EE 6885 Statistical Pattern Recognition

Fall 2005
Prof. Shih-Fu Chang
http://www.ee.columbia.edu/~sfchang

Lecture 8 (10/5/05)

- Reading
  - Nonparametric Estimation
    - DHS Chap. 4.1-4.3
  - Paper: Bayesian Classifier with VQ and Parzen Window
    - A. Vailaya, M. Figueiredo, A. Jain, and HJ Zhang, "A Bayesian Framework for Semantic Classification of Outdoor Vacation Images," IEEE Trans. Image Processing, Vol. 10, No. 1, pp. 157-172, Jan. 2001.
- Homework #3, due Oct. 12th 2005
- Midterm Exam
  - Oct. 24th 2005 Monday 1pm-2:30pm (90mins)

Review

- Problem of Dimensionality
- Turk '78 example

$p(x \mid \omega_1) = N(\mu_1, I)$

$p(x \mid \omega_2) = N(\mu_2, I)$

where $\mu_1 = -\mu_2 = \mu = \{(1/i)^{1/2}, i = 1...n\}$

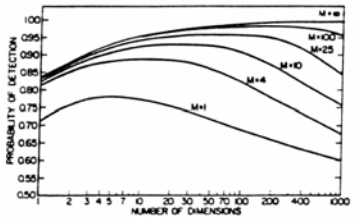
equal prior $P(\omega_1) = P(\omega_2) = 1/2$

- If true parameters are known, high dimensionality helps

$P_e \to 0$ when $n \to \infty$

- If true parameters are unknown, curse of dimensionality

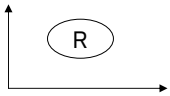
$\lim_{n\to\infty} P_e = 0.5$



- Required training data grows exponentially with dimension.
- Interpoint distances are all large and roughly equal in high dimensions
- Most samples are on the convex hull of the training set.

Nonparametric Techniques

- Assumptions about the underlying distributions may be incorrect.
- General approach: estimate the density directly.



$p(x) \simeq \frac{k/n}{V}$, where k: # points falling in R, V: volume of R

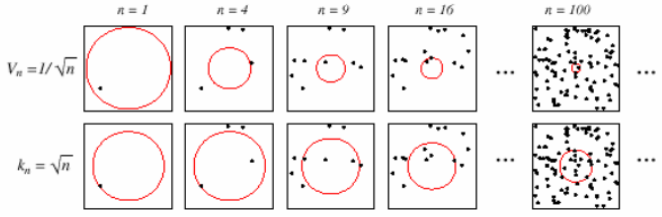
form a sequence of R_n: $p_n(x) \simeq \frac{k_n/n}{V_n}$

For $p_n(x) \to p(x)$, required conditions: $\lim_{n\to\infty} V_n = 0$; $\lim_{n\to\infty} k_n/n = 0$ $\lim_{n\to\infty} k_n = \infty$

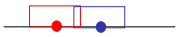
- Two approaches:

1: control and shrink the volumn V_n, e.g., $1/\sqrt{n} \to$ Parzen window

2: control k_n, e.g., $\sqrt{n} \to k_n$ nearest-neighbor method



Parzen Window



Let R_n be a d-dimensional hpercube: $V_n = (h_n)^d$

starts with the unit window function $\varphi(u)$:

points falling in R_n

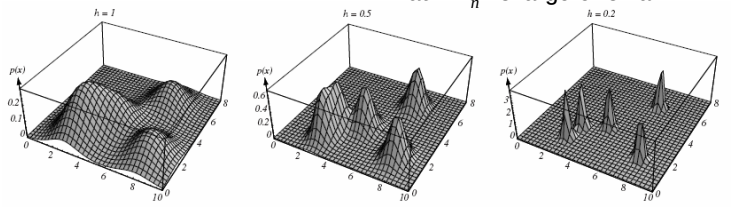
$$\varphi(u) = \begin{cases} 1 & |u_j| \le \frac{1}{2} \quad j = 1,\dots,d \\ 0 & \text{otherwise} \end{cases}$$

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

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

Superposition of n estimate functions

What if h_n is large or small?



- Check conditions on $\varphi(.)$and V_n for $p_n(x)$ to converge to p(x)

Convergence Property

mean

$$E[p_n(x)] = \int \frac{1}{V_n}\varphi\left(\frac{x - v}{h_n}\right) p(v)dv = \int \delta_n(x - v)\,p(v)dv$$

convolution of $\delta(.)$ and $p(.)$

variance

$$\sigma_n^2(x) = \frac{1}{nV_n}\int \frac{1}{V_n}\varphi^2\left(\frac{x - v}{h_n}\right) p(v)dv - \frac{1}{n}\bar{p}_n^2(x) \le \frac{\sup(\varphi(.))\,\bar{p}_n(x)}{nV_n}$$

- Use large n and small V_n to achieve accurate mean and reduce estimator variance

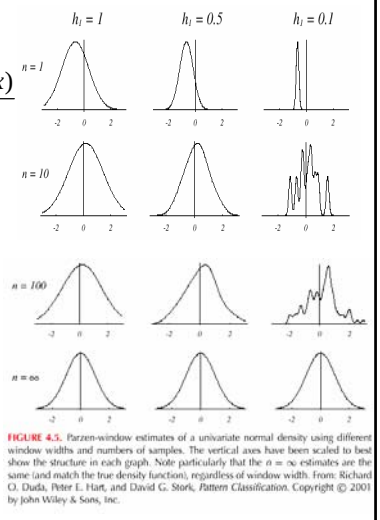
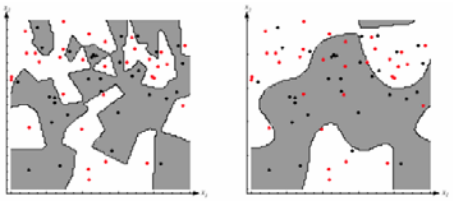


FIGURE 4.3. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- Classification: decision boundaries depend on window function and sample size



Probabilistic Neural Network (PNN)

- Use Gaussian function as window function

$$\varphi\left(\frac{x - x_k}{h_n}\right) \propto \exp(-(x - x_k)^t (x - x_k)/2\sigma^2)$$

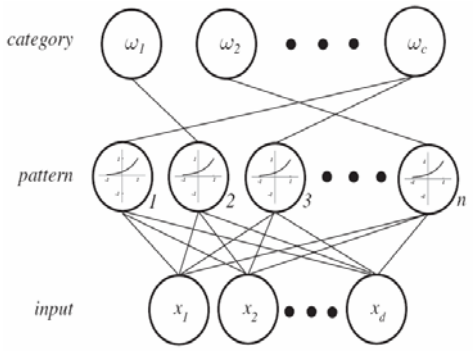
σ controls the window width

$$= \exp(-(\underbrace{x^t x}_{=1} + \underbrace{x_k{}^t x_k}_{=1} - 2x^t x_k)/2\sigma^2)$$

normalize

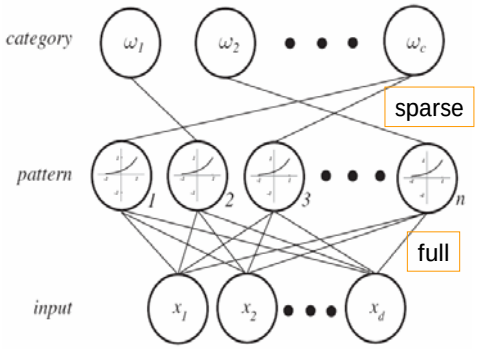
$$= \exp(net_k - 1)/\sigma^2),$$

where $net_k = x^t x_k$



PNN (cont.)

- Training Algorithm
  - Normalize each pattern x of the training set to 1
  - Place the first training pattern on the input units and copy it to the first pattern node
  - Set the weights linking the input units and the first pattern units : $w_1 = x_1$
  - Make a single connection from the first pattern unit to the category unit corresponding to the known class of that pattern
  - Repeat the process for all remaining training patterns by setting the weights such that $w_k = x_k$ $(k = 1, 2, \ldots, n)$



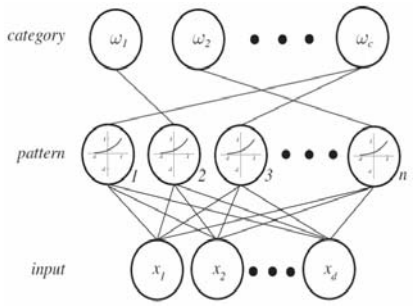
PNN Classification Algorithm

- Normalize test pattern x, place at input layer
- Compute net activation at pattern node k

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

- Sum contributions from pattern units to each output node

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



- Find the class with max posterior

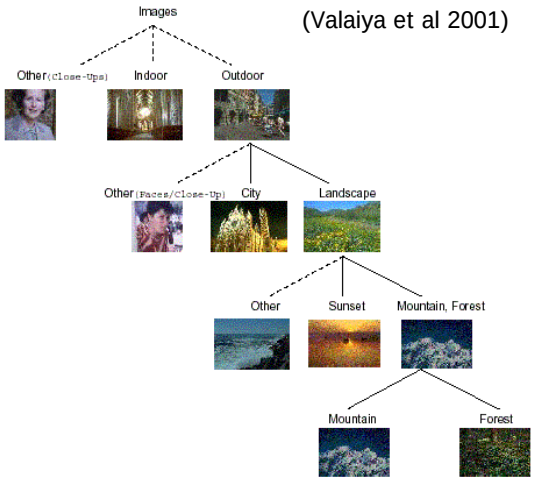
$$\text{classify } x \text{ to } class \leftarrow \arg\max_j P(\omega_j \mid x)$$

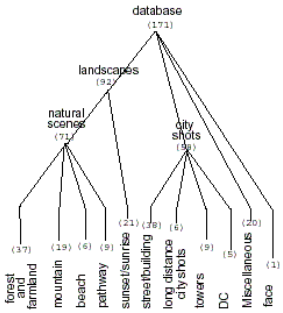
  - Complexity
    - Space $O((n+1)d)$
    - Time $O(nd)$ or $O(1)$
  - Pros and Cons?

Application: Bayesian Image Classification

Exclusive classes –
no rejections nor overlap

(Valaiya et al 2001)





Bayesian Image Classifiers with Parzen Window

- Features: color/edge histogram, coherence histogram

Feature independence $f_{\mathbf{X}}(\boldsymbol{x} \mid \omega) \equiv f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega) = \prod_{i=1}^{M} f_{\mathbf{Y}^{(i)}}(\boldsymbol{y}^{(i)} \mid \omega).$

MAP Classification $\hat{\omega} = \delta(\boldsymbol{x}) = \arg\max_{\omega \in \Omega} \{p(\omega \mid \boldsymbol{y})\} = \arg\max_{\omega \in \Omega} \{f_{\mathbf{Y}}(\boldsymbol{y} \mid \omega)\, p(\omega)\}.$

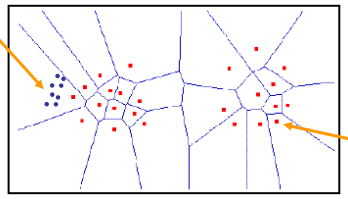
- Probability density estimation through Vector Quantization

$$VQ: \; y \to \hat{y}_i \text{ if } D(y, \hat{y}_i) \le D(y, \hat{y}_j), \forall j \ne i, \; j = 1, \ldots, q$$

$\hat{y}_i$: codebook vectors

- Design codebook by distortion minimization (Euclidean or Mahala. Dist.)

Multiple samples in a cell



Voronoi cells and points from VQ

codeword

VQ density estimation & MDL

Parzen Window: Approximate density with proportion of sample data in each cell

$$f_{\mathbf{Y}^{(i)}}(\boldsymbol{y}^{(i)} \mid \omega) \approx \frac{m_j^{(i)}}{Vol(S_j^{(i)})},$$ Piecewise constant

$$f_{\mathbf{Y}^{(i)}}(\boldsymbol{y}^{(i)} \mid \omega) \propto \sum_{j=1}^{q} m_j^{(i)} * \exp(-\|\boldsymbol{y}^{(i)} - \boldsymbol{v}_j^{(i)}\|^2/2).$$ GMM

- Difference from the standard Parzen Window? Why?
- What happens if codebook size q increases?
  - Likelihood increases; Model size increases (overfitting)
- Consider the total data length for describing the data and model
  - Minimal Description Length (MDL)

MDL optimization principle $\hat{q} = \arg\min \{L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) + L(\boldsymbol{\theta}_{(q)})\}$

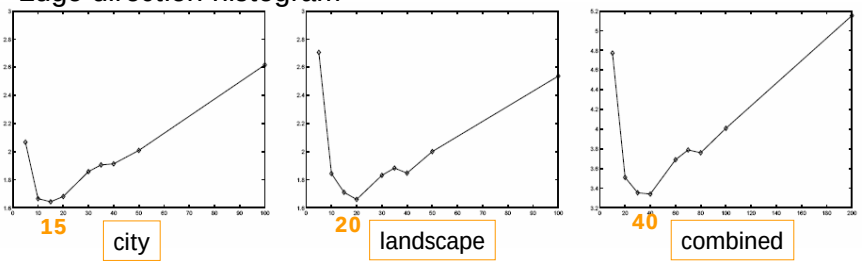
Optimal sample length given model $L(\mathcal{Y}|\boldsymbol{\theta}_{(q)}) = -\sum_{j=1}^{n} \log f(\boldsymbol{y}^{(i)}(j)|\omega, \boldsymbol{\theta}_{(q)})$

Model description length

$$\begin{cases} L(\boldsymbol{\theta}_{(q)}) = (\zeta(q)/2) \log n \\ \zeta(q) = \{q + q \dim(\boldsymbol{y}^{(i)})\} \end{cases}$$

Experiment: Optimal model size

- Edge direction histogram

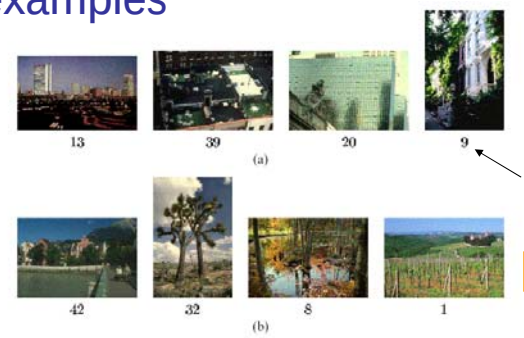


- Why the optimal model size increases when combining data sets?
- Edge information important for detecting “city” images

Test Data	EDH	EDCV	CH	CCV	EDH & CH	EDH & CCV	EDCV & CH	EDCV & CCV
Training Set	94.7	96.7	83.7	83.5	94.8	95.4	96.4	96.9
Test Set	92.0	92.7	75.4	76.0	92.5	92.8	93.4	93.8
Entire Database	93.4	94.7	79.6	79.8	93.7	94.1	94.9	95.3

- Color information important for discriminating landscape subclasses

Failure examples



city

Confidence score posterior prob.

landscape

- Key features:
  - Reduce the sample size by VQ
  - Reduce memory size and computational cost
  - Estimate local density by GMM Parzen Window
- Possible improvements:
  - Additional classes; non-hierarchical classes
  - Multiple binary classifiers, e.g., “city” vs. “no-city”

Interesting Issue:
Discovering image classes through sorting task (Valaiya et al 98)

- What categories to classify?
- Human subjects to sort images to groups (unconstrained)
- Define pair-wise distance matrix
- Perform hierarchical clustering
- Cut at different levels to obtain salient concepts