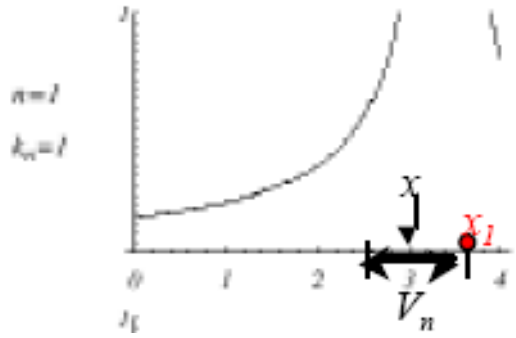
- K_n - Nearest neighbor estimation
  - Goal: a solution for the problem of the unknown "best" window function
    - Let the cell volume be a function of the training data
    - Center a cell about x and let it grows until it captures k_n samples $(k_n = f(n))$
    - k_n are called the k_n nearest-neighbors of x
  - 2 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 stop until higher density regions are reached

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

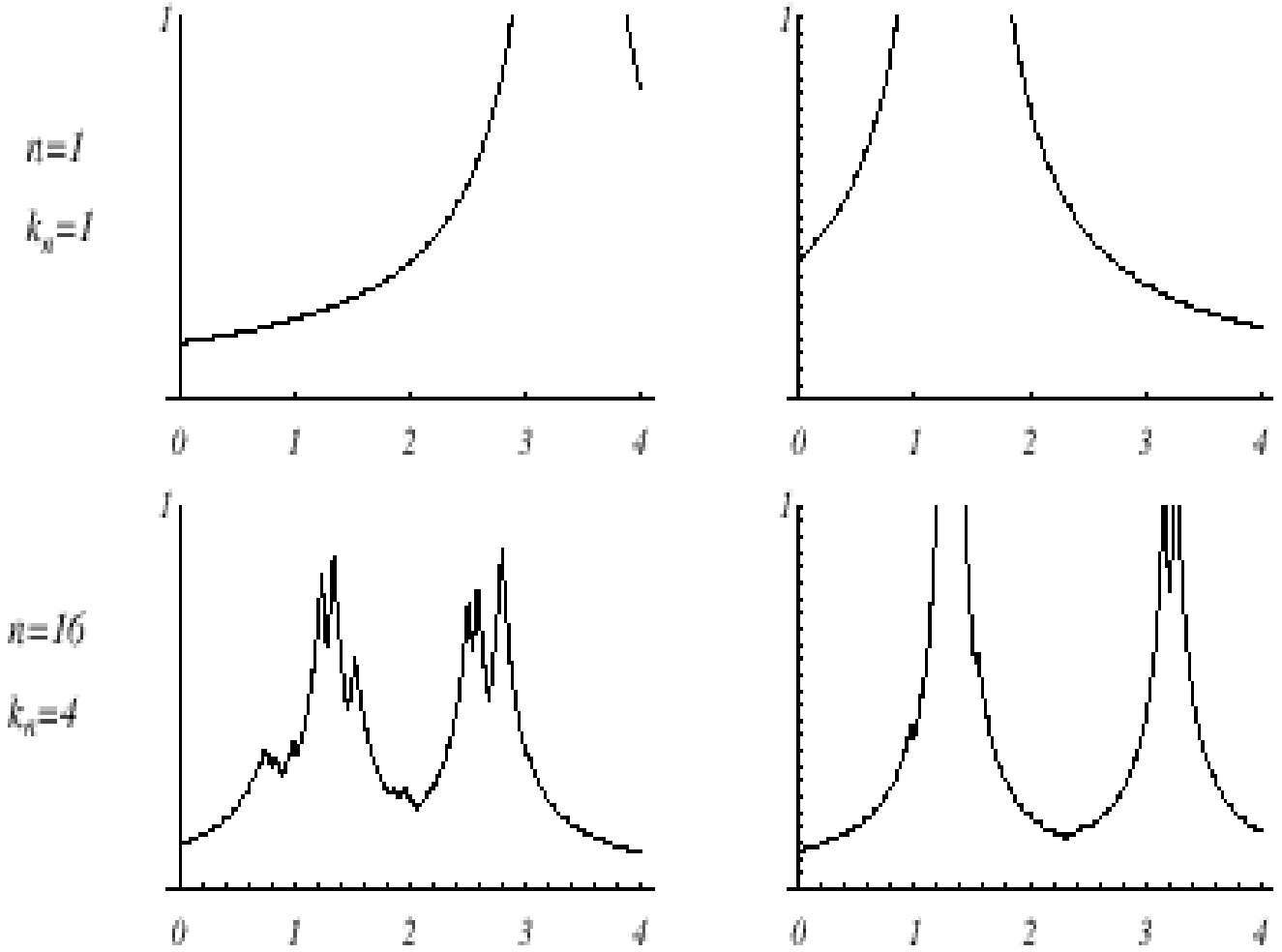
One dimensional case

For $k_n = \sqrt{n} = 1$; the estimate becomes:

$$P_n(x) = k_n / n.V_n = 1 / V_1 = 1 / 2|x-x_1|$$



a Gaussian and a bimodal distribution.



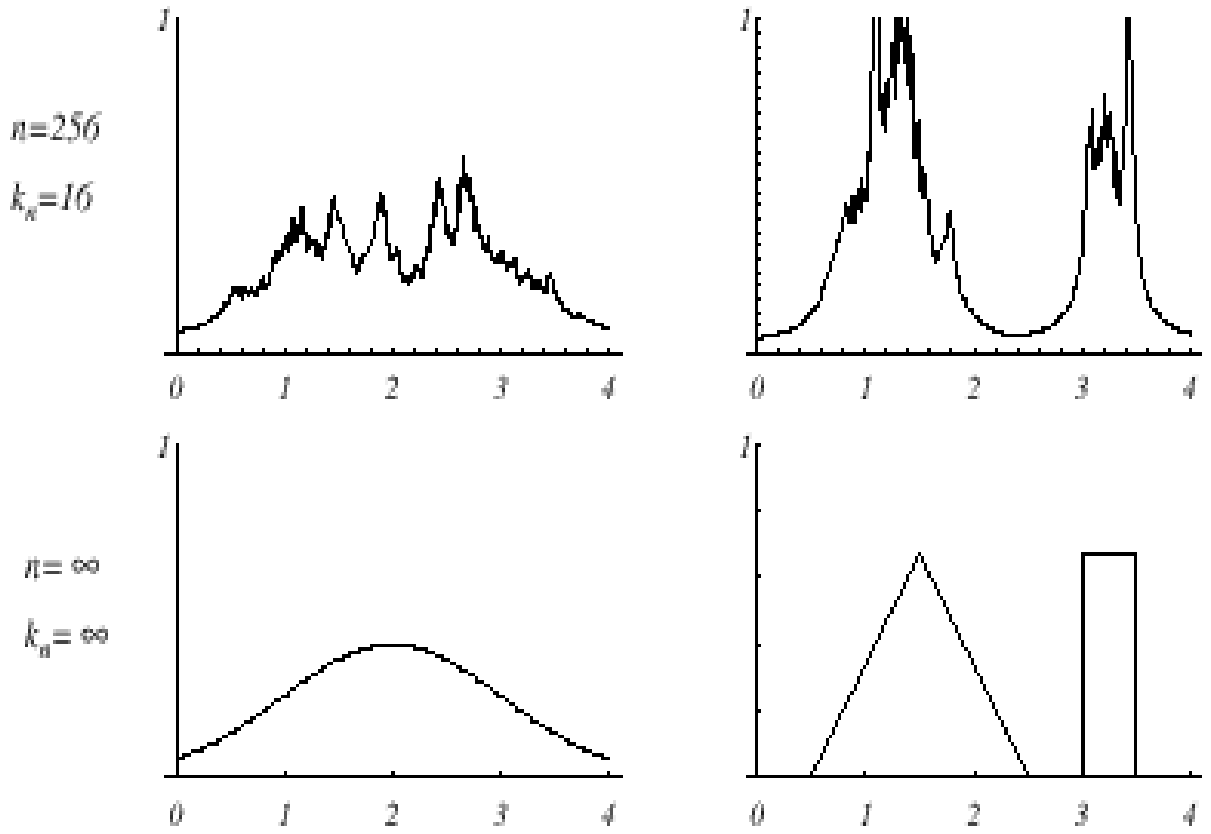
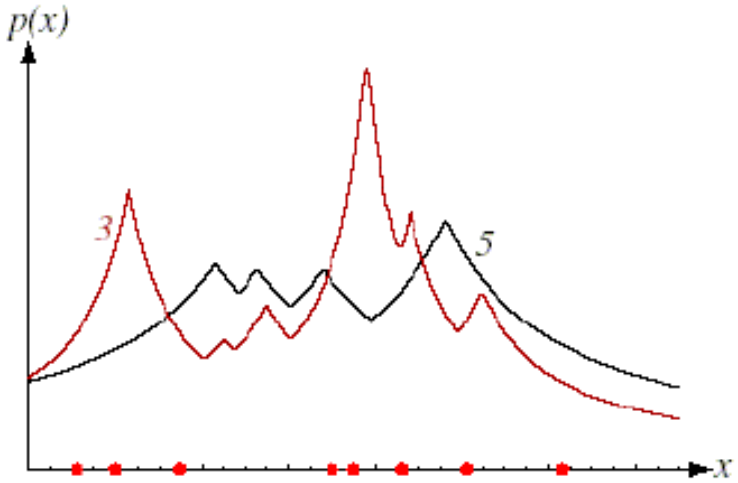
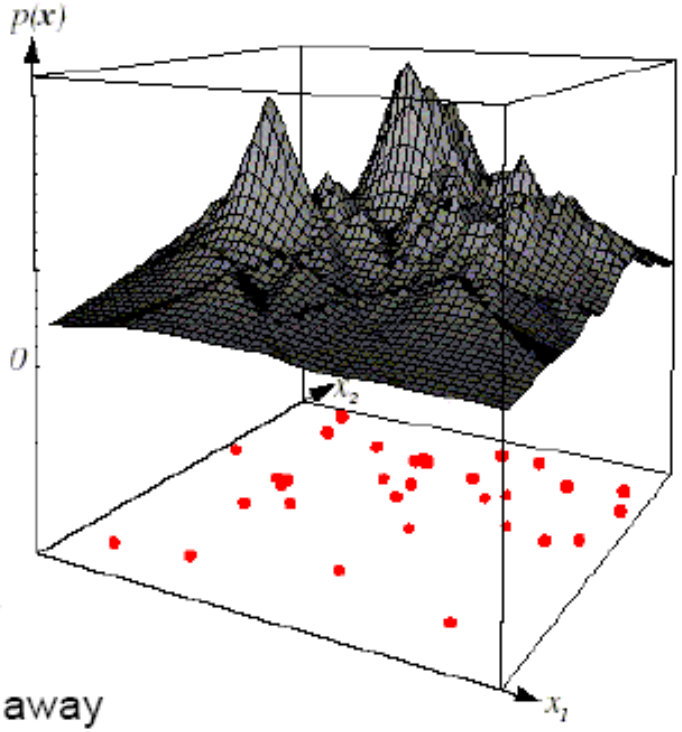


FIGURE 4.12. Several *k*-nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite *n* estimates can be quite "spiky." From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.

- kNN estimate for $k = 3$ and $k = 5$ when data consists of 8 points in one-dimensional feature space.



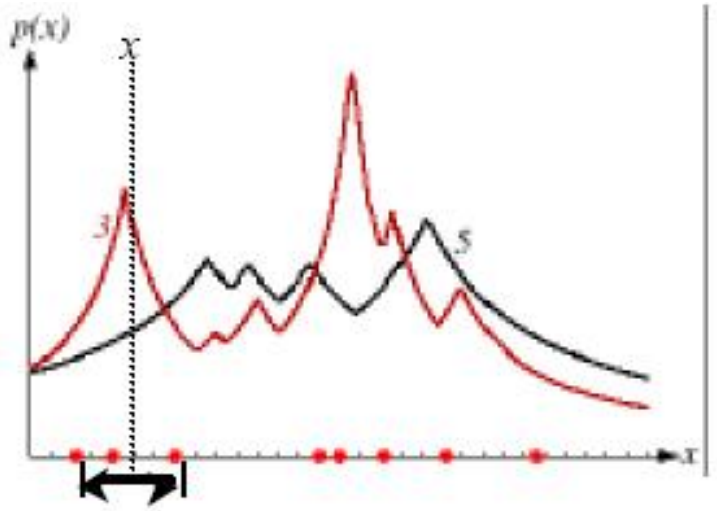
- kNN estimate for $k = 5$ when data consists of points in a 2D feature space.



- Observations:
  - While $p_n(x)$ is a continuous function, its gradient is not.
  - Slope discontinuities occur along lines away from the positions of the prototypes points.

$$k_8 = 3$$

$$p_8(x) \approx \frac{3/8}{V}$$



- Estimation of a-posteriori probabilities
  - Goal: estimate $P(w_i \mid x)$ from a set of n labeled samples
    - Let's place a cell of volume V around x and capture k samples
    - k_i samples amongst k turned out to be labeled w_i then:

$$p_n(x, w_i) = k_i / n.V$$

An estimate for $p_n(w_i \mid x)$ is:

$$p_n(w_i \mid x) = \frac{p_n(x, w_i)}{\sum_{j=1}^{j=c} p_n(x, w_j)} = \frac{k_i}{k}$$

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

$$p_n(\omega_i \mid \mathbf{x}) = \frac{P(\omega_i)\, p_n(\mathbf{x} \mid \omega_i)}{p_n(x)} = \frac{P(\omega_i)\, p_n(\mathbf{x} \mid \omega_i)}{\sum_{j=1}^{c} P(\omega_j)\, p_n(\mathbf{x} \mid \omega_j)}$$

$$p_n(\omega_i \mid \mathbf{x}) = \frac{\frac{k_i/n}{V}}{\sum_{j=1}^{c} \frac{k_j/n}{V}} = \frac{k_i}{k}$$

- k_i/k is the fraction of the samples within the cell that are labeled w_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

q The nearest –neighbor rule

- q Let $D_n = \{x_1, x_2, ..., x_n\}$ be a set of n labeled prototypes
- q Let $x' \hat{I} D_n$ 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'
- q The nearest-neighbor rule leads to an error rate greater than the minimum possible: the Bayes rate
- q 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!)
- q If n ® ¥, it is always possible to find x' sufficiently close so that:

$$P(w_i \mid x') @ P(w_i \mid x)$$

Example:

$x = (0.68, 0.60)^t$

Prototypes	Labels	A-posteriori probabilities estimated
(0.50, 0.30)	w_2	*0.25*
	w_3	*0.75*
(0.70, 0.65)	w_5	*0.70*
	w_6	*0.30*

Decision: w_5 is the label assigned to x

If $P(w_m \mid x) \approx 1$, then the nearest neighbor selection is almost always the same as the Bayes selection

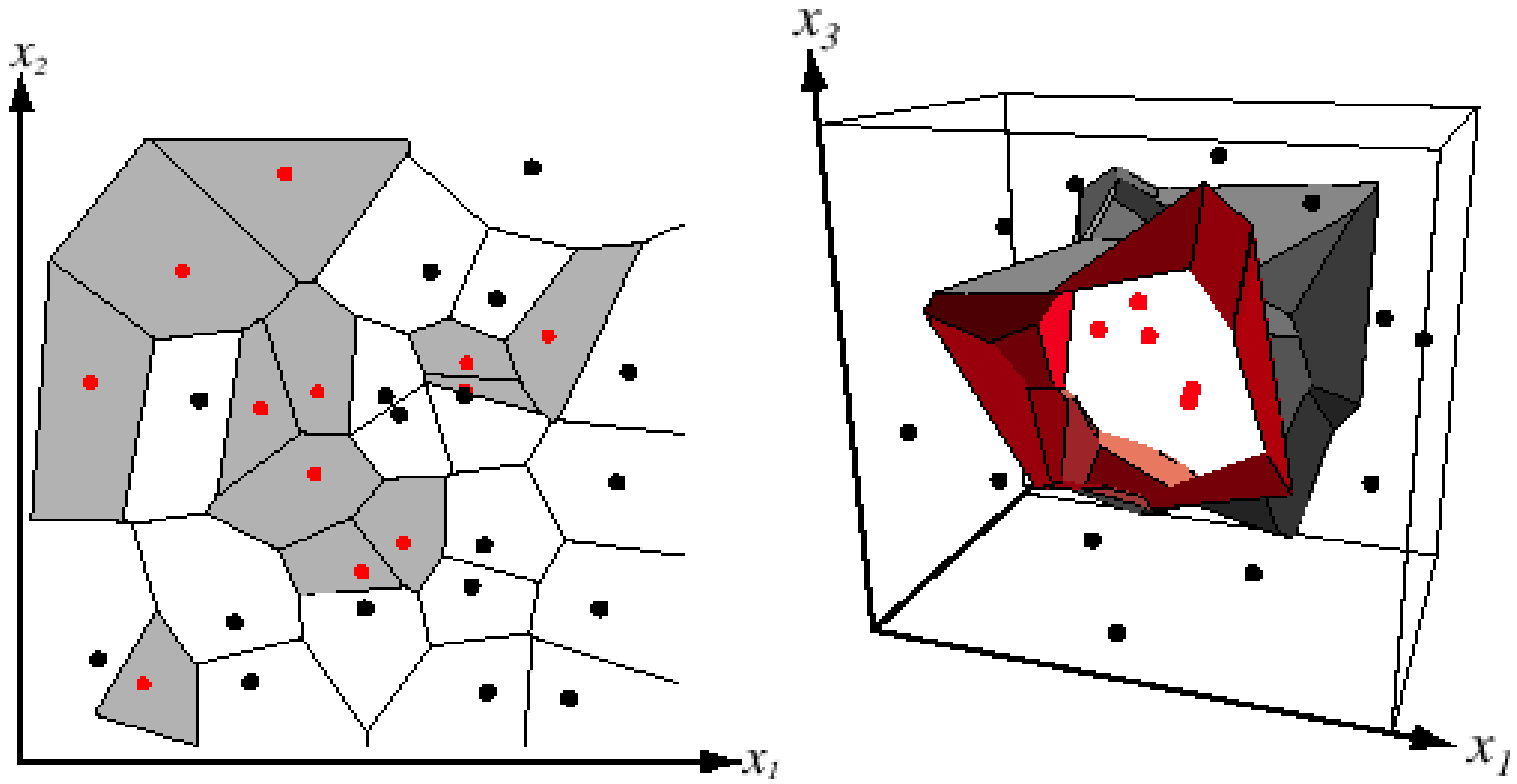


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal.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

- 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

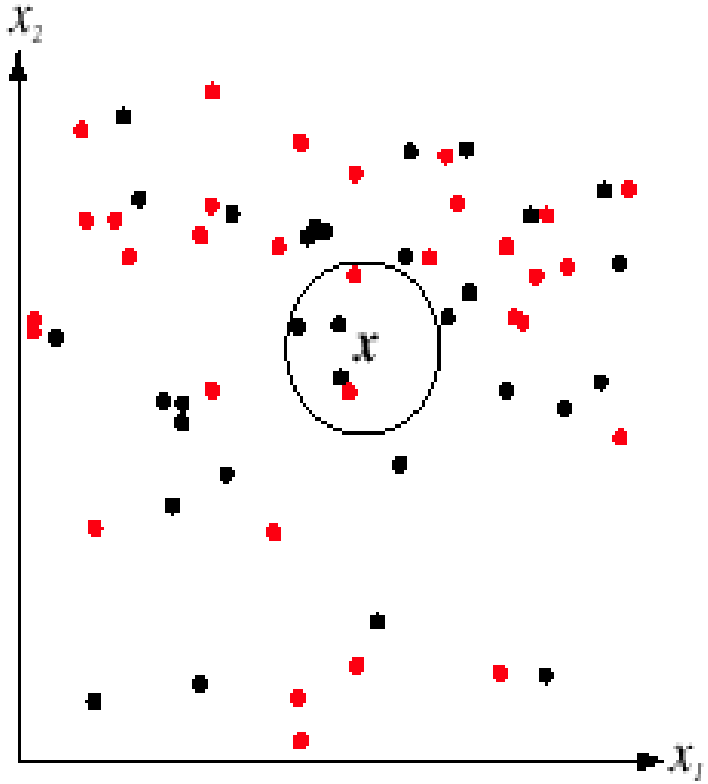


FIGURE 4.15. The k-nearest-neighbor query starts at the test point $\mathbf{x}$ and grows a spherical region until it encloses k training samples, and it labels the test point by a majority vote of these samples. In this $k = 5$ case, the test point $\mathbf{x}$ would be labeled the category of the black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

Example:

k = 3 (odd value) and $x = (0.10, 0.25)^t$

Prototypes	Labels
(0.15, 0.35)	w_1
(0.10, 0.28)	w_2
(0.09, 0.30)	w_5
(0.12, 0.20)	w_2

Closest vectors to x with their labels are:

$\{(0.10, 0.28, w_2); (0.12, 0.20, w_2); (0.15, 0.35, w_1)\}$

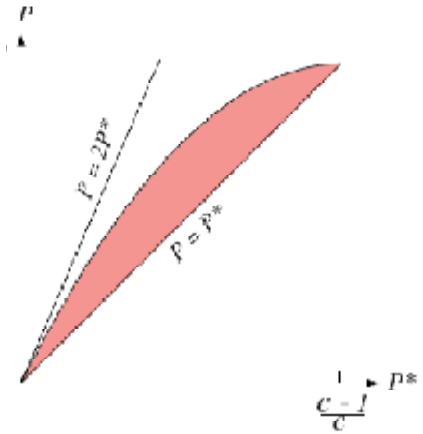
One voting scheme assigns the label w_2 to x since w_2 is the most frequently represented

Error Bounds

$$P(\omega_m \mid x) = \max_i P(\omega_i \mid x)$$

$$P(e \mid x) = 1 - P(\omega_m \mid x)$$

$$P^* = \int P(e \mid x) p(x) dx$$



$$P^* \le P \le P^* \left(2 - \frac{c}{c-1} P^* \right) \le 2P^*$$

FIGURE 4.14. Bounds on the nearest-neighbor error rate P in a c-category problem given infinite training data, where P^* is the Bayes error (Eq. 52). At low error rates, the nearest-neighbor error rate is bounded above by twice the Bayes rate. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

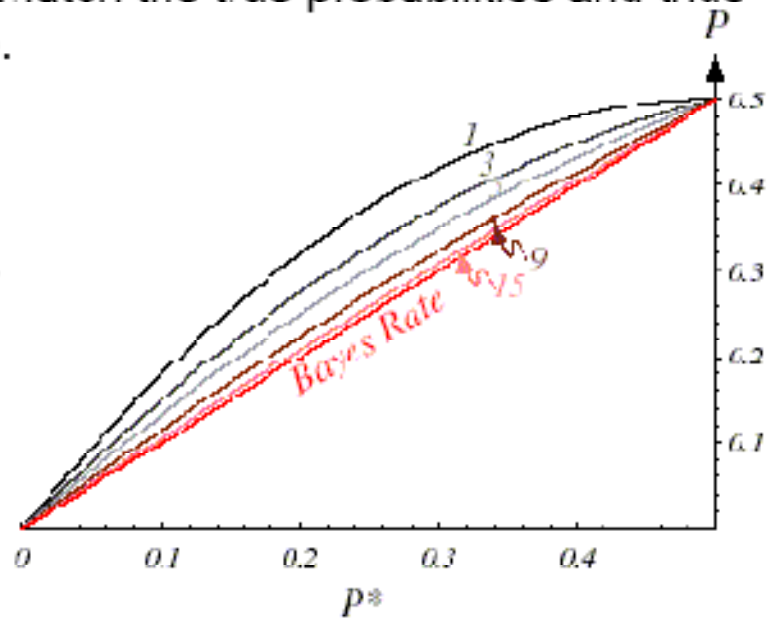
- P and P* have the values between 0 and 1-1/c (equal probability)
- If P* is small, so P will be small and close to P*
- If P* is large, so P will be close to P*

ANALYSIS OF kNN RULE

- **Error Rate:** Bayes decision rule classifies x into class C_p if, and only if $P(C_p|x) = max_j\ P(C_j|x)$.
  - It can be demonstrated that the kNN error rate is bounded above by twice the Bayes rate.
  - When $P(C_p|x)$ is closed to unity, the nearest neighbour selection is always matching the Bayes selection.
  - When $k \to \infty$ the estimated probabilities match the true probabilities and thus the error rate is equal to the Bayes rate.

- **Space Complexity:** high.

- **kNN Editing:** The high complexity of nearest-neighbour method can be reduced by removing those prototypes surrounded by prototypes of the same category.



Effect of Scaling

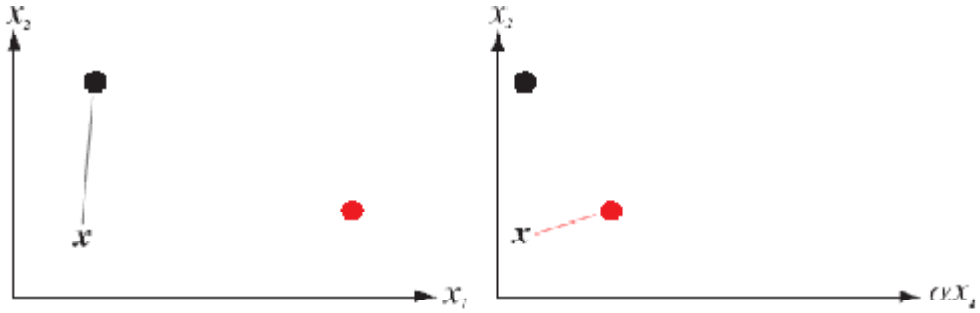


FIGURE 4.18. Scaling the coordinates of a feature space can change the distance relationships computed by the Euclidean metric. Here we see how such scaling can change the behavior of a nearest-neighbor classifer. Consider the test point x and its nearest neighbor. In the original space (left), the black prototype is closest. In the figure at the right, the x_1 axis has been rescaled by a factor 1/3; now the nearest prototype is the red one. If there is a large disparity in the ranges of the full data in each dimension, a common procedure is to rescale all the data to equalize such ranges, and this is equivalent to changing the metric in the original space. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example of Metrics

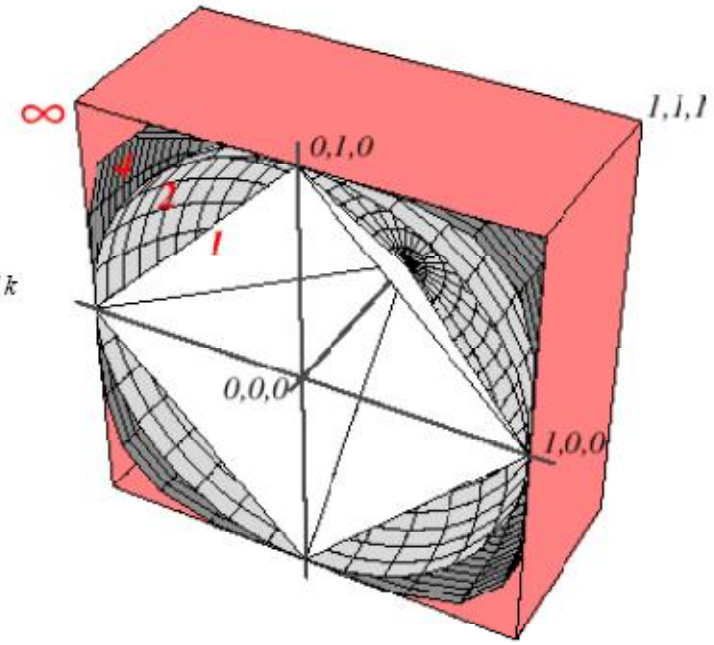
$$a = (a_1, \ldots, a_n)$$
$$b = (b_1, \ldots, b_n)$$

Euclidean $$d_e(a,b) = \sqrt{\sum_{i=1}^{n} (b_i - a_i)^2}$$

Minkowski $$L_k(a,b) = \left[\sum_{i=1}^{n} |b_i - a_i|^k \right]^{1/k}$$

Manhattan $$L_1(a,b) = \sum_{i=1}^{n} |b_i - a_i|$$

$$L_\infty(a,b) = \max_i |b_i - a_i|$$



Effect of Transformations

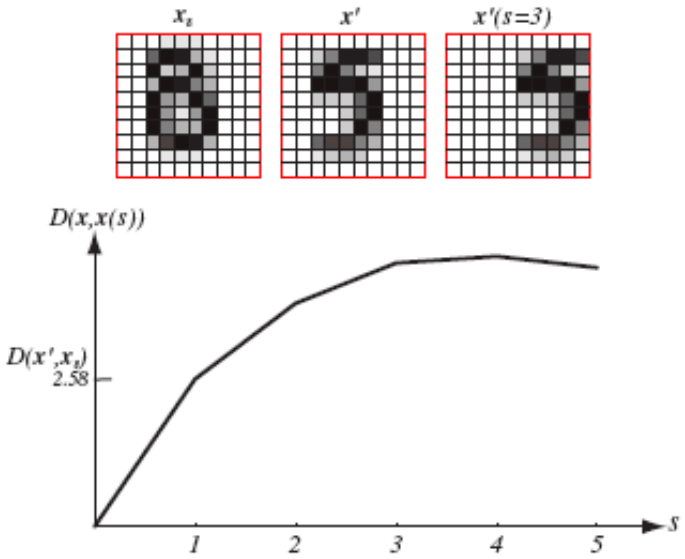


FIGURE 4.20. The uncritical use of Euclidean metric cannot address the problem of translation invariance. Pattern $\mathbf{x}'$ represents a handwritten 5, and $\mathbf{x}'(s = 3)$ represents the same shape but shifted three pixels to the right. The Euclidean distance $D(\mathbf{x}', \mathbf{x}'(s = 3))$ is much larger than $D(\mathbf{x}', \mathbf{x}_8)$, where $\mathbf{x}_8$ represents the handwritten 8. Nearest-neighbor classification based on the Euclidean distance in this way leads to very large errors. Instead, we seek a distance measure that would be insensitive to such translations, or indeed other known invariances, such as scale or rotation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

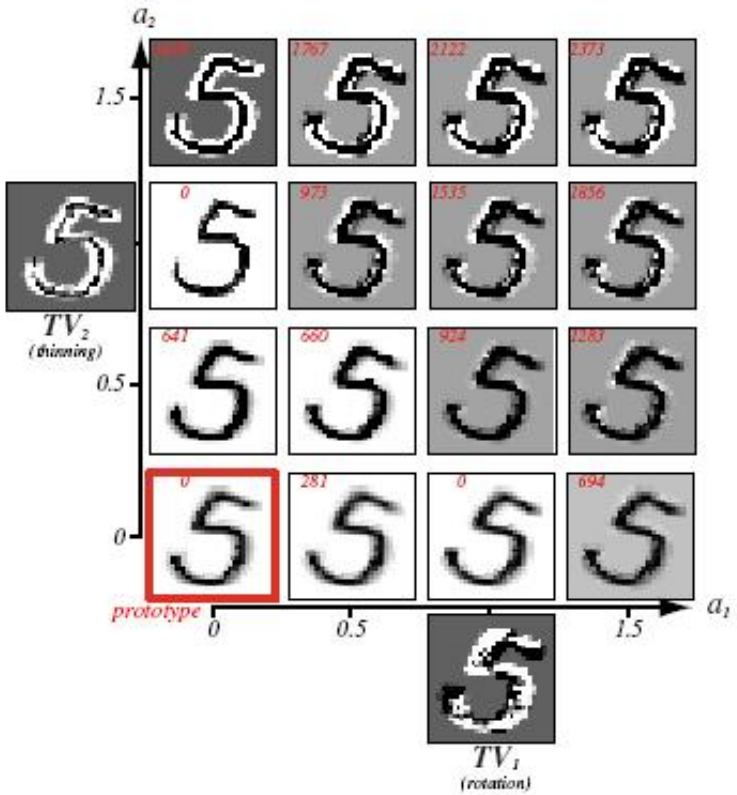


FIGURE 4.21. The pixel image of the handwritten 5 prototype at the lower left was subjected to two transformations, rotation, and line thinning, to obtain the tangent vectors $\mathbf{TV}_1$ and $\mathbf{TV}_2$; images corresponding to these tangent vectors are shown outside the axes. Each of the 16 images within the axes represents the prototype plus linear combination of the two tangent vectors with coefficients a_1 and a_2. The small red number in each image is the Euclidean distance between the tangent approximation and the image generated by the unapproximated transformations. Of course, this Euclidean distance is 0 for the prototype and for the cases $a_1 = 1, a_2 = 0$ and $a_1 = 0, a_2 = 1$. (The patterns generated with $a_1 + a_2 > 1$ have a gray background because of automatic grayscale conversion of images with negative pixel values.) From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.