Parzen Windows

Richard O. Duda, Peter E. Hart, David G. Stork

Pattern Classification - 1st edition: 1973, 2nd edition: 2000

February 15, 2012

Introduction

- Parzen Windows $\simeq$ Kernel density estimation
- Goal: given a set of d-dimensional samples, estimate the underlying probability distribution

Introductory example: hypercube

- We can use a d-dymensional hypercube as a window function to describe the distribution probability

$$V_n = h_n^d.$$

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

where h_n^d is the length of each edge and u_j is u's j^{th} normalized position inside the hypercube

- Now, probability distribution can be estimated as

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

General window functions

- For $p_n(x)$ to be a proper density function, we can use any window function satisfying

$$\varphi(\mathbf{x}) \geq 0$$
$$\int \varphi(\mathbf{u})\, d\mathbf{u} = 1$$

- Probability distribution can be written as

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

Parzen windows: width (h_n) effect

- There's little justification for a specific h_n size or a specific window function if no knowledge about the underlying distribution is available
- Usually, a Gaussian distribution is used for statistichal independence

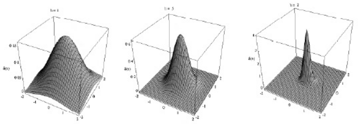


Figure 4.3: Examples of two-dimensional circularly symmetric normal Parzen windows $\varphi(\mathbf{x}/h)$ for three different values of h. Note that because the $\delta_k(\cdot)$ are normalized, different vertical scales must be used to show their structure.

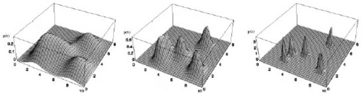


Figure 4.4: Three Parzen-window density estimates based on the same set of five samples, using the window functions in Fig. 4.3. As before, the vertical axes have been scaled to show the structure of each function.

Convergence of the mean

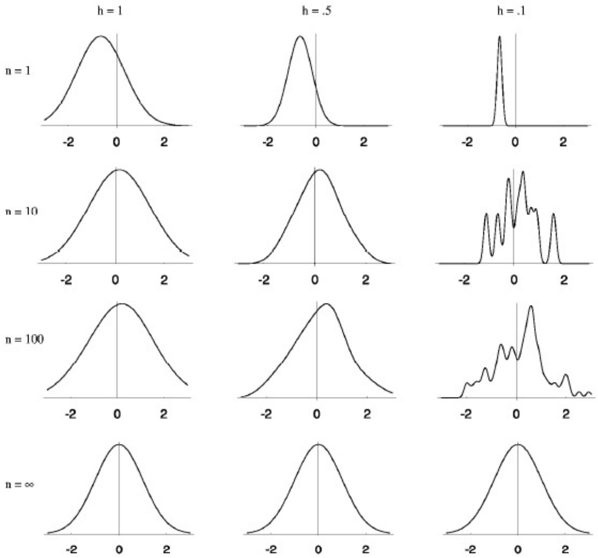
$$\begin{aligned}
\bar{p}_n(\mathbf{x}) &= \mathcal{E}[p_n(\mathbf{x})] \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathcal{E}\Big[\frac{1}{V_n}\varphi\Big(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\Big)\Big] \\
&= \int \frac{1}{V_n}\varphi\Big(\frac{\mathbf{x}-\mathbf{v}}{h_n}\Big)\, p(\mathbf{v})\, d\mathbf{v} \\
&= \int \delta_n(\mathbf{x}-\mathbf{v})p(\mathbf{v})\, d\mathbf{v}.
\end{aligned}$$

Convergence of the variance

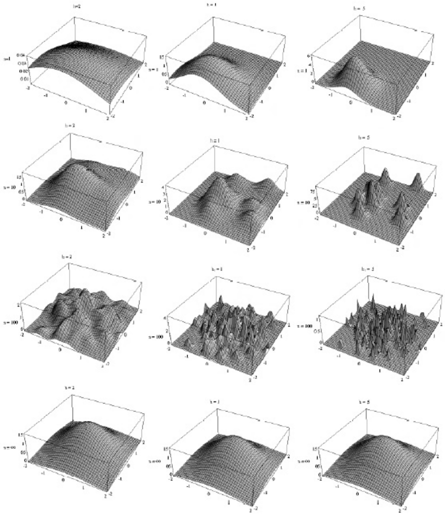
$$\sigma_n^2(\mathbf{x}) = \sum_{i=1}^{n} \mathcal{E}\left[\left(\frac{1}{nV_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right) - \frac{1}{n}\bar{p}_n(\mathbf{x})\right)^2\right]$$
$$= n\,\mathcal{E}\left[\frac{1}{n^2V_n^2}\varphi^2\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)\right] - \frac{1}{n}\bar{p}_n^2(\mathbf{x})$$
$$= \frac{1}{nV_n}\int \frac{1}{V_n}\varphi^2\left(\frac{\mathbf{x}-\mathbf{v}}{h_n}\right) p(\mathbf{v})\, d\mathbf{v} - \frac{1}{n}\bar{p}_n^2(\mathbf{x})$$

$$\sigma_n^2(\mathbf{x}) \leq \frac{\sup(\varphi(\cdot))\ \bar{p}_n(\mathbf{x})}{nV_n}$$

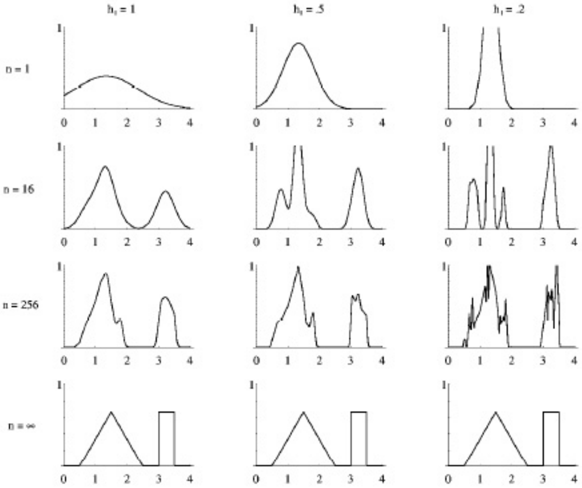
Illustrations I



Illustrations II



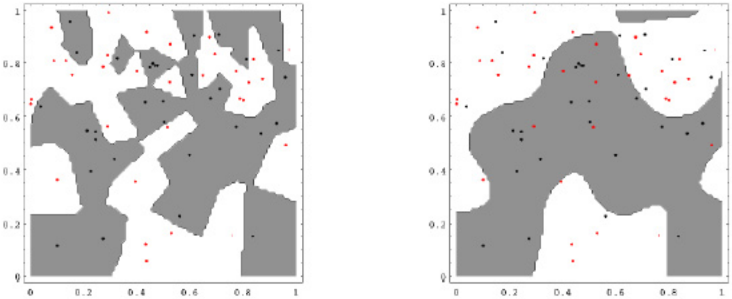
Illustrations III



Classification example

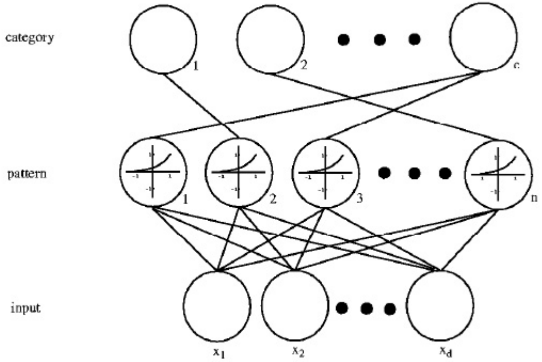
- A different estimate of the probability distribution can be built for each class
- Given a sample, the estimated probabilities to belong to each class can be computed
- The highest probability can be chosen as the classifier's output

2-class classification example: h_n size effect



Probabilistic Neural Network I

- Fast implementation of parzen-window classification



Algorithm 1 (PNN training)

<u>**begin**</u> <u>**initialize**</u> $j = 0, n = \#\text{patterns}$
    <u>**do**</u> $j \leftarrow j + 1$
        normalize : $x_{jk} \leftarrow x_{jk} / \left(\sum_{i}^{d} x_{ji}^2 \right)^{1/2}$
        train : $w_{jk} \leftarrow x_{jk}$
        <u>**if**</u> $\mathbf{x} \in \omega_i$ <u>**then**</u> $a_{ic} \leftarrow 1$
    <u>**until**</u> $j = n$

Algorithm 2 (PNN classification)

<u>**begin**</u> <u>**initialize**</u> $k = 0, \mathbf{x} = \text{test pattern}$
    <u>**do**</u> $k \leftarrow k + 1$
        $z_k \leftarrow \mathbf{w}_k^t \mathbf{x}$
        <u>**if**</u> $a_{kc} = 1$ <u>**then**</u> $g_c \leftarrow g_c + \exp[(z_k - 1)/\sigma^2]$
    <u>**until**</u> $k = n$
  <u>**return**</u> $class \leftarrow \arg\max_i g_i(\mathbf{x})$
<u>**end**</u>