Unsupervised Learning of Probabilistic Grammar-Markov Models (PGMM) for Object Categories

Long Zhu, Yuanhau Chen, Alan Yuille

Stephen W. Thomas

April 7, 2009

Department of Computer Science
University of Arizona
Overview
Motivation

- **Overall goal**: *unsupervised* learning of probabilistic models for generating and parsing images

- **Idea**: build on PCFG (thanks Federico!) and MRF

- Want to bridge gap between ML natural language grammars and computer vision
  - But, images are far more complex than sentences;
  - have cluttered background and unknown object pose and unknown aspect;
  - and input of images is far more complex.
Background

- **Probabilistic Context Free Grammars (PCFG)**
  - **Idea**: Add probabilities to the rules of a CFG. Can determine:
    - likelihood of a sentence given a grammar
    - most likely parse of a sentence
  - **Good things**: Can recursively support arbitrary number of leaf nodes, since it’s a tree
  - **Bad things**: Assumes subtrees are independent—hard to model spatial relationships

- **Markov Random Fields (MRF) (a.k.a. Markov Networks)**
  - **Idea**: A graphical model with undirected edges and nodes that have the Markov property (conditional independence)
  - **Good things**: Can depend on other parts of the model, and thus can model spatial relationships
  - **Bad things**: It’s a “rigid” graph structure, so it’s hard to model a variable number of features
Problem Statement

*INPUT*

Set of images

- Unknown pose
- Unknown aspect
- Object may or may not be present

*OUTPUT*

Probabilistic Model for Object Categories
Approach

• **Step 1.** Create *Attribute Features (AF)* to represent images

**Process to get the features:**

1. Apply Kadir-Brady saliency detector to select circular regions
2. Apply SIFT operator to obtain Lowe’s feature descriptor
3. Perform PCA on the appearance attributes to obtain 15-dimensional subspace
• **Step 2.** Represent images by a triplet of AF’s called *oriented triplets*

**Benefits of triplets:**

1. Invariant to scale and orientation
2. Lends itself well to dynamic programming solutions
3. Good for *structure pursuit*, since we can combine two trees to make another
Approach

Fig. 5. This figure illustrates the features and triplets without orientation (left two panels) and oriented triplets (next two panels).

Why do they need to be oriented?
Approach

- Learning task is broken into two phases
  - Learning the structure of the model (harder)
  - Learning the parameters of the model
    - Grammatical parameters, spatial parameters
Definition of PGMM

- A graph $G = (V, E)$
  - $V$ contains three types of nodes: OR, AND, and LEAF

- Correspond to different aspects of the object
- Correspond to clique
- Correspond to AF’s

Background node
Set of object leaves
A PGMM specifies the probability distribution of the AF's observed in an image in terms of a parse graph $y$ and model parameters $\Omega, \omega$ for the grammar and MRF respectively.
Model Description

- The observed image features are thus:

\[ x = \{(z_a, A_a, \theta_a) : \text{ s.t. } u_a = 1\} \]

- Joint distribution over these features is computed by

\[
P(x, y, \omega, \Omega) = \sum_{(z_a, A_a, \theta_a) \text{ s.t. } u_a=0} P(u, z, A, \theta, y, \omega, \Omega)
\]
Explanation of Model Details

- Generating leaf nodes
- Generating the observable leaf nodes
- Generating the positions and orientations of leaf nodes
- The appearance distribution
- The correspondence problem
Generating Leaf Nodes

- Specifies how many AF’s are present in the images
- Output of $y$ is the set of numbered leaf nodes
- Specified by a set of production rules:

\[
\begin{align*}
S & \rightarrow \{B, O\} \text{ with prob } 1, \\
O & \rightarrow \{O_i : i = 1, \ldots, \rho\} \text{ with prob } \Omega_i^R, \ i = 1, \ldots, \rho \\
O_i & \rightarrow \{n_a, N_{a+1,a+2}\} \text{ with prob. } 1, \ a = \beta_i^\Omega, \\
N_{a,a+1} & \rightarrow \{n_a, N_{a+1,a+2}\} \text{ with prob. } 1, \ \beta_i^\Omega + 1 \leq a \leq \beta_{i+1}^\Omega - 4. \\
N_{\beta_i^\Omega-3,\beta_i^\Omega-2} & \rightarrow \{n_{\beta_i^\Omega-3}, n_{\beta_i^\Omega-2}, n_{\beta_i^\Omega-1}\} \text{ with prob } 1, \\
B & \rightarrow \{n_{\beta_{\rho+1}}, \ldots, n_{\beta_{\rho+1}+m}\} \text{ with prob } \Omega^B e^{-m\Omega_B} (m = 0, 1, 2\ldots).
\end{align*}
\]
Generating the Observable Leaf Nodes

- Specifies whether objects are observable in the image
  - Occlusion or low detector response

- Observability of nodes is independent:

\[
P(u|y, \omega^g) = \prod_{a \in L_O(y)} \lambda_a^{u_a} (1 - \lambda_a)^{(1-u_a)} = \exp \left\{ \sum_{a \in L_O(y)} \left\{ \delta_{u_a,1} \log \lambda_a + \delta_{u_a,0} \log(1 - \lambda_a) \right\} \right\}
\]

Parameter of Bernoulli distribution
Delta Function
Generating Position, Orientations of Leaf Nodes

- Distribution of the spatial positions $z$ and orientations $\Theta$ of leaf nodes
- Distribution is required (by authors) to satisfy two properties
  - Invariant to pose
  - Easily computable
- Approximate by:

$$P(z, \theta|y, \omega^g) = K \times P(l(z, \theta)|y, \omega^g)$$

$$P(l|y, \omega^g) = \frac{1}{Z} \exp \left\{ \sum_{a \in \text{Cliques}(y)} \psi_a(l(z_a, \theta_a, z_{a+1}, \theta_{a+1}, z_{a+2}, \theta_{a+2}), \omega^g_a) \right\}$$
The Appearance Distribution

- Appearances of background nodes are generated from a uniform distribution.
- Appearances of object nodes are generated by:

\[
P(A_a | \omega^A_a) = \frac{1}{\sqrt{2\pi} |\Sigma_{A,a}|} \exp\left\{-\frac{1}{2}(A_a - \mu^A_a)^T(\Sigma^A_a)^{-1}(A_a - \mu^A_a)\right\}
\]

\[\omega^A_a = (\mu^A_a, \Sigma^A_a)\]

Normal Distribution with parameters \((\mu^A_a, \Sigma^A_a)\)
The Priors

\[ P(\Omega), P(\omega^A), P(\omega^g) \]

- All set to be uniform distributions
Learning and Inference

- **Inference.** Estimate parse tree $y$ from input $x$
  - Parameters are fixed. Solve:

\[
(y^*, V^*) = \underset{y, V}{\text{arg max}} \ P(y, V|x, \omega, \Omega) \\
= \underset{y, V}{\text{arg max}} \ P(x, \omega, \Omega, y, V)
\]
Learning and Inference

- **Parameter Learning.** Model is known, but we need to estimate parameters
  - Estimate by MAP using EM:

\[
(\omega^*, \Omega^*) = \arg \max_{\omega, \Omega \in W} P(\omega, \Omega | x) \propto P(x | \omega, \Omega) P(\omega, \Omega)
\]

\[
= \arg \max_{\omega, \Omega \in W} P(\omega, \Omega) \prod_{\tau \in \Lambda} \sum_{y_{\tau}, V_{\tau}} P(y_{\tau}, V_{\tau}, x_{\tau} | \omega, \Omega).
\]

- **Structure Learning.** Learning the model structure.
  - Strategy is to grow the tree structure of PGMM by adding new aspect nodes or adding new cliques to existing aspect nodes
  - After adding something new, compute the score:

\[
\text{score} = \max_{\omega, \Omega} P(\omega, \Omega) \prod_{\tau \in \Lambda} \sum_{y_{\tau}} \sum_{V_{\tau}} P(x_{\tau}, y_{\tau}, V_{\tau} | \omega, \Omega)
\]
Dynamic Programming

- Plays a key role in all three phases of learning in PGMM
- In fact, structure of PGMM was designed so that dynamic programming would be practical
EM for Parameter Learning

- Use EM to estimate the parameters:

\[ P(\omega, \Omega | \{x_\tau\}) = \sum_{\{y_\tau\}, \{V_\tau\}} P(\omega, \Omega, \{y_\tau\}, \{V_\tau\} | \{x_\tau\}) \]

\[ P(\omega, \Omega, \{y_\tau\}, \{V_\tau\} | \{x_\tau\}) = \frac{1}{Z} P(\omega, \Omega) \prod_{\tau \in \Lambda} P(y_\tau, V_\tau | x_\tau, \omega, \Omega) \]

Define a free energy:

\[ F[q(\ldots), \omega, \Omega] = \sum_{\{y_\tau\}, \{V_\tau\}} q(\{y_\tau\}, \{V_\tau\}) \log q(\{y_\tau\}, \{V_\tau\}) \]

\[ - \sum_{\{y_\tau\}, \{V_\tau\}} q(\{y_\tau\}, \{V_\tau\}) \log P(\omega, \Omega, \{y_\tau\}, \{V_\tau\} | \{x_\tau\}) \]
EM for Parameter Learning

**E-Step:**

\[ q^t(y_t, V_t) = P(y_t, V_t | \omega^t, \Omega^t, x_t) \]

Gives the distribution of the aspects and correspondences, given the current estimates of the parameters

**M-Step:**

\[ \omega^{t+1}, \Omega^{t+1} = \arg \min_{\omega, \Omega} \left\{ - \sum_{y_t, V_t} q^t(y_t, V_t) \log P(\omega, \Omega, y_t, V_t | x_t) \right\} \]

Gives parameters given the current distribution of the aspects and correspondences
Structure Pursuit

• Adding new triplets to the PGMM
  – Use clustering algorithms to define a triplet vocabulary
  – Triplet vocabulary is used to propose ways to grow PGMM
  – Evaluated by how well PGMM fits data
Experimental Results

- Inference algorithm is fast (under 5 seconds)
- Can learn and infer when pose varies
- Works on training datasets with both images with object in it, and without object in it
Experimental Results
Experimental Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Localization Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faces</td>
<td>96.3</td>
</tr>
<tr>
<td>Motorbikes</td>
<td>98.6</td>
</tr>
<tr>
<td>Airplanes</td>
<td>91.5</td>
</tr>
</tbody>
</table>

Stephen W. Thomas  
Unsupervised Learning of PGMMs for Object Learning
Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale Normalized</td>
<td>97.8</td>
</tr>
<tr>
<td>Rotation Only</td>
<td>96.3</td>
</tr>
<tr>
<td>Rotation + Scale</td>
<td>96.3</td>
</tr>
</tbody>
</table>
Backups/Reference
### TABLE I

**THE NOTATIONS USED FOR THE PGMM.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Lambda$</td>
<td>the set of images</td>
</tr>
<tr>
<td>$x_i = (z_i, \theta_i, A_i)$</td>
<td>an attributed feature (AF)</td>
</tr>
<tr>
<td>${x_i : i = 1, \ldots, N_\tau}$</td>
<td>attributed features of image $I_\tau$</td>
</tr>
<tr>
<td>$N_\tau$</td>
<td>the number of features in image $I_\tau$</td>
</tr>
<tr>
<td>$z_i$</td>
<td>the location of the feature</td>
</tr>
<tr>
<td>$\theta_i$</td>
<td>the orientation of the feature</td>
</tr>
<tr>
<td>$A_i$</td>
<td>the appearance vector of the feature</td>
</tr>
<tr>
<td>$y$</td>
<td>the topological structure</td>
</tr>
<tr>
<td>$a$</td>
<td>the index</td>
</tr>
<tr>
<td>$n_a$</td>
<td>the leaf nodes of PGMM</td>
</tr>
<tr>
<td>$C_a = {n_a, n_{a+1}, n_{a+2}}$</td>
<td>a triplet clique</td>
</tr>
<tr>
<td>$\overline{I}_C()$</td>
<td>the invariance triplet vector of clique $C$</td>
</tr>
<tr>
<td>$u = {u_a}$</td>
<td>observability variables</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>the parameters of grammatical part</td>
</tr>
<tr>
<td>$\omega$</td>
<td>$(\omega^p, \omega^A)$</td>
</tr>
<tr>
<td>$\omega^p$</td>
<td>the parameters of spatial relation of leaf nodes</td>
</tr>
<tr>
<td>$\omega^A$</td>
<td>the parameters of appearances of the AF's</td>
</tr>
<tr>
<td>$V = {v(a)}$</td>
<td>the correspondence variables</td>
</tr>
</tbody>
</table>