Object Detection with Bayesian Networks

SS 2008 – Bayesian Networks

Multimedia Computing, Universität Augsburg
Rainer.Lienhart@informatik.uni-augsburg.de
www.multimedia-computing.{de,org}
Overview

- **Object Detection** (Object Class Recognition) vs. **Object Recognition**

- Basic Techniques
  - Template matching
  - Multi-scale search

- Bayesian approach
Many Uses

- User Interfaces
- Interactive Agents
- Security Systems
- Video Compression
- Image Database Analysis
Why Face Detection is Difficult?

- **Pose**: Variations are due to the relative camera-face pose (frontal, 45 degree, profile, upside-down), and some facial features such as an eye or the nose may become partially or wholly occluded.

- **Presence/absence of structural components**: Facial features such as beards, mustaches, and glasses present/absent & great variability in shape, color, and size.

- **Facial Expression**:

- **Occlusion**: Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.

- **Image orientation**: Face images directly vary for different rotations about the camera’s optical axis.

- **Image Conditions** (during image formation): Lighting (spectra, spatial distribution, intensity) & camera characteristics (sensor response, lenses, aspect ratio)
Template Matching

\[ E = \| \| - \| \|^2 \]
Multi-Scale Search (MSS)

- Search at multiple scales (and poses)

- Approaches:
  - Multiple templates (of different sizes for various scales); comparison at original scale of image
  - Single templates (at reference size); comparison at multiple scales (by using an image pyramid)
  - Easy scalable template (to create various scales); comparison at original scale of image

- Note: “Template” can be replaced by features derived from a template

- Image Pyramid
  - Decimate image by constant factor
  - Efficient search
Complexity Analysis

Given:
• Image size $W \times H$ (320x240, 720x480)
• Window size $w \times h$ (16x16 to 64x64)
• Scale factor $f$ (1/1.1, 1/1.2, 1/sqrt(2))

Approx. Calculation:

# of location $= W \cdot H \cdot \frac{f^{n+1} - 1}{f - 1}$

with

$n = \min\left(\left\lfloor \log \frac{w}{W} \right\rfloor, \left\lfloor \log \frac{h}{H} \right\rfloor \right)$

Example today
• $W \times H = 320 \times 240$, $w \times h = 24 \times 24$, $f = 1/1.2$

$\Rightarrow n = 12$

$\Rightarrow \# = 5.44 \times W \times H$

$= 417,792$

Example tomorrow
• $W \times H = 720 \times 480$, $w \times h = 24 \times 24$, $f = 1/1.1$

$\Rightarrow n = 31$

$\Rightarrow \# = 10.48 \times W \times H$

$= 3,621,888$
Generic Approach to Detection

Given:

- Meta-search algorithm for a
  - fixed scale
  - fixed position

Goal here:

Create a Bayesian object detector
Brute-Force

\[ P(x_{1,1}, x_{1,2}, \ldots, x_{14,14}) \]

<table>
<thead>
<tr>
<th>(1,1)</th>
<th>(1,2)</th>
<th>...</th>
<th>(14,14)</th>
<th>classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>Non-object</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>Non-object</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>35</td>
<td>45</td>
<td>...</td>
<td>28</td>
<td>Object</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>255</td>
<td>255</td>
<td>...</td>
<td>255</td>
<td>Non-object</td>
</tr>
</tbody>
</table>

Infeasible because \[256^{196} = 2^{1568}\] entries
Classification Model

Random variables: $X=(X_{11}, X_{12}, \ldots, X_{HxW})$

Instance of $X$: $x=(x_{11}, x_{12}, \ldots, x_{HxW})$

Class Label (Object/Non-Object): $Y=\{1,0\}$

A priori knowledge:

$P_0(X) := P( X \mid y=0)$

$P_1(X) := P( X \mid y=1)$

Prior:

$P(Y)$

Most probable hypothesis:

$y_{MAP} = \arg \max_{y \in Y} P(y \mid x)$

$= \arg \max_{y \in Y} \frac{P(x \mid y)P(y)}{P(x)}$

$= \arg \max_{y \in Y} P_y(x)P(y)$

Classification Function

$\Rightarrow f(x) = \begin{cases} 1 & \text{if } \frac{P_1(x)P(y=1)}{P_0(x)P(y=0)} > 1 \\ 0 & \text{otherwise} \end{cases}$

$\Rightarrow f(x) = \begin{cases} 1 & \text{if } \frac{P_1(x)P(y=0)}{P_0(x)P(y=1)} = \lambda' \\ 0 & \text{otherwise} \end{cases}$

$\Rightarrow f(x) = \begin{cases} 1 & \text{if } \log \frac{P_1(x)}{P_0(x)} > \lambda = \log \lambda' \\ 0 & \text{otherwise} \end{cases}$
Required joint probability distribution
\[ P(X|y) = P_y(X) = (X_{11}, X_{12}, \ldots, X_{HxW}|y) \]

→ BN efficient tool to model joint probability distributions

→ Approach

\[ P_y(x) = \prod_{i=\{(1,1),\ldots,(H,W)\}} P_y(X_i = x_i | p_{a_i}) \]

→ Find network structure maximizing the Kullback-Leibler divergence
Kullback-Leibler Divergence

From Wikipedia, the free encyclopedia.

:= relative entropy := quantity measuring difference between two probability distributions

→ Not a metric (violates triangle inequality) & not symmetric

The KL divergence between two probability distributions \( p \) and \( q \) is defined as

\[
KL(p, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}
\]

for distributions of a discrete/continuous variable:

\[
KL(p, q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, dx
\]

It can be seen from the definition that

\[
KL(p, q) = - \sum_x p(x) \log q(x) + \sum_x p(x) \log p(x) = H(p, q) - H(p)
\]

denoting by \( H(p, q) \) the cross entropy of \( p \) and \( q \), and by \( H(p) \) the entropy of \( p \). As the cross-entropy is always greater than or equal to the entropy, this shows that the Kullback-Leibler divergence is nonnegative, and furthermore \( KL(p, q) \) is zero iff \( p = q \).

In Bayesian statistics the KL divergence can be used as a measure of the "distance" between the prior distribution and the posterior distribution.
Possible Solution

\[ KL(P_0(X), P_1(X)) = \sum_x P_0(x) \log \frac{P_0(x)}{P_1(x)} \]
\[ = \sum_x P_0(x) \log \left( \frac{\prod_i P_0(x_i | pa_i)}{\prod_i P_1(x_i | pa_i)} \right) \]
\[ = \sum_x P_0(x) \log \frac{\prod_i P_0(x_i | pa_i)}{\prod_i P_1(x_i | pa_i)} = \sum_x \sum_i P_0(x) \log \frac{P_0(x_i | pa_i)}{P_1(x_i | pa_i)} \]
\[ = \sum_i \sum_{x_i, pa_i} P_0(x_i, pa_i) \log \frac{P_0(x_i | pa_i)}{P_1(x_i | pa_i)} \]

Why not \( KL( P_1(X), P_0(X) ) \)?

Maximization is equivalent to maximum branching problem according to

“T. V. Pham, M. Worring, A. Smeulders. “Face detection by aggregated Bayesian Network Classifiers”. (http://staff.science.uva.nl/~vietp/)
\[ f(x) = \begin{cases} 
1 & \text{if } \log \frac{P_1(x)}{P_0(x)} > \lambda \\
0 & \text{otherwise} 
\end{cases} \]

\[ = \begin{cases} 
1 & \text{if } \log \frac{P_1(x_i | pa_i)}{P_0(x_i | pa_i)} > \lambda \\
0 & \text{otherwise} 
\end{cases} \]

\[ = \begin{cases} 
1 & \text{if } \sum_i \log \frac{P_1(x_i | pa_i)}{P_0(x_i | pa_i)} > \lambda \\
0 & \text{otherwise} 
\end{cases} \]