EE 6885 Statistical Pattern Recognition

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

Lecture 9 (10/10/05)

Reading
- Nonparametric Estimation
  - DHS Chap. 4.1-4.3
- Nearest Neighbor Estimation, Distance Metrics
  - DHS Chap. 4.4-4.5, 4.6
- Paper: Bayesian Classifier with VQ and Parzen Window

Homework #3, due Oct. 12th 2005

Midterm Exam
- Oct. 24th 2005 Monday 1pm-2:30pm (90mins)
  - Open books/notes, no computer
Review: Nonparametric Techniques

- General approach: estimate the density directly. Form local region sequence $R_n$: $p_n(x) = \frac{k_n}{n V_n}$

$$
p_n(x) \rightarrow p(x) \quad \lim_{n \to \infty} V_n = 0, \quad \lim_{n \to \infty} k_n / n = 0, \quad \lim_{n \to \infty} k_n = \infty
$$

- Parzen Window

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

$$
\mathbb{E}[p_n(x)] = \int \delta_n(x - v) p(v) dv
$$

$$
\sigma_n^2(x) \leq \frac{\sup(\phi(\cdot)) p_n(x)}{n V_n}
$$

- Parallel implementation

$$
\phi\left( \frac{x - x_{i-1}}{h_n} \right) \propto \exp\left(-\frac{1}{2} \left( x - x_{i-1} \right)^2 / \sigma^2 \right) = \exp(x_i^2 / \sigma^2)
$$

- Complexity

space: $O((n + 1)d)$, time: $O(nd)$

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
  
  \[ f(x | \omega) \equiv f_y(y | \omega) = \prod_{i=1}^{M} f_{\omega_i}(y^{(i)} | \omega). \]

- Feature independence
  
  \[ \hat{\omega} = \delta(x) = \arg \max_{\omega \in \Omega} \{ p(\omega | y) \} = \arg \max_{\omega \in \Omega} \{ f_y(y | \omega) p(\omega) \}. \]

- MAP Classification

- Probability density estimation through Vector Quantization
  
  \[ VQ: \ y \rightarrow \hat{y}, \text{if } D(y, \hat{y}_i) \leq D(y, \hat{y}_j), \forall j \neq i, j = 1, \ldots, q \]

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

Multiple samples in a cell

Voronoi cells and points from VQ

\[ \hat{y}_i: \text{codebook vectors} \]

VQ density estimation & MDL

- Parzen Window: Approximate density with proportion of sample data in each cell
  
  \[ f_{\omega}(y^{(i)} | \omega) \approx \frac{m_j^{(i)}}{V_{\omega}(S_j^{(i)})}, \]

  - Piecewise constant

  \[ f_{\omega}(y^{(i)} | \omega) \approx \sum_{j=1}^{q} m_j^{(i)} \log(-||y^{(i)} - \mathbf{v}_j^{(i)}||^2/2). \]

  - GMM

- Difference from the standard Parzen Window?

- 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

- Optimal data length given model

- Model description length

  \[ L(\mathbf{\Theta}^{(q)}) = (\zeta(q)/2) \log n \]

  \[ \zeta(q) = \{ q + q \dim(y^{(i)}) \} \]
Experiment: Optimal model size

- Edge direction histogram

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

<table>
<thead>
<tr>
<th>Test Data</th>
<th>EDH</th>
<th>EDCV</th>
<th>CH</th>
<th>CCV</th>
<th>EDH &amp; CH</th>
<th>EDCV &amp; CH</th>
<th>EDCV &amp; CCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
<td>94.7</td>
<td>96.7</td>
<td>83.3</td>
<td>83.5</td>
<td>94.8</td>
<td>94.4</td>
<td>96.4</td>
</tr>
<tr>
<td>Test Set</td>
<td>92.0</td>
<td>92.7</td>
<td>75.4</td>
<td>76.0</td>
<td>92.5</td>
<td>92.8</td>
<td>93.4</td>
</tr>
<tr>
<td>Entire Database</td>
<td>93.4</td>
<td>94.7</td>
<td>79.6</td>
<td>79.8</td>
<td>93.7</td>
<td>94.1</td>
<td>94.9</td>
</tr>
</tbody>
</table>

- Color information important for discriminating landscape subclasses

Failure examples

- Key features:
  - Reduce the number of estimate functions by VQ
  - Reduce memory size and computational cost
  - Estimate local density by GMM Parzen Window

- Possible improvements:
  - Did not use cross validation to assess performance and choose q
  - Add classes; non-hierarchical classes
  - Multiple binary classifiers, e.g., “city” vs. “no-city”
Interesting Issue:
Discovering image classes through sorting task (Valaiva et al 98)

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

---

$k_n$-Nearest-Neighbor

$\displaystyle p_n(x) = \frac{k_n}{N} \Rightarrow k_n = \sqrt{N}$

- Necessary and sufficient conditions for
  \[ \lim_{n \to \infty} k_n/n = 0 \quad \lim_{n \to \infty} k_n = \infty \]
- Example:
  - Peak of $p_n(x)$ often are away from sample points.

  For classification, estimate $p(x)$ for each class $\omega_i$

  $\displaystyle p_i(x, \omega_i) = \frac{k_i}{n} \quad p_n(\omega_i | x) = \frac{\sum_{j=1}^k p_n(x, \omega_j)}{k}$

  - Use labels of neighbors to est. posteriors

---
Error Rate of Nearest Neighbor Classifier

- When \( k=1 \), nearest neighbor

\[
P^* \leq \lim_{n \to \infty} P_n(e) \leq P^* \left(2 - \frac{c}{c-1} P^*\right)
\]

where \( c \) : # of classes, \( P^* \) : Bayesian Error Prob.

\[
P^* = \int P^*(e \mid x) p(x) dx
\]

\[
P^*(e \mid x) = 1 - \max_i P(\omega_i \mid x)
\]

- When \( k>1 \), two-category case (\( c=2 \))

Compared to random guess?

Deriving the error bound ...

Assume \( n \) samples : \((x_1, \theta_1), (x_2, \theta_2), \ldots, (x_n, \theta_n)\)

Assume \( x'_n \) is the nearest neighbor to \( x \)

\[
P_n(e \mid x, x'_n) = 1 - \sum_{i=1}^{c} P(\theta = \theta_i, \theta'_n = \omega_i \mid x, x'_n) = 1 - \sum_{i=1}^{c} P(\omega_i \mid x) P(\omega_i \mid x'_n)
\]

\[
\lim_{n \to \infty} P_n(e \mid x) = \lim_{n \to \infty} \int P_n(e \mid x, x'_n) p(x'_n) dx'_n = \lim_{n \to \infty} \int P_n(e \mid x, x'_n) \delta(x'_n - x) dx'_n
\]

\[
= \int \left[1 - \sum_{i=1}^{c} P(\omega_i \mid x) P(\omega_i \mid x'_n)\right] \delta(x'_n - x) dx'_n = 1 - \sum_{i=1}^{c} P^2(\omega_i \mid x)
\]

\[
P = \lim_{n \to \infty} P_n(e \mid x) = \lim_{n \to \infty} \int P_n(e \mid x) p(x) dx = \int [1 - \sum_{i=1}^{c} P^2(\omega_i \mid x)] p(x) dx
\]

maximized when \( P(\omega_i \mid x) \) are equal except the largest

\[
\Rightarrow \quad P^* \leq \lim_{n \to \infty} P_n(e) \leq P^* \left(2 - \frac{c}{c-1} P^*\right)
\]
Distance Metrics

- Nearest neighbor rules need distance metrics
- Required properties of a metric
  1. non-negativity: \( D(a,b) \geq 0 \)
  2. reflexivity: \( D(a,a) = 0 \) iff \( a = b \)
  3. symmetry: \( D(a,b) = D(b,a) \)
  4. triangular inequality: \( D(a,b) + D(b,c) \geq D(c,a) \)
      \( D(a,b) \geq D(c,a) - D(b,c) \)

- Minkowski Metric
  - Euclidean
  - Manhattan
  - \( L_\infty \)

- Tanimono Metric
  - sets of elements
  - Point-point distance not useful

\[ L_k(a,b) = \left( \sum_{i=1}^{d} |a_i - b_i|^k \right)^{1/k} \]

\[ D_{\text{tanimono}}(S_1,S_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}} = \frac{(n_1 - n_{12}) + (n_2 - n_{12})}{n_1 + n_2 - n_{12}} \]

useful in indexing

EE6887-Chang 9-13