A temporary version for the 2018-04-11 lecture.

Nonparametric probability density estimation

Václav Hlaváč

Czech Technical University in Prague

Czech Institute of Informatics, Robotics and Cybernetics

166 36 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic

http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

also Center for Machine Perception, http://cmp.felk.cvut.cz

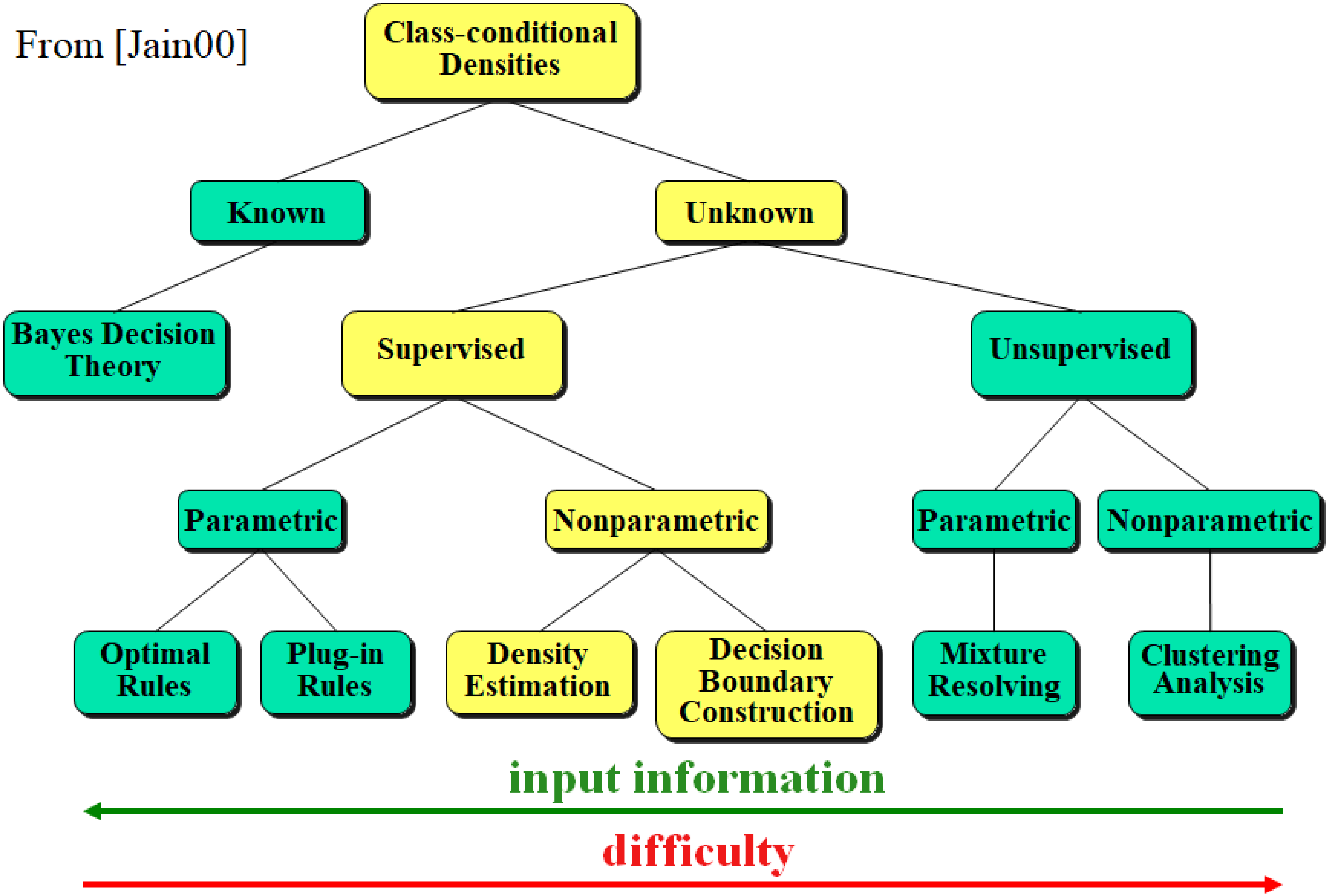
Courtesy: Vojtěch Franc, Min-Ling Zhang. Book Duda, Hart, Stork 2001.

Outline of the talk:

- Decision making methods taxonomy.
- Max. likelihood vs. MAP methods.
- Histogramming as a core idea.
- Towards non-parametric estimates.
- Parzen window method.
- k_n-nearest-neighbor method.

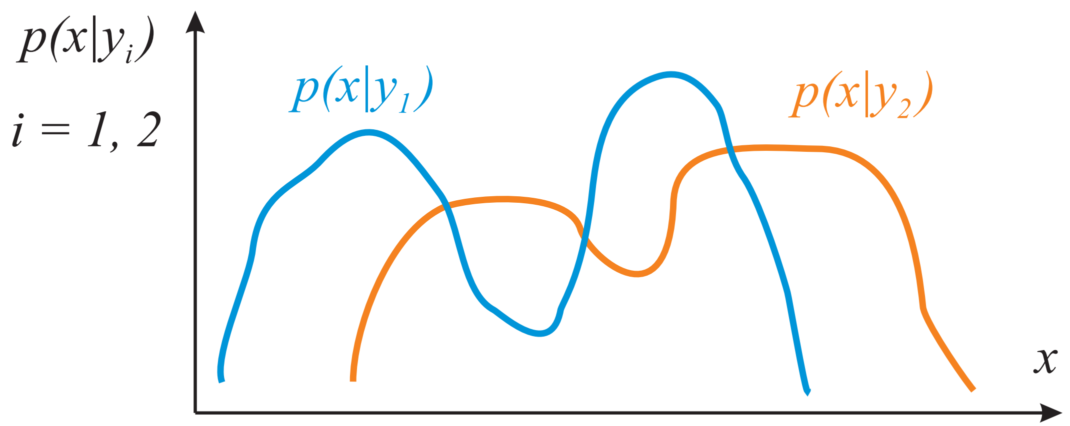
Decision making methods taxonomy according to statistical models

From [Jain00]



Unimodal and multimodal probability densities

- Parametric methods are good for estimating parameters of unimodal probability densities.
- Many practical tasks correspond to multimodal probability densities, which can be only rarely modeled as a mixture of unimodal probability densities.

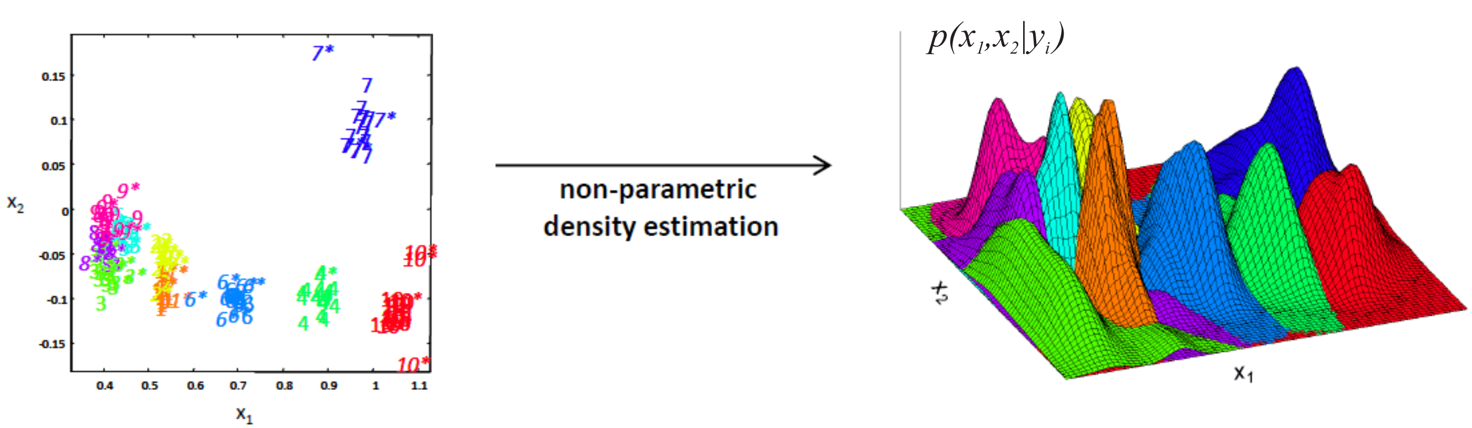


- Nonparametric method can be used for multimodal densities without the requirement to assume a particular type (shape) of the probability distribution.

There is the price to pay: more training data is needed.

Nonparametric density estimation

- Consider the observation $x \in X$ and the hidden parameter $y \in Y$ (a class label in a special case).
- In Naïve Bayes classification and in the parametric density estimation methods, it was assumed that either
  - The likelihoods $p(x|y_i)$ were known, or
  - their parametric form was known (cf. parametric density estimation methods explained already).
- Instead, nonparametric density estimation methods obtain the needed probability distribution from data without assuming a particular form of the underlying distribution.



Courtesy: Ricardo Gutierrez-Osuna, U of Texas

Nonparametric density estimation methods; two task types

- There are two groups of methods enabling to estimate the probability density function:

  1. The likelihood, i.e. the probability density $p(x|y_i)$ depends on a particular hidden parameter y_i. The (maximal) likelihood is estimated using sample patterns, e.g. a by the histogram method, Parzen window method.

  2. Maximally aposteriori probability (MAP) $p(y_i|x)$ methods, e.g. nearest neighbor methods.

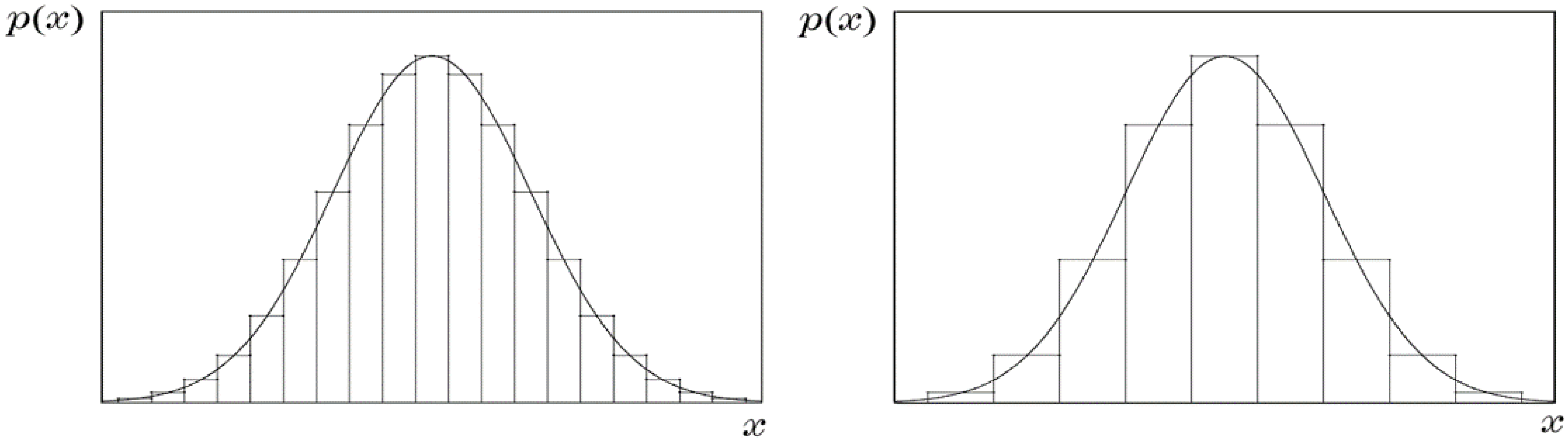
     MAP methods bypass the probability density estimation. Instead, they estimate the decision rule directly.

Idea = counting the occurrence frequency ⇒ histogram

- Divide the sample (events) space to quantization bins of the width h.
- Approximate the probability distribution function at the center of each bin by the fraction of points in the dataset that fall into a corresponding bin,

$$\hat{p}(x) = \frac{1}{h} \frac{\text{count of samples in the particular bin}}{\text{total number of samples}}$$

- The histogram method requires defining two parameters, the bin width h and the starting position of the first bin.



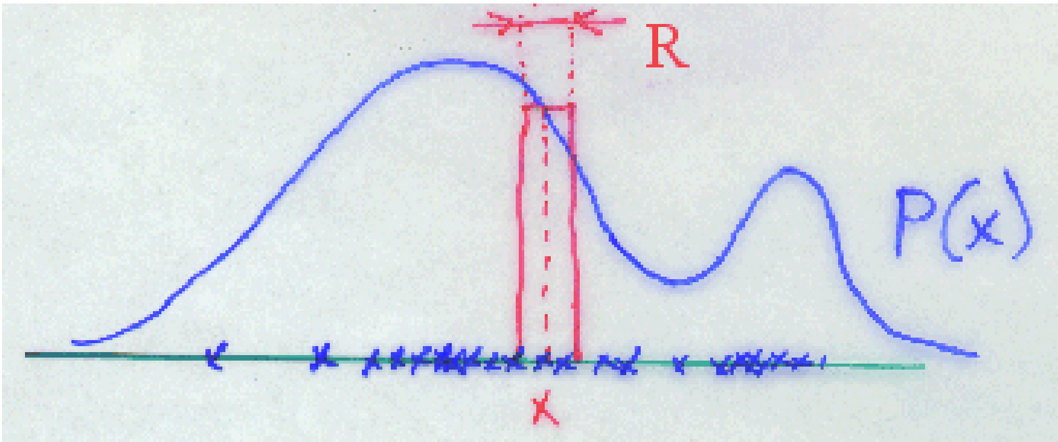
(a) (b)

Disadvantages of histogram-based estimates

- Discontinuities in the probability distribution estimates depend on the quantization bins density instead of the probability itself.
- Curse of dimensionality:
  - A fine representation requires many quantization bins.
  - The number of bins grows exponentially with the number of dimensions.
  - When not enough data is available, most of quantization bins remain empty.
- These disadvantages make the histogram-based probability density estimate useless with the exception of the fast data visualization in dimension 1 or 2.

Nonparametric estimates, ideas (1)

- Consider a dataset $X \in \mathcal{X}$, $X = \{x_1, x_2, \ldots, x_m\}$.
- Consider outcomes of experiments, i.e. samples x of a random variable.



- The probability that the sample x appears in a bin R (or more generally in a region R in multidimensional case) is $P = \Pr[x \in R] = \int_R p(x')\,\mathrm{d}x'$.
- Probability P is a smoothed version of the probability x.
- Inversely, the value $p(x)$ can be estimated from the probability P.

Nonparametric estimates, ideas (2)

- Suppose that n samples (vectors) $x_1, x_2, \ldots x_n$ are drawn from the probability distribution. We are interested, which k of these vectors fall in the particular discretization bin. Such a situation is described by the binomial distribution.
- A binomial experiment is a statistical experiment with the following properties:
  - The experiment consists of n repeated trials.
  - Each trial can result in just two possible outcomes (e.g. success, failure; yes, no; In our case, if a sample x_i, $i = 1, \ldots n$, falls in a particular discretization bin).
  - The trials are independent, i.e. the outcome of a trial does not effect other trials.
  - The probability of success P is the same on every experiment.

Nonparametric estimates, ideas (3)

- The probability that k of n samples fall in the particular discretization bin is given by the binomial distribution

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad 0 \le k \le n\,,$$

where the binomial coefficient, i.e. the number of combinations is $\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$ for $k \le n$ and zero otherwise.

Note that a k-combination is a selection of k items from a collection of n items, such that the order (unlike permutations) of selection does not matter.

- Binomial distribution is rather sharp at its expected value. It can be expectated that $\frac{k}{n}$ will be a good estimate of the probability P and consequently of the probability density p.
- The expected value $\mathcal{E}(k) = nP$; Consequently, $P = \frac{\mathcal{E}(k)}{n}$.

Nonparametric estimates, ideas (4)

- x is a point within the quantization bin R. We repeat from slide 8:
  $P = \Pr[x \in R] = \int\limits_R p(x')\,\mathrm{d}x'$.

- Let assume the quantization bin R is small; V is the volume enclosed by R. $p(\cdot)$ hardly varies within R. $P \simeq p(x)\,V$.

- $P = \frac{\mathcal{E}(k)}{n}$ and $P \simeq p(x)\,V$. Consequently, $p(x) = \frac{\frac{\mathcal{E}}{n}}{V}$.

- X follows the binomial probability distribution, see slide 10. X peaks sharply about $\mathcal{E}(X)$ for large enough n.

- Let k be the actual value of X after observing the i.i.d. examples $x_1, x_2, \ldots x_n$. The consequence is that $k \simeq \mathcal{E}[X]$.

- It implies from the previous two items: $p(x) = \frac{\frac{k}{n}}{V}$.

Parzen windows vs. k_n-nearest neighbor

- We like to show the explicit relation to number of dataset elements n (training samples in a special case in pattern recognition). We will denote the related quantities by the subscript n.

- Recall:

  R is the quantization bin. k is the number of samples falling into R.

  $p(x)$ is the probability that the sample x falls into the bin R.

$$R \quad \rightarrow \quad R_n \quad \text{(containing } x\text{)}$$

$$p(x) = \frac{\frac{k}{n}}{V} \quad \rightarrow \quad p_n(x) = \frac{\frac{k_n}{n}}{V_n}$$

Two basic probability density methods can be introduced:

- Parzen windows method: Fix the volume V_n and determine k_n.

- k_n-nearest-neighbor method: fix k_n and determine V_n.

Parzen Windows

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$ Fix V_n, and then determine k_n

Assume $\mathcal{R}_n$ is a d-dimensional hypercube (超立方体)

The length of each edge is h_n

$$V_n = h_n^d$$

Emanuel Parzen (1929-)

Determine k_n with window function (窗口函数), a.k.a. kernel function (核函数), potential function (势函数), etc.

Parzen Windows (Cont.)

Window function: $$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq 1/2; \quad j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$

$\varphi(\mathbf{u})$ defines a **unit hypercube** centered at the origin

$$\varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) = 1 \iff$$ $\mathbf{x}_i$ falls within the hypercube of volume V_n centered at $\mathbf{x}$

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

Parzen Windows (Cont.)

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

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

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

An average of functions of $\mathbf{x}$ and $\mathbf{x}_i$

$\varphi(\cdot)$ is not limited to be the hypercube window function of Eq.9 [pp.164]

$\varphi(\cdot)$ could be any pdf function:

$$\varphi(\mathbf{u}) \geq 0$$

$$\int \varphi(\mathbf{u})\, d\mathbf{u} = 1$$

Parzen Windows (Cont.)

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

$\varphi(\cdot)$ being a pdf function $\Rightarrow$ $p_n(\cdot)$ being a pdf function

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

Integration by substitution (换元积分)

Let $\mathbf{u} = (\mathbf{x}-\mathbf{x}_i)/h_n$

$$= \frac{1}{nV_n}\sum_{i=1}^{n}\int h_n^d\,\varphi(\mathbf{u})\;d(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n}\int \varphi(\mathbf{u})\,d(\mathbf{u}) = 1$$

window function (being pdf) $\varphi(\cdot)$ + window width h_n + training data $\mathbf{x}_i$ $\Rightarrow$ Parzen pdf $p_n(\cdot)$

Parzen Windows (Cont.)

Parzen pdf: $$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \qquad (V_n = h_n^d)$$

$\varphi(\cdot)$ being a pdf function $\Longrightarrow$ $p_n(\cdot)$ being a pdf function

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

What is the effect of h_n ("window width") on the Parzen pdf?

- $p_n(\mathbf{x})$: **superposition** (叠加) of n interpolations (插值)
- $\mathbf{x}_i$: contributes to $p_n(\mathbf{x})$ based on its "**distance**" from $\mathbf{x}$ (i.e. "$\mathbf{x}$-$\mathbf{x}_i$")

Parzen Windows (Cont.)

The effect of h_n ("window width")

$$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right) = \underbrace{\frac{1}{h_n^d}}_{\text{Affects the amplitude}}\varphi\underbrace{\left(\frac{\mathbf{x}}{h_n}\right)}_{\text{Affects the width}}$$

Affects the *amplitude* (vertical scale, 幅度)

Affects the *width* (horizontal scale, 宽度)

What do "amplitude" and "width" mean for a function?

For $\varphi(\mathbf{u})$:

$|\varphi(\mathbf{u})| \leq a$ *(amplitude)* $\Rightarrow$ $|\delta_n(\mathbf{x})| \leq (1/h_n^d) \cdot a$

$|u_j| \leq b_j$ *(width)* $(j = 1, \ldots, d)$ $\Rightarrow$ $|x_j| \leq h_n \cdot b_j \ (j = 1, \ldots, d)$

For $\delta_n(\mathbf{x})$:

Parzen Windows (Cont.)

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

$\delta_n(\cdot)$ being a pdf function

$$\int \delta_n(\mathbf{x})\,d\mathbf{x} = \int \frac{1}{h_n^d}\varphi\left(\frac{\mathbf{x}}{h_n}\right)\,d\mathbf{x}$$

Integration by substitution

Let $\mathbf{u} = \mathbf{x}/h_n$

$$= \int \frac{1}{h_n^d}\cdot\varphi\left(\mathbf{u}\right)\cdot h_n^d\,d\mathbf{u} = \int \varphi\left(\mathbf{u}\right)\,d\mathbf{u} = 1$$

Case I

$h_n \uparrow$

- Function amplitude (vertical scale) $\downarrow$
- Function width (horizontal scale) $\uparrow$

Case II

$h_n \downarrow$

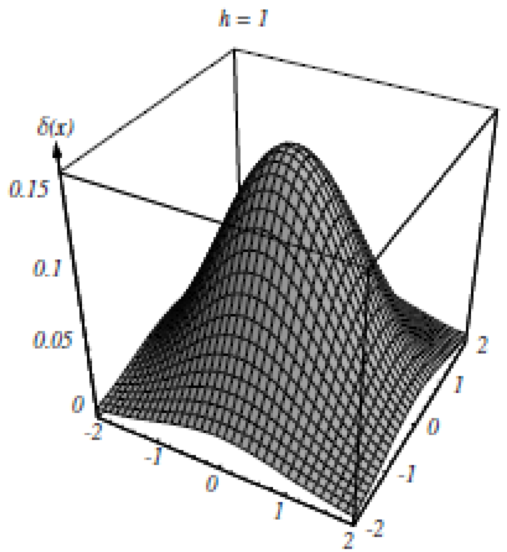
- Function amplitude (vertical scale) $\uparrow$
- Function width (horizontal scale) $\downarrow$

Parzen Windows (Cont.)

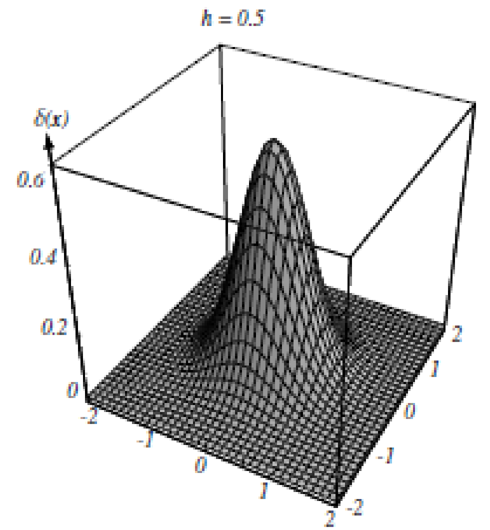
$$\delta_n(\mathbf{x}) = \frac{1}{h_n^d}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

Suppose $\varphi(\cdot)$ being a 2-d *Gaussian pdf*

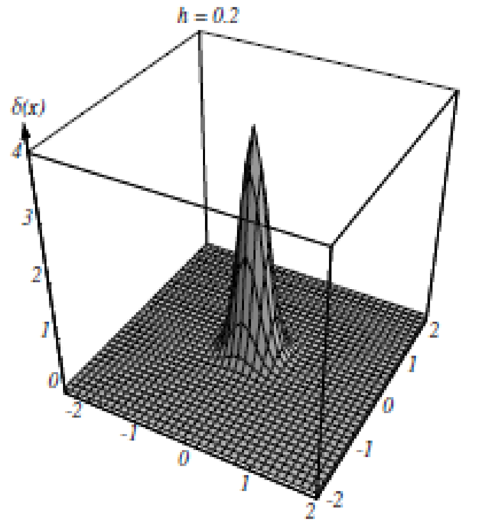
The shape of $\delta_n(\mathbf{x})$ with decreasing values of h_n



h=1.0



h=0.5



h=0.2

Parzen Windows (Cont.)

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i),\ \text{where}\ \delta_n(\mathbf{x}) = \frac{1}{h_n^d}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

- h_n very large ➔ $\delta_n(\mathbf{x})$ being *broad* with *small amplitude*

  $p_n(\mathbf{x})$ will be the superposition of n broad, slowly changing (慢变) functions, i.e. being *smooth* (平滑) with *low resolution* (低分辨率)

- h_n very small ➔ $\delta_n(\mathbf{x})$ being *sharp* with *large amplitude*

  $p_n(\mathbf{x})$ will be the superposition of n sharp pulses (尖脉冲), i.e. being *variable/unstable* (易变) with *high resolution* (高分辨率)

A compromised value (折衷值) of h_n should be sought for limited number of training examples

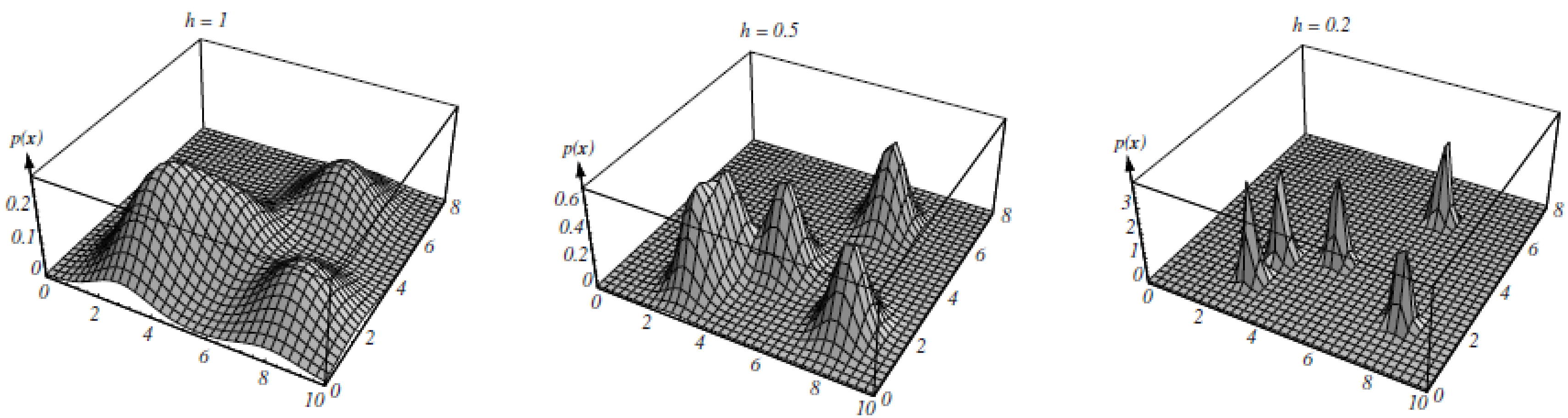
Parzen Windows (Cont.)

More illustrations:
Subsection 4.3.3 [pp.168]

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i), \text{ where } \delta_n(\mathbf{x}) = \frac{1}{h_n^d}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

Suppose $\varphi(\cdot)$ being a 2-d *Gaussian pdf* and *n*=5

The shape of $p_n(\mathbf{x})$ with decreasing values of h_n



h=1.0 *h*=0.5 *h*=0.2

k_n-Neareast-Neighbor

$$p_n(\mathbf{x}) = \frac{k_n/n}{V_n}$$ Fix k_n, and then determine V_n

specify k_n ➔ center a cell about **x** ➔ grow the cell until capturing k_n *nearest examples* ➔ return cell volume as V_n

The principled rule to specify k_n [pp.175]

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

$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$

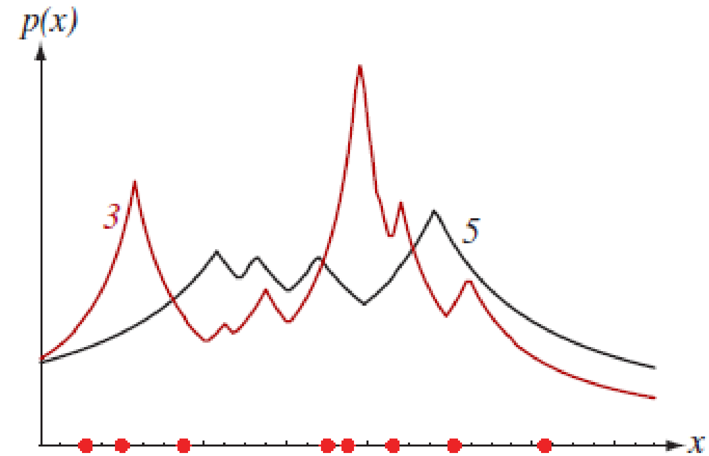
➔ A rule-of-thumb choice for k_n: $k_n = \sqrt{n}$

k_n-Neareast-Neighbor (Cont.)

Eight points in one dimension (n=8, d=1)

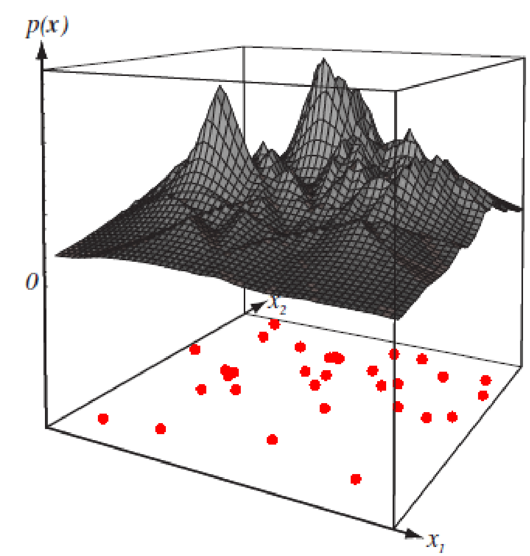
red curve: k_n=3

black curve: k_n=5



Thirty-one points in two dimensions (n=31, d=2)

black surface: k_n=5



Summary

- Basic settings for nonparametric techniques
  - Let the data speak for themselves
    - Parametric form not assumed for class-conditional pdf
  - Estimate class-conditional pdf from training examples

    ➔ Make predictions based on Bayes Formula
- Fundamental result in density estimation

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

n: # training examples

V_n: volume of region $\mathcal{R}_n$ containing $\mathbf{x}$

k_n: # training examples falling within $\mathcal{R}_n$

Summary (Cont.)

- Parzen Windows: Fix V_n ➔ Determine k_n
  - Effect of h_n (window width): A compromised value for a fixed number of training examples should be chosen

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

$\varphi(\cdot)$ being a pdf function ➔ $p_n(\cdot)$ being a pdf function

window function (being pdf) $\varphi(\cdot)$ + window width h_n + training data $\mathbf{x}_i$ ➔ Parzen pdf $p_n(\cdot)$

Summary (Cont.)

- k_n-nearest-neighbor: Fix k_n ➔ Determine V_n

specify k_n ➔ center a cell about **x** ➔ grow the cell until capturing k_n *nearest examples* ➔ return cell volume as V_n

The principled rule to specify k_n [pp.175]

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

$$\lim_{n\to\infty} \frac{k_n}{n} = 0$$

A rule-of-thumb choice for k_n: $k_n = \sqrt{n}$