Non-Parametric Techniques

Jacob Hays
Amit Pillay
James DeFelice

4.1, 4.2, 4.3

Parametric vs. Non-Parametric

- Parametric
  - Based on Functions (e.g Normal Distribution)
  - Unimodal – Only one peak
  - Unlikely real data confines to function
- Non-Parametric
  - Based on Data
  - As many peaks as Data has
  - Methods for both $p(w_j \mid x)$ and $P(w_j \mid x)$

Density Estimation

- Probability a vector **x** will fall in region *R*.

$$P = \int_{\Re} p(x')dx' \qquad (1)$$

- Assume n samples, identically distributed. By a Binomial equation, Probability that k samples are in Region R is

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \qquad (2)$$

- Expected value for k = nP, so P ≈ k/n

Density Estimation

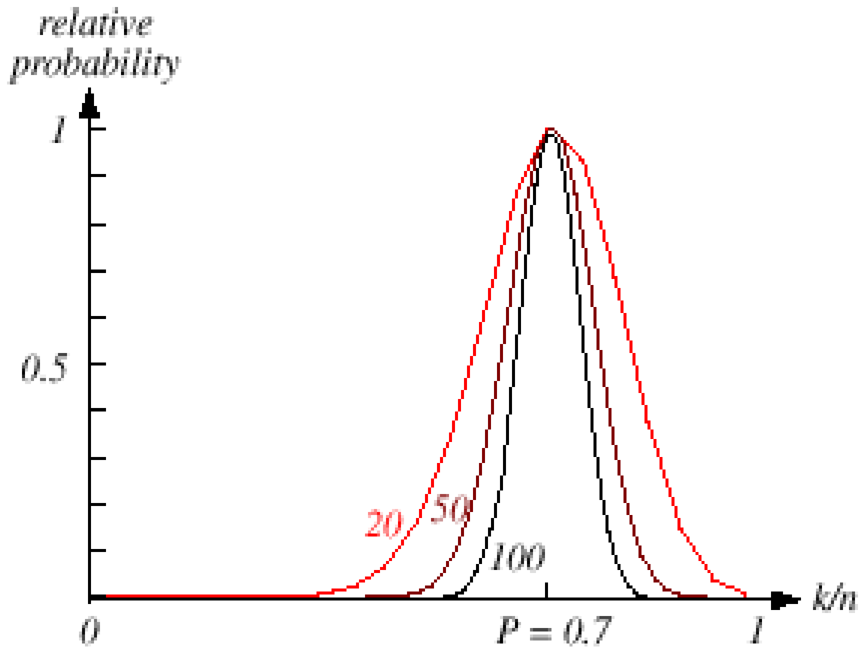
- For large n, k/n is a good estimate for P
- If p(x) is continuous, and p does not vary in R

$$P = \int_{\Re} p(x')dx' \cong p(x)V \qquad (4)$$

- Where V is volume of R
- Combine with P = k/n

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

If volume V is fixed, and n is increased towards ∞, P(x) converges to the average p of that volume.



It peaks at the true probability, which is 0.7, and with infinite n, will converge to 0.7.

Density Estimation

- If n is fixed, and V approaches zero, V will become so small it has zero samples, or reside directly on a point, making $p(x) \approx 0$ or ∞
- In Practice, can not allow volume to become too small, since data is limited.
  - If you use a non-zero V, estimation will have some variance in k/n from actual.
- In theory, with unlimited data, can get around limitations

Density Est. with Infinite data

- To get the density at x. Assume a sequence of regions (R_1 , R_2 , … R_n) that all contain **x**. In R_i the estimate uses i samples
- V_n is volume of R_n , k_n is the number of samples in R_n . $p_n(x)$ is the nth estimate for n.
  - $p_n(x) = k_n / n / V_n$
  - Goal is to get $p_n(x)$ to converge to $p(x)$

Convergence of $p_n(x)$ to $p(x)$

- $p_n(x)$ converges to $p(x)$ if the following is true

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

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

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

- Region R covers negligible space
- p(x) is average of infinite samples (unless p(x) = 0)
- The samples of k, are a negligible amount of the whole set n. n gets bigger faster then k does.

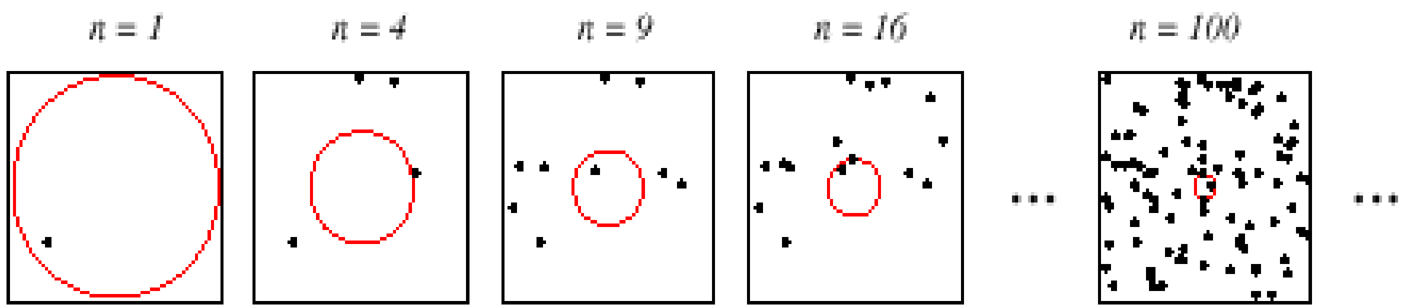
Satisfying conditions

- Two common methods to satisfy conditions that both converge
- Volume of Region R based on n
  - Parzen Windows
  - $V_n = 1 / \sqrt{n}$
- Number of points in region (k) based on n
  - k_n nearest neighbors
  - $k_n = \sqrt{n}$

Example

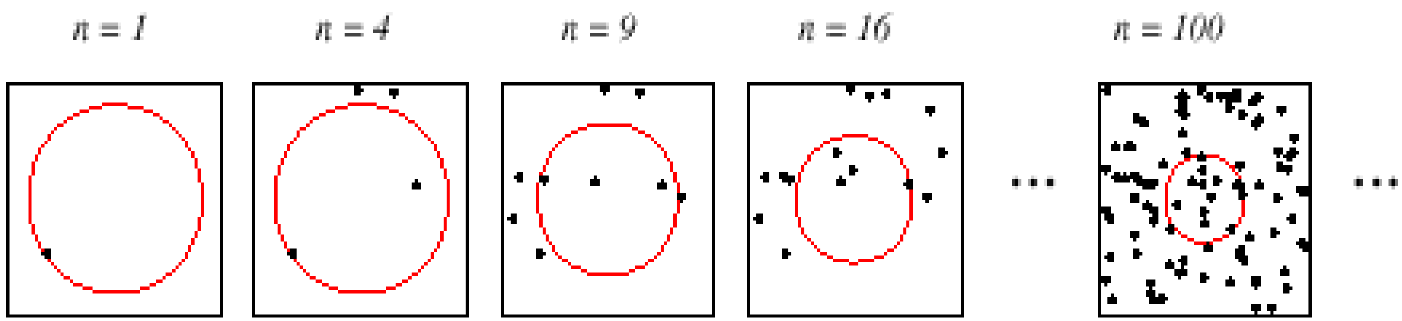
- Volume based on n

$V_n = 1/\sqrt{n}$



- Volume based on k_n

$k_n = \sqrt{n}$



Parzen Windows

- Assume Rn is d-dimensional hypercube
- h_n length of an edge of that cube
- Volume of cube is $V_n = h_n^d$
- Need to determine k_n (number of samples that fall within R_n)

Parzen Windows

- Define a "window" function that tells us if a sample is in R_n:

$$V_n = h_n^d \ (\mathrm{h_n} : \text{length of the edge of } \Re_n)$$

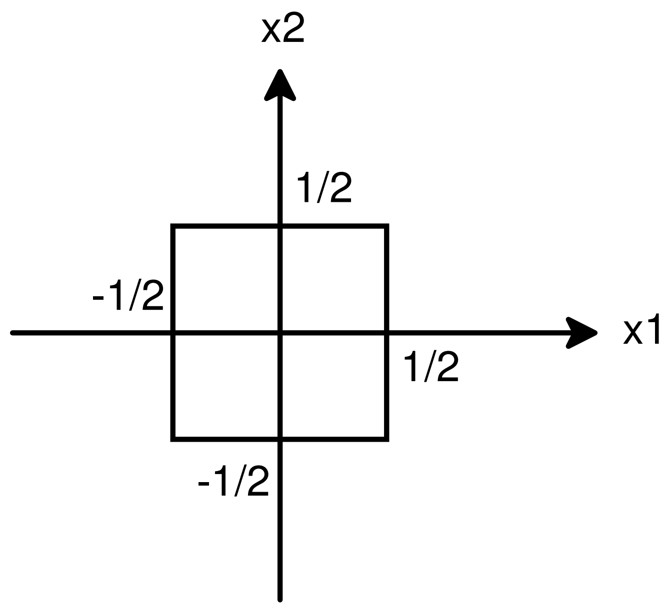
Let $\varphi(\mathrm{u})$ be the following window function :

$$\varphi(\mathrm{u}) = \begin{cases} 1 & \left|\mathrm{u}_j\right| \leq \dfrac{1}{2} \quad \mathrm{j} = 1, \ldots, \mathrm{d} \\ 0 & \text{otherwise} \end{cases}$$

Example

- Assume d = 2, $h_n = 1$
- $\varphi((x\text{-}x_i)/h_n) = 1$ if x_i falls within R_n

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



Parzen Windows

$$\varphi(\mathrm{u}) = \begin{cases} 1 & |\mathrm{u}_{\mathrm{j}}| \le \frac{1}{2} \quad \mathrm{j} = 1, \ldots, \mathrm{d} \\ 0 & \text{otherwise} \end{cases}$$

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

- Number of samples in R_n computed as k_n
- Derive new $p_n(x)$
  - Earlier, $p_n(x) = (k_n/n)/V_n$, now redefined as

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

Generalize $\varphi(x)$

- $p_n(x)$ is average of functions of x and samples x_i
- Window function $\varphi(x)$ is being used for interpolation
  - Each x_i contributes to $p_n(x)$ according to its distance from x
- We'd like $\varphi(x)$ to be a legitimate density function

$$\varphi(v) \geq 0$$

$$\int \varphi(v)\,dv = 1$$

Window Width

- Remember that:

$$V_n = h_n^d \ (\mathrm{h_n} : \text{length of the edge of } \Re_\mathrm{n})$$

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

- New definition:

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

- h_n clearly affects the amplitude and width of delta function

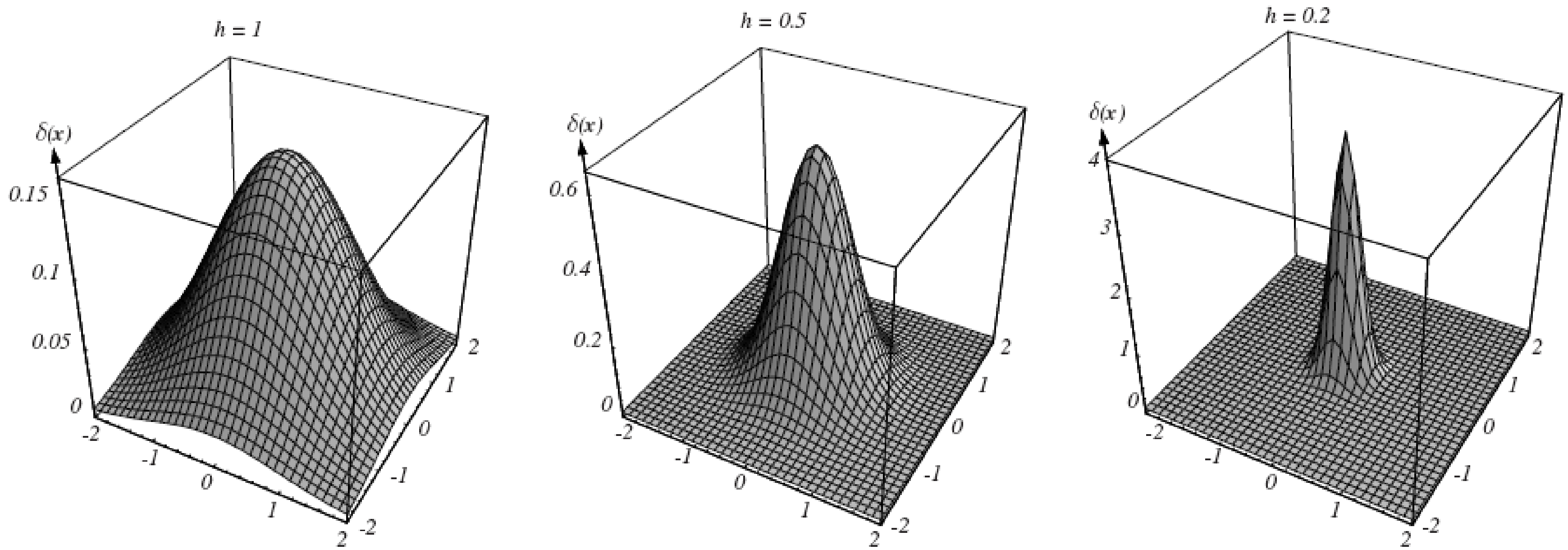
$$p_n(x) = \frac{1}{n}\sum_{i=1}^{i=n}\delta_n(x - x_i)$$

Window Width

- Very large h_n
  - Small amplitude of delta function
  - x_i must be far from x before $\delta_n(x-x_i)$ changes from $\delta_n(0)$
  - $p_n(x)$ is superposition of a broad, slowly changing function (out of focus)
  - Too little resolution
- Very small h_n
  - Large amplitude of delta function
  - Peak of $\delta_n(x-x_i)$ is large, occurs near $x=x_i$
  - $p_n(x)$ is superposition of sharp pulses (erratic, noisy)
  - Too much statistical instability

Window Width

- For any h_n distribution is normalized



$$\int \delta_n(x - x_i)dx = \int \frac{1}{V_n}\varphi\left(\frac{x - x_i}{h_n}\right)dx = \int \varphi(u)du = 1$$

Convergence

- With limited samples, best we can do for h_n is compromise
- With unlimited samples, we can let V_n slowly approach zero as n increases, and $p_n(x) \rightarrow p(x)$
- For any fixed x, $p_n(x)$ depends on r.v. samples (x_1, x_1,.. x_n) …

$$p_n(x) \text{ has some mean } \overline{p}_n(x) \text{ and variance } \sigma_n^2(x)$$

Convergence of mean

$$\overline{p}_n(x) = \int \delta_n(x-v)p(v)dv$$

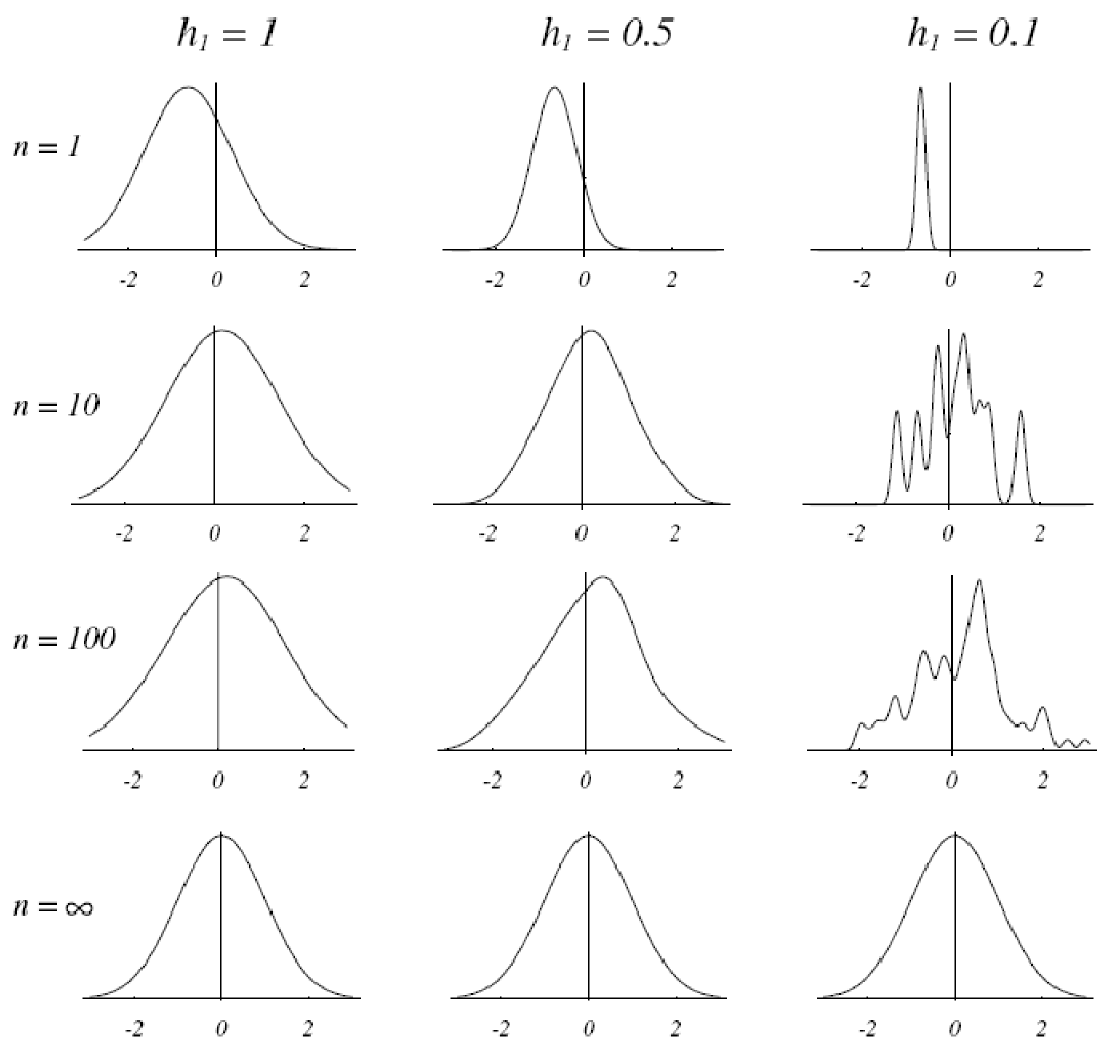
- Expected value of estimate is averaged value of unknown, true density *p(x)*
  - "blurred" or "smoothed" version of *p(x)* as seen through the averaging window
- Limits, as n → ∞
  - $V_n \rightarrow 0$
  - $nV_n \rightarrow \infty$
  - $\delta_n(x-v)$ → delta function centered at *x*
  - expected value of estimate → true density

Convergence of variance

- For any n
  - expected value of estimate → true density
    - if we let $V_n \rightarrow 0$
    - for some set of n samples estimate is useless ("spiky")
  - need to consider variance
    - Should let $V_n \rightarrow 0$ slower than n → ∞

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

Example



Illustration

- The behavior of the Parzen-window method
  - Case where $p(x) \rightarrow N(0,1)$

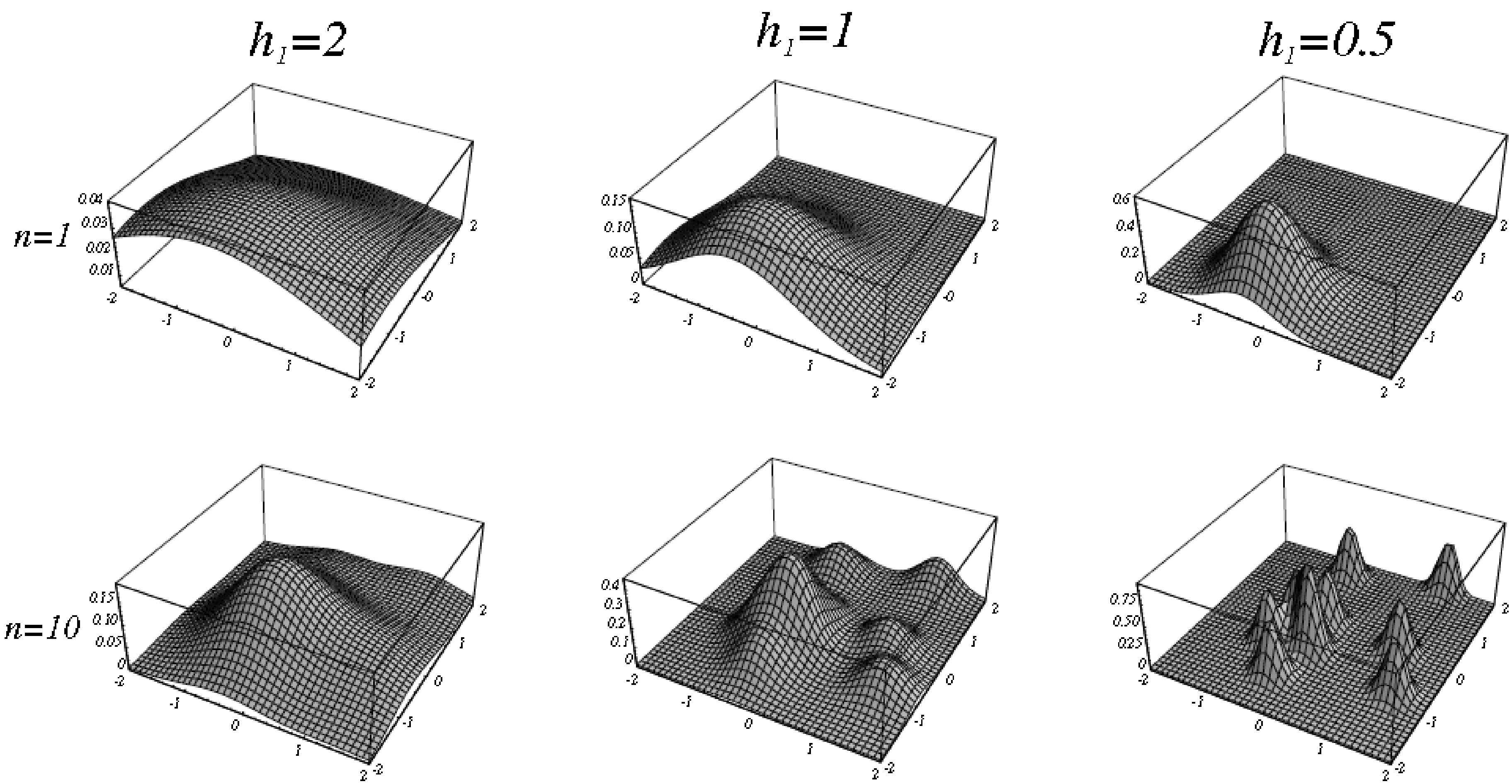
    Let $\varphi(u) = (1/\sqrt{2\pi})\exp(-u^2/2)$ and $h_n = h_1/\sqrt{n}$ $(n>1)$

    (h_1: known parameter)

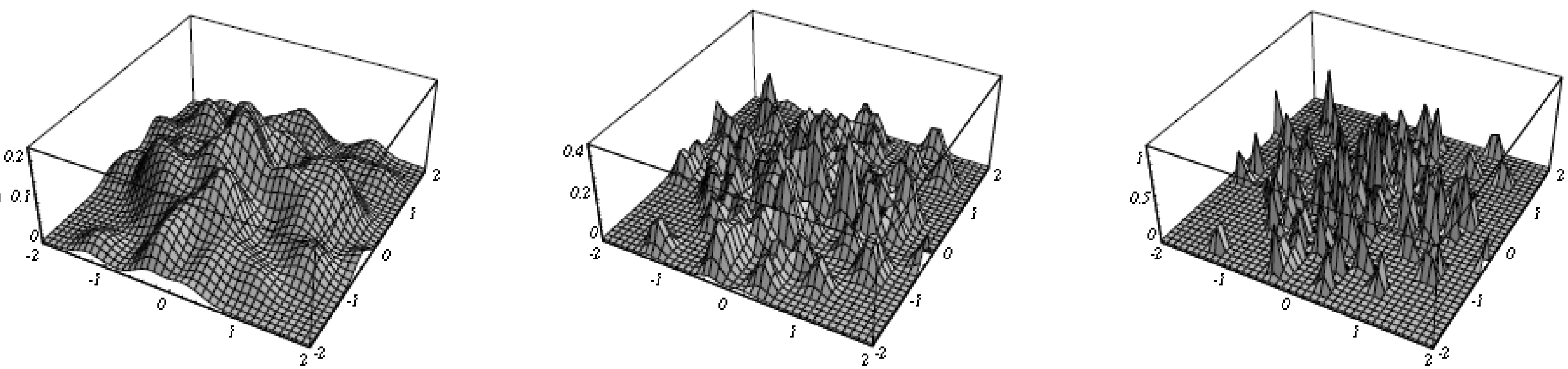
    Thus:

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

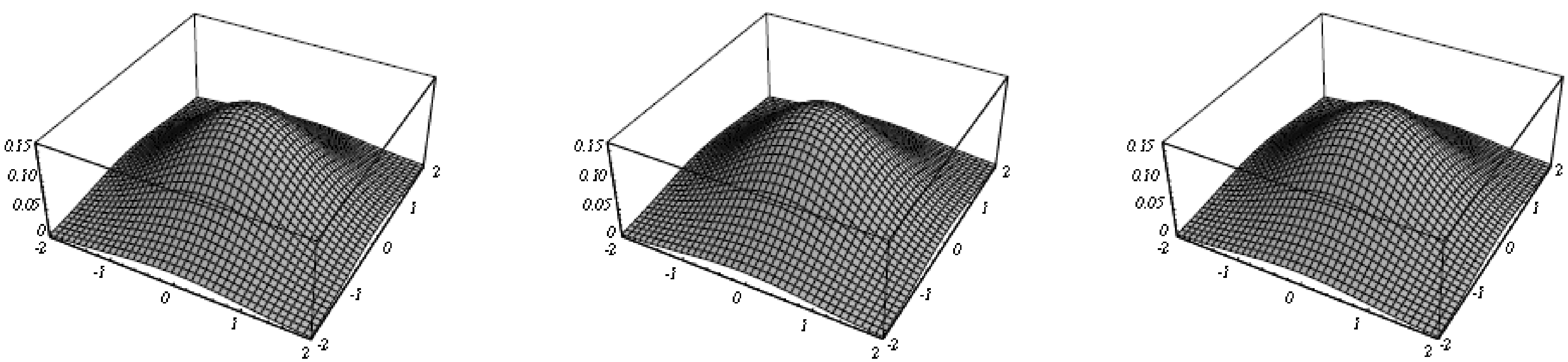
is an average of normal densities centered at the samples x_i.


h1=2
h1=1
h1=0.5
n=1
n=10
0.04
0.03
0.02
0.01
0.15
0.10
0.05
0.6
0.4
0.2
0.4
0.3
0.2
0.1
0.75
0.50
0.25

$n=100$



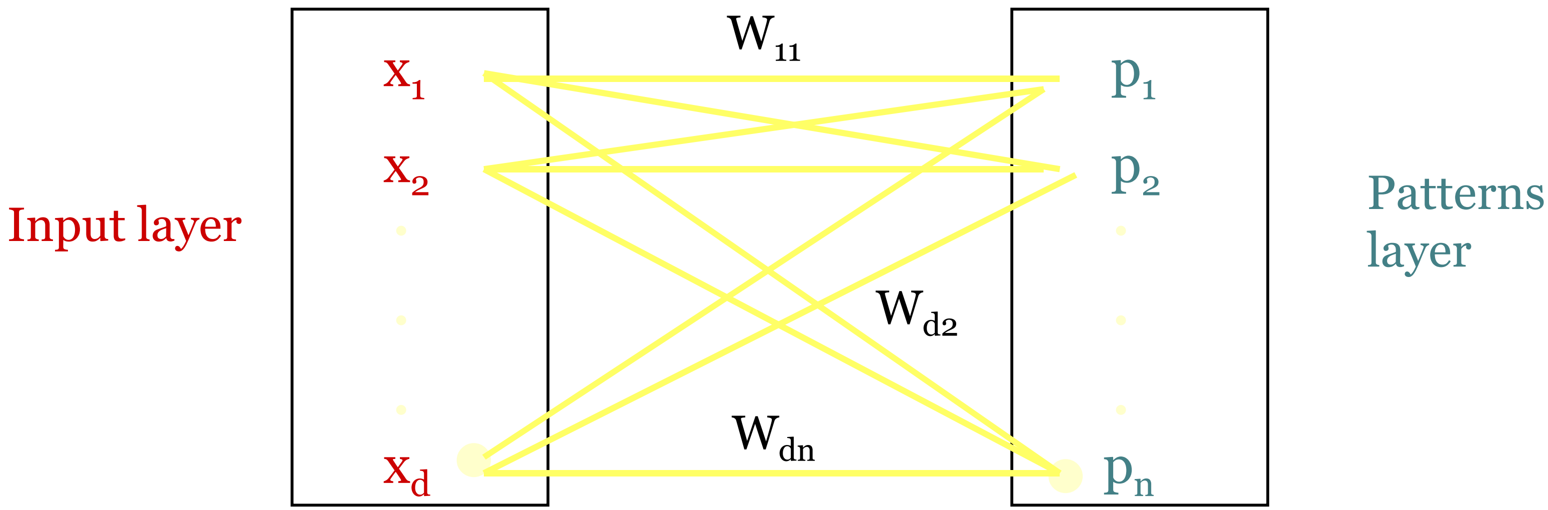
$n=\infty$



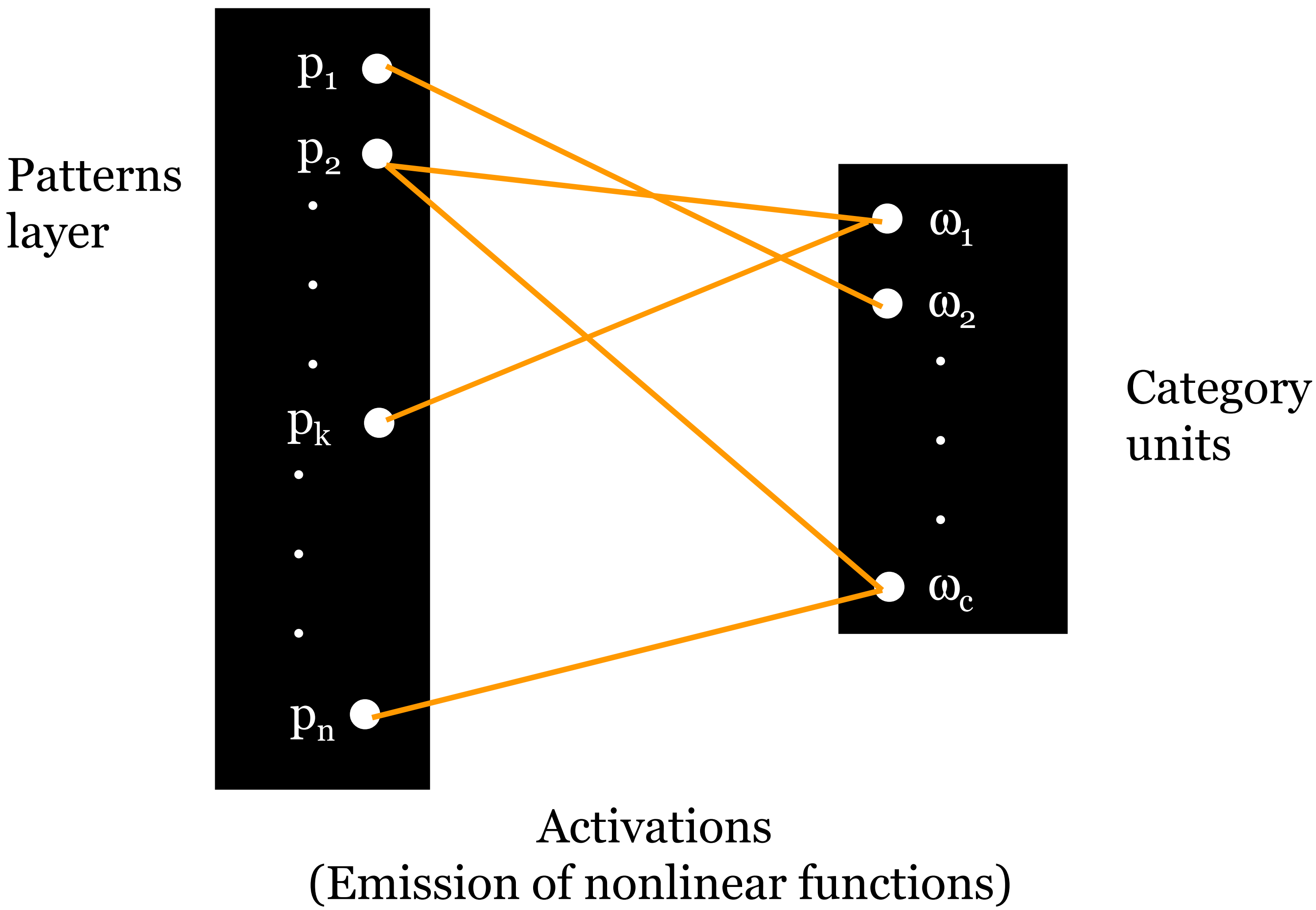
Probabilistic Neural Network

- The PNN for this case has
- 1. d input units comprising the input layer,
- 2. n pattern units comprising of the pattern layer,
- 3. c category units
- Each input unit is connected to each pattern unit.
- Each pattern unit is connected to one and only one

  category unit corresponding the category of the training sample.
- The connections from the input to pattern units have modifiable weights w which will be learned during the training.



Input layer
x_1
x_2
x_d
W_{11}
W_{d2}
W_{dn}
p_1
p_2
p_n
Patterns layer


Patterns layer
p_1
p_2
p_k
p_n
ω_1
ω_2
ω_c
Category units
Activations
(Emission of nonlinear functions)

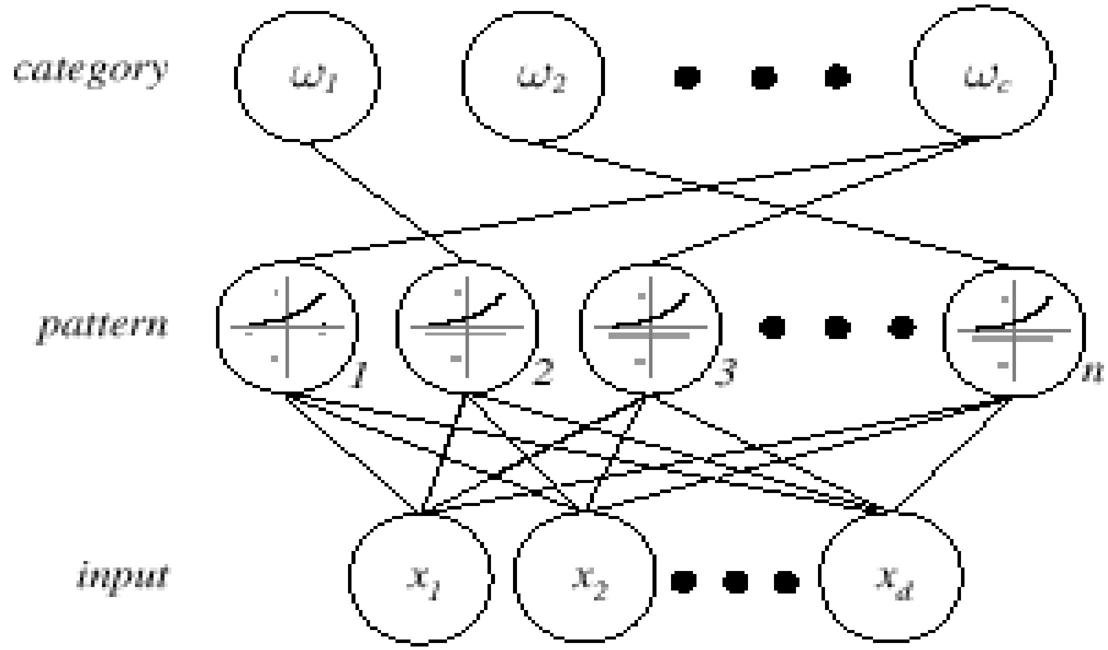


FIGURE 4.9. A probabilistic neural network (PNN) consists of d input units, n pattern units, and c category units. Each pattern unit forms the inner product of its weight vector and the normalized pattern vector $\mathbf{x}$ to form $z = \mathbf{w}^t\mathbf{x}$, and then it emits $\exp[(z-1)/\sigma^2]$. Each category unit sums such contributions from the pattern unit connected to it. This ensures that the activity in each of the category units represents the Parzen-window density estimate using a circularly symmetric Gaussian window of covariance $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is the $d \times d$ identity matrix. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.

PNN training

- The training procedure is simple consisting of three simple steps.
- 1) Normalize the training feature vectors so that $||x_i|| = 1$ for all i = 1…n.
- 2) Set the weight vector $w_i = x_i$ for all i = 1…n. w_i consists of weights connecting the input units to the *ith* pattern unit.
- 3) Connect the pattern unit *i* to the category unit corresponding to the category of x_i for all i = 1…n.

PNN Classification

1. Normalize the test pattern x and place it at the input units
2. Each pattern unit computes the inner product in order to yield the net activation

$$net_k = w_k^t . x$$

and emit a nonlinear function

$$f(net_k) = exp\left[\frac{net_k - 1}{\sigma^2}\right]$$

3. Each output unit sums the contributions from all pattern units connected

$$P_n(x \mid \omega_j) = \sum_{i=1}^{n} \varphi_i \propto P(\omega_j \mid x)$$

4. Classify by selecting the maximum value of $P_n(x \mid \omega_j)$ $(j = 1, \ldots, c)$

Advantages of PNN

- Speed of learning
  - Since $W_k=X_k$, it requires a single pass thru training
- Time complexity
  - For parallel implementation its O(1) as inner product can be done in parallel
- New training patterns can be incorporated quite easily

Summary

- Non parametric estimation can be applied to any random distribution of data
- Parzen window method provide a better estimation of pdf
- Estimation depends upon no. of samples and Parzen window size
- PPN gives an efficient Parzen window method implementation

References

- R.J. Schalkoff. (1992) Pattern Recognition: Statistical, Structural, and Neural Approaches, Wiley.*
- Pattern Classification (2nd Edition) by R.O. Duda, P. E. Hart and D. Stork, Wiley 2001