Recursive Compositional Models: Modeling, Inference and Learning.

Alan Yuille
Stat. Dept. of UCLA
Long (Leo) Zhu: PhD Thesis
Talk Plan

• (I) Discuss Standard Models for Deformable Objects.
• Representation/Inference/Learning.
• (II) Describe their limitations and motivate:
• (III) Recursive Compositional Models.
Task: Detect Deformable Objects.

- Example: Detecting Hands by Deformable Templates (Coughlan et al.).
Deformable Template

- A Deformable Template can be formulated as a Bayesian model. 
  \[ P(D|W)P(W) \] – \( W \) is configuration of template 
  \( D \) is the data (image). 
  Prior \( P(W) \) – probable geometry 
  Imaging Model \( P(D|W) \).

Inference: estimate most probable \( W^* \).
Prior is Markov Random Field

- \( q \) denotes position and orientation of point.
- \( h \) is an occlusion process.
- \( s \) is both variables.

\[
P(q_1 \cdots q_N) = P(q_1) \prod_{i=1}^{N-1} P(q_{i+1}|q_i)
\]

\[
P(h_1 \cdots h_N) = P(h_1) \prod_{i=1}^{N-1} P(h_{i+1}|h_i)
\]

\[
P(s_1, s_2, \cdots s_N) = P(q_1 \cdots q_N)P(h_1 \cdots h_N),
\]
Imaging Term:

- Extract Features (edges and corners) from the image.

\[ D(x) = (I_e(x), C(x), \theta_I(x)). \]

\[ P(D|s_1 \cdots s_N) = \prod_{\text{all pixels } y} P(D(y)|s_1 \cdots s_N) \]

where \( P(D(y)|s_1 \cdots s_N) \) is set to \( P_{on}(D(y)|\theta_1) \), \( P_{triple}(D(y)) \) or \( P_{off}(D(y)) \).

\[ P_{off}(D(y)) = P_{off}^{e}(I_e(y))P_{off}^{c}(C(y))P_{off}^{a}(\theta_i). \]

\[ P_{triple}(D(y)) = P_{triple}^{e}(I_e(y))P_{triple}^{c}(C(y))P_{triple}^{a}(\theta_i), \]

\[ P_{on}(D(y)|\theta_i) = P_{on}^{e}(I_e(y))P_{on}^{c}(C(y))P_{on}^{a}(\theta_i - \theta_I(y)), \]
Images and Feature Maps

- Examples of edges and corners.
Inference by Dynamic Programming

• Perform the MAP estimate:

\[ MAP = \arg\max_{s_1 \cdots s_N} \left\{ \prod_{i=1}^{N} \frac{P_{on}(s_i)}{P_{off}(s_i)} \right\} P(s_1 \cdots s_N), \]

• Compute by Dynamic Programming (DP) by computing partial paths recursively:

\[ f_i(s_1) = \left\{ \prod_{j=1}^{i} \frac{P_{on}(s_j)}{P_{off}(s_j)} \right\} P(s_1, s_2, \cdots, s_i). \]
Results

• MAP estimates of hand configuration.
Limitations

• These types of models (c.f. pictorial structures) can be effective but have limitations.

• (I) They have only short range spatial interactions (Markov) -- limited modeling.

• (II) Dynamic Programming proceeds linearly from thumb to fifth finger -- unintuitive.

• But if we add dense spatial interactions then: (a) what algorithms can perform
Long (Leo) Zhu’s PhD Thesis

- More like a research program than a thesis.
- 9 peer reviewed conference publications.
- 1 journal publication.
- 3 journal papers in review.
- 2 journal papers in preparation.
Recursive Compositional Models of Objects (and Images)

• Goals: construct models, which enable learning, and perform efficient inference.

Unified Approach – one model can perform several vision tasks.
General Applicability – applicable to patterns in general (not just images and objects).

Applications to Vision Tasks on Benchmarked Datasets

- **Object Categorization, Segmentation and Recognition**
  - Caltech-101
- **Deformable Object Detection, Segmentation and Parsing**
  - Weizmann Horse
  - Multi-view Face Alignment
- **Articulated Object Parsing**
  - Berkeley Baseball Human Body
- **Image Parsing: scene labeling**
  - Microsoft Research
Recursive Compositional Models

The three ingredients:

(II) Inference – pruned dynamic programming, compositional inference.
(III) Learning of Structure and Parameters.
   Supervised – structure perceptron/max margin.
   Unsupervised – compositional learning.
Modeling/Representation

• Recursive Composition of elementary components. Tree Structure.

• Triplets
Graph Structure (Tree).

- Probability on graphs

The Graph Structure. \( G = (V, E) \) – \( V \) vertices – \( E \) edges.

Parent-child structure: \( ch(\nu) \) children of \( \nu \).

\( V^{LEAF} \) leaf nodes.

\( V_{\nu} \) subtree of descendants of \( \nu \).

Two types of nodes: \textit{AND node} and \textit{OR nodes}.

Variable defined on the graph.

State variable \( z_\mu \). E.g. position, orientation, scale.

\( z_{ch(\mu)} = \{z_\nu : \nu \in ch(\mu)\} \) states of the children \( \mu \).

\( D(z_\mu) \) subregion of image corresponding to \( \mu \).
Probability Distribution

• Exponential Model. Prior and Data Terms.

\[ P(z|\mathbf{I}) = \frac{1}{Z(\alpha, \mathbf{I})} \exp\left\{ - \sum_{\mu \in V} \tilde{\alpha}_\mu \cdot \tilde{\phi}(z_\mu, z_{ch(\mu)}) - \sum_{\mu \in V} \tilde{\alpha}_\mu^D \cdot \tilde{\phi}_\mu^D(\mathbf{I}, D(z_\mu)) \right\}. \]

Prior potentials: \( \tilde{\phi}(z_\mu, \{z_\nu : \nu \in ch(\mu)\}) \)

Data potentials: \( \tilde{\phi}_D(\mathbf{I}, D(z_\mu)), \text{ for all } \mu \in V. \)

Potentials (sufficient statistics) have coefficients \( \tilde{\alpha}_\mu, \tilde{\alpha}_\mu^D \) (to be learnt).
Probabilistic Formulation

• Example of the probabilities.

\[ y_v = (P_x, P_y, \theta, s)_v \]
AND/OR Graph Model

• A novel AND/OR graph is proposed to model enormous articulated poses.
• Learning is performed in a supervised manner.
• Applications: Human Body Parsing
AND/OR Graph for Human Body
Advantages of Recursive Compositional Models

• (I) Enables modeling of geometric regularities and appearance cues at different scales.

• (II) Rich representation – enables different visual tasks to be performed by same model.

• (III) Enables effective inference and learning algorithms.

• NOTE: Summing out all nodes except the LEAVES gives a dense flat model
Inference

• Estimate most probable configuration:

\[ z^* = \arg \min_z E(z|I) \]

\[ E(z|I) = \sum_{\mu \in V} \bar{\alpha}_\mu \cdot \bar{\phi}(z_\mu, z_{ch(\mu)}) + \sum_{\mu \in V} \bar{\alpha}_\mu^D \cdot \bar{\phi}_D(I, D(z_\mu)). \]

• Energy can be computed recursively for subtrees:

\[ E_v(z_{des(\nu)}|I) = \sum_{\mu \in V_v} \bar{\alpha}_\mu \cdot \bar{\phi}(z_\mu, z_{ch(\nu)}) + \sum_{\mu \in V_v} \bar{\alpha}_\mu^D \cdot \bar{\phi}_D(I, D(z_\mu)). \]

\[ E_v(z_{des(\nu)}|I) = \sum_{\rho \in ch(\nu)} E_{\rho}(z_{des(\rho)}|I) + \bar{\alpha}_\nu \cdot \bar{\phi}(z_\nu, z_{ch(\nu)}) + \bar{\alpha}_\nu^D \cdot \bar{\phi}_D(I, D(z_\nu)). \]

• Enables Dynamic Programming (poly time).
Inference.

- **Dynamic Programming** (pruned)
  - (i) keep a list of possible states of child nodes of the tree,
  - (ii) propose states of the parent nodes. Prune by spatial relationships, overlaps, etc.
- Complexity: polynomial in size of image and hierarchy.
PseudoCode

• Pseudocode for Dynamic Programming

Input: \( \{MP_{\nu,1}\} \). Output: \( \{MP_{\nu,L}\} \). \( \oplus \) denotes the operation of combining two proposals.

Loop: \( l = 1 \) to \( L \), for each node \( \nu \) at level \( l \)

• IF \( \nu \) is an OR node

1. Union: \( MP_{\nu,b} = \bigcup_{\rho \in T_{\nu,a}=1,...,MP_{\rho,a}^{-1}} MP_{\rho,a}^{-1} \)

• IF \( \nu \) is an AND node

1. Composition: \( P_{\nu,b} = \bigoplus_{\rho \in T_{\nu,a}=1,...,MP_{\rho,a}^{-1}} MP_{\rho,a}^{-1} \)

2. Pruning: \( P_{\nu,a} = \{P_{\nu,a}|E(\Lambda_{\nu,a}) > K_l\} \)

3. Local Maximum: \( (MP_{\nu,a}, CL_{\nu,a}) \) = LocalMaximum(\( \{P_{\nu,a}\}, \epsilon_W \))

where \( \epsilon_W \) is the size of the window \( W_{\nu}^l \) defined in space, orientation, and scale.
Learning

• Different Types of Learning depending on amount of information available.
• Two Cases:
  • (i) Object boundary labeled. (Supervised).
  • (II) Object is somewhere in image (Weak Supervision).
  • (III) One of several objects may be in the image (Unsupervised).
Supervised Learning

• Boundary specified on training data.

• Two algorithms used for learning:
  • (i) Structure-Perceptron.
  • (ii) Structure max-margin.
• Both are discriminative learning (not MLE) for computational reasons.

  Both require an inference algorithm (pruned DP).
Structure-Perceptron

- Generalization of standard perceptron
- (binary outcome).
  (simple, but often effective).

**Input:** A set of training images with ground truth \((x^i, y^i)\) for \(i = 1..N\). Initialize parameter vector \(\alpha = 0\). For \(t = 1..T, i = 1..N\)

- find the best state of the model on the \(i\)'th training image with current parameter setting, i.e., \(y^* = \arg\max_y \psi(x^i, y) \cdot \alpha\)
- Update the parameters: \(\alpha = \alpha + \psi(x^i, y^i) - \psi(x^i, y^*)\)
- Store: \(\alpha^{t,i} = \alpha\)

**Output:** Parameters \(\gamma = \sum_{t,i} \alpha^{t,i}/NT\)
Structure Max-Margin

- Generalization of Support-Vector Machine

Training example $i$ ground-truth $z^T(i)$.
Error measure $\Delta(z^T_\mu(i), z_\mu(i)) = 1$
if $\text{dist}(z^T_\mu(i), z_\mu(i)) \geq \sigma$ and $\Delta(\cdot, \cdot) = 0$ otherwise.

Full error measure:
$L(z^T(i), z(i)) = \sum_{\nu \in \text{VAND}} \Delta(z^T_\mu(i), z_\mu(i)) + \sum_{\nu \in \text{VLEAF}} \Delta(z^T_\mu(i), z_\mu(i))$.

Find weights $\{\alpha, \alpha^D\}$ to minimize the criterion:
$\frac{1}{2}|\alpha|^2 + C \sum_i \zeta_i$

s.t. $\{\vec{\alpha}_\mu \cdot \phi(z^T) + \vec{\alpha}^D \cdot (\vec{\phi}(I, z^T(i)))\} - \{\vec{\alpha}_\mu \cdot \phi(z) + \vec{\alpha}^D \cdot (\vec{\phi}(I, z(i)))\} \geq L(z^T(i), z(i)) - \zeta_i, \forall i, z.$
Unsupervised Structure Learning

• Procedure: Bottom-Up and Top-Down

• Three principles:
  – Recursive Composition: combine elementary structures (danger combinatorial explosion)
  – Suspicious Coincidence
  – Competitive Exclusion

• Complexity: linear in the height of a hierarchy (empirically)
10 images for training
Bottom-Up Learning

Repeat from low levels to high levels
1. **Composition**: combine instances from level L
2. **Clustering**: compose concepts at level L+1
3. **Parsing**: get responses of concepts
4. **Suspicious Coincidence**: prune out non-frequent concepts
5. **Competitive Exclusion**: prune out the similar concepts

Until no new compositions are formed (The number of layers is automatically decided by the algorithm)
From Generic Features to Object Structures

- Unified descriptors
- Unified learning: bridge the gap between the generic features and specific object structures

<table>
<thead>
<tr>
<th>Level 0</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Level 0" /></td>
<td><img src="image" alt="Level 1" /></td>
<td><img src="image" alt="Level 2" /></td>
<td><img src="image" alt="Level 3" /></td>
<td><img src="image" alt="Level 4" /></td>
</tr>
</tbody>
</table>
Top-Down Learning

• Fill in the missing parts caused by 1) competitive exclusion 2) suspicious coincidence.

• Exar
Results
Multi-Level Computational Complexity

• Explore a huge number of candidate

<table>
<thead>
<tr>
<th>L</th>
<th>Composit.</th>
<th>Clusters</th>
<th>Prune</th>
<th>Com. Exe.</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>167431</td>
<td>14684</td>
<td>262</td>
<td>4</td>
<td>1s</td>
</tr>
<tr>
<td>2</td>
<td>2034851</td>
<td>741662</td>
<td>995</td>
<td>48</td>
<td>117s</td>
</tr>
<tr>
<td>3</td>
<td>2135467</td>
<td>1012777</td>
<td>305</td>
<td>116</td>
<td>254s</td>
</tr>
<tr>
<td>4</td>
<td>236955</td>
<td>72620</td>
<td>30</td>
<td>2</td>
<td>99s</td>
</tr>
</tbody>
</table>

![Graph showing the relationship between level and cluster size](image1)

![Graph showing the relationship between level and concept instances](image2)
Segmentation and Parsing

• Segmentation (Accuracy)
  – the proportion of the correct pixel labels (object or non-object)

• Parsing (Average Position Error)
  – the average distance between the positions of leaf nodes of the ground truth and those estimated in the parse tree

<table>
<thead>
<tr>
<th>Methods</th>
<th>Testing</th>
<th>Segmentation</th>
<th>Parsing</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchical</td>
<td>228</td>
<td>94.7</td>
<td>16</td>
<td>23s</td>
</tr>
<tr>
<td>Ren (Berkeley)</td>
<td>172</td>
<td>91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winn (LOCUS)</td>
<td>200</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Levin and Weiss</td>
<td>N/A</td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kumar (OBJ CUT)</td>
<td>5</td>
<td>96</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

38
Multi-view Face Alignment

• Show the versatility of our framework
• Alignment: AAM (5.7) vs. Our method (6.0)
Human Body Parsing
Image Model

- **Input Image:** \( x \)
- **State:** (Tree structure)
- **Hierarchical Log-Linear Form**

\[
P(y \mid x; \alpha) = \frac{1}{Z(x; \alpha)} \exp \left\{ - \sum_{i=1..6} E_i(x, y) \right\}
\]

\[
E(x, y) = -\psi(x, y) \cdot \alpha
\]

\[
y = (s, c) = (\text{template, object})
\]

- \( \psi_1(x, s, c) = \log p(o \mid x) \)
- \( \psi_2(x, s, c) \)
- \( \psi_3(s, c) = \sum \delta(o^r_a, o^r_b) \)
- \( \psi_4(c_a, c_b) \)
- \( \psi_5(s) = \log p(s) \)
- \( \psi_6(s, c) = \log p(s, c) \)
Image Parsing: Segmentation-Recognition Template

- **S-R** pair: chicken-and-egg of Seg. and Rec.
- **Local:** *Shape* and *Appearance* of object part
- **Global:** *Gist* of scene (Spatial layout);
- **Multi-level** coarse-to-fine
Image Parsing

• Task: Image Segmentation and Scene Labeling

[Images of scenes with labeled objects]
Comparisons

• Implementation Details

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Classe</th>
<th>Siz</th>
<th>Training</th>
<th>Training</th>
<th>Testing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC</td>
<td>21</td>
<td>591</td>
<td>45%</td>
<td>55h</td>
<td>30s</td>
</tr>
</tbody>
</table>

• Comparisons

<table>
<thead>
<tr>
<th></th>
<th>TextonBoost Shotton et al.</th>
<th>PLSA-MRF Berbeek and Trigg</th>
<th>Mspatch Yang et al. CMU</th>
<th>Classifier only</th>
<th>HIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>57.7</td>
<td>64</td>
<td>61.8</td>
<td>67.2</td>
<td>74.5</td>
</tr>
<tr>
<td>Global</td>
<td>72.2</td>
<td>69 (Classifier)</td>
<td>73.5</td>
<td>75.1</td>
<td>75.9</td>
</tr>
<tr>
<td></td>
<td>72.2</td>
<td>69 (Classifier)</td>
<td>73.5</td>
<td>75.1</td>
<td>81.4</td>
</tr>
</tbody>
</table>
Feasibility of scaling up

• Short-term goal: 100 objects and 1000 images

• CPU and memory costs:
  – 10 images: 5 minutes, 320 Megabytes
  – 20 images: 10 minutes, 550 Megabytes
  – 50 images: 60 minutes, 1900 Megabytes
  – 1000 images: 2 days, 40 gigabytes
  (Prediction)
Summary

• New Class of Models:
  • Recursive Compositional Models.
  • Efficient learning and inference algorithms which exploit the recursive structure.
• Proof of concept on many different vision tasks on benchmarked datasets.
## Comparisons

<table>
<thead>
<tr>
<th>Methods</th>
<th>Testing</th>
<th>Segmentation</th>
<th>Parsing</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised</td>
<td>316</td>
<td>93.3</td>
<td></td>
<td>17s</td>
</tr>
<tr>
<td>Structure-</td>
<td>228</td>
<td>94.7</td>
<td>16</td>
<td>23s</td>
</tr>
<tr>
<td>Ren (Berkeley)</td>
<td>172</td>
<td>91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Winn (LOCUS)</td>
<td>200</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Levin and Weiss</td>
<td>N/A</td>
<td>95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kumar (OBJ CUT)</td>
<td>5</td>
<td>96</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Horse images with keypoints](image-url)