5. Non-Parametric Techniques

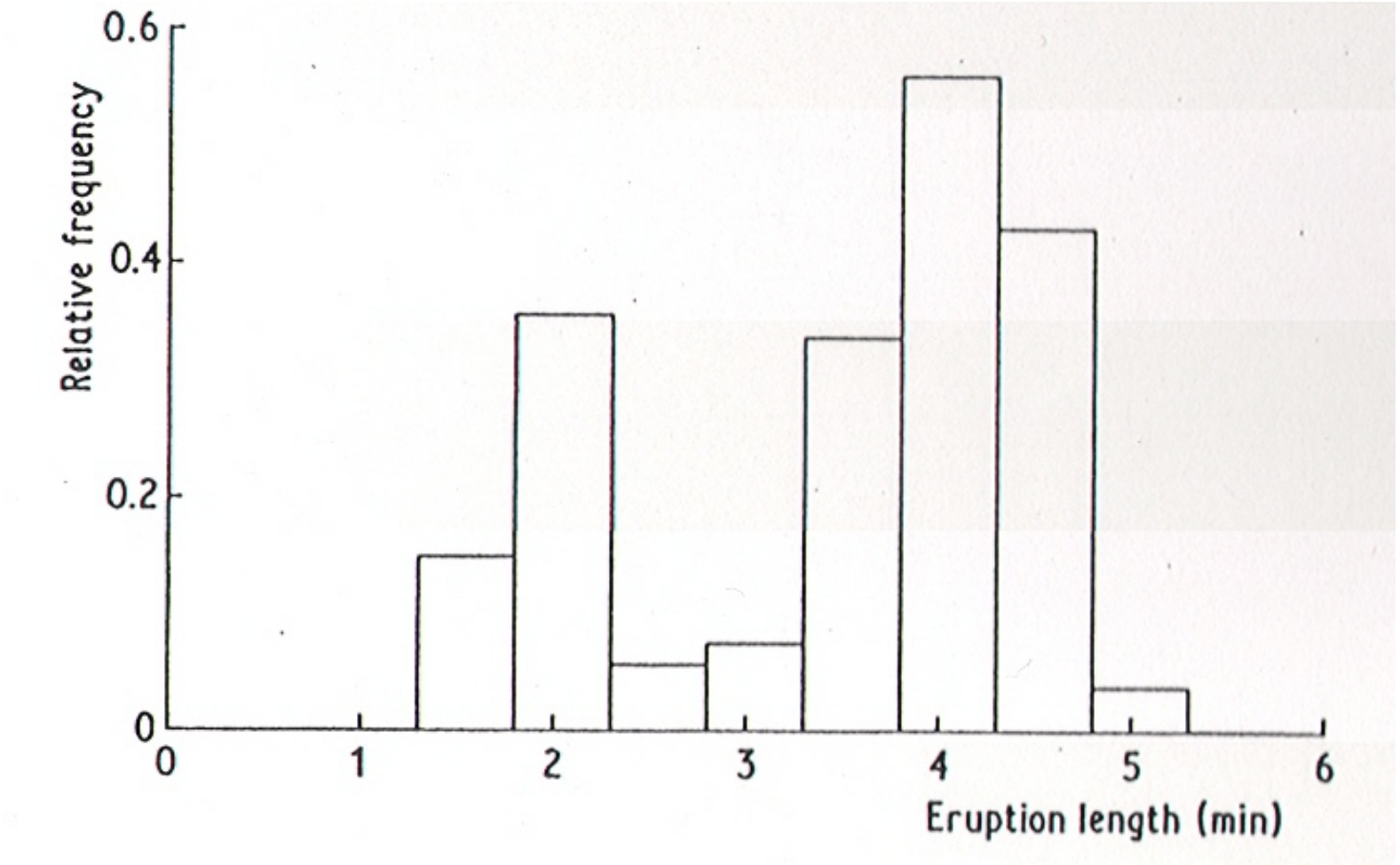
Non-Parametric Techniques

- 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
- We will consider
  - Parzen Density Estimation
  - K_n Nearest Neighbor Estimation
  - k-Nearest Neighbor Rule

Histogram Estimation

- Normalized histogram density estimation is perhaps the simplest density estimation approach
- Histogram density estimation has the main shortcomings that it is not smooth
- Other approaches are needed to overcome this problem

Histogram Estimation – Example 1D



Histogram Estimation – Example 2D

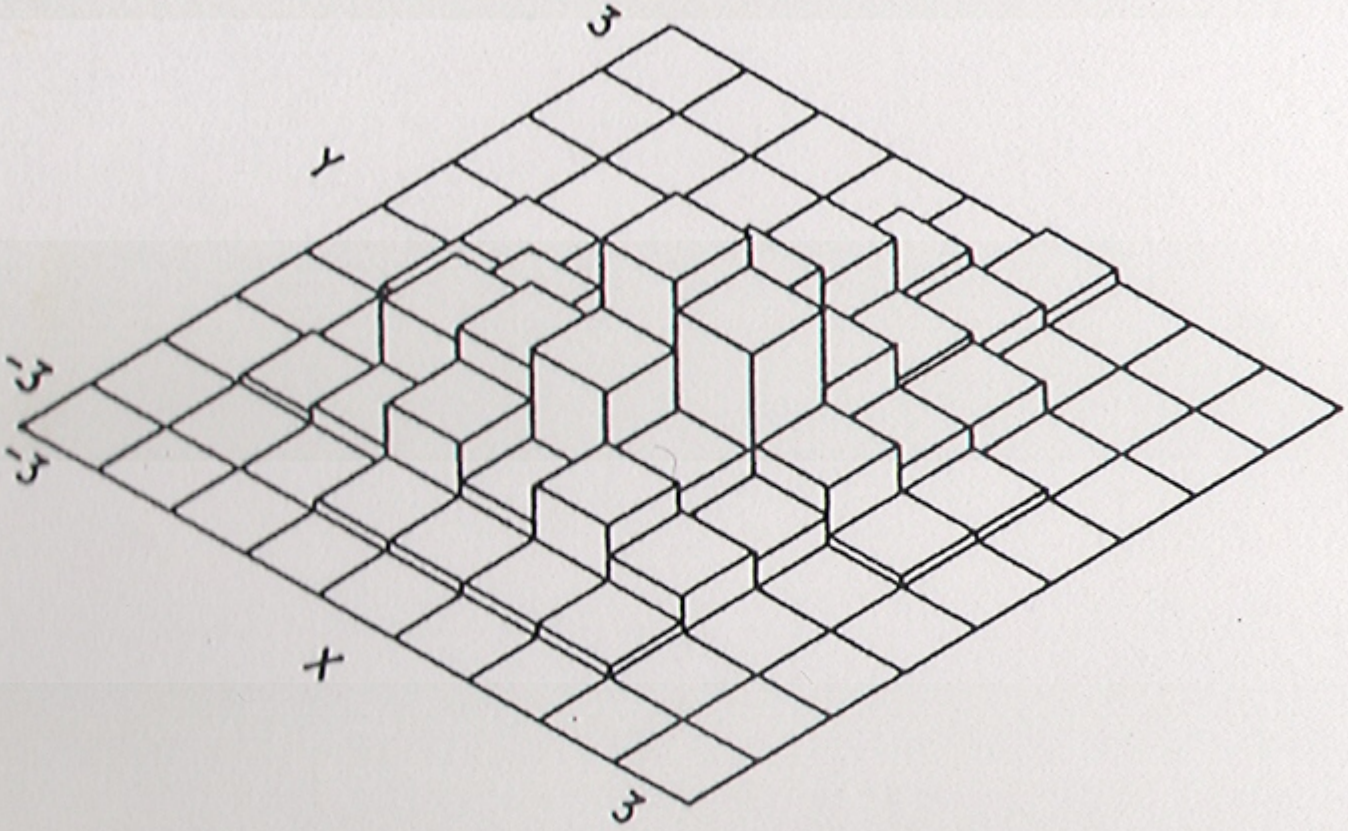


Fig. 4.2 *A typical bivariate histogram. Reproduced from Scott (1982) with the permission of the author.*

Density Estimation

- ***Basic idea:***
  Probability that a vector 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(x) if we have a sample of size n; therefore, the probability that k points fall in R is then:

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

and the expected value for k is:

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

Density Estimation

ML estimation of P = θ

- $\underset{\theta}{\text{Max}}\left(P_k \mid \theta\right)$ **is reached for** $\hat{\theta} = \frac{k}{n} \cong P$

•Therefore, the ratio k/n is a good estimate for the probability P and hence for the density function p.

•p(x) is continuous and 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 x is a point within R and V is the volume enclosed by R.

Density Estimation

- **Combining equations (1) , (3) and (4) yields:**

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

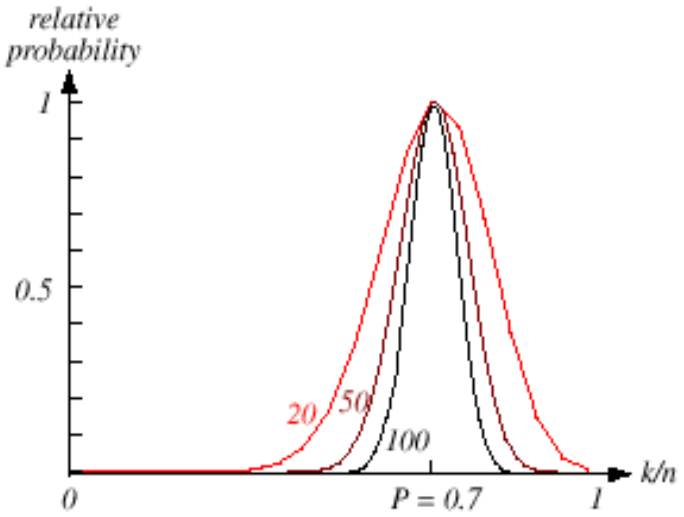


FIGURE 4.1. The relative probability an estimate given by Eq. 4 will yield a particular value for the probability density, here where the true probability was chosen to be 0.7. Each curve is labeled by the total number of patterns n sampled, and is scaled to give the same maximum (at the true probability). 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

Justification of equation (4)

$$\int_{\Re} p(x')dx' \cong p(x)V \qquad (4)$$

We assume that *p(x)* is continuous and that region *R* is so small that p does not vary significantly within *R*. Since *p(x)* = constant, it is not a part of the integral.

Density Estimation

$$\int_{\Re} p(x')dx' = p(x')\int_{\Re} dx' = p(x')\int_{\Re} 1_{\Re}(x)dx' = p(x')\mu(\Re)$$

Where: $\mu(R)$ is: a surface in the Euclidean space R^2

a volume in the Euclidean space R^3

a hypervolume in the Euclidean space R^d

Density Estimation

Since $p(x) \cong p(x')$ = constant, therefore in the Euclidean space R^3:

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

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

Density Estimation

- **Condition for convergence**

The fraction *k/(nV)* is a space averaged value of *p(x).*

***p(x)* is obtained only if V approaches zero.**

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

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

Density Estimation

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

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

Density Estimation

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 and a certain amount of averaging of the density p(x)**

Density Estimation

Theoretically, if an unlimited number of samples is available, we can circumvent this difficulty

To estimate the density of x, we form a sequence of regions

$R_1, R_2, \ldots$ containing 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 $p_n(x)$ be the n^{th} estimate for $p(x)$:

$$p_n(x) = (k_n/n)/V_n$$

Density Estimation

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

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

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

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

Density Estimation

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

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

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

This is called "the *Parzen-window* estimation method"

2) Specify k_n as some function of n, such as $k_n = \sqrt{n}$; the volume V_n is grown until it encloses k_n neighbors of x. This is called "the *k_n-nearest neighbor* estimation method"

Density Estimation

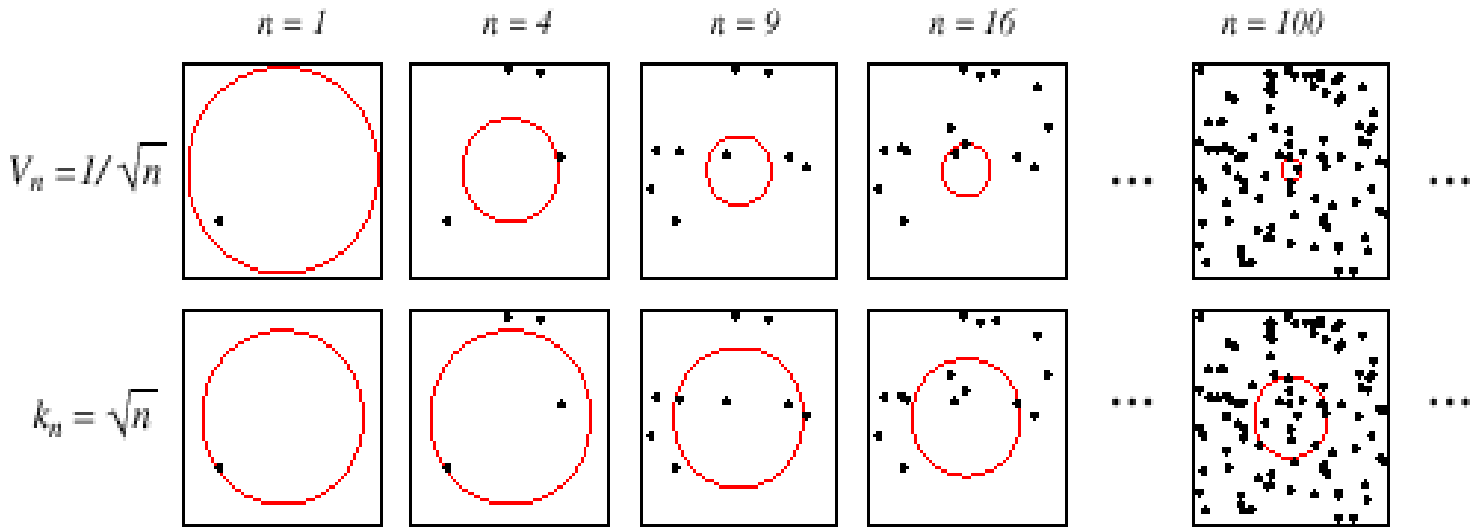


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the 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 Density Estimation

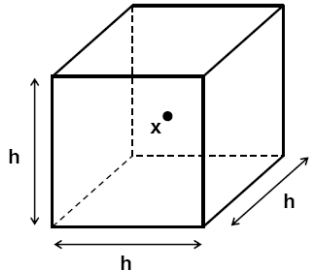
Properties:

- Also known as kernel estimator or Parzen windows
- Can be used for multiple features
- Window width is an important parameter in the Parzen Density Estimation.
- The width is usually found by trial and error

Parzen Density Estimation

Use the *Parzen-window approach* to estimate densities:

- **Assume that the region R_n is a d-dimensional hypercube**



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

Let $\varphi(u)$ be the following window function:

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

$\varphi((x-x_i)/h_n)$ is equal to unity if x_i falls within the hypercube of volume V_n centered at x. It is equal to zero otherwise.

Parzen Density Estimation

- **The number of samples in this hypercube is:**

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

By using $p(x) \cong \frac{k/n}{V}$ **we obtain the following estimate:**

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

$p_n(x)$ estimates $p(x)$ as an average of functions of x and the samples (x_i) $(i = 1,\ldots,n)$. These functions φ can be general.

Parzen Density Estimation

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

The window function is being used for *interpolation* – each sample contributing to estimate in accordance with its distance from x

Effect of the window width on $p_n(x)$

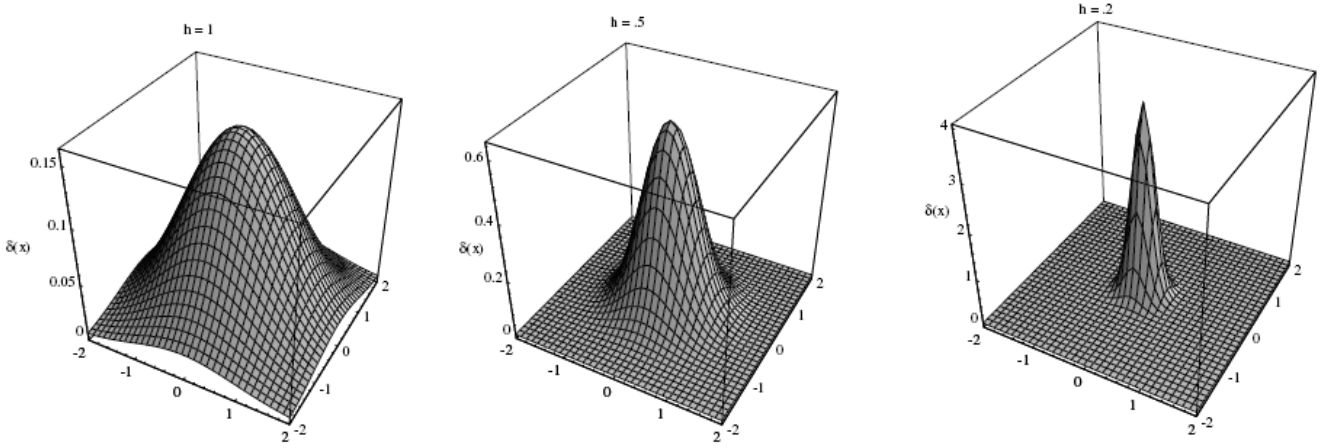


Figure 4.3: Examples of two-dimensional circularly symmetric normal Parzen windows $\varphi(\mathbf{x}/h)$ for three different values of h. Note that because the $\delta_k(\cdot)$ are normalized, different vertical scales must be used to show their structure.

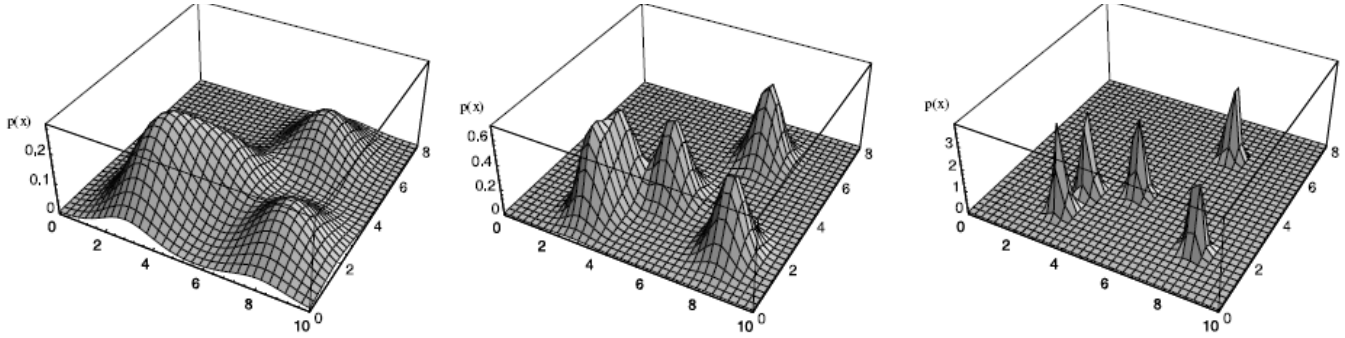


Figure 4.4: Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each function.

Parzen Density Estimation

- **The behavior of the Parzen-window method**
  Case where p(x) →N(0,1)

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

$$h_n = h_1/\sqrt{n}$$ **(h_1: known parameter)**

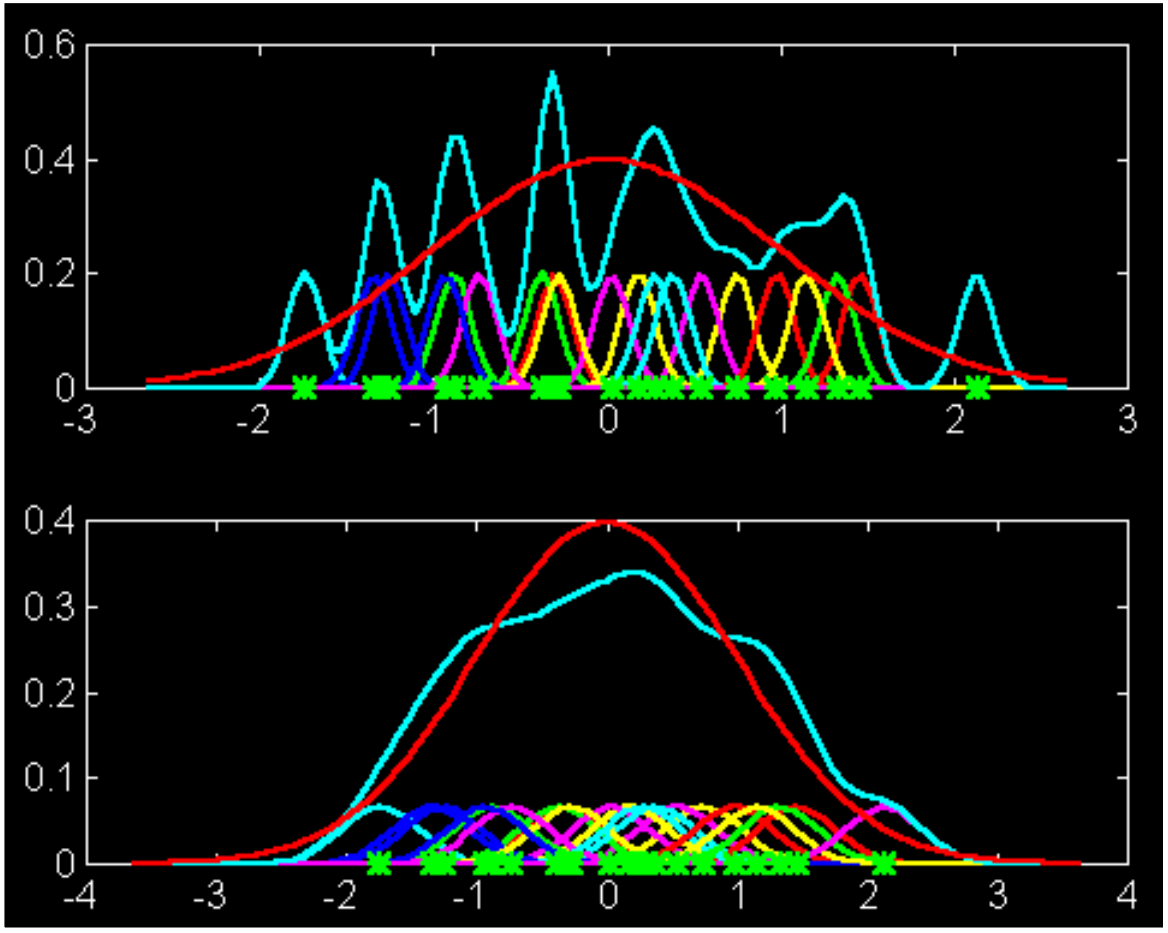
$$p_n(x) = \frac{1}{n}\sum_{i=1}^{i=n} \frac{1}{h_n}\varphi\left(\frac{x - x_i}{h_n}\right)$$

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

Parzen Density Estimation

h = 0.1

h = 0.3



Parzen Density Estimation

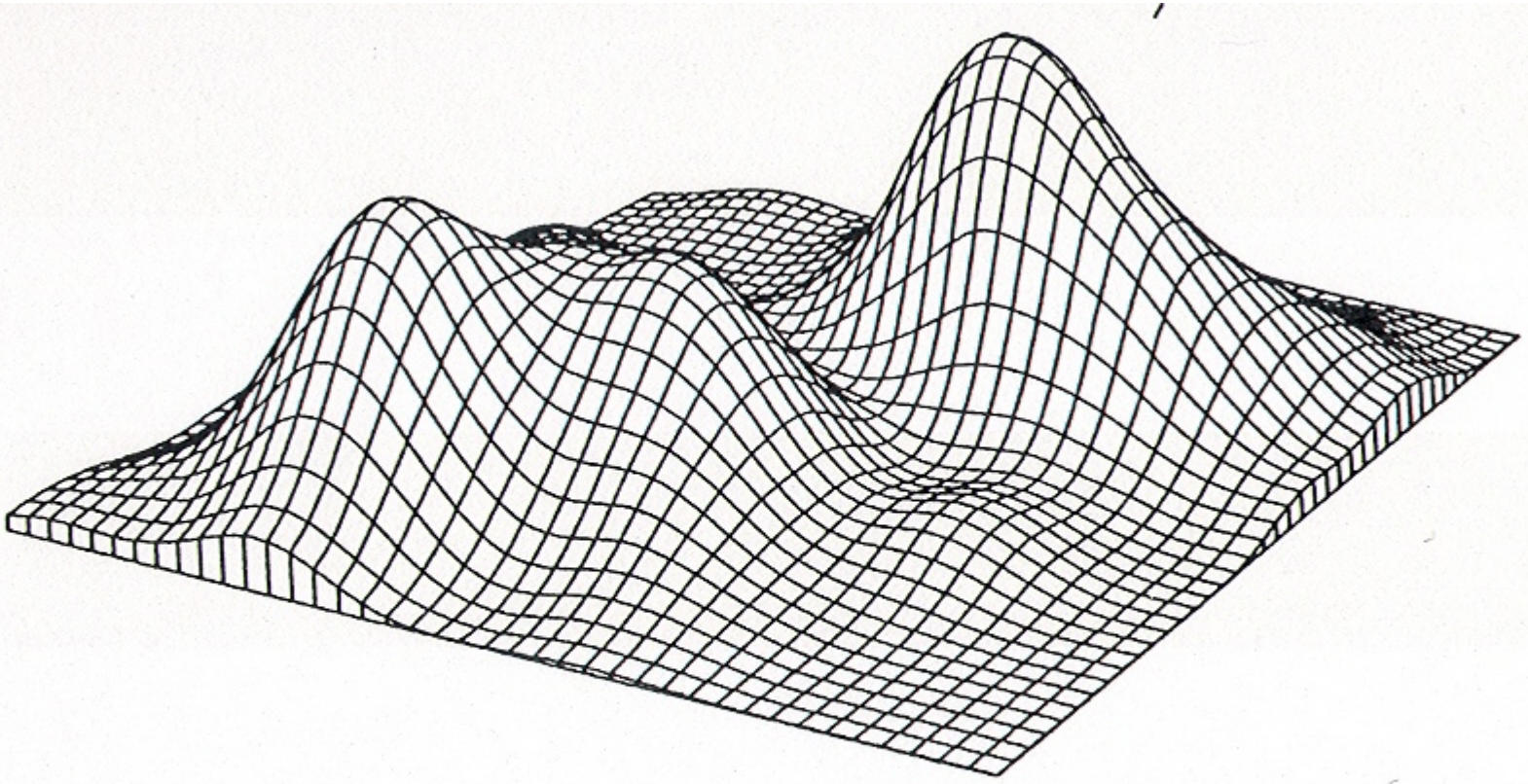


Fig. 4.3 *Density estimate, displayed on the square* $(\pm 3, \pm 3)$, *for 100 observations from bivariate normal mixture, window width 1.2.*

Parzen Density Estimation

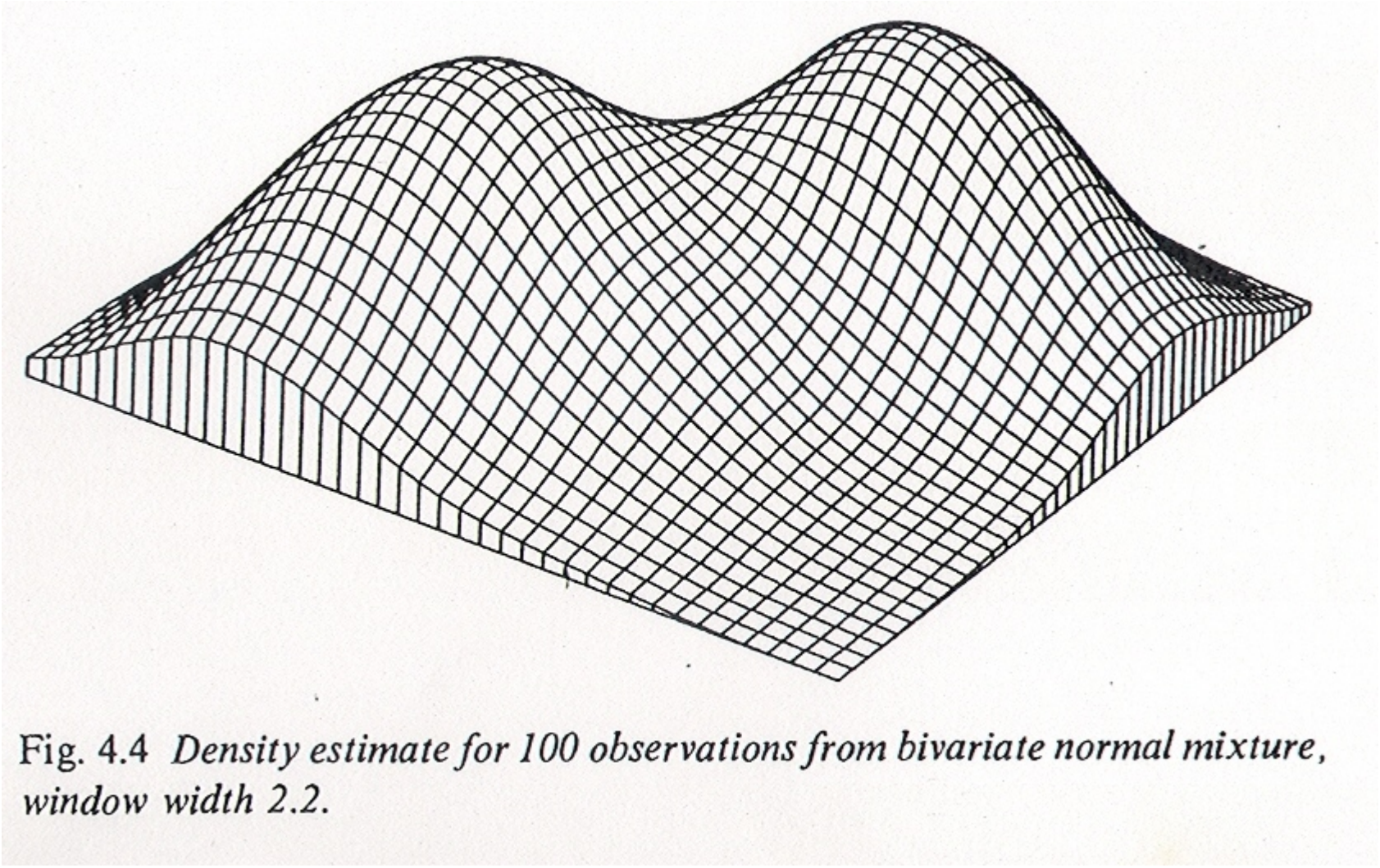


Fig. 4.4 *Density estimate for 100 observations from bivariate normal mixture, window width 2.2.*

Parzen Density Estimation

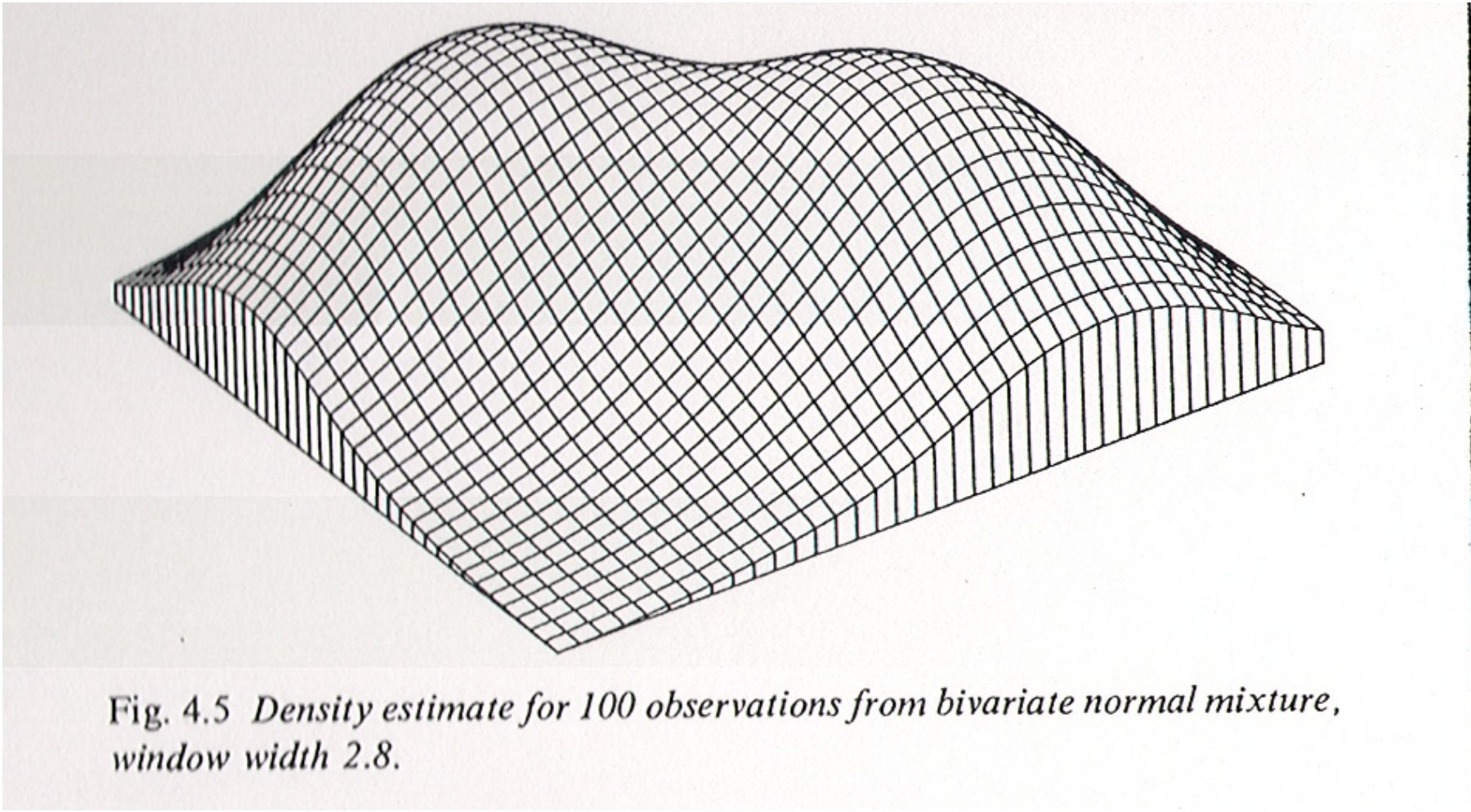


Fig. 4.5 *Density estimate for 100 observations from bivariate normal mixture, window width 2.8.*

Parzen Density Estimation

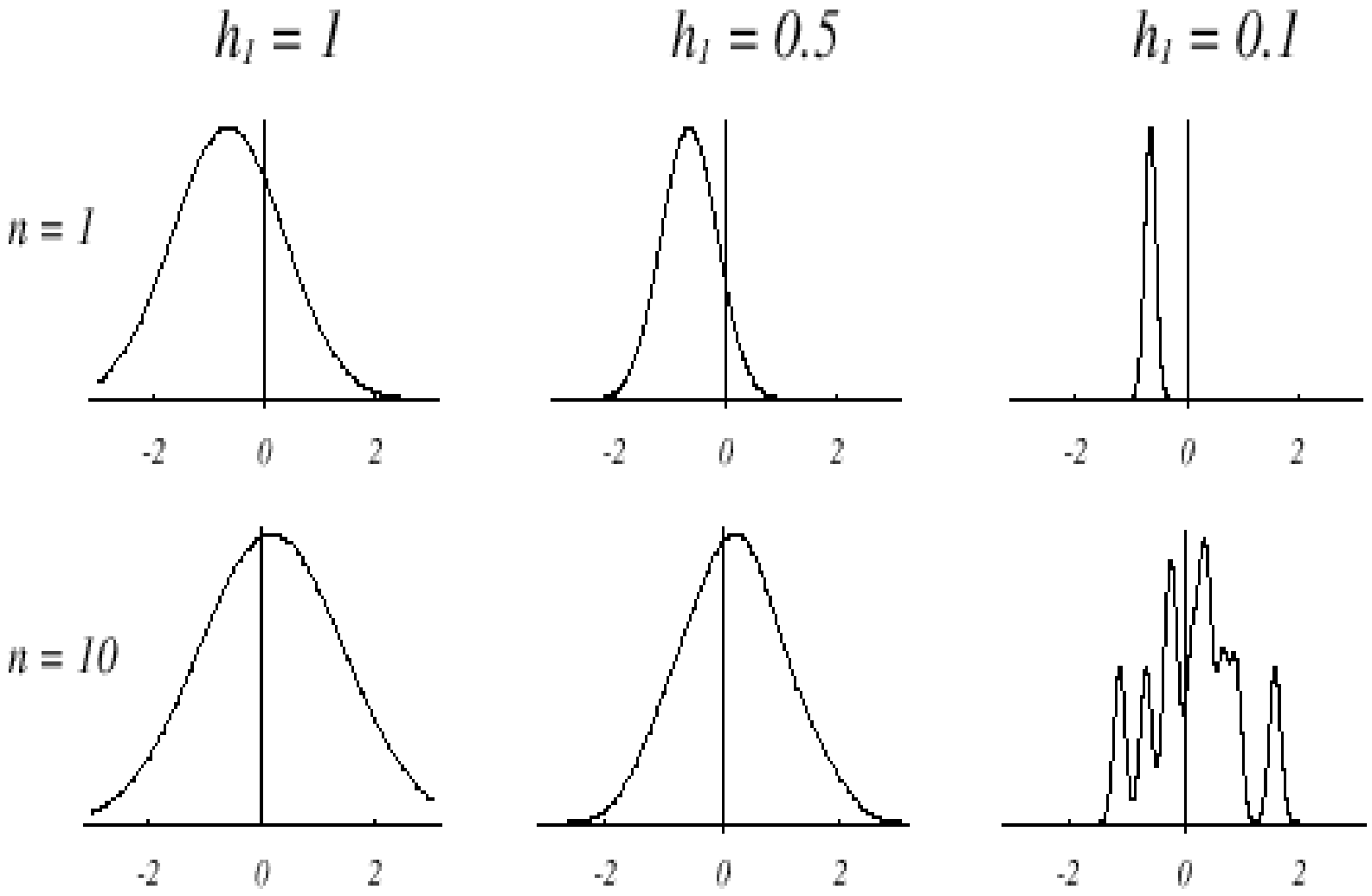
- **Numerical results:**

For n = 1 and h_1=1

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

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

Parzen Density Estimation



Parzen Density Estimation

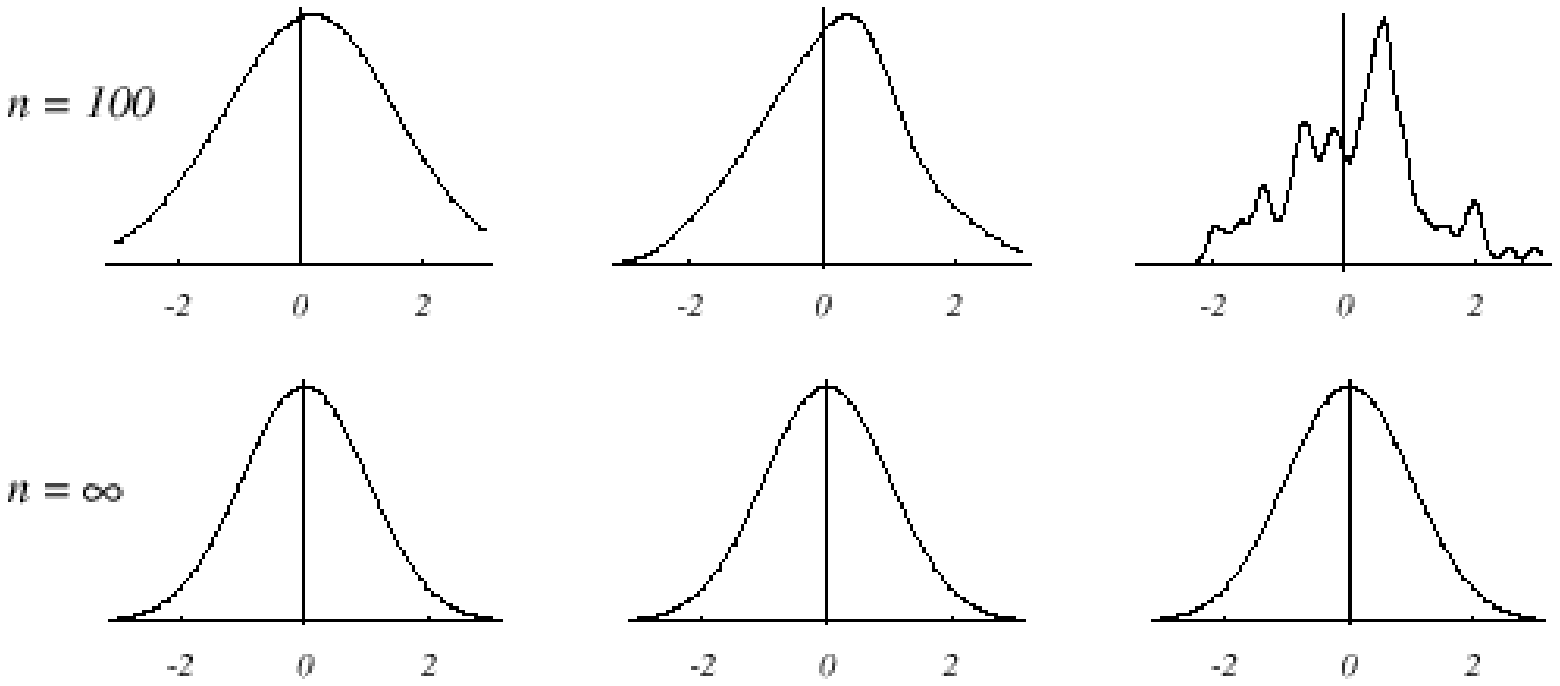
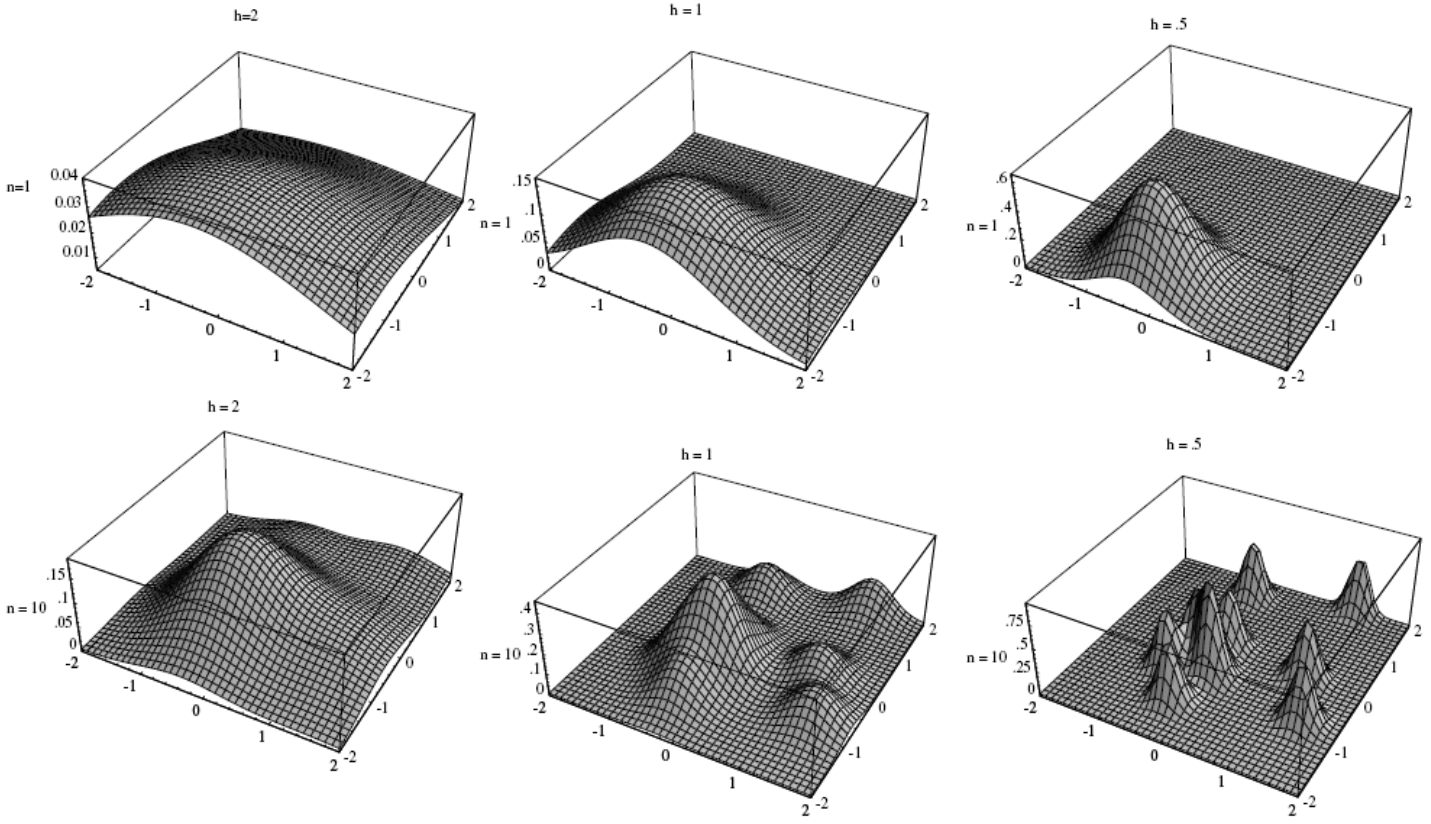


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless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

Parzen Density Estimation

Analogous results are also obtained in two dimensions:



Parzen Density Estimation

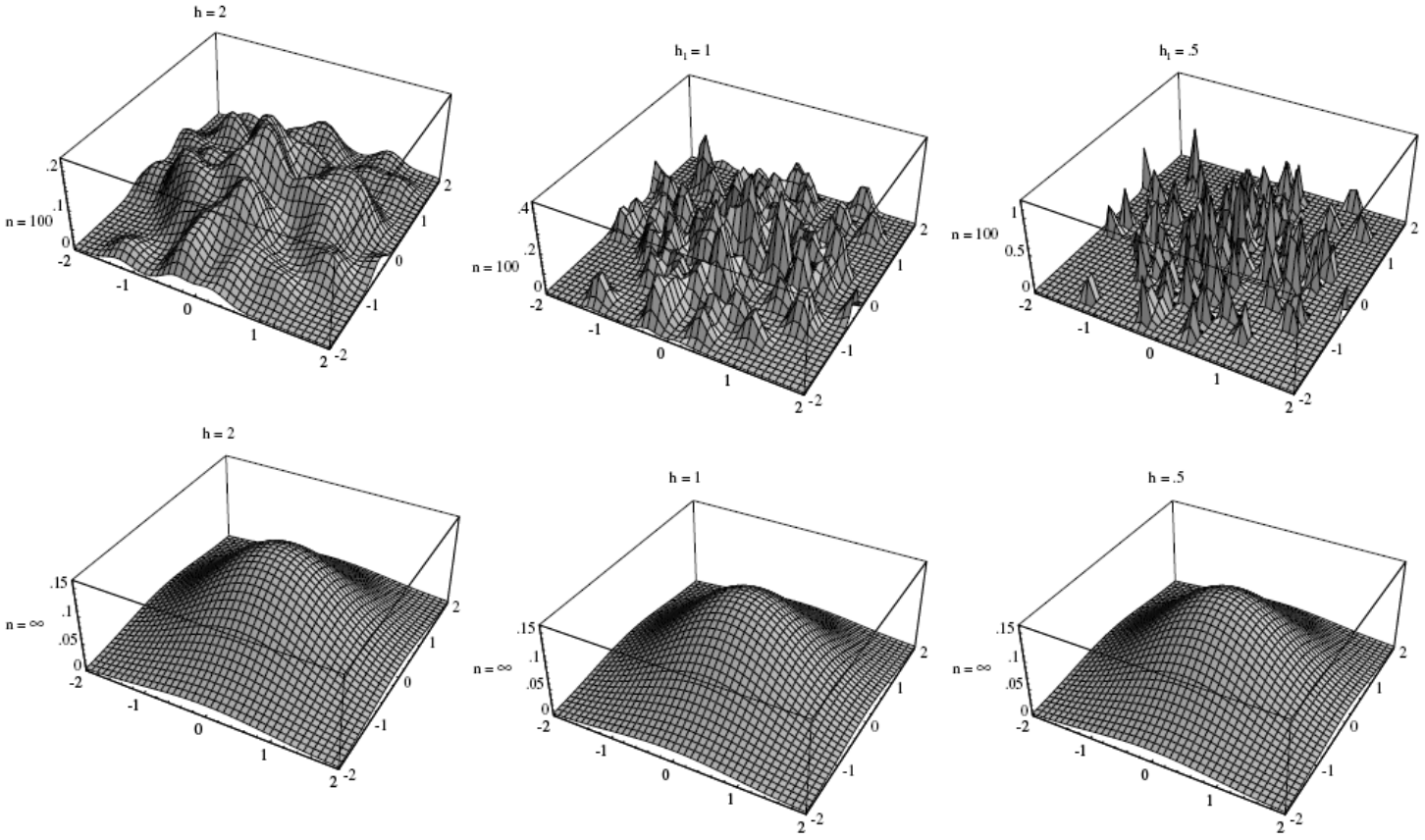
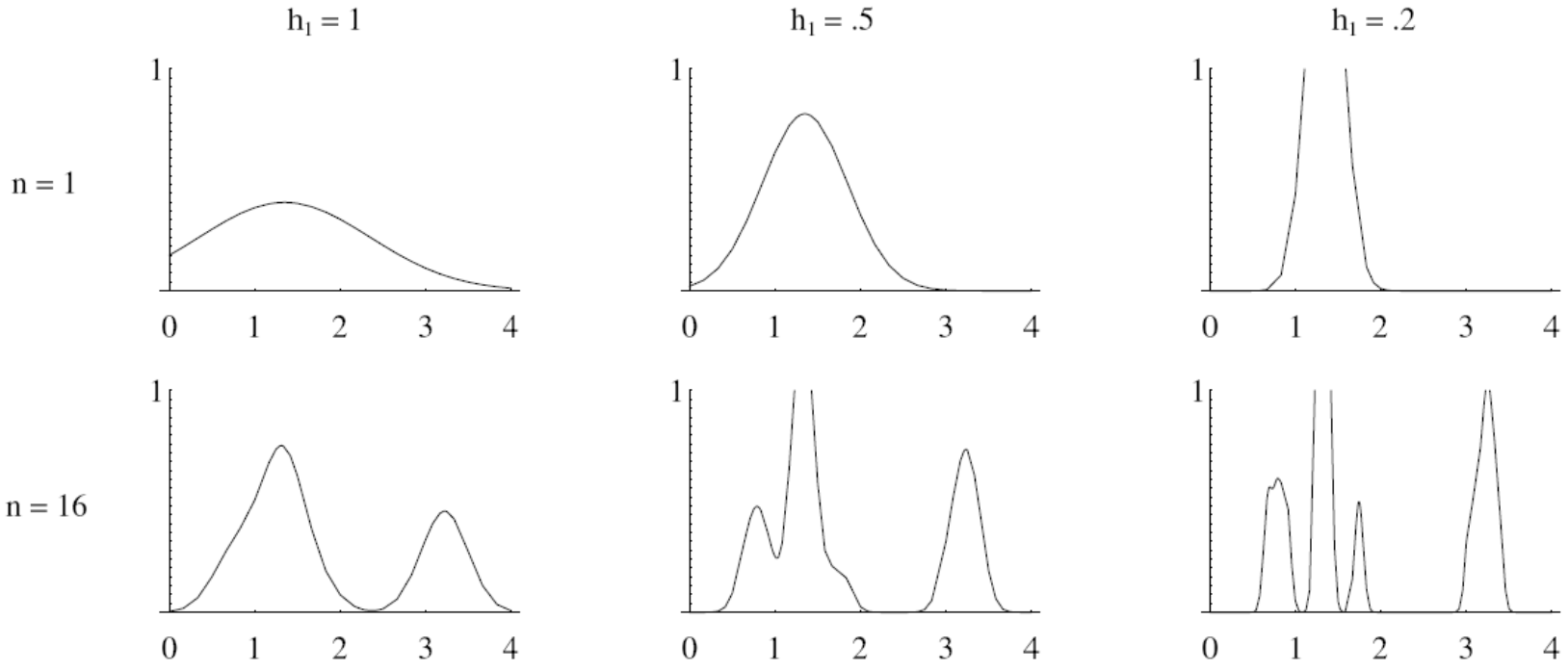


Figure 4.6: Parzen-window estimates of a bivariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true generating distribution),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

Parzen Density Estimation

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



Parzen Density Estimation

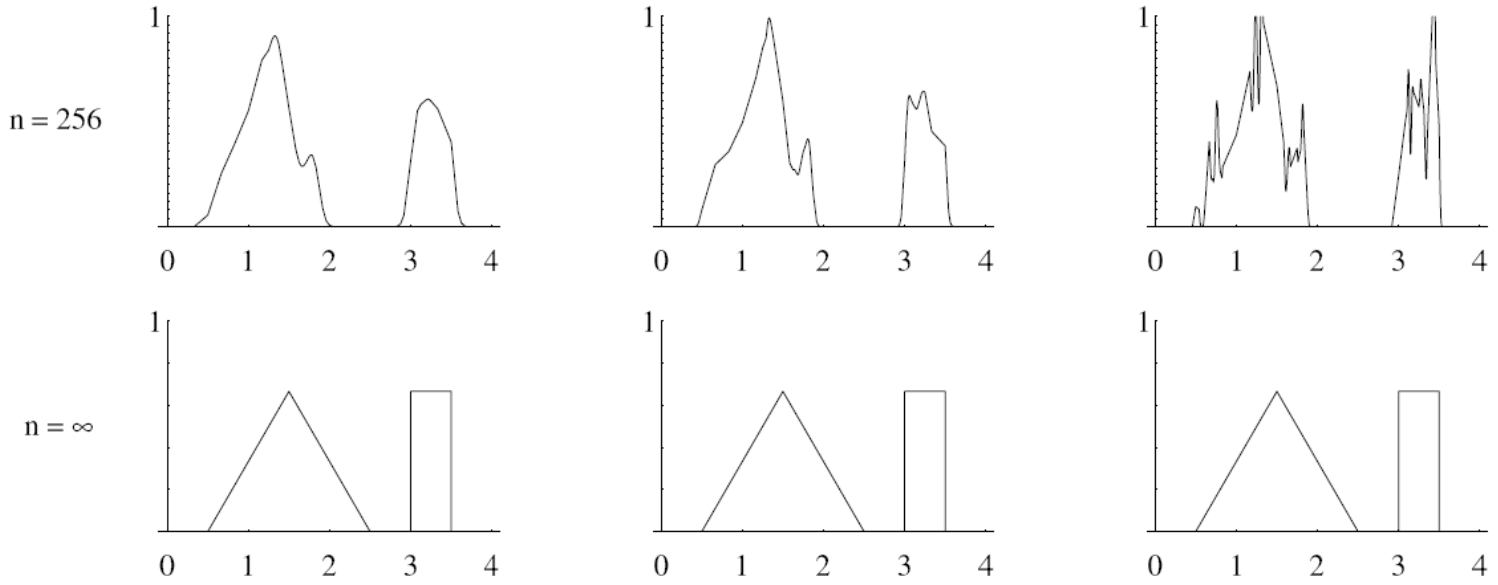


Figure 4.7: Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true generating distribution), regardless of window width h.

Parzen Density Estimation

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 the window function as illustrated in the following figure

Parzen Density Estimation

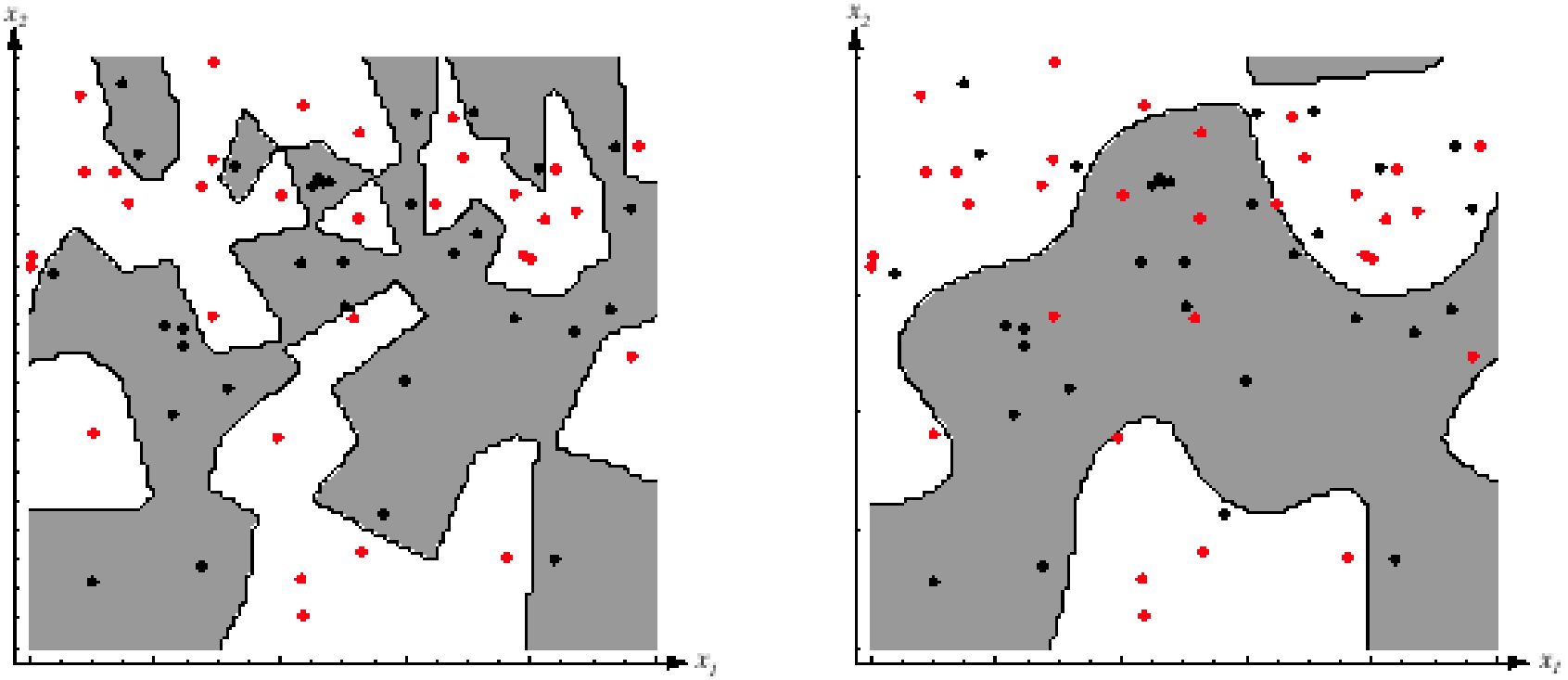


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the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

Table 4.2 *Sample size required (accurate to about 3 significant figures) to ensure that the relative mean square error at zero is less than 0.1, when estimating a standard multivariate normal density using a normal kernel and the window width that minimizes the mean square error at zero*

Dimensionality	*Required sample size*
1	4
2	19
3	67
4	223
5	768
6	2 790
7	10 700
8	43 700
9	187 000
10	842 000

Parzen Density Estimation – Exercise

Let $p(x) \sim U(0, a)$ be uniform from 0 to a, and let a Parzen window be defined as $\varphi(x) = e^{-x}$ for $x > 0$ and 0 for $x \leq 0$.

Show that the mean of such a Parzen-window estimate is given by

$$\bar{p}_n(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{a}(1 - e^{-x/h_n}) & 0 \leq x \leq a \\ \frac{1}{a}(e^{a/h_n} - 1)e^{-x/h_n} & a \leq x. \end{cases}$$

K_n Nearest Neighbor Estimation

Goal: a solution for the problem of the unknown "best" window function

Let the cell volume be a *function of the training data*.
Center a cell about x and let it grow until it captures k_n samples ($k_n = f(n)$)

k_n are called the k_n nearest-neighbors of x

Two possibilities can occur:

Density is high near x. Therefore, the cell will be small which provides a good resolution.

Density is low. Therefore, the cell will grow large and not stop until higher density regions are reached.

We can obtain a family of estimates by setting $k_n = k_1\sqrt{n}$ and choosing different values for k_1

K_n Nearest Neighbor Estimation

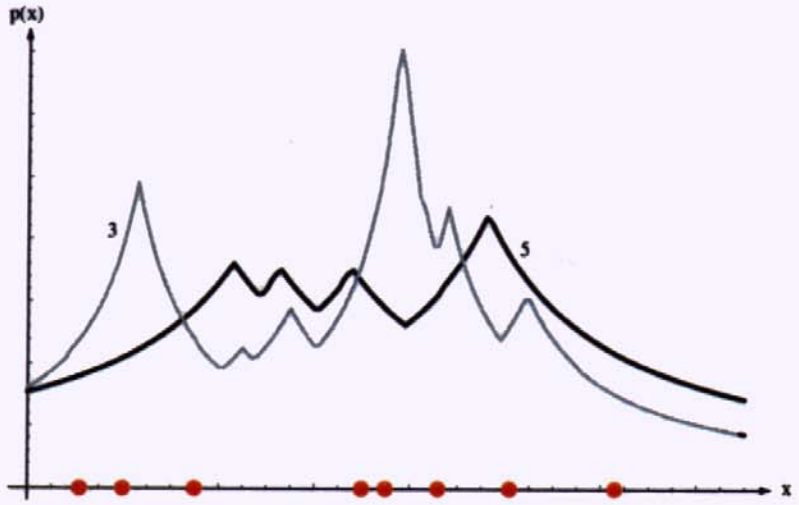


Figure 4.10: Eight points in one dimension and the k-nearest-neighbor density estimates, for $k = 3$ and 5. Note especially that the discontinuities in the slopes in the estimates generally occur *away* fom the positions of the points themselves.

K_n Nearest Neighbor Estimation

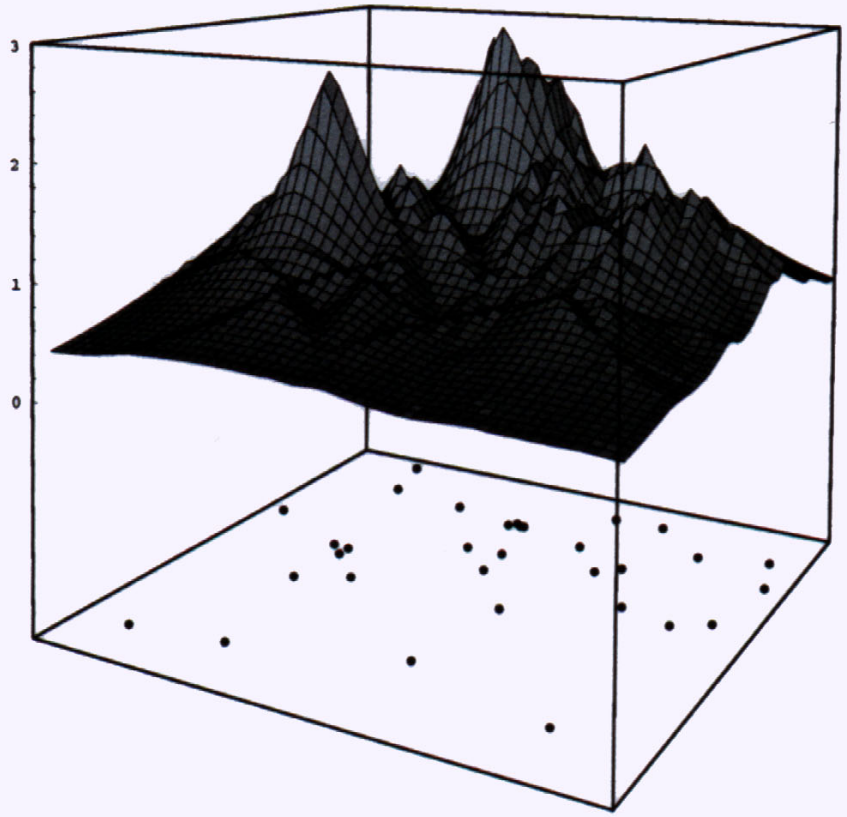


Figure 4.11: The k-nearest-neighbor estimate of a two-dimensional density for $k = 5$. Notice how such a finite n estimate can be quite "jagged," and that discontinuities in the slopes generally occur along lines away from the positions of the points themselves.

Illustration: K_n Nearest Neighbor Estimation

- **Previous example for Parzen**
- **For n = 1 and $k_n = \sqrt{n} = 1$; the estimate becomes:**

$P_n(x) = 1 / V_1$

$= 1 / 2|x-x_1|$

K_n Nearest Neighbor Estimation

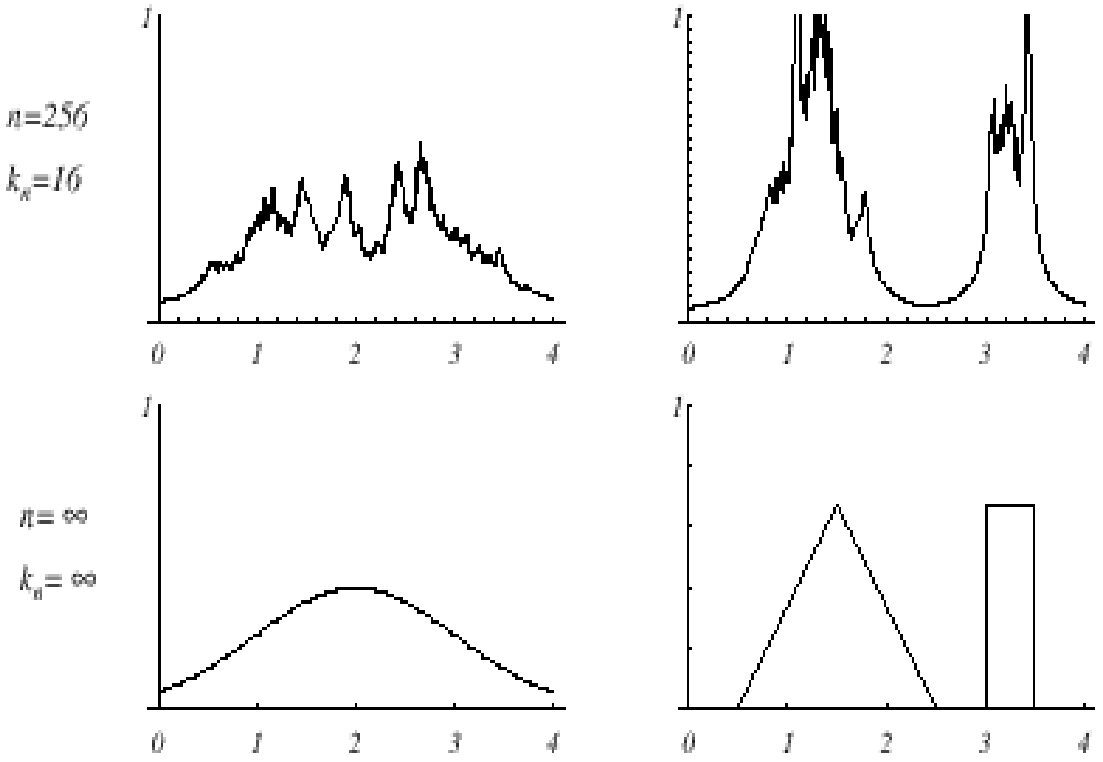
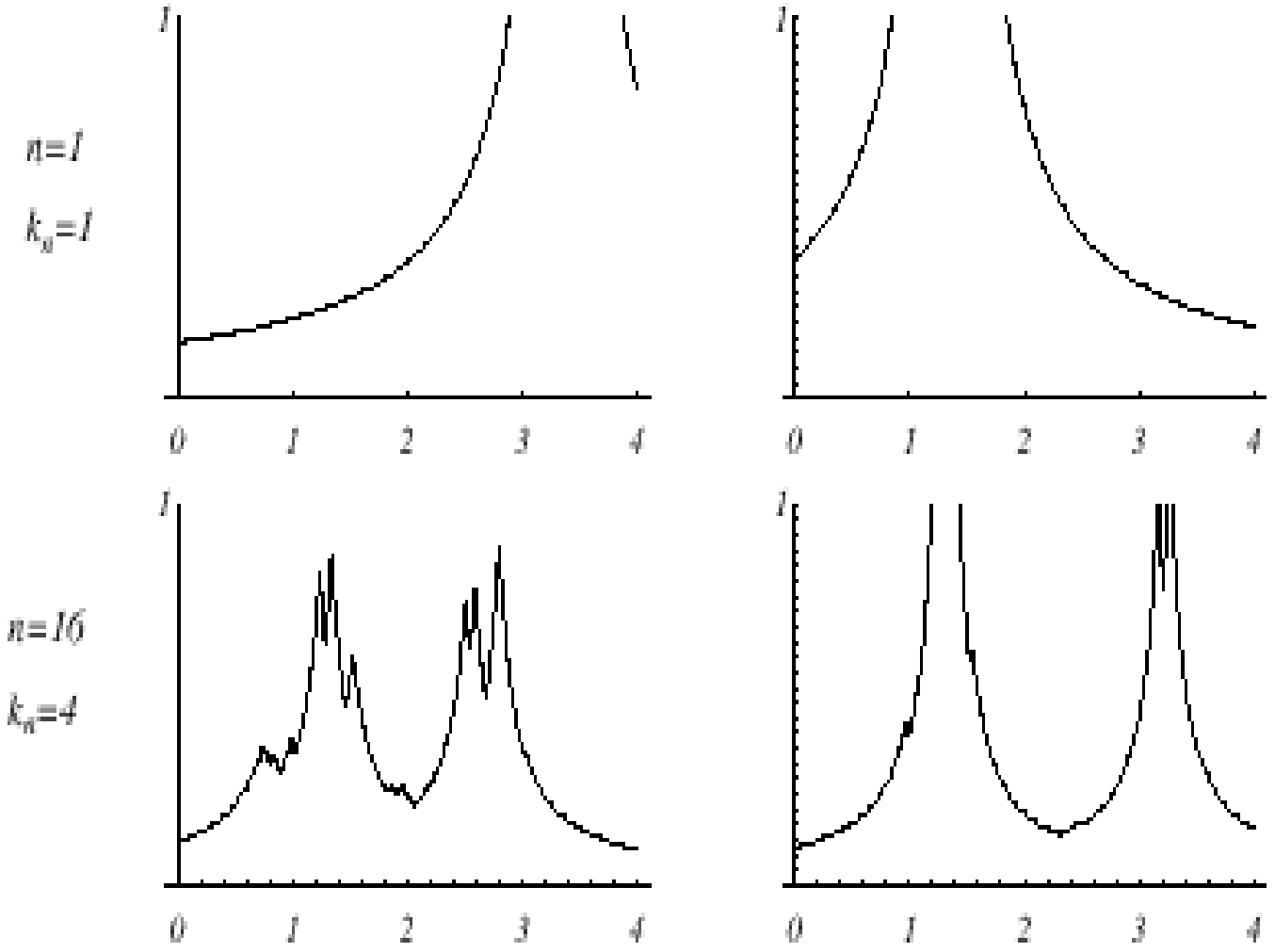


FIGURE 4.12. Several k-nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite n estimates can be quite "spiky." From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

K_n Nearest Neighbor Estimation



Estimation of a-posteriori probabilities

- Goal: estimate $P(\omega_i \mid x)$ from a set of n labeled samples
- Let us place a cell of volume V around x and capture k samples
- k_i samples amongst k turned out to be labeled ω_l then:

$$p_n(\mathbf{x}, \omega_i) = \frac{k_i/n}{V}$$

- An estimate for $P_n(\omega_i \mid x)$ is:

$$P_n(\omega_i|\mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k}$$

Estimation of a-posteriori probabilities

- k_i/k is the fraction of the samples within the cell that are labeled ω_i
- For minimum error rate, the most frequently represented category within the cell is selected
- If k is large and the cell sufficiently small, the performance will approach the best possible

The Nearest Neighbor Rule

- Let $D_n = \{x_1, x_2, \ldots, x_n\}$ be a set of n labeled prototypes
- Let $x' \in D_n$ be the closest prototype to a test point x then the *nearest-neighbor rule* for classifying x is to assign it the label associated with x'
- The nearest-neighbor rule leads to an error rate greater than the minimum possible: the Bayes rate
- If the number of prototype is large (unlimited), the error rate of the nearest-neighbor classifier is never worse than twice the Bayes rate (it can be demonstrated)
- If $n \rightarrow \infty$, it is always possible to find x' sufficiently close so that:

$$P(\omega_i \mid x') \cong P(\omega_i \mid x)$$

Example

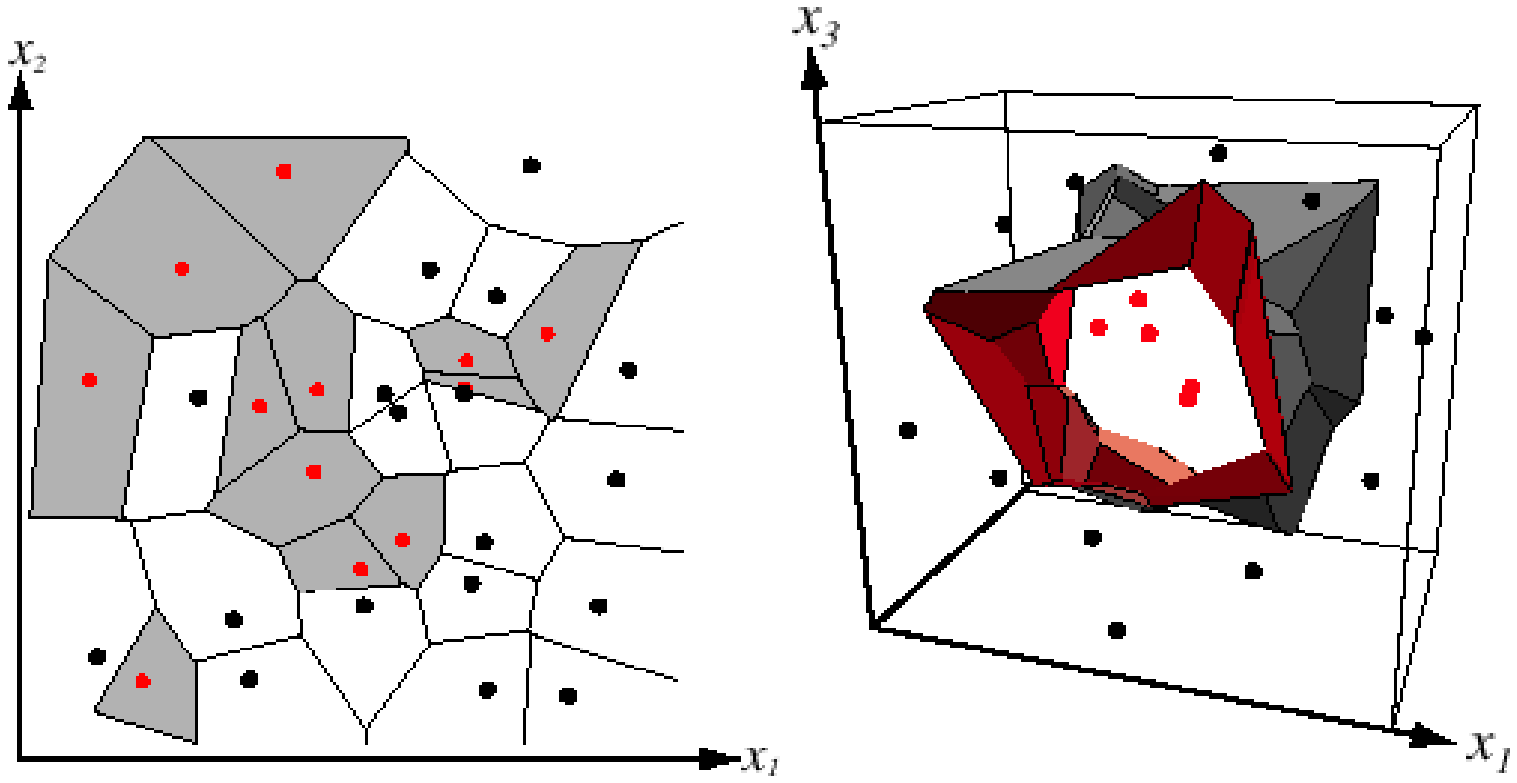
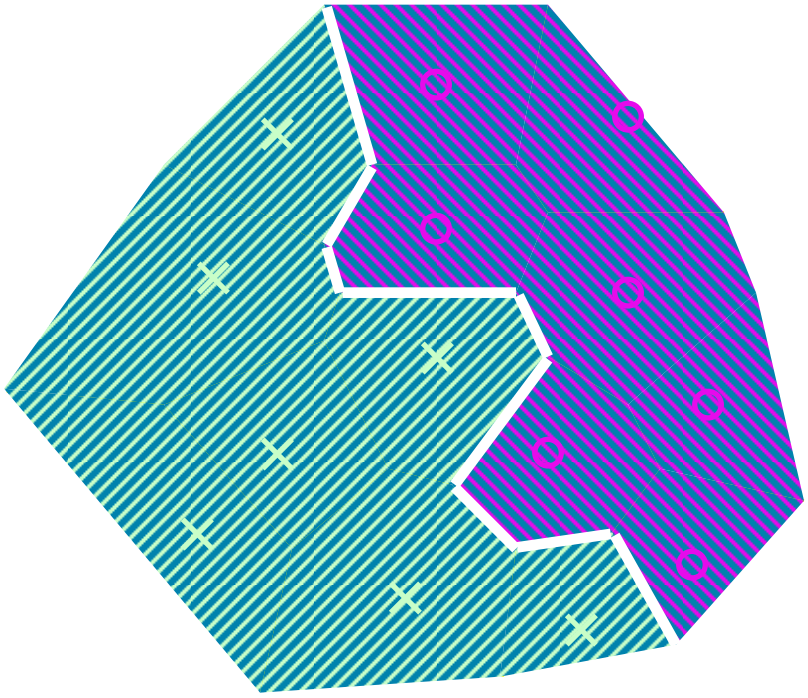


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Decision Boundary for NNR

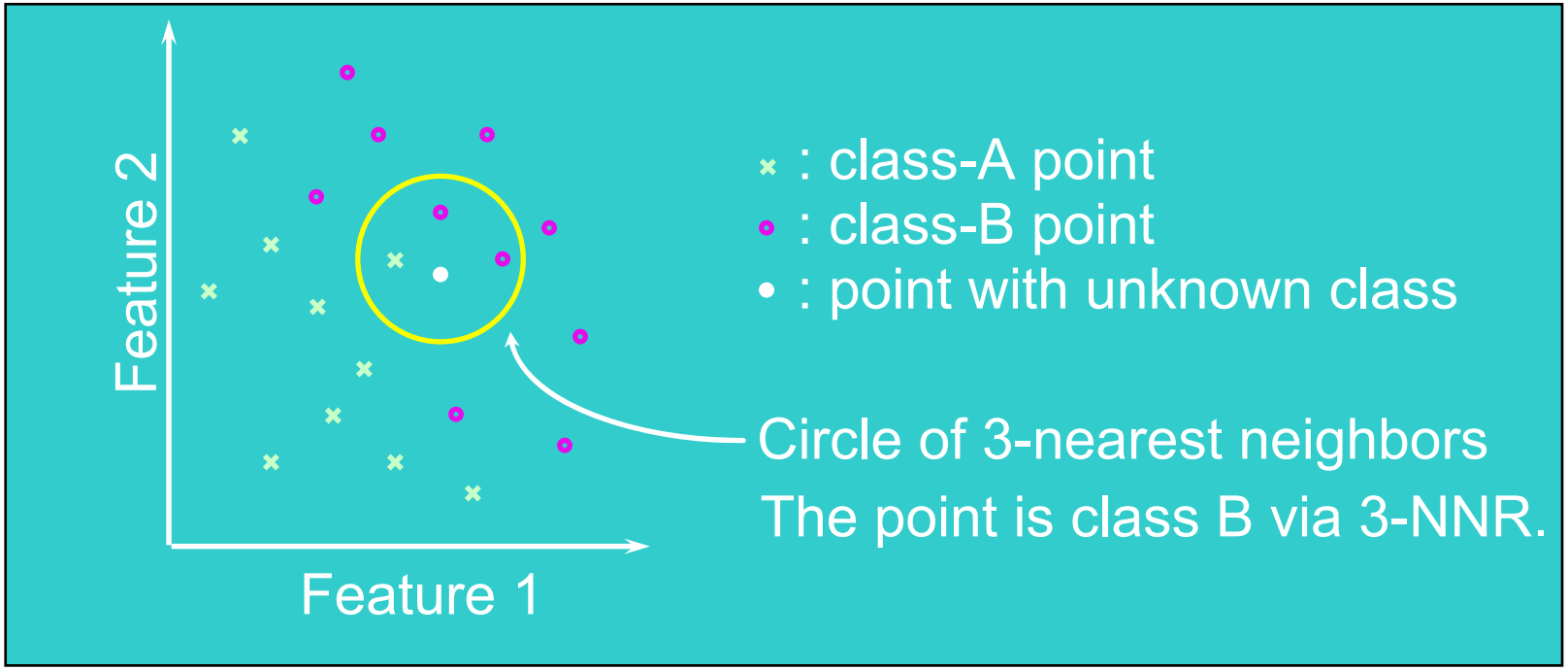
Voronoi diagram: piecewise linear boundary



K-Nearest Neighbor Rule (K-NNR)

Steps:

1. Find the first k nearest neighbors of a given point.
2. Determine the class of the given point by a voting mechanism among these k nearest neighbors.



The k-Nearest Neighbor Rule

- **Goal: Classify x by assigning it the label most frequently represented among the k nearest samples and use a voting scheme**

The k-Nearest Neighbor Rule

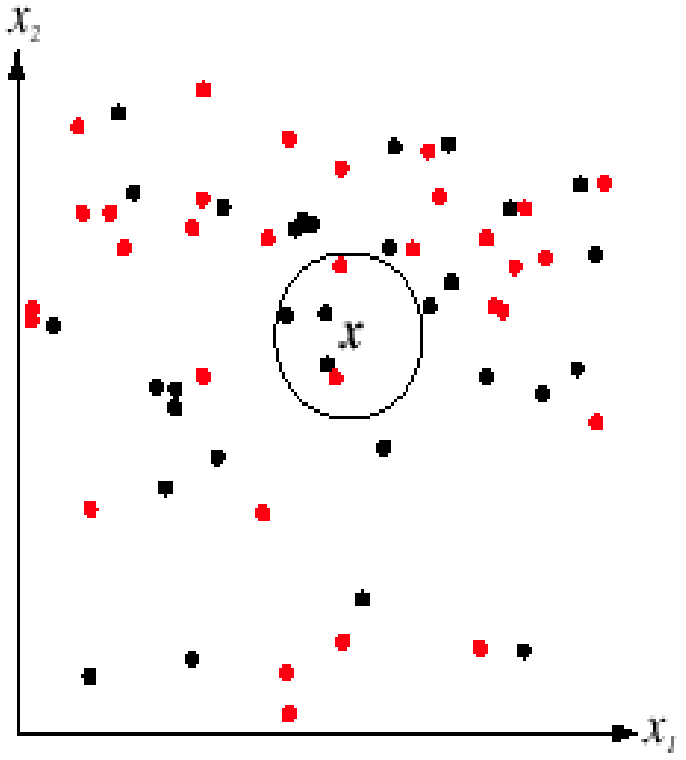


FIGURE 4.15. The k-nearest-neighbor query starts at the test point $\mathbf{x}$ and grows a spherical region until it encloses k training samples, and it labels the test point by a majority vote of these samples. In this $k = 5$ case, the test point $\mathbf{x}$ would be labeled the category of the black points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example

- **Example: k = 3 (odd value) and x = $(0.10, 0.25)^T$**

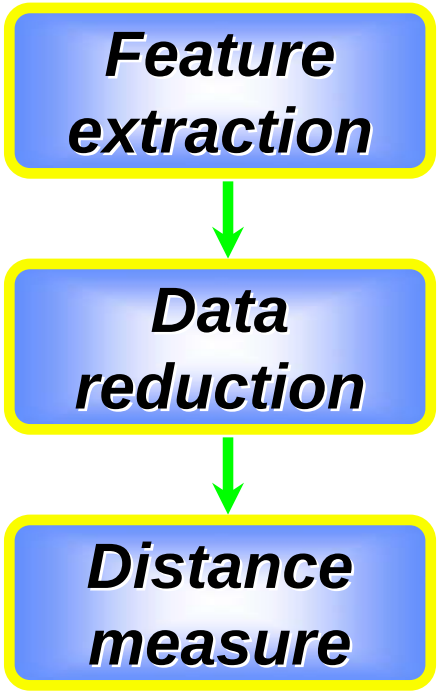
Prototype	Labels
(0.15,0.35)	ω_1
(0.10,0.28)	ω_2
(0.09,0.30)	ω_5
(0.12,0.20)	ω_2

- **Closest vectors to x with their labels are: {(0.10, 0.28, ω_2); (0.12, 0.20, ω_2); (0.09, 0.30, ω_5)}**

One voting scheme assigns the label ω_2 to x since ω_2 is the most frequently represented

Flowchart for Nearest Neighbor

General flowchart:



Particle example:

From image to features

None

Distance Computation

Example

A real-world application, word pronunciation, is used to exemplify how the classifier learns and classifies:

http://demo.viidea.com/aaai07_bosch_knnc/

The k-Nearest Neighbor Rule

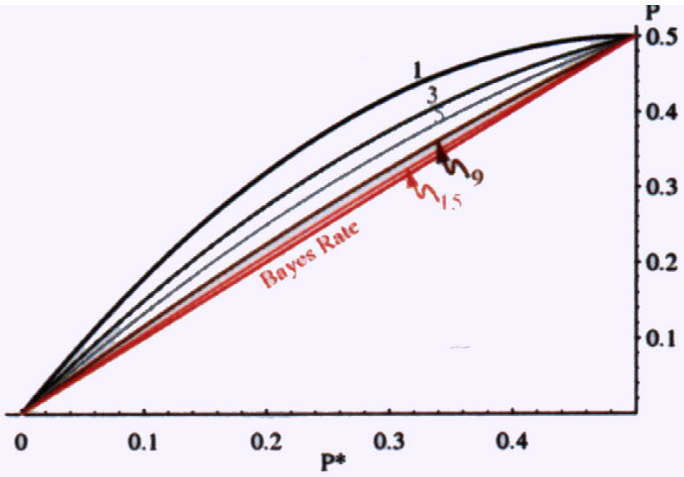


Figure 4.16: The error-rate for the k-nearest-neighbor rule for a two-category problem is bounded by $C_k(P^*)$ in Eq. 55. Each curve is labelled by k; when $k = \infty$, the estimated probabilities match the true probabilities and thus the error rate is equal to the Bayes rate, i.e., $P = P^*$.

The Error for the k-Nearest Neighbor Rule

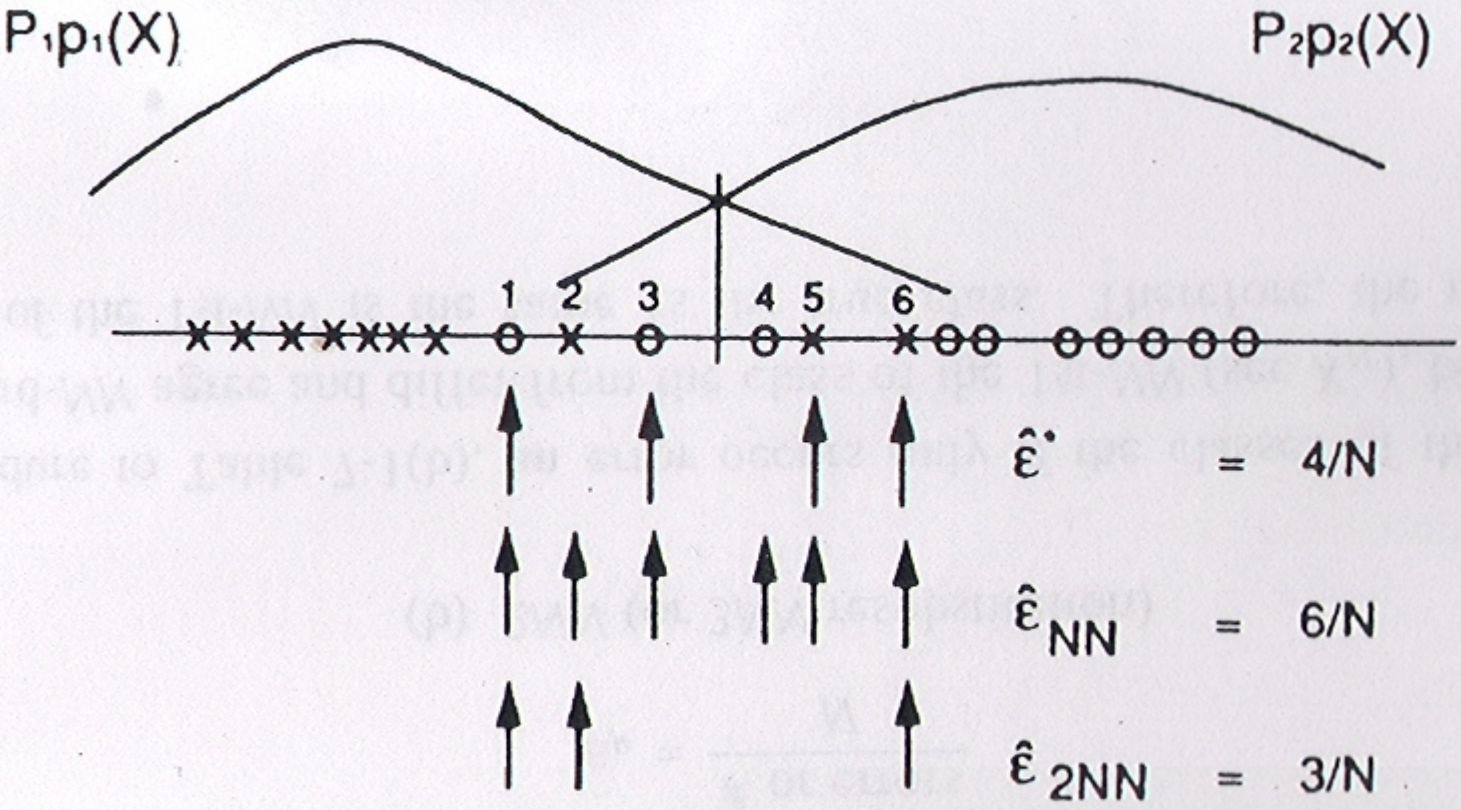


Fig. 7-3 Example of *kNN* classification.

Distance Metrics

L-d norms (aka Minkowski distance):

$$d_d(\vec{x}) = \left(\sum_i |x_i|^d \right)^{1/d}$$

- **d = 1: City block distance, Manhattan metric, taxicab distance** $d_1(\vec{x}) = \sum_i |x_i|$
- **d = 2: Euclidean distance** $d_2(\vec{x}) = \sqrt{\sum_i |x_i|^2}$
- **d = inf: maximum distance metric**

$$d_\infty(\vec{x}) = \max_i |x_i|$$

Error Estimation – k-NN

Resubstitution error:

Data		1st *NN*		2nd *NN*		3rd *NN*		Classification	Correct or Error
	ω		ω		ω		ω		
X_1	1	X_1	1	X_3	1	X_{10}	1	1	Correct
X_2	2	X_2	2	X_{18}	1	X_{25}	2	2	Correct
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
X_N	1	X_N	1	X_{35}	2	X_{536}	2	2	Error

$$\hat{\varepsilon}_R = \frac{\#\text{ or errors}}{N}$$

(b) 2*NN* (or 3*NN* resubstitution)

Error Estimation – k-NN

Leave one out error:

3*NN* ERROR ESTIMATION PROCEDURES

Data		1st *NN*		2nd *NN*		3rd *NN*		Classification	Correct or Error
	ω		ω		ω		ω		
X_1	1	X_3	1	X_{10}	1	X_{23}	2	1	Correct
X_2	2	X_{18}	1	X_{25}	2	X_{36}	1	1	Error
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
X_N	1	X_{35}	2	X_{536}	2	X_{366}	2	2	Error

$$\hat{\varepsilon}_L = \frac{\#\text{ of errors}}{N}$$

(a) 3*NN* (or 3*NN* leave-one-out)