Nonparametric Classifiers
KNN

Chapter 4
Inel 5046

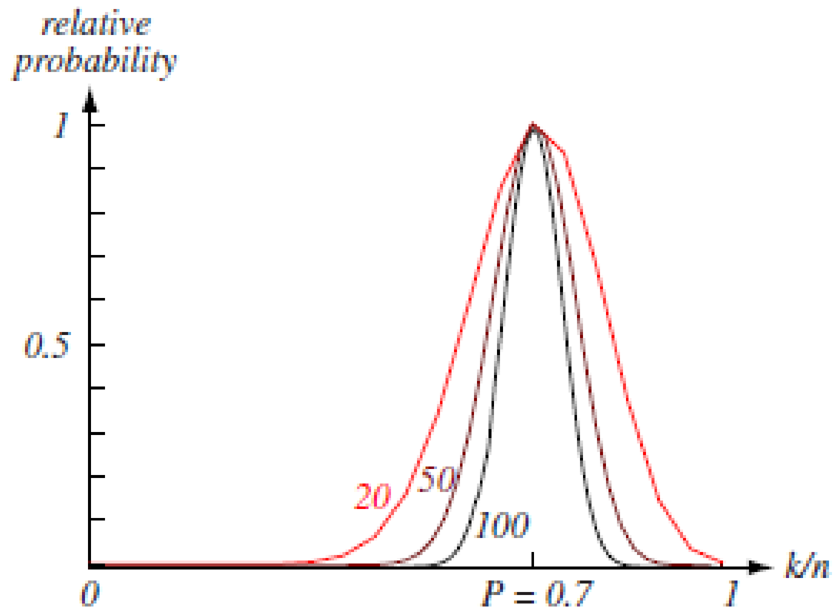


FIGURE 4.1. The relative probability an estimate given by Eq. 4 will yield a particular value for the probability density, here where the true probability was chosen to be 0.7. Each curve is labeled by the total number of patterns n sampled, and is scaled to give the same maximum (at the true probability).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

Kernel Density Estimate

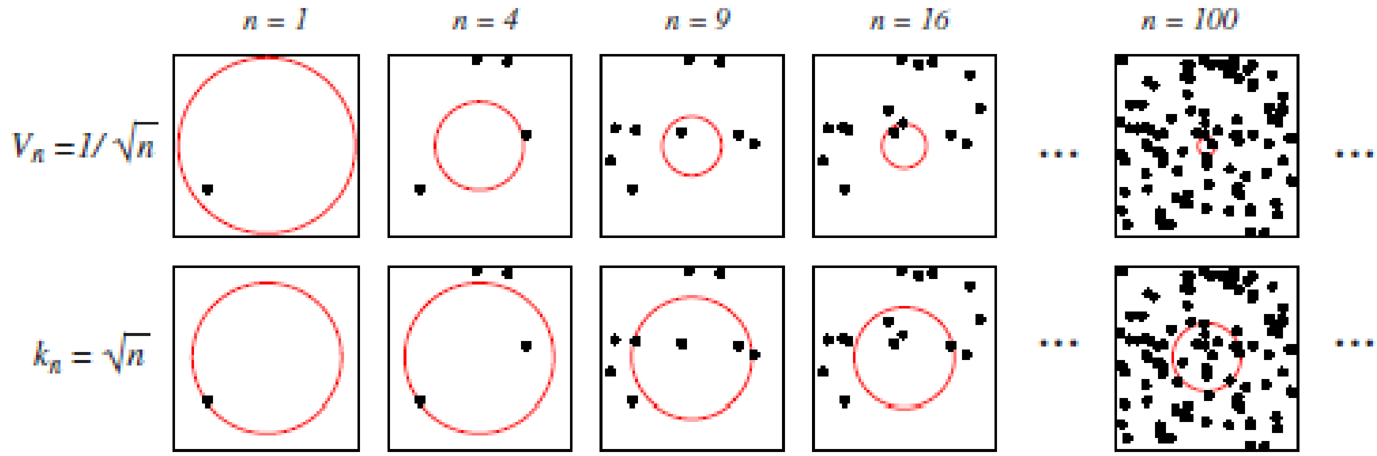


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the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

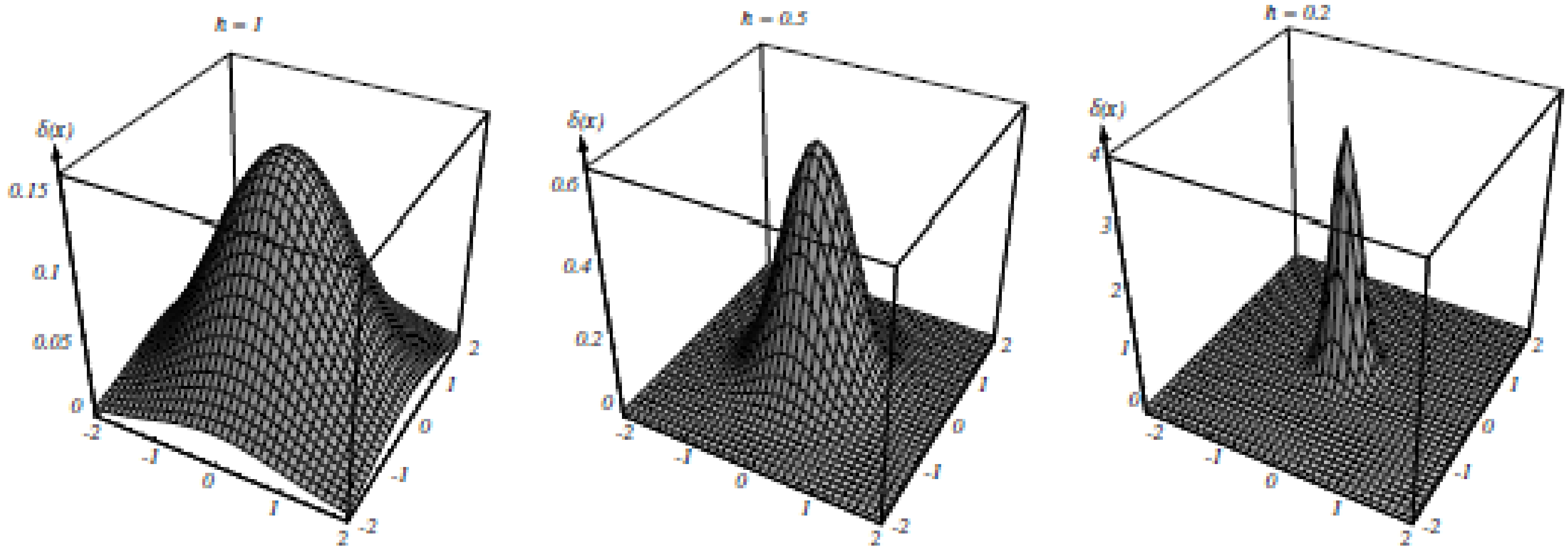


FIGURE 4.3. Examples of two-dimensional circularly symmetric normal Parzen windows for three different values of h. Note that because the $\delta(\mathbf{x})$ are normalized, different vertical scales must be used to show their structure. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern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

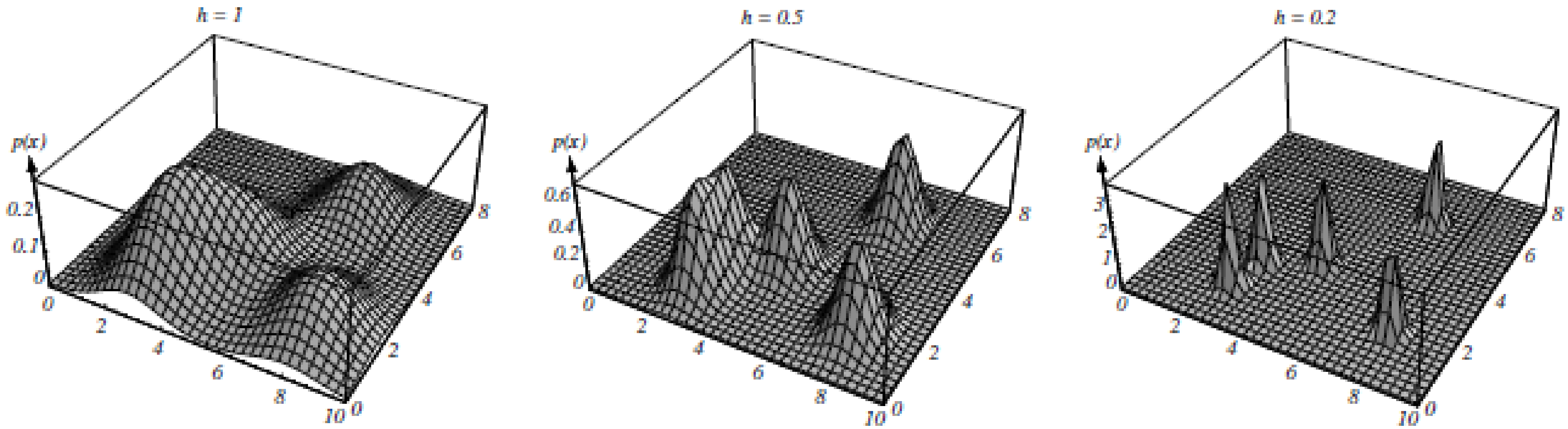


FIGURE 4.4. Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.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

Illustration

- P(x) is a 0 mean, unit-variance, univariate normal density
- For n=10 and h1=0.1 the contributions of the individual samples are clearly discernible,
- As n gets larger, the ability of pn(x) to resolve variations in p(x) increases
- Many samples are required to get an accurate estimate
- Similar results in 2 dimensions

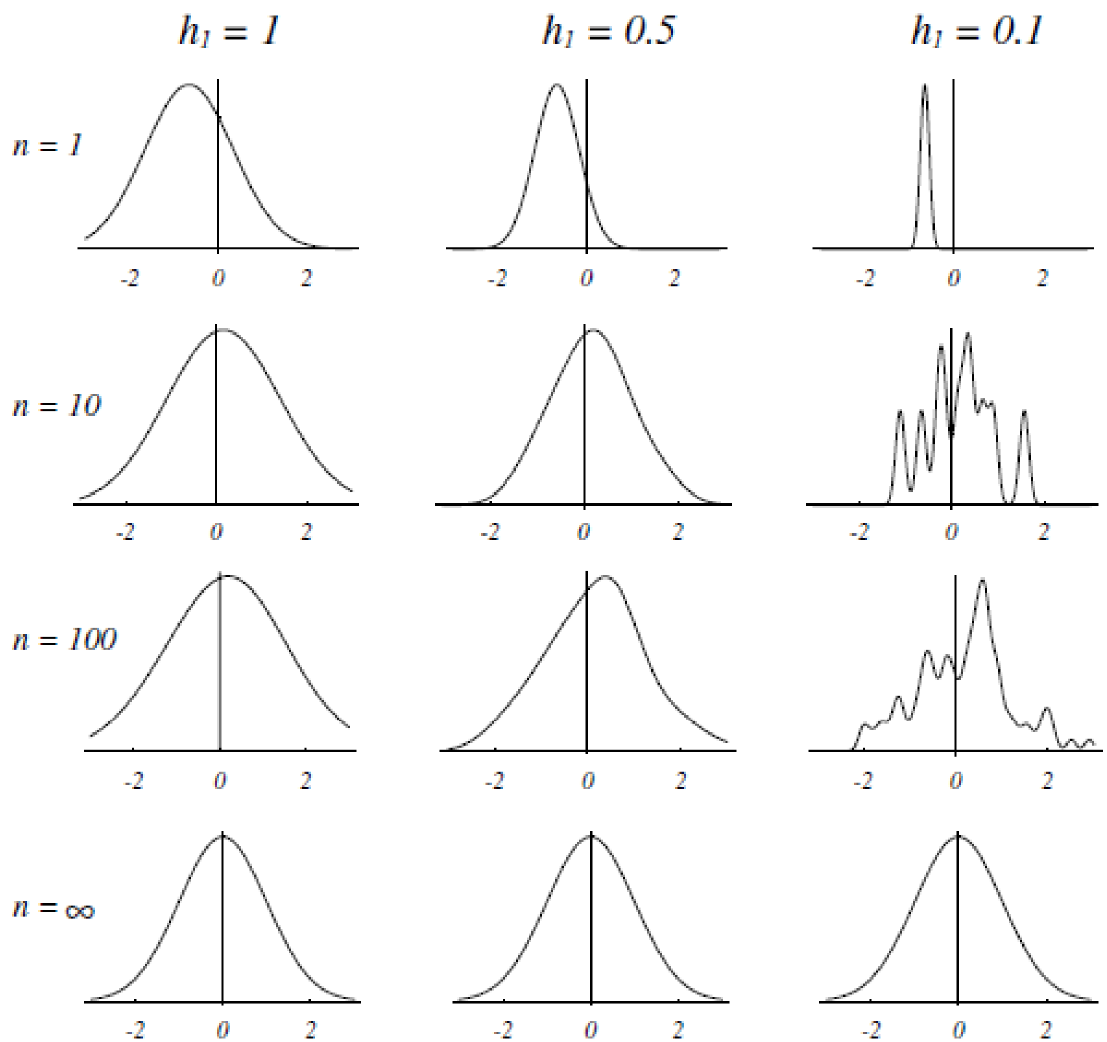
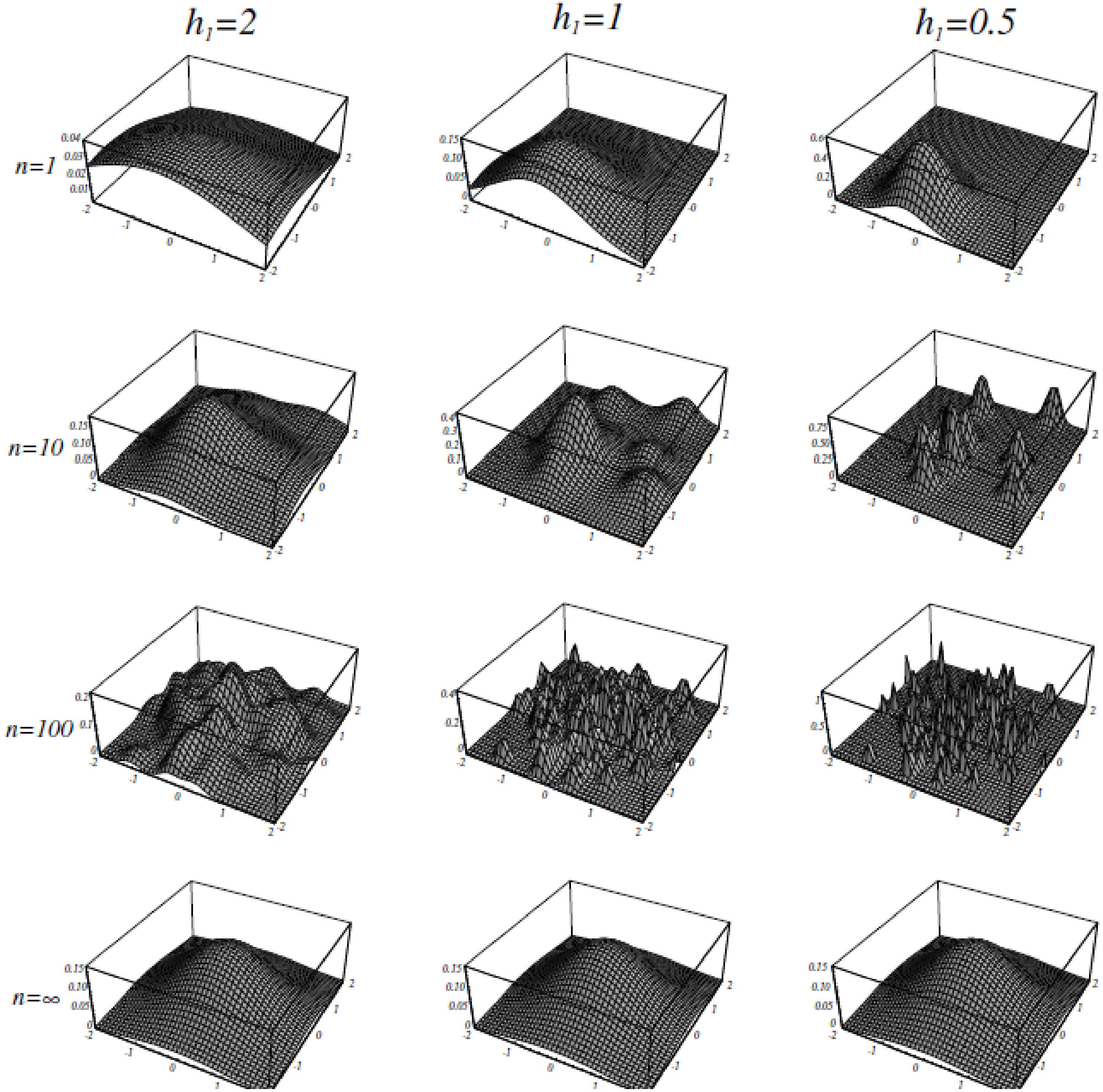


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless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


h_1=2
h_1=1
h_1=0.5
n=1
n=10
n=100
n=∞

- Unknown density is a mixture of uniform and triangle density
- n=1 tells more about window function than about unknown density
- For n=16, no estimate is good
- For n=256 and h1 =1 results are beginning to be acceptable

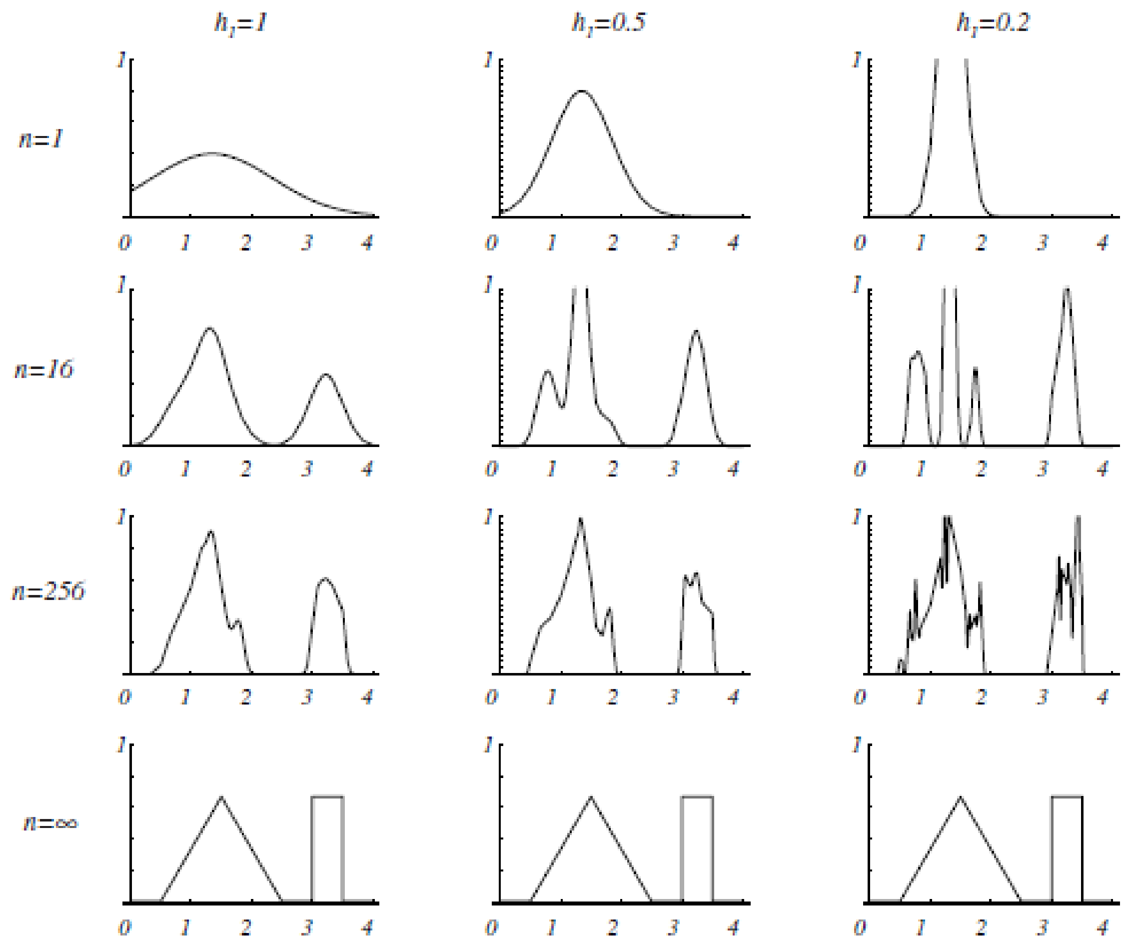


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern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

Classification Example

- The densities for each category is estimated; classify a test point by the label corresponding to the maximum posterior
- Adv of density estimation: their generality
- With enough samples, method converges to arbitrarily complex target density
- Disadv-requirements for computation time and storage; data sample requirement is large; Huges phenomenon-more dimension requires more training samples
- To overcome this incorporate correct knowledge about data

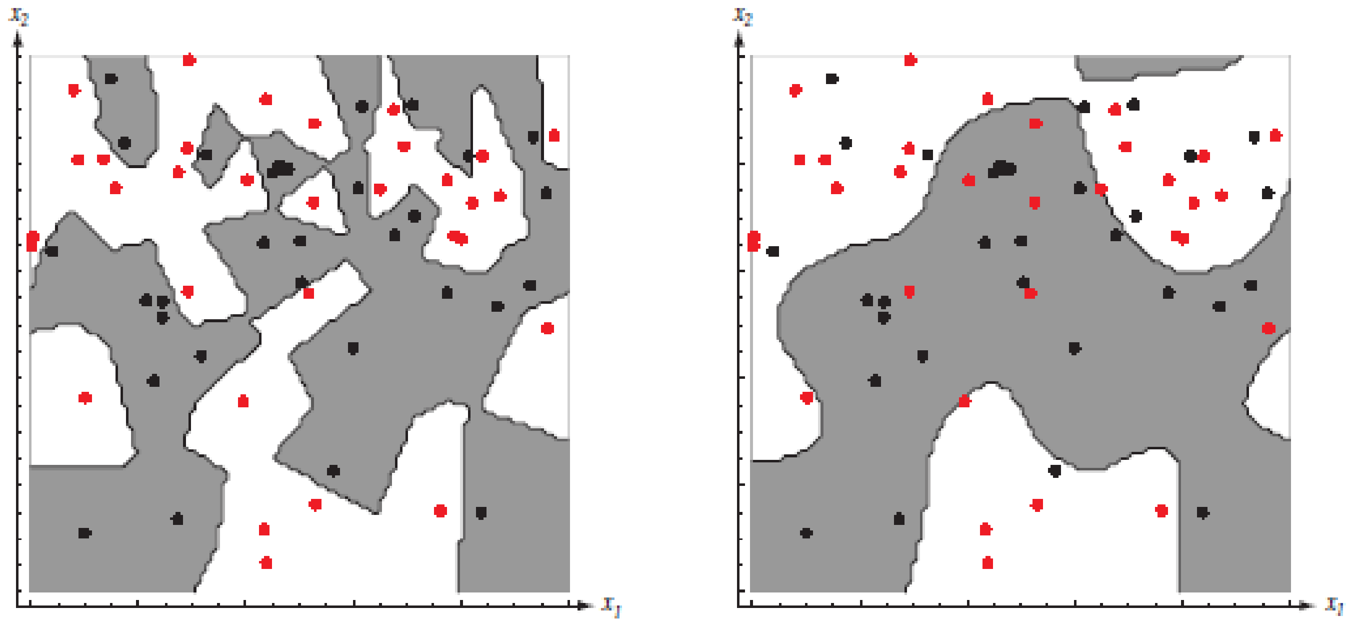
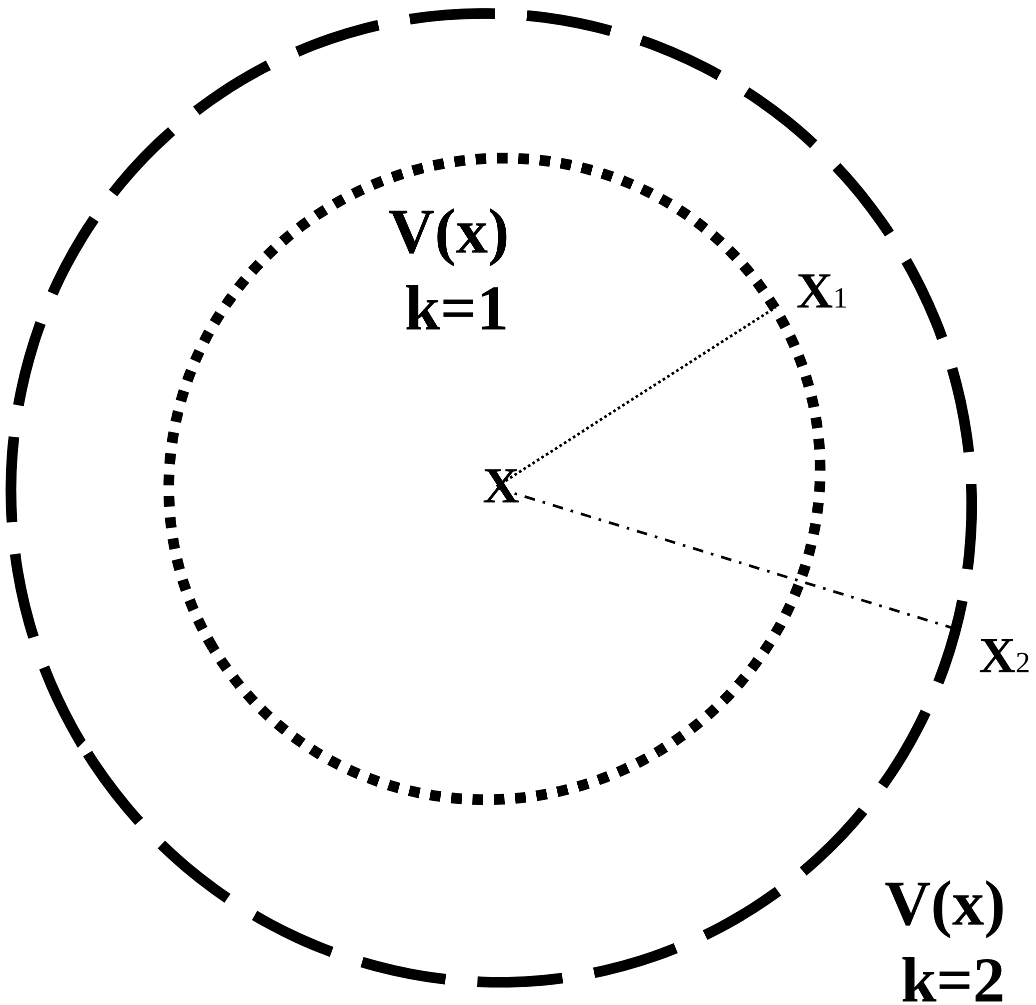


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the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

K-Nearest Neighbor
Density Estimation



- Remedy to ‘best’ window function is to let the cell volume be a function of training data
- If the density is high near x, the cell will be relatively small, which leads to good resolution

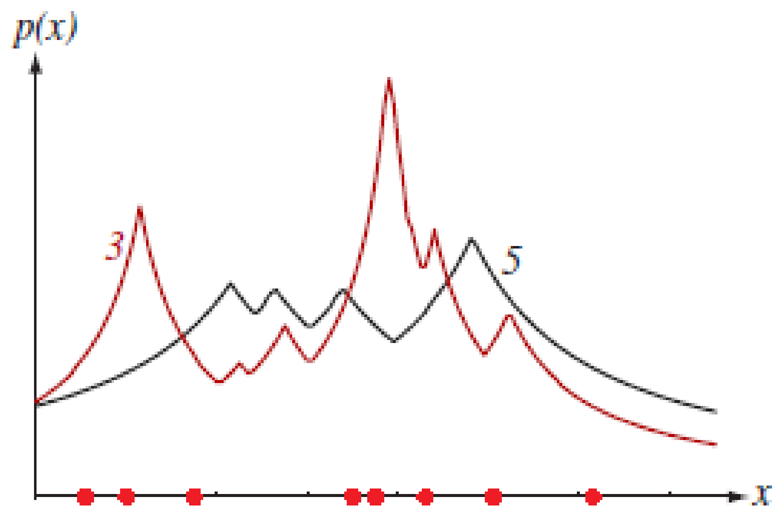


FIGURE 4.10. Eight points in one dimension and the k-nearest-neighbor density estimates, for $k = 3$ and 5. Note especially that the discontinuities in the slopes in the estimates generally lie *away* from the positions of the prototype points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

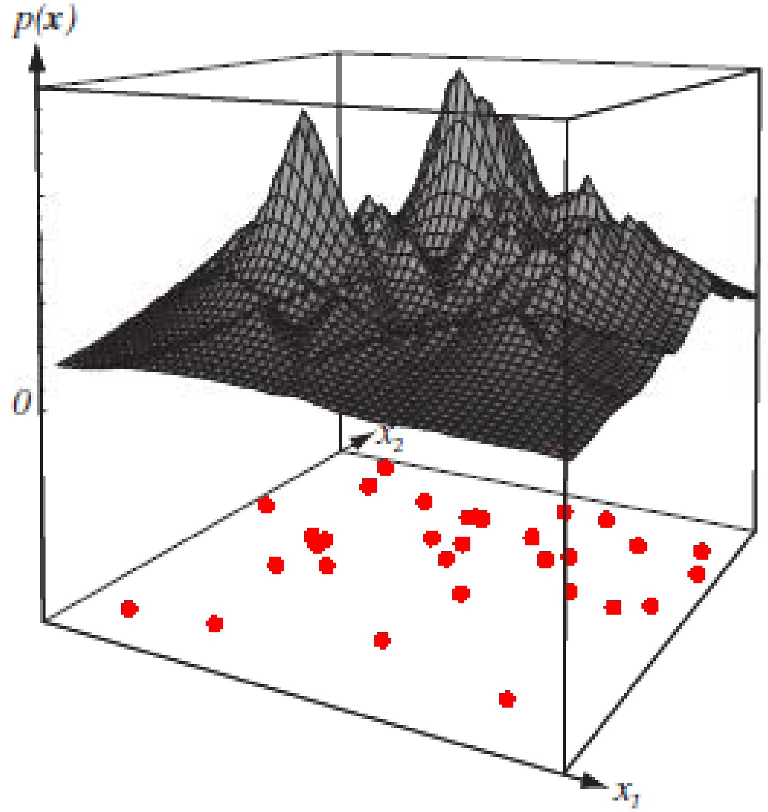


FIGURE 4.11. The k-nearest-neighbor estimate of a two-dimensional density for $k = 5$. Notice how such a finite n estimate can be quite "jagged," and notice that discontinuities in the slopes generally occur along lines away from the positions of the points themselves. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

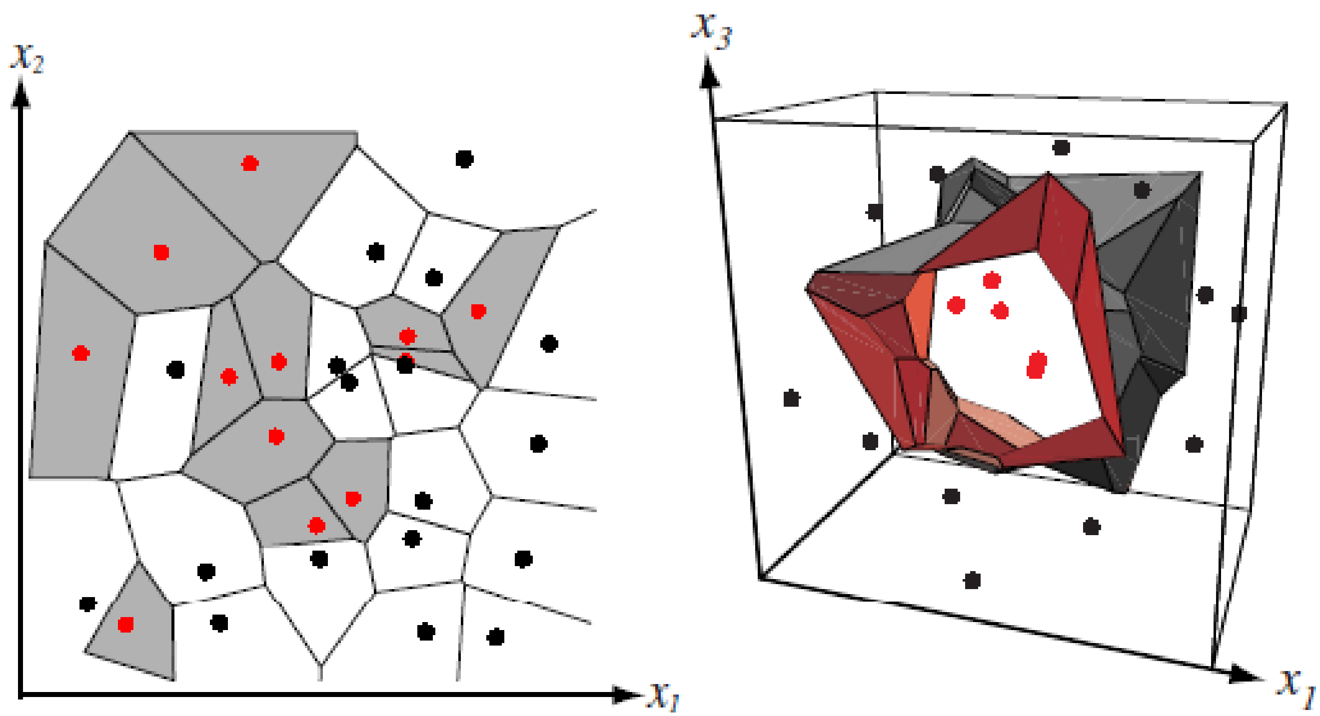


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal.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

Computational complexity of k-NN

- We have n labeled training samples, in d dimensions
- Each distance calculated is O(d), this search for nearest neighbor is $O(dn^2)$
- A parallel implementation is O(1) in time and O(n) in space
- Three methods to reduce computational burden- partial distance, prestructuring (create search tree) and editing
- Editing-eliminate useless prototypes by editing, pruning or condensing-eliminate prototypes surrounded by training points of the same category label

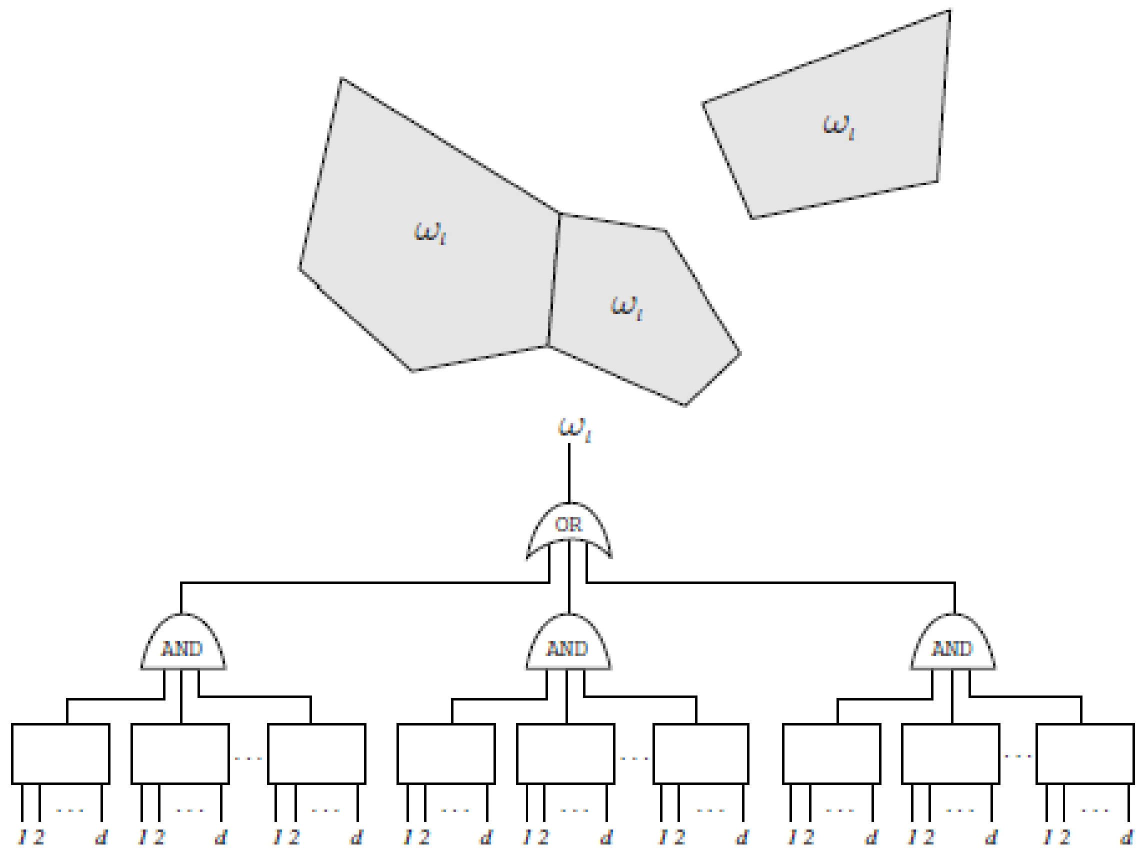


FIGURE 4.17. A parallel nearest-neighbor circuit can perform search in constant—that is, $O(1)$—time. The d-dimensional test pattern $\mathbf{x}$ is presented to each box, which calculates which side of a cell's face $\mathbf{x}$ lies on. If it is on the "close" side of every face of a cell, it lies in the Voronoi cell of the stored pattern, and receives its label. In the case shown, each of the three AND gates corresponds to a single Voronoi cell. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Nearest Neighbor Editing

- **Begin** **initialize** j←0, D←data set, n← # prototypes
  - Construct the full Voronoi diagram of D
  - **Do** j←j+1; for each prottype x’j
    - Find the Voronoi neighbors of x’j
    - **If** any neighbor is not from the same class as x’j, then mark x’j
  - **Until** j=n
- Discard all points that a`re not marked
- Construct the Voronoi diagram of the remaining (marked) prototypes
- **end**

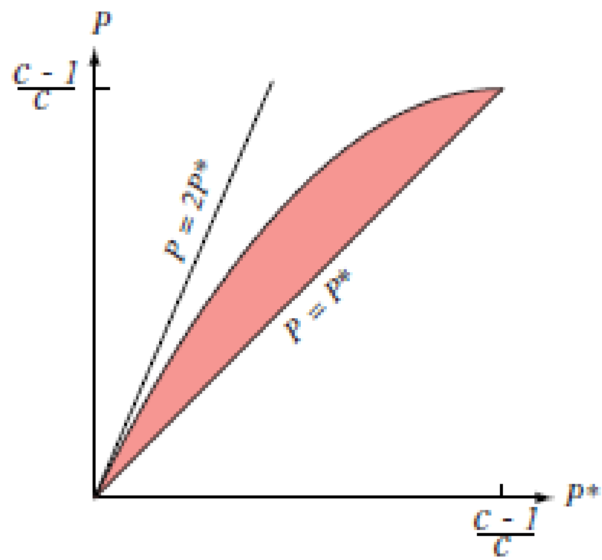


FIGURE 4.14. Bounds on the nearest-neighbor error rate P in a c-category problem given infinite training data, where P^* is the Bayes error (Eq. 52). At low error rates, the nearest-neighbor error rate is bounded above by twice the Bayes rate. From: Richard O. Duda, Peter E. Hart, and David G. Stork,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

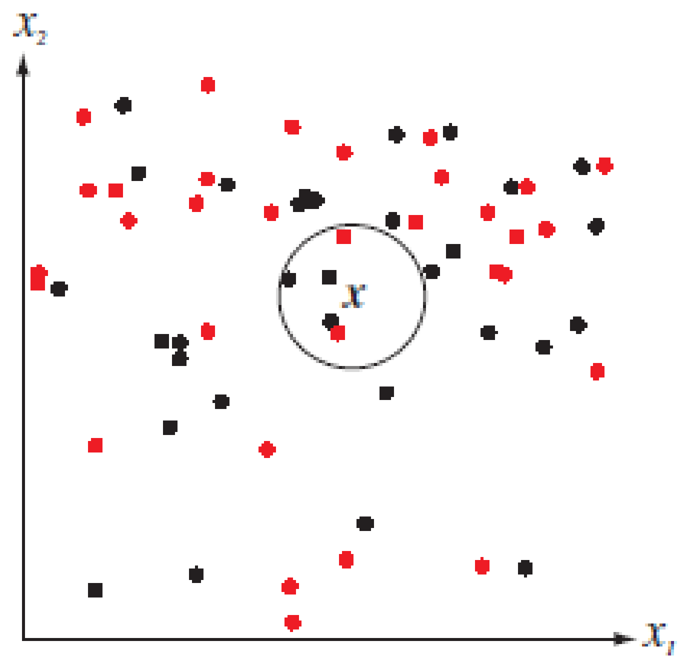


FIGURE 4.15. The k-nearest-neighbor query starts at the test point **x** and grows a spherical region until it encloses k training samples, and it labels the test point by a majority vote of these samples. In this $k = 5$ case, the test point **x** would be labeled the category of the black points. From: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

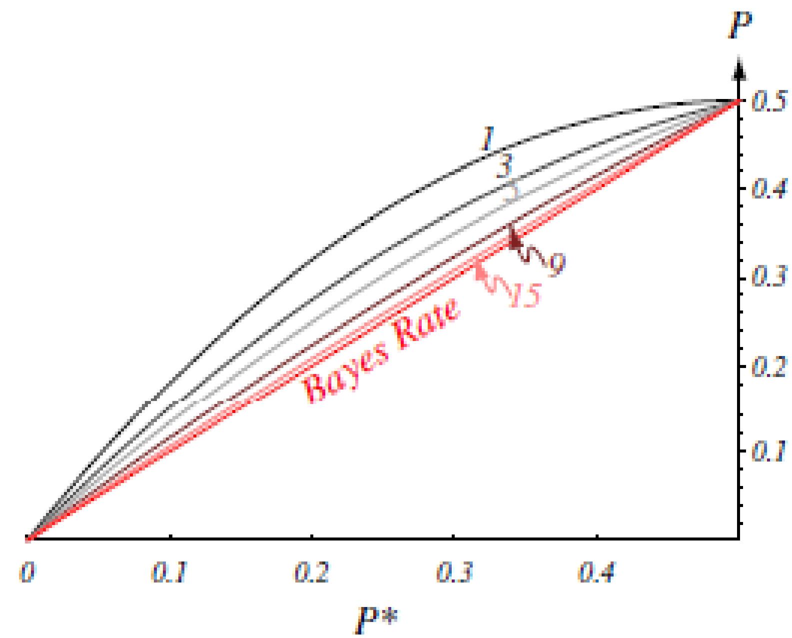


FIGURE 4.16. The error rate for the k-nearest-neighbor rule for a two-category problem is bounded by $C_k(P^*)$ in Eq. 54. Each curve is labeled by k; when $k = \infty$, the estimated probabilities match the true probabilities and thus the error rate is equal to the Bayes rate, that is, $P = P^*$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

kNN Approach Classifier

$$g_i(x) = \hat{f}\left(x / w_i\right)$$

$$g_i(x) = f(x) = \frac{k_i / n_i}{V_i(x)}$$

$$g_i(\mathbf{X}) \geq g_j(\mathbf{X}) \qquad \forall j = 1, 2, \ldots, m$$

$$V_i(x) = K \cdot dist_i^d(x)$$

d = dimensionality

$$g_i(\mathbf{X}) \geq g_j(\mathbf{X}) \qquad \forall j = 1, 2, \ldots, m$$

$$V_i(\mathbf{X}) \leq V_j(\mathbf{X}) \qquad \forall j = 1, 2, \ldots, m$$

$$dist_i^d(\mathbf{X}) \leq dist_j^d(\mathbf{X}) \qquad \forall j = 1, 2, \ldots, m$$

$$dist_i(\mathbf{X}) \leq dist_j(\mathbf{X}) \qquad \forall j = 1, 2, \ldots, m$$

kNN Approach Classifier

- Nearest neighboor-assign the same label as that of the nearest training pixel.

- k nearest neighboor-assign label according to the majority label of k nearest-neighboor trainingh pixels.

What kind of distance?

- Euclidean
- Malahanobis

Five Good Distances

Name	Formula
Euclidean	$\|\mathbf{x}-\mathbf{v}\|_2 = \sqrt{(\mathbf{x}-\mathbf{v})^{\mathbf{T}}(\mathbf{x}-\mathbf{v})}$
City Block	$\|\mathbf{x}-\mathbf{v}\|_1 = \sum_{\mathbf{j}=1}^{\mathbf{d}} \left\| \mathrm{x}_{\mathbf{j}} - \mathrm{v}_{\mathbf{j}} \right\|$
Mahalanobis	$\|\mathbf{x}-\mathbf{v}\|_{\Sigma^{-1}} = \sqrt{(\mathbf{x}-\mathbf{v})^{\mathbf{T}}\Sigma^{-1}(\mathbf{x}-\mathbf{v})}$
Diagonal	$\|\mathbf{x}-\mathbf{v}\|_{\mathbf{D}^{-1}} = \sqrt{(\mathbf{x}-\mathbf{v})^{\mathbf{T}}\mathbf{D}^{-1}(\mathbf{x}-\mathbf{v})}$
Sup or Max	$\|\mathbf{x}-\mathbf{v}\|_\infty = \max_{\mathrm{j}}\left\{\left\|\mathbf{x}_{\mathbf{j}} - \mathbf{v}_{\mathbf{j}}\right\|\right\}$

Graphic Example