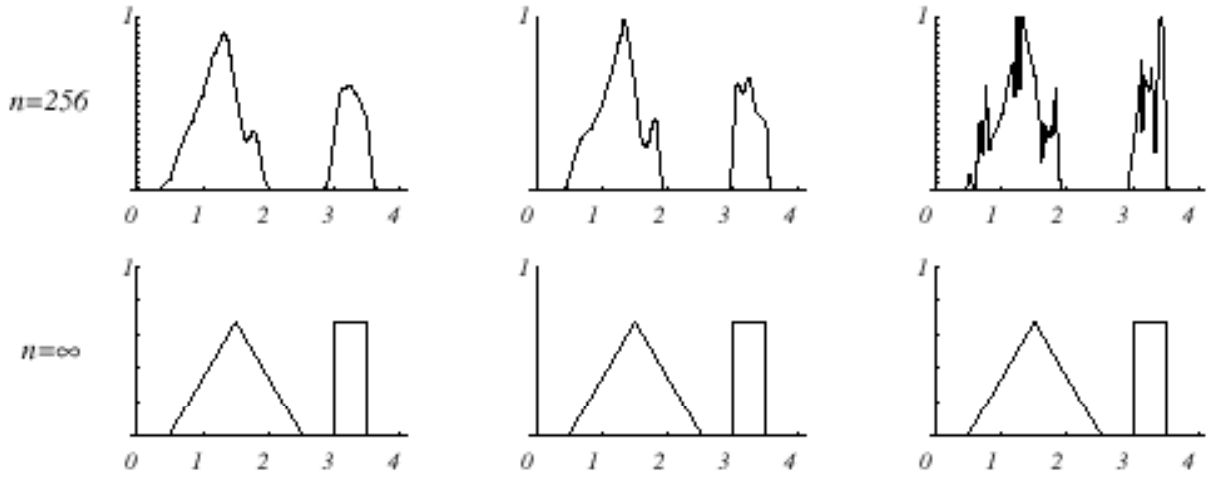


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern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

Parzen Windows

- Design the 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.
  - Training error can be reduced arbitrarily by making the window width sufficiently small, but it may not work well with novel patterns in tests.
  - Generally Gaussian windows are good choices, but no window width is particularly good.

Parzen Windows

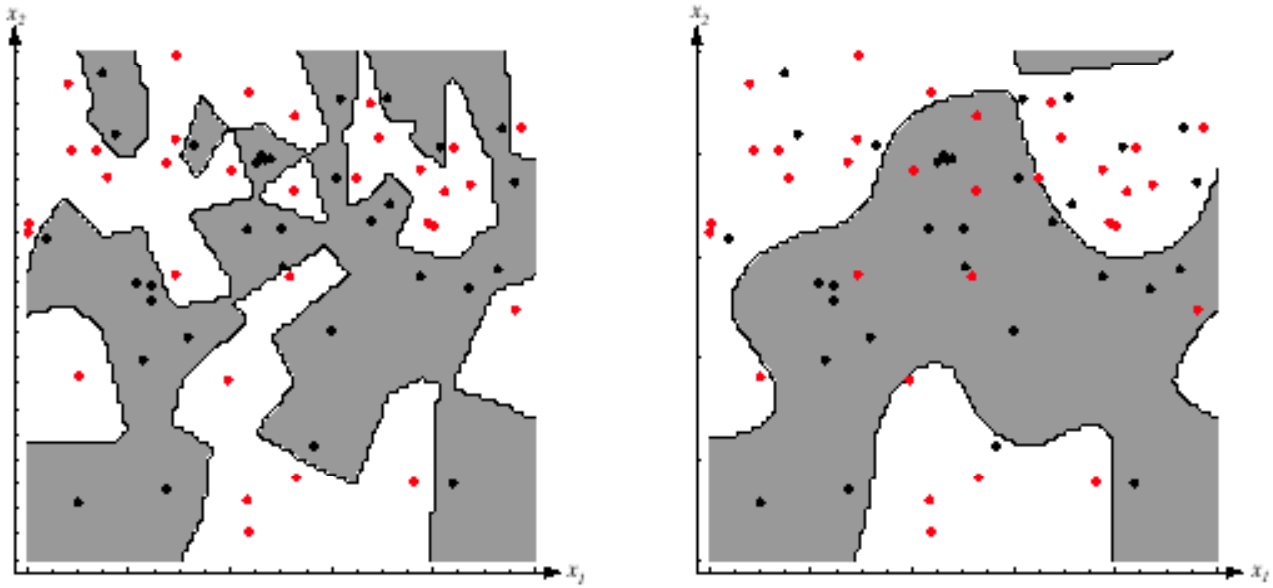


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the 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.

Parzen Windows

- Parzen window method can be implemented as a Probabilistic Neural Network (PNN).
  - Parallel implementation trading space complexity for time complexity.
  - Compute a Parzen estimate based on *n* patterns (samples), each patterns with *d* features sampled from *c* classes. This PNN has
    - *d* input units, *n* pattern units, and *c* category units
    - each input unit is connected to all pattern units, and each pattern unit is connected to a category unit.
    - The connections between the input units and the pattern units represent modifiable weights.

Parzen Windows

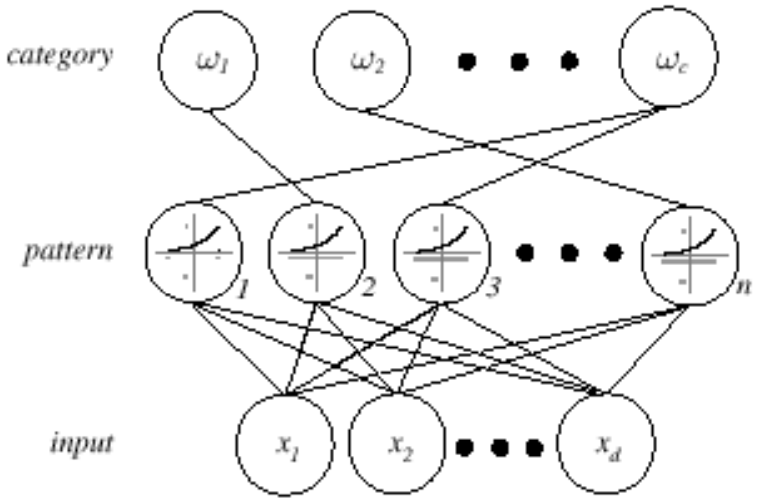


FIGURE 4.9. A probabilistic neural network (PNN) consists of d input units, n pattern units, and c category units. Each pattern unit forms the inner product of its weight vector and the normalized pattern vector $\mathbf{x}$ to form $z = \mathbf{w}^t\mathbf{x}$, and then it emits $\exp[(z-1)/\sigma^2]$. 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. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Parzen Windows

– PNN training algorithm

1. Normalize each pattern $\boldsymbol{x_k} = [x_{k1}, x_{k2}, \ldots, x_{kd}]^t$ of the training set so that
$$\sum_{i=1}^{d} x_{ki}^2 = 1$$
2. Place the first training pattern $\boldsymbol{x_1} = [x_{11}, x_{12}, \ldots, x_{1d}]^t$ on the input units
3. Set the weights linking the input units and the first pattern units such that: $\boldsymbol{w_1} = \boldsymbol{x_1}$, i.e. $[w_{11}, w_{12}, \ldots, w_{1d}]^t = [x_{11}, x_{12}, \ldots, x_{1d}]^t$

Parzen Windows

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 $\boldsymbol{w_k} = \boldsymbol{x_k}$ i.e. $[w_{k1}, w_{k2}, \ldots, w_{kd}]^t = [x_{k1}, x_{k2}, \ldots, x_{kd}]^t$ for $k = 1, 2, \ldots, n$

Parzen Windows

– PNN classification algorithm

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

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

and emit a nonlinear active function

$$f(net_k) = e^{(net_k - 1)/\sigma^2}$$

where σ related to the Gaussian window width. The active function here must be exponential to implement Parzen window method

Parzen Windows

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

$$g_i = \sum_{\{k:k \to i\}} e^{(net_k - 1)/\sigma^2}$$

4. Classify by selecting the maximum value of g_i for $i = 1, \ldots, c$

Parzen Windows

– The corresponding Gaussian Window function

- Consider an (unnormalized) Gaussian window centered on the position of one of the training pattern $\boldsymbol{w_k}$, assume the effective window width $\boldsymbol{h_n}$ is a constant, the window function is

$$\varphi\left(\frac{x-w_k}{h_n}\right) \propto e^{-(x-w_k)^t(x-w_k)/2\sigma^2}$$

$$= e^{-(x^t x + w^t w - 2x^t w_k)/2\sigma^2}$$

$$= e^{(net_k - 1)/\sigma^2}$$

$$(\because x^t x = w^t w = 1, \text{ they are normalized})$$

Parzen Windows

- Advantages of this PNN
  - The speed of learning is fast. The learning is simple, only one single pass through all training samples. But the space complexity $O((n+1)d)$ can be significant.
  - New training samples can be easily incorporated into a previously trained classifier, important for on-line learning.