Nonparametric probability density estimation

Václav Hlaváč

Czech Technical University in Prague

Czech Institute of Informatics, Robotics and Cybernetics

160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic

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

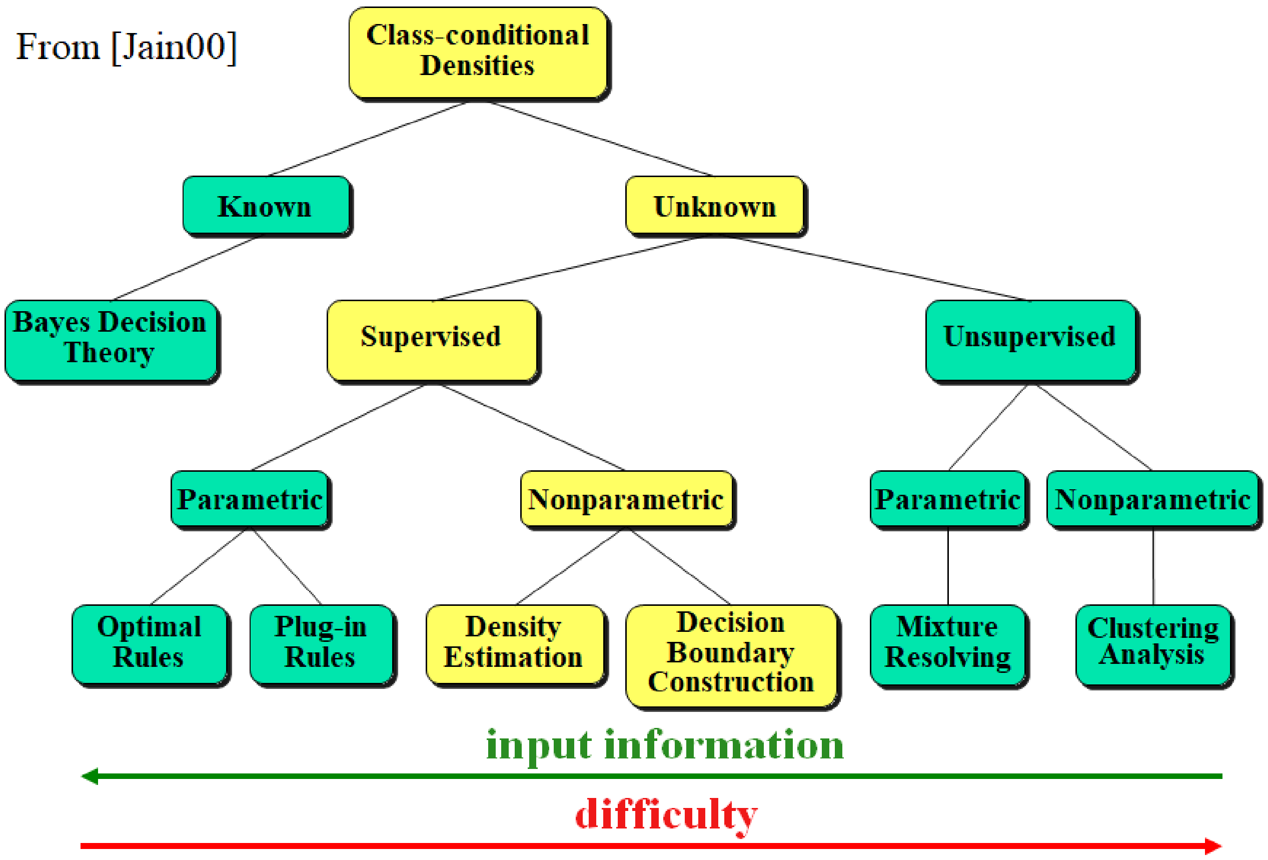
also Center for Machine Perception, `http://cmp.felk.cvut.cz`

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

Outline of the talk:

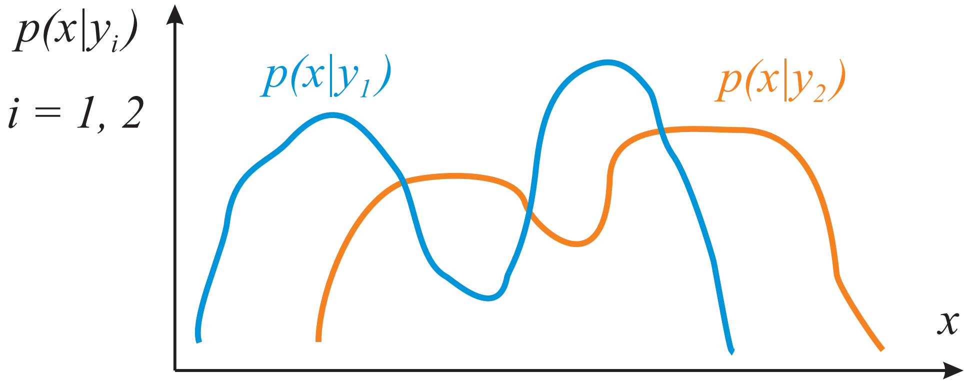
- Decision making methods taxonomy.
- Max. likelihood vs. MAP methods.
- Histogramming as a core idea.
- Towards non-parametric estimates.
- Parzen window method.
- k_n-nearest-neighbor method.

Decision making methods taxonomy according to statistical models



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.

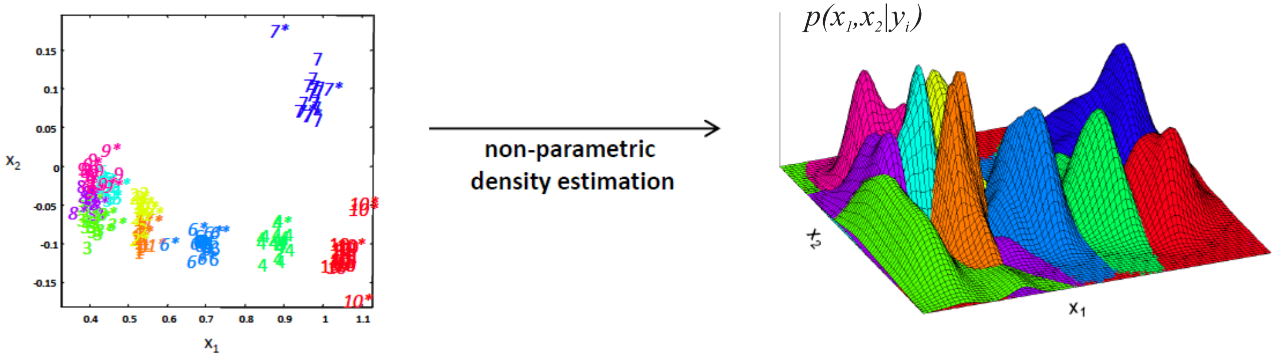


- Nonparametric method can be used for multimodal densities without the requirement to assume a particular type (shape) of the probability distribution.

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 the Naïve Bayes classification and in the parametric density estimation methods, we assume knowing either
  - The likelihoods (also class-conditional probabilities) $p(x|y_i)$, or
  - their parametric form (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 class-conditional 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 (also called the kernel smoothing function).

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

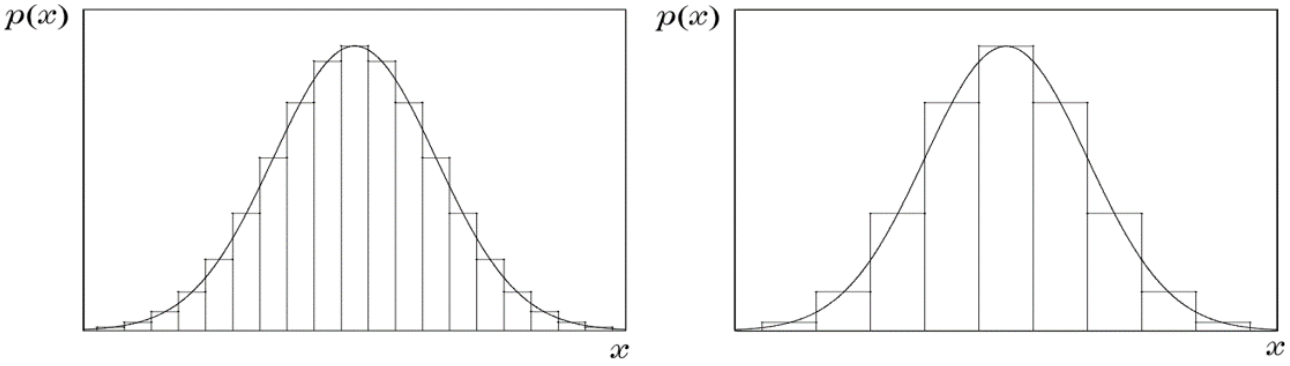
     MAP methods bypass the probability density estimation. Instead, they estimate the decision rule directly.

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. h is the width of the bin.

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

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



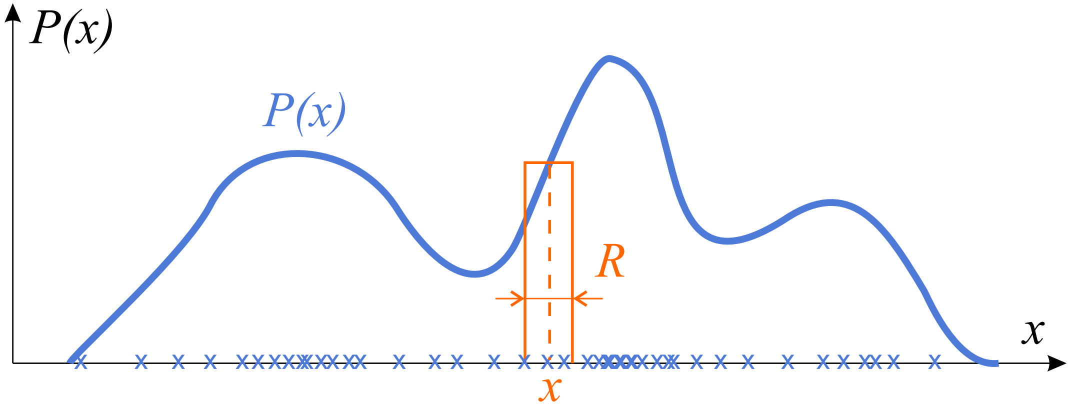
(a) (b)

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 $P = \Pr[x \in R] = \int\limits_R p(x')\, \mathrm{d}x'$.
- Probability P is a smoothed version of the probability distribution $p(x)$.
- 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 the number n of elements in the dataset (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_n is the number of samples falling into R.
  $p(x)$ is the probability that the sample x falls into the bin R.

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

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

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 window (1)

- $p_n(x) = \frac{\frac{k_n}{n}}{V_n}$; Fix the volume V_n and determine k_n.
- Assume R_n is a d-dimensional hypercube. The length of each edge is h_n. It implies $V_n = h_n^d$.
- Determine k_n with a Parzen window function (also called kernel smoothing function or potential function).
- One possiblity: a hypercube window function

Emanuel Parzen
(1929-2016)
Photo from 2006

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

- $\varphi(\mathbf{u})$ defines a unit hypercube centered at the origin. $\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) = 1$, i.e., $\mathbf{x}_i$ 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 window (2)

- Combining $p_n(x) = \frac{\frac{k_n}{n}}{V_n}$ and $k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$ results in Parsen pdf

  $p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$, i.e. an average of functions of $\mathbf{x}$ and $\mathbf{x}_i$.

  $V_n = h_n^d$; $\varphi(\cdot)$ is a pdf function $\Rightarrow p_n$ is also a pdf function.
- The window function $\varphi(\cdot)$ is not limited to a hypercube window function from Slide 13. $\varphi(\cdot)$ can be any probability distribution function; $\varphi(\mathbf{u}) \geq 0$; $\int \varphi(\mathbf{u})\,\mathrm{d}\mathbf{u} = 1$.
- $\int p_n(x)\,\mathrm{d}\mathbf{x} = \frac{1}{nV_n} \sum_{i=1}^{n} \int \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) \mathrm{d}\mathbf{x} = \left(\text{integration by substitution } \mathbf{u} = \frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) =$

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

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

Parzen window, superposition, distance

- Parsen probability distribution function (repeated from Slide 14): $p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$
- Simplification by the substitution $\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$ yields $p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \delta_n(\mathbf{x}-\mathbf{x}_i)$
  - $p_n(\mathbf{x})$ is a superposition of n interpolants.
  - $\mathbf{x}_i$ contributes to $p_n(\mathbf{x})$ based on its “distance” from $\mathbf{x}$, i.e. $\mathbf{x}-\mathbf{x}_i$.

What is the effect of the window width h_n on the Parzen probability distribution function?

What is the effect of "window width" h_n on the Parzen probability density function?

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

- $\frac{1}{h_n^d}$ affects the amplitude (also vertical scale).
- $\frac{\mathbf{x}}{h_n}$ affects the width (also horizontal scale).

For $\varphi(()u)$:		For $\delta_n(\varphi(\mathbf{x})$:
$\lvert\varphi(\mathbf{i})\rvert \leq a$ (amplitude)	$\Rightarrow$	$\lvert\delta_n(\mathbf{x})\rvert \leq \frac{a}{h_n}$
$\lvert u_j\rvert \leq b_j$ (width), $j = 1, \ldots, d$.	$\Rightarrow$	$\lvert x_j\rvert \leq h_n \cdot b_j$, $j = 1, \ldots, d$.

$$\int \delta_n(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \frac{1}{h_n^d}\,\varphi\left(\frac{\mathbf{x}}{h_n}\right)\mathrm{d}\mathbf{x} = \left(\text{integration by substitution } \mathbf{u} = \frac{\mathbf{x}}{h_n}\right) =$$

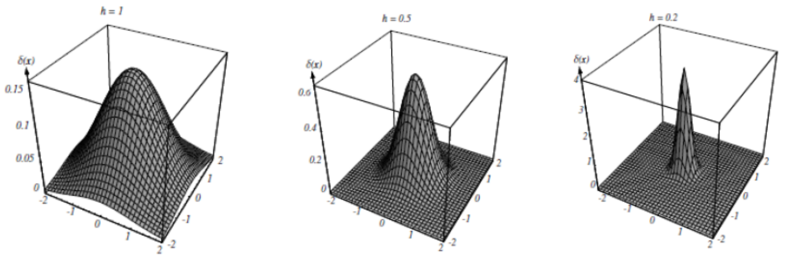
$$\int \frac{1}{h_n^d}\,\varphi(\mathbf{u})\,h_n^d \mathrm{d}\mathbf{u} = \int \varphi(\mathbf{u})\mathrm{d}\mathbf{u} = 1$$

Effect of “window width: h_n” (2)

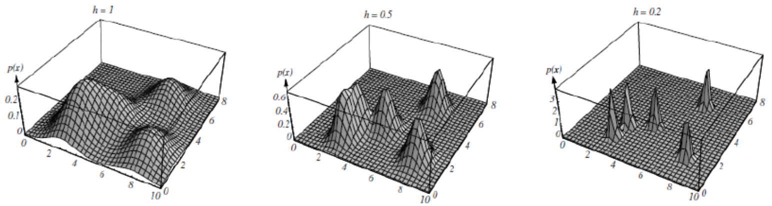
Case one:
If h_n increases $\Rightarrow$ the amplitude (vertical scale) decreases and the function width (horizontal scale) increases.

Case two:
If h_n decreases $\Rightarrow$ the amplitude (vertical scale) increases and the function width (horizontal scale) increases.

Example 1: The influence of h on the shape of $\delta_n(x)$ for a single 2D Gaussian



Example 2: The influence of h on the shape of $\delta_n(x)$ consisting of five 2D Gaussians



Effect of "window width": h_n (3)

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

- If h_n is very large then $\delta_n(\mathbf{x})$ is broad with small amplitude. p_n is a superposition of n broad, smooth functions with low resolution.
- If h_n is very small then $\delta_n(\mathbf{x})$ is sharp with large amplitude. p_n is a superposition of n sharp functions with high resolution.

One has to find a compromise value of h_n for limited number of training examples.

k_n-Nearest Neighbor

- Parsen probability distribution function $p_n(\mathbf{x})$, cf. Slide 14,

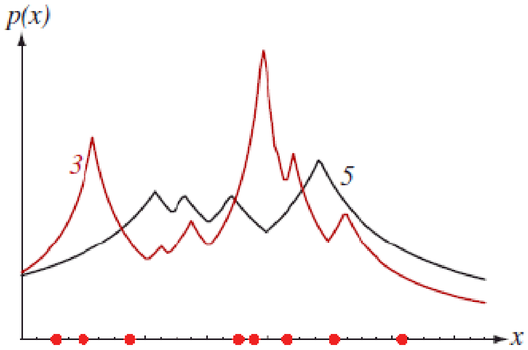
$$p_n(\mathbf{x}) = \frac{\frac{k_n}{n}}{V_n}$$

- Fix the number of data occurrences k_n in a quantization bin. $\Rightarrow$ Determine the volume V_n. of the quantization bin.
- The procedure:
  Specify k_n $\rightarrow$ Center a cell about $\mathbf{x}$ $\rightarrow$ Grow the cell until capturing k_n nearest examples $\rightarrow$ Return the cell volume V_n.
- The principled rule to specify k_n, page 175 Duda, Hart, Stork 2001:
  $\lim\limits_{n\to\infty} k_n = \infty$; $\lim\limits_{n\to\infty} \frac{k_n}{n} = \infty$
- A rule of thumb for the choice for k_n: $k_n = \sqrt{n}$.

k_n-Nearest Neighbor, examples

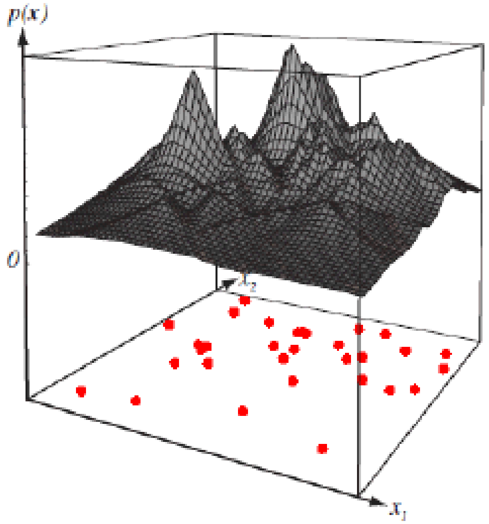
Example 1:
Eight points in one dimension; $(n = 1;\, d = 1)$.

- red curve: $k_n = 3$
- black curve: $k_n = 5$



Example 2:
31 points in two dimensions; $(n = 31;\, d = 2)$

- Black surface: $k_n = 5$



Summary (1)

- Let the data speak for themselves.
- Parametric methods are not considered for class-conditional probability $p(x|y_i)$ (also likelihood) density functions because it can be a multimodal function. Notation reminder: $x \in X$ is the observation and $y \in Y$ is the hidden parameter (class label in the more special case).
- Estimate the class-conditional pdf from training examples. Make predictions based on Bayes formula.
- Fundamental result in probability density function estimation:

$$p_n = \frac{\frac{k_n}{n}}{V_n}, \text{ where}$$

  - V_n is a volume of region R_n containing $\mathbf{x}$,
  - n is the number of training examples,
  - k_n is the number of training examples falling within R_n.

Summary (2), Parzen window

- Notation reminder: n is the number of elements in the dataset. k_n is the number of data occurrences in a particular quantization bin. V_n is the volume of this bin $\varphi(\cdot)$ is a Parzen window function.
- Fix the volume V_n of the quantization bin $\Rightarrow$ Determine the number of data occurrences k_n in a bin.
- Effect of the Parzen window width h_n. A compromised value for a fixed number of training samples has to be determined.
- Parzen window function $\varphi(\cdot)$ is a pdf function $\Rightarrow$ p_n is also a pdf function.

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

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

Summary (3), k_n-nearest neighbor

- Parsen probability distribution function $p_n(\mathbf{x})$, cf. Slide 14,

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

- Fix the number of data occurrences k_n in a quantization bin. $\Rightarrow$ Determine the volume V_n. of the quantization bin.
- The procedure:

  Specify k_n $\rightarrow$ Center a cell about $\mathbf{x}$ $\rightarrow$ Grow the cell until capturing k_n nearest examples $\rightarrow$ Return the cell volume V_n.
- A rule of thumb for the choice for k_n: $k_n = \sqrt{n}$.