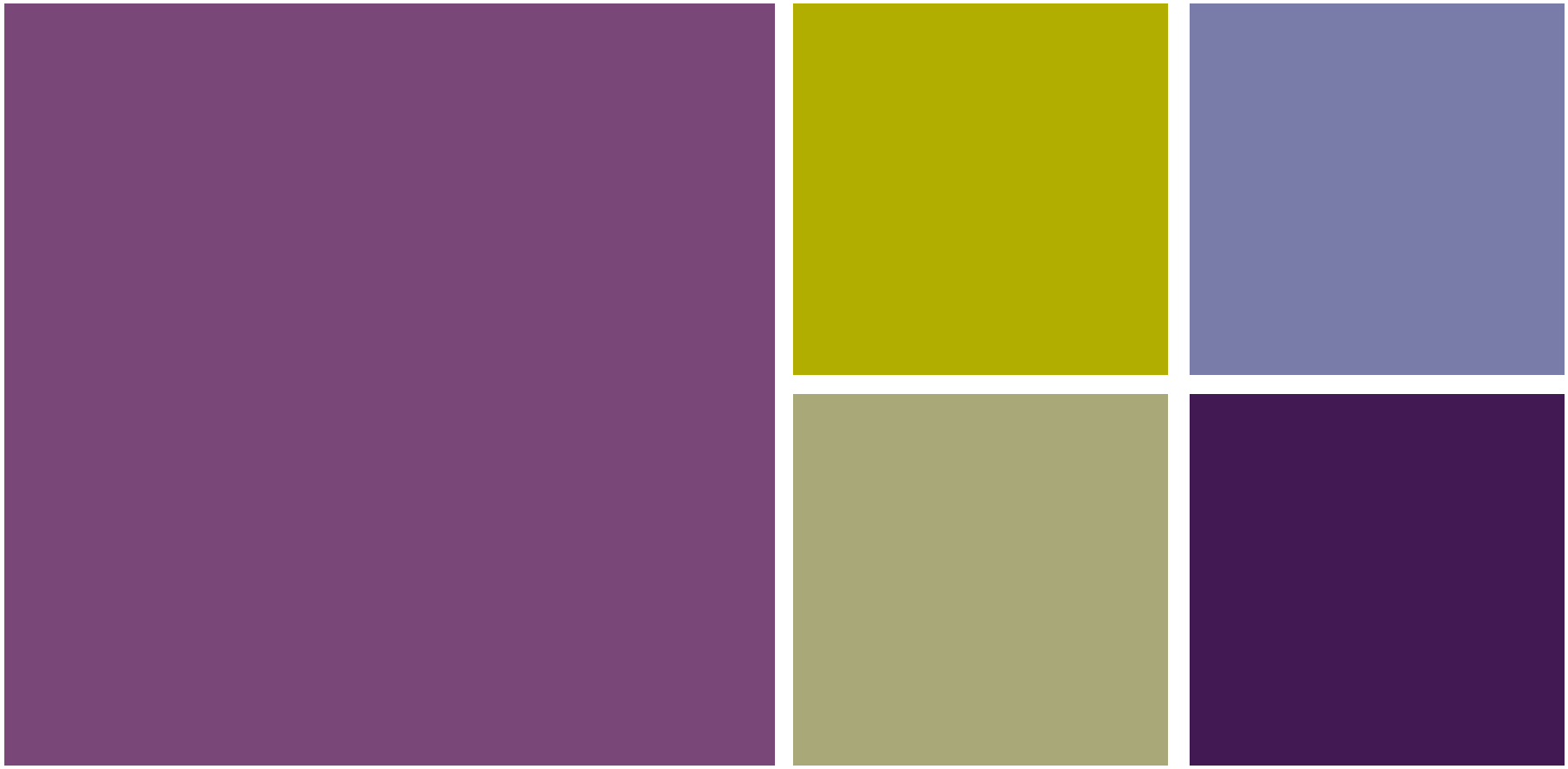


Non-Parameteric Classifiers

Pattern Recognition and Image Analysis

Dr. Manal Helal – Fall 2014

Lecture 7

Overview

- Density Estimation
- Parzen Windows
- K_n Nearest Neighbours

Non-Parametric Probability Distributions

- The common parametric forms rarely fit the densities actually encountered in practice, because they are unimodal (having a single local maximum), while practical problems are multi-modal densities.
- Furthermore, representing a high dimensional density as the product of one-dimensional densities is not accurate for practical problems.
- Nonparametric procedures assume arbitrary empirical distributions with unknown densities to provide estimates of probability density functions (pdf) or cumulative distribution functions (cdf) based on sample data.
- You can use these approaches when the data cannot be described accurately by the supported parametric distributions.

Non-Parametric Methods Types

1. To estimate the density functions $p(\mathbf{x}|\omega_j)$
   - If these estimates are satisfactory, they can be substituted for the true densities when designing the classifier (generative model).
2. To estimate the a posteriori probabilities $P(\omega_j|\mathbf{x})$.
   - This is closely related to nonparametric design procedures such as the nearest-neighbor rule, which bypass probability estimation and go directly to decision functions (discriminative model). Probabilistic neural networks will be covered later on.
3. To transform the feature space in the hope that it may be possible to employ parametric methods in the transformed space.
   - For Example the Fisher linear discriminant, which provides an important link between the parametric techniques and the adaptive non-parametric techniques.

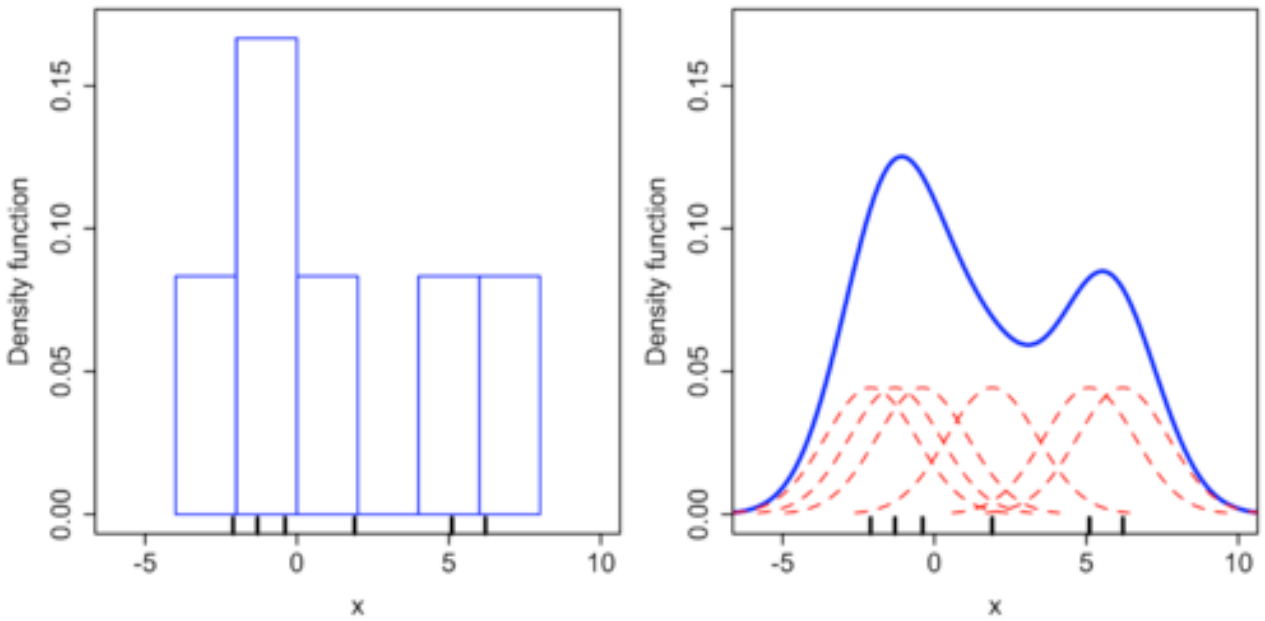
Histograms vs. Density Estimation

- Histograms:
  - To draw a histogram of a given 6 data points: $x_1 = -2.1, x_2 = -1.3, x_3 = -0.4, x_4 = 1.9, x_5 = 5.1, x_6 = 6.2$. The x-axis is divided in bins of size 2, the y axis is incremented by 1/12 each time an x value falls in the x-axis interval.

$$\hat{p}(\mathbf{x}) = \frac{m}{Vn},$$

  - The histogram estimate is not a very good way to estimate densities, especially so when there are many features. Particularly, it leads to discontinuous density estimates

- For the kernel density estimate, we place a normal kernel with variance 2.25 (indicated by the red dashed lines) on each of the data points x_i. The kernels are summed to make the kernel density estimate (solid blue curve).



Background - combinations

- Consider the problem of selecting k labeled balls without replacement from an urn containing n balls.
- In how many different ways may we select those k balls? i.e. combinations (subsets) containing k elements.
- The answer is $$C(n,k) = \left(\begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$$
- n! = n · (n − 1) · · · 2 · 1, (convention 0! = 1)

Density Estimation

- The probability P that a vector x will fall in a region R is given by:

$$P = \int_{\mathcal{R}} p(\mathbf{x}')\ d\mathbf{x}'.$$

- For n i.i.d samples of x, the probability that k of these n fall in R is given by the binomial law:

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

- The expected value for k is: $E[k] = \sum_{k=0}^{n} k P_k = nP$.

- ML estimation of P = θ $\underset{\theta}{Max}(P_k \mid \theta)$ 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.

- If we now assume that p(x) is continuous and that the region R is so small that p does not vary appreciably within it, we can write

$$\int_{\mathcal{R}} p(\mathbf{x}')\ d\mathbf{x}' \simeq p(\mathbf{x})V,$$

where x is a point within R and V is the volume enclosed by R.

- Combining the previous equations, we can estimate p(X) as:

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

- If we fix V and increase the samples, the ratio k/n will converge (in probability) as desired, but we have only obtained an estimate of the space-averaged value of p(x),

$$\hat{p}(\mathbf{x}) = \frac{\hat{k}}{nV} = \frac{\int_R p(\mathbf{x})d\mathbf{x}}{\int_R d\mathbf{x}}. \qquad \frac{P}{V} = \frac{\int_{\mathcal{R}} p(\mathbf{x}')\ d\mathbf{x}'}{\int_{\mathcal{R}} d\mathbf{x}'}.$$

- As V approaches 0, we are obtaining p(x) not an estimate of it. So, fixing n, and let V approaches zero, will create regions with no samples ($p(x) \simeq 0$) is useless, and regions with one or more samples coincide at x, the estimate diverges to infinity, which is useless as well.

- Practically, we have to accept a certain amount of variance in the ratio k/n and a certain amount of averaging of the density p(x).

- To estimate the density at x for unlimited samples, we form a sequence of regions $R_1, R_2, \ldots$, containing x — the first region to be used with one sample, the second with two, and so on. Let V_n be the volume of R_n, k_n be the number of samples falling in R_n, and $p_n(x)$ be the n^{th} estimate for p(x):

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}.$$

- Three conditions for $\lim_{n\to\infty} p_n(x)$ is to converge to $p(x)$

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

assures us that the space averaged P/V will converge to p(x), provided that the regions shrink uniformly and that p(·) is continuous at x.

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

only makes sense if p(x) ≠ 0, assures us that the frequency ratio will converge (in probability) to the probability P .

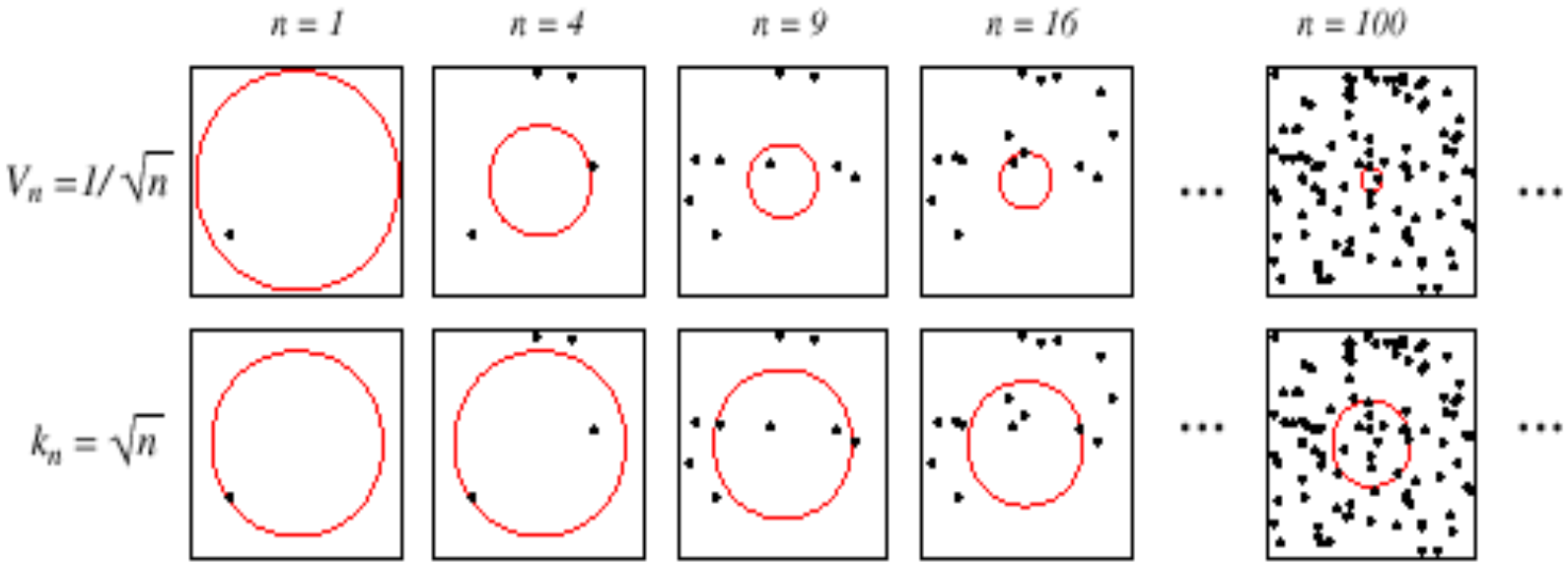
$$\lim_{n\to\infty} k_n/n = 0.$$

necessary if $p_n(x)$ is to converge at all. Although a huge number of samples will eventually fall within the small region R_n, they will form a negligibly small fraction of the total number of samples.

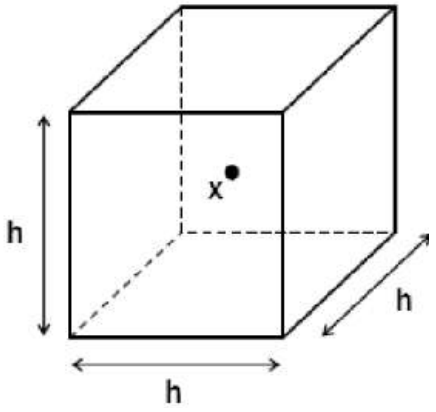
How to Obtain the sequence of Regions?

- Parzen Windows:
  - Shrink an initial region by specifying the volume V_n as some function of n, such as $V_n = 1/\sqrt{n}$.
  - Check if the random variables k_n and k_n/n behaves properly, such that $p_n(x)$ converges to $p(x)$.

- k_n-Nearest- Neighbor Estimation
  - 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



Parzen Windows



- Also called kernel density estimation (KDE) and Parzen–Rosenblatt window.
- Assume that the region R_n is a d-dimensional hypercube.
- If h_n is the length of the side of the hypercube, its volume is given by $V_n = h_n^d$.
- We can obtain an analytic expression for kn - the number of samples falling into the hypercube - by defining the following **window function**:

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

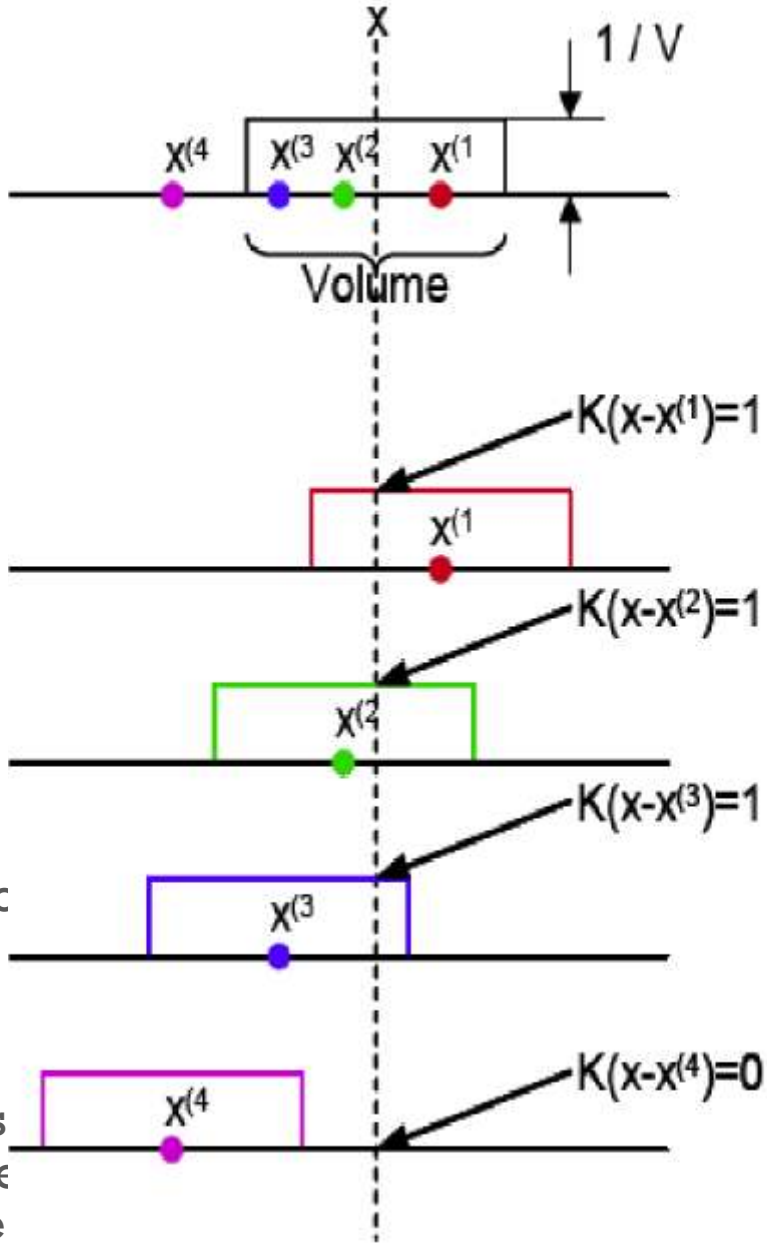
  - That is ϕ has the value one inside and the value zeros outside the unit hypercube centered at origin. $\varphi((x-x_i)/h_n)$ is equal to unity if x_i falls within the hypercube of volume V_n centered at x and equal to zero otherwise. It is the kernel function k(.) that is typically unimodal
  - h > 0 is a smoothing parameter called the bandwidth (window or kernel width).

- The number of samples in this hypercube is:

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

By substituting k_n, we obtain the following Parzen-window density estimate:

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

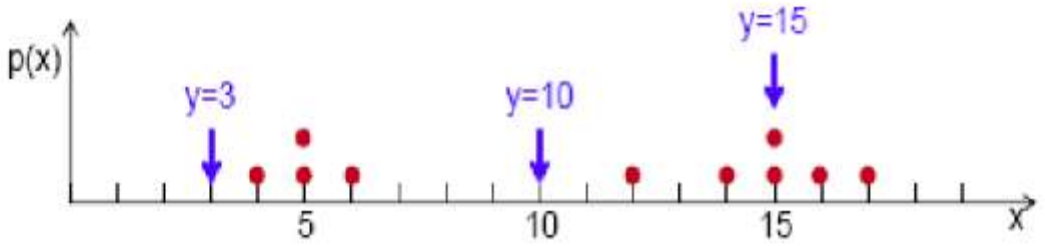


$P_n(x)$ estimates $p(x)$ as an average of functions of x and the samples (x_i) $(i = 1, \ldots, n)$. These functions φ can be general!

The estimate $p_n(x)$ is an average of (window) functions. Usually the window function has its maximum at the origin and its values become smaller when we move further away from the origin. Then each training sample is contributing to the estimate in accordance with its distance from x.

Numeric exercise

- **Given the dataset below, use Parzen windows to estimate the density p(x) at y=3,10,15. Use a bandwidth of h=4**
  - $X = \{x^{(1)}, x^{(2)}, \ldots x^{(N)}\} = \{4, 5, 5, 6, 12, 14, 15, 15, 16, 17\}$
- **Solution**
  - Let's first draw the dataset to get an idea of what numerical results we should expect



$$K(\mathbf{u}) = \begin{cases} 1 & |u_j| < 1/2 \\ 0 & \text{otherwise} \end{cases}$$

  - Let's now estimate p(y=3):

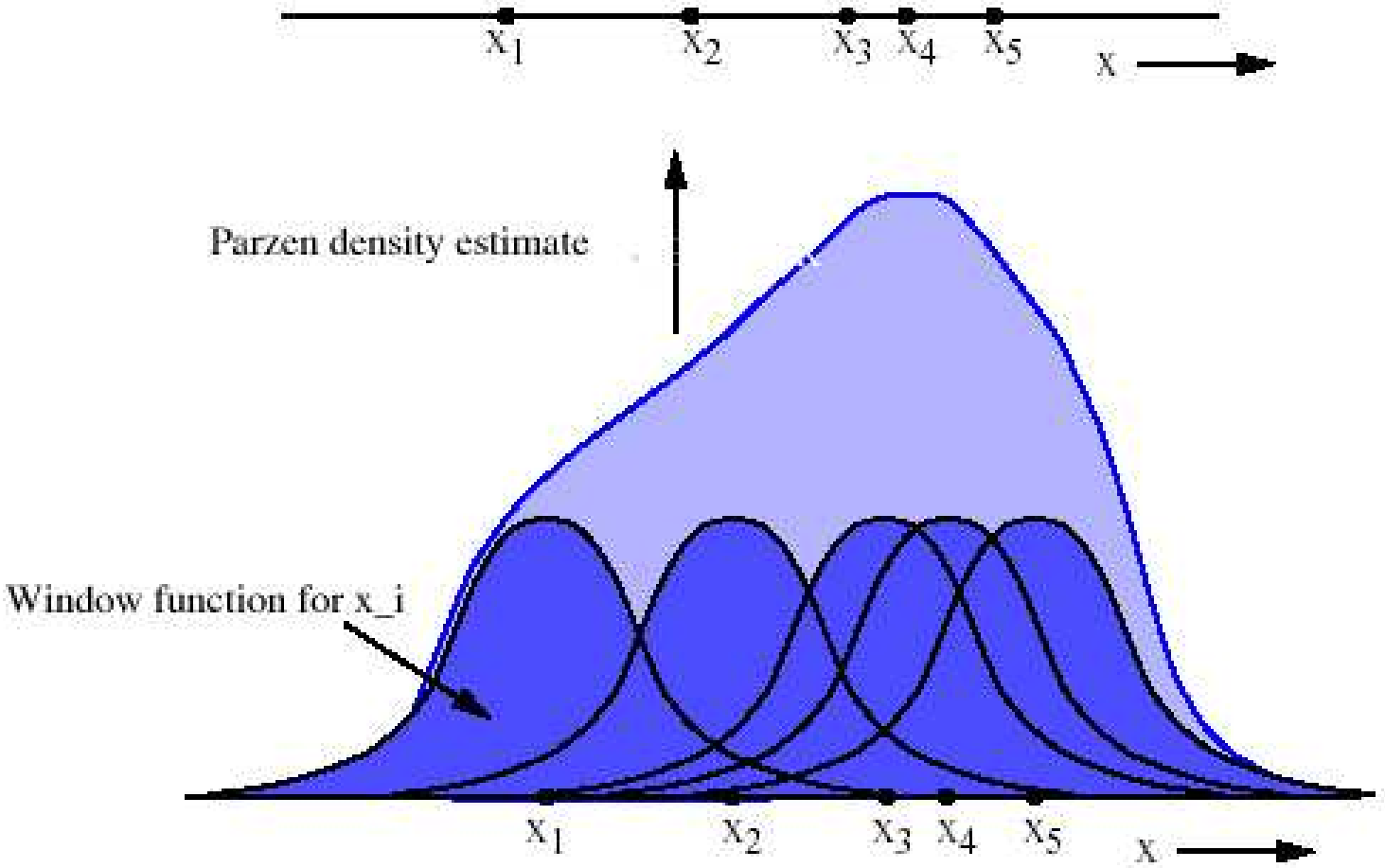
$$p_{KDE}(y=3) = \frac{1}{Nh^D}\sum_{n=1}^{N} K\left(\frac{y-x^{(n)}}{h}\right) = \frac{1}{10\times 4^1}\left[K\underbrace{\left(\frac{3-4}{4}\right)}_{-1/4} + K\underbrace{\left(\frac{3-5}{4}\right)}_{-1/2} + K\underbrace{\left(\frac{3-5}{4}\right)}_{-1/2} + K\underbrace{\left(\frac{3-6}{4}\right)}_{-1} + \ldots + K\underbrace{\left(\frac{3-17}{4}\right)}_{-13/4}\right] =$$

$$= \frac{1}{10\times 4^1}[1+0+0+0+0+0+0+0+0+0] = \frac{1}{10\times 4} = 0.025$$

  - Similarly

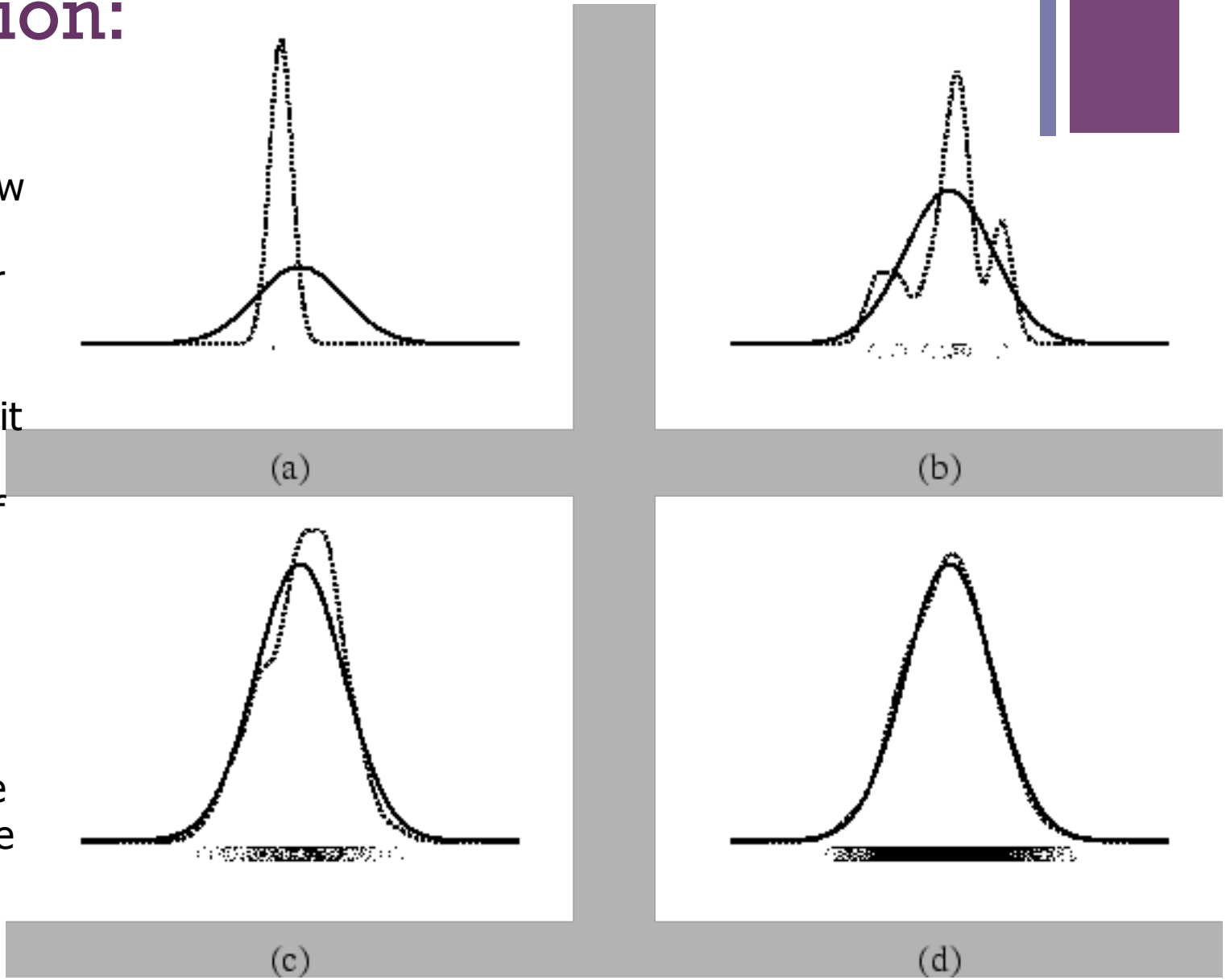
$$p_{KDE}(y=10) = \frac{1}{10\times 4^1}[0+0+0+0+0+0+0+0+0+0] = \frac{0}{10\times 4} = 0$$

$$p_{KDE}(y=15) = \frac{1}{10\times 4^1}[0+0+0+0+0+1+1+1+1+0] = \frac{4}{10\times 4} = 0.1$$


x1
x2
x3
x4
x5
x
Parzen density estimate
Window function for x_i
x1
x2
x3
x4
x5
x

Gaussian Kernel Parzen Window Estimation:

The Parzen-window PDF estimate (dotted curve), for a Gaussian PDF (solid curve) with zero mean and unit variance, with a Gaussian kernel of $\sigma = 0.25$ and a sample size of (a) 1, (b) 10, (c) 100, and (d) 1000. The circles indicate the observations in the sample.



Window functions

- We want $p_n(x)$ to be legitimate density, i.e. 1) $p_n(x) \geq 0$ for all x, and 2) $\int p_n(x)dx = 1$.
- If we maintain the relation $h^d_n = V_n$, this is guaranteed if the window function is a legitimate density:
- A popular choice for the window function is the Gaussian:

$$\varphi(\mathbf{u}) = \frac{1}{(2\pi)^{(d/2)}} \exp[-0.5\mathbf{u}^T\mathbf{u}].$$

- Note that V_n and h_n do not have a geometric interpretation anymore, they are related to the window function whose support usually spans the entire feature space.

Window width

- How should we select the window width h_n?
  If h_n is too large, the density estimate $p_n(x)$ will be very smooth and 'out-of-focus'.

- If h_n is too small, the estimate $p_n(x)$ will be just superposition of n sharp pulses centered at training samples, i.e. an erratic noisy estimate of the true density.

- In practice, we have to seek some acceptable compromise since the number of training samples is always limited and we may not be able to affect the number of available training samples.

- In practice, one selects h_1 and then asserts that $h_n = h_1/\sqrt{n}$.
  The selection of h_1 can be problematic.

Convergence of Parzen window estimates

- With an unlimited number of training samples it is possible to let V_n approach zero, and have $p_n(\mathbf{x})$ converge to $p(\mathbf{x})$.
- By convergence we mean the convergence in mean-square sense i.e. that for all $\mathbf{x}$

$$1.\ \lim_{n\to\infty} E[p_n(\mathbf{x})] = p(\mathbf{x}),$$
$$2.\ \lim_{n\to\infty} Var[p_n(\mathbf{x})] = 0.$$

- That means we want to obtain correct estimates on the average, and the variance within these estimates should be negligible as the number of training samples approaches infinity. Expectations are taken with respect to the sequence (of length n) of training samples.

Convergence of Parzen window estimates

In order to guarantee the convergence, we must place conditions on the unknown density, the window function $\phi(\mathbf{x})$, and the window width h_n:

- The density function must be continuous.
- The window function must be bounded and legitimate density.
- The values of the window function must be negligible at infinity.
  - $V_n \rightarrow 0$ when $n \rightarrow \infty$.
  - $nV_n \rightarrow \infty$ when $n \rightarrow \infty$.

Illustration

The behavior of the Parzen-window method

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

  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:

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

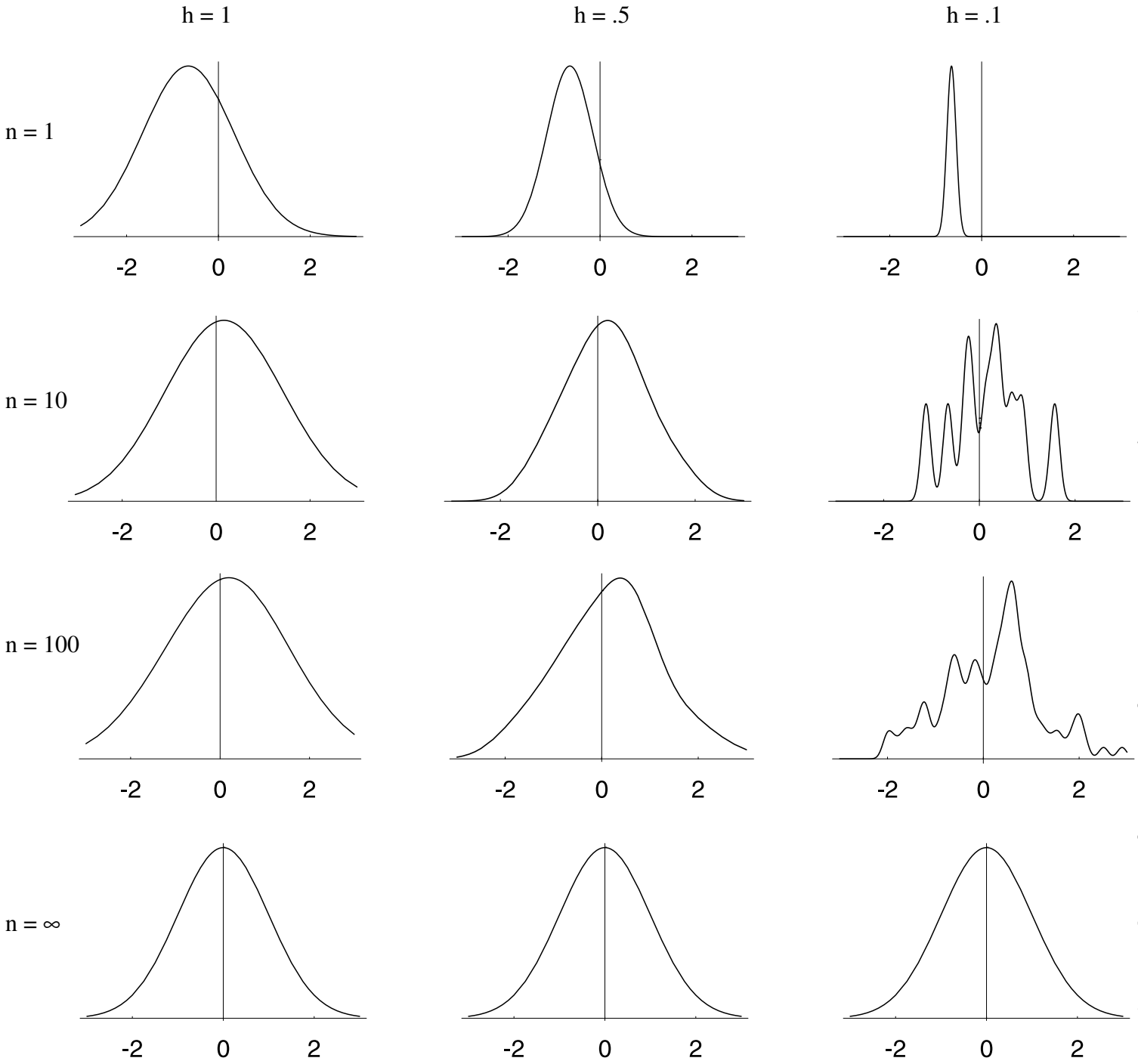
  is an average of normal densities centered at the samples x_i.

- Numerical results:

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

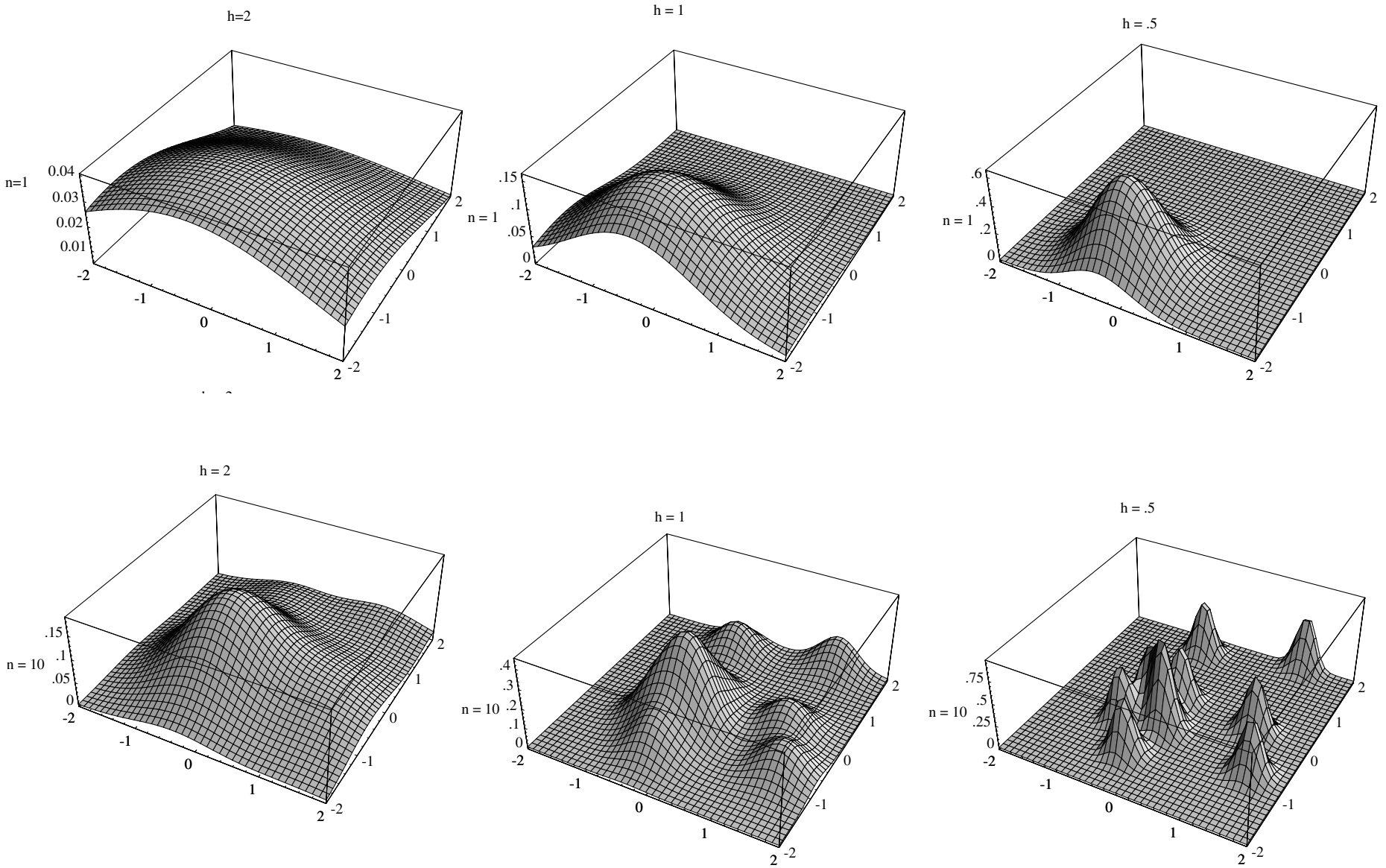
$$p_1(x) = \varphi(x - x_1) = \frac{1}{\sqrt{2\pi}} e^{-1/2(x - x_1)^2} \rightarrow N(x_1, 1)$$

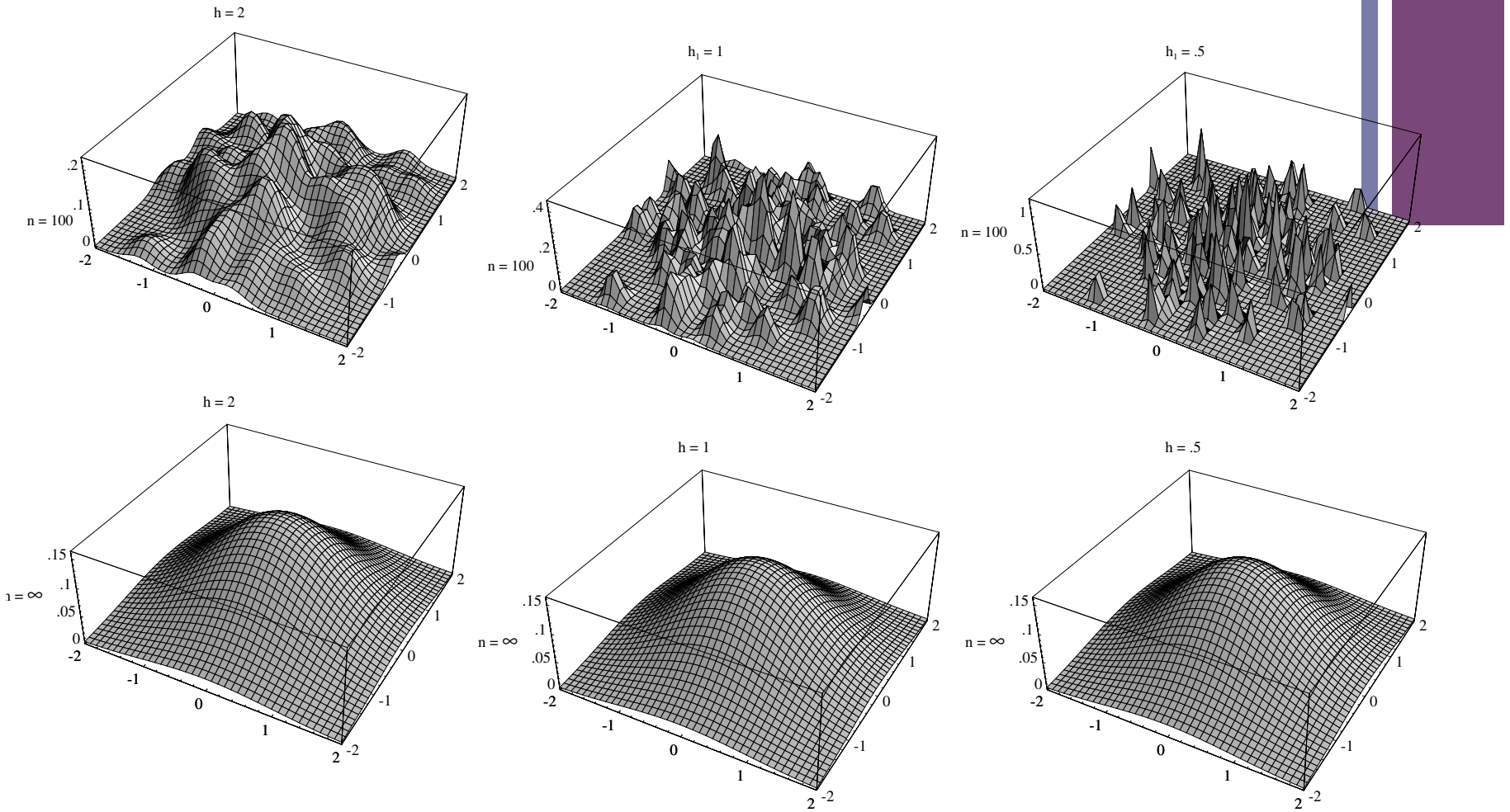
For $n = 10$ and $h = 0.1$, the contributions of the individual samples are clearly observable !



Parzen-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 = ∞ estimates are the same (and match the true generating function), regardless of window width h.

Analogous results are also obtained in two dimensions as illustrated:





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 = ∞ estimates are the same (and match the true generating distribution), regardless of window width h.

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 window function as illustrated in the following figure.

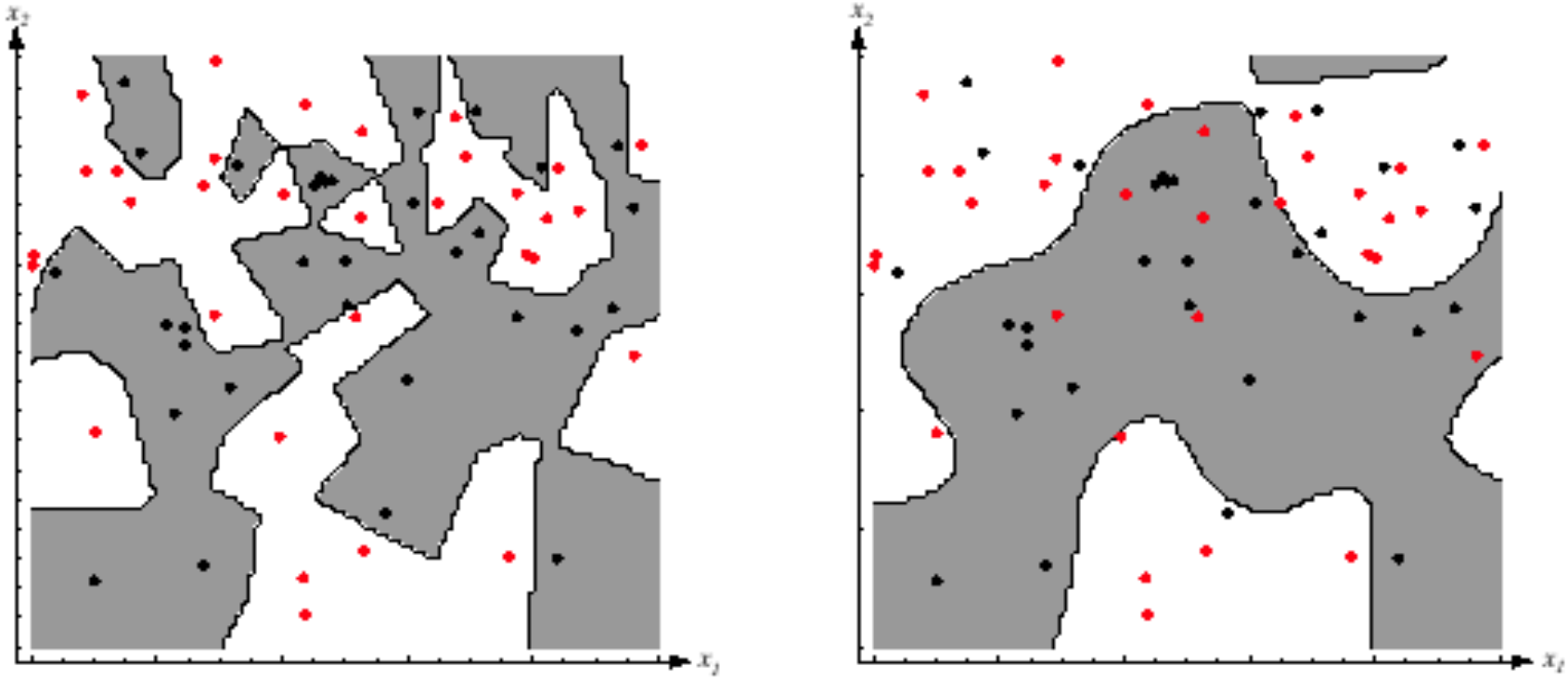


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h. At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the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

- Goal: a solution for the problem of the unknown "best" window function.
- Approach: Estimate density using real data points.
- 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 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 $k_n=k_1/\sqrt{n}$ and choosing different values for k_1.

Estimation of A Posteriori Probabilities

- Goal: estimate $P(\omega_i | x)$ 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 ω_i then: $p_n(x, \omega_i) = (k_i/n)/V$.
- A reasonable estimate for $P(\omega_i | x)$ is:

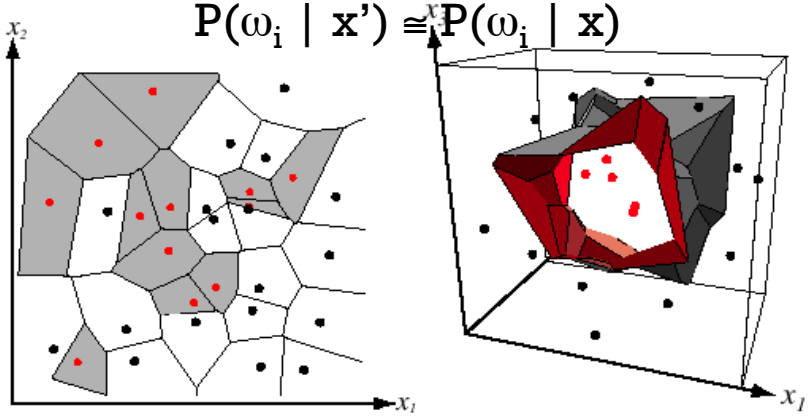
$$p_n(\omega_i \mid \mathbf{x}) = \frac{p_n(\mathbf{x}, \omega_i)}{\sum_{j=1}^{j=c} p_n(\mathbf{x}, \omega_j)} = \frac{k_i}{k}$$

- 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

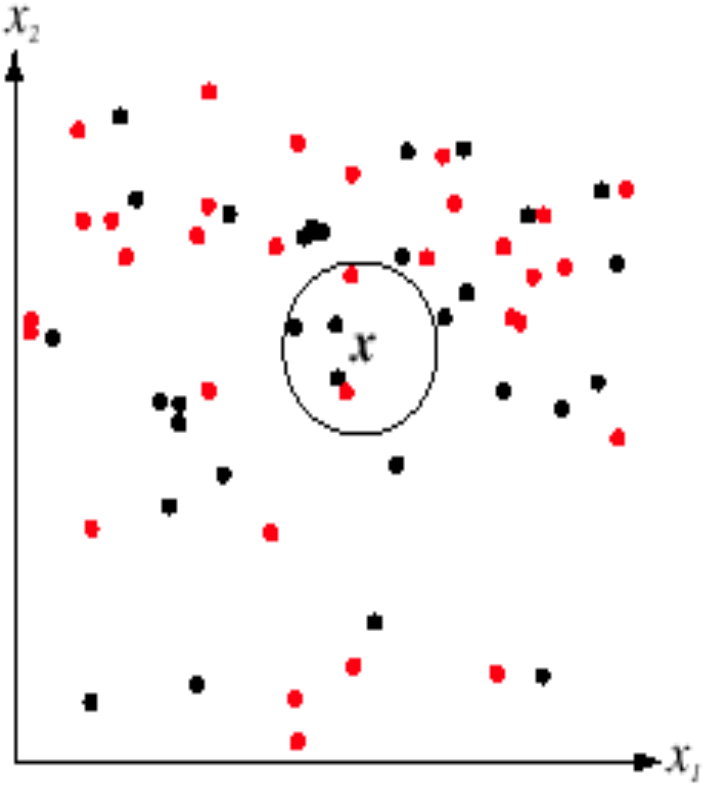
- Let $D_n = \{x_1, x_2, \ldots, x_n\}$ be a set of n labeled prototypes.
- Let $x' \in D_n$ be the closest prototype to a test point x.
- The nearest-neighbor rule for classifying x is to assign it the label associated with x'
- The nearest-neighbor rule leads to an error rate greater than the minimum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

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



- **This produces a Voronoi tesselation of the space, and the individual decision regions are called Voronoi cells.**
- **For large data sets, this approach can be very effective but not computationally efficient.**

The K-Nearest-Neighbor Rule



- The k-nearest neighbor rule is a straightforward modification of the nearest neighbor rule that builds on the concept of majority voting.
- Query starts at the data point, x, and grows a spherical region until it encloses k training samples.
- The point is labeled by a majority vote of the class assignments for the k samples.
- For very large data sets, the performance approaches the Bayes error rate.

- The computational complexity of this approach can be high. Each distance calculation is O(d), and thus the search is $O(dn^2)$.
- A parallel implementation exists that is O(1) in time and O(n) in space.
- Tree-structured searches can gain further efficiency, and the training set can be "pruned" to eliminate "useless" prototypes (complexity: $O(d^3 n^{(d/2)} \ln(n))$).

Properties of Metrics

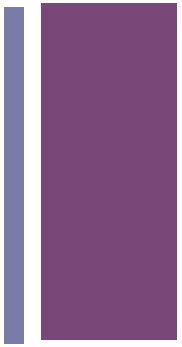
- Nonnegativity: $D(\mathbf{a},\mathbf{b}) \geq 0$
- Reflexivity: $D(\mathbf{a},\mathbf{b}) = 0 \quad iff \quad \mathbf{a} = \mathbf{b}$
- Symmetry: $D(\mathbf{a},\mathbf{b}) = D(\mathbf{b},\mathbf{a})$
- Triangle Inequality: $D(\mathbf{a},\mathbf{b}) + D(\mathbf{b},\mathbf{c}) \geq D(\mathbf{a},\mathbf{c})$
- Euclidean Distance: $D(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} (a_k - b_k)^2 \right)^{1/2}$
- The Minkowski metric is a generalization of a Euclidean distance:

$$L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} |a_k - b_k|^k \right)^{1/k}$$

and is often referred to as the L_k norm.

- The L_1 norm is often called the city block distance because it gives the shortest path between a and b, each segment of which is parallel to a coordinate axis.

Summary



- Motivated nonparameteric density estimation.
- Introduced Parzen windows.
- Introduced k-nearest neighbor approaches.
- Discussed properties of a good distance metric.

Exercise

- Do Example 1.7.1:3, and 1.8.1:2, not using a randomly generated data as shown, but on your project's dataset.