Structured Prediction Cascades

Ben Taskar

David Weiss, Ben Sapp, Alex Toshev

University of Pennsylvania
Supervised Learning

Learn $h : \mathcal{X} \rightarrow \mathcal{Y}$ from $(x^1, y^1) \ldots (x^n, y^n)$

- **Regression:**
  $$\mathcal{Y} = \mathbb{R}$$

- **Binary Classification:**
  $$\mathcal{Y} = \{-, +\}$$

- **Multiclass Classification:**
  $$\mathcal{Y} = \{1, \ldots, k\}$$

- **Structured Prediction:**
  $$\mathcal{Y} = \{1, \ldots, k\}^l$$
Handwriting Recognition

\[ x \quad \rightarrow \quad \text{structured} \quad \rightarrow \quad y \]
Ce n'est pas un autre problème de classification.

‘This is not another classification problem.’
Pose Estimation

\[ x \rightarrow y \]
Structured Models

\[ h(x) = \arg \max_{y \in \mathcal{Y}(x)} \text{score}(x, y) \]

Complexity of inference depends on “part” structure

\[ \text{score}(x, y) = \theta^\top f(x, y) = \sum_p \theta^\top f(x_p, y_p) \]

parts = cliques, productions
Supervised Structured Prediction

Model: \( score(x, y) = \theta^\top f(x, y) \)

Data:
\[
(x^1, y^1), \ldots, (x^n, y^n)
\]

Learning:
- Discriminative estimation of \( \theta \)
- Intractable/impractical for complex models

Prediction:
- \( \arg \max_{y \in Y(x)} \theta^\top f(x, y) \)
- Intractable/impractical for complex models

3rd order OCR model = 1 mil states * length
Berkeley English grammar = 4 mil productions * length^3
Tree-based pose model = 100 mil states * # joints
Approximation vs. Computation

• Usual trade-off: approximation vs. estimation
  – Complex models need more data
  – Error from over-fitting

• New trade-off: approximation vs. computation
  – Complex models need more time/memory
  – Error from inexact inference

• This talk: enable complex models via cascades
An Inspiration: Viola Jones Face Detector

Scanning window at every location, scale and orientation
Classifier Cascade

- Most patches are non-face
- Filter out easy cases fast!!
- Simple features first
- Low precision, high recall
- Learned layer-by-layer
- Next layer more complex
Related Work

• Global Thresholding and Multiple-Pass Parsing. J Goodman, 97
• A maximum-entropy-inspired parser. E. Charniak, 00.
• Coarse-to-fine n-best parsing and MaxEnt discriminative reranking, E Charniak & M Johnson, 05
• TAG, dynamic programming, and the perceptron for efficient, feature-rich parsing, X. Carreras, M. Collins, and T. Koo, 08
• Coarse-to-Fine Natural Language Processing, S. Petrov, 09
• Coarse-to-fine face detection. Fleuret, F., Geman, D, 01
• Robust real-time object detection. Viola, P., Jones, M, 02
• Progressive search space reduction for human pose estimation, Ferrari, V., Marin-Jimenez, M., Zisserman, A, 08
What’s a Structured Cascade?

• What to filter?
  – Clique assignments

• What are the layers?
  – Higher order models
  – Higher resol. models

• How to learn filters?
  – Novel convex loss
  – Simple online algorithm
  – Generalization bounds
Trade-off in Learning Cascades

- **Accuracy**: Minimize the number of errors incurred by each level

- **Efficiency**: Maximize the number of filtered assignments at each level
Max-marginals(Sequences)

Score of an output: \( \theta_x(y) = \theta^\top f(x, y) \)

Max marginal: \( \theta_x^*(y_c) \equiv \max_{y': y'_c = y_c} \theta_x(y) \)
Filtering with Max-marginals

- Set threshold $t$
- Filter clique assignment $y_c$ if $\theta^*_x(y_c) \leq t$

\[
\theta_x(abcc) \leq t
\]
Filtering with Max-marginals

• Set threshold \( t \)
• Filter clique assignment \( y_c \) if \( \theta^*_x(y_c) \leq t \)

\[
\theta_x(abcc) \leq t
\]

Remove edge \( bc \)
Why Max-marginals?

- Valid path guarantee
  - Filtering leaves at least one valid global assignment

- Faster inference
  - No exponentiation, $O(k)$ vs. $O(k \log k)$ in some cases

- Convex estimation
  - Simple stochastic subgradient algorithm

- Generalization bounds for error/efficiency
  - Guarantees on expected trade-off
Choosing a Threshold \( \theta_x^*(y_c) \leq t \)

- Threshold must be specific to input \( x \)
  - Max-marginal scores are not normalized
- Keep top-K max-marginal assignments?
  - Hard(er) to handle and analyze
- Convex alternative: \textbf{max-mean-max function}:

\[
t_x(\alpha) = \alpha \max_{y'} \theta_x(y') + (1 - \alpha) \frac{1}{m} \sum_{y_c} \theta_x^*(y_c)
\]

- \textbf{max score}
- \textbf{mean max marginal}
  - \( m = \# \text{edges} \)
Choosing a Threshold (cont)

\[ t_x(\alpha) = \alpha \max_{y'} \theta_x(y') + (1 - \alpha) \frac{1}{m} \sum_{y_c} \theta_x^*(y_c) \]

- **Max marginal scores**
- **Mean max marginal**
- **Score of truth**
- **Max score**
- **Range of possible thresholds**
  \( \alpha = \text{efficiency level} \)
Example OCR Cascade

Mean ≈ 0  Max = 368  α = 0.55  Threshold ≈ 204
Example OCR Cascade
Example OCR Cascade
## Example OCR Cascade

<table>
<thead>
<tr>
<th>b</th>
<th>h</th>
<th>k</th>
<th>Mean ≈ 1412</th>
<th>α = 0.44</th>
</tr>
</thead>
<tbody>
<tr>
<td>1440</td>
<td>1558</td>
<td>1575</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ba</th>
<th>ha</th>
<th>ka</th>
<th>be</th>
<th>he</th>
<th>Max = 1575</th>
<th>Threshold ≈ 1502</th>
</tr>
</thead>
<tbody>
<tr>
<td>1440</td>
<td>1558</td>
<td>1575</td>
<td>1413</td>
<td>1553</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ab</th>
<th>ad</th>
<th>ag</th>
<th>ea</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1400</td>
<td>1480</td>
<td>1575</td>
<td>1397</td>
<td>1393</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ac</th>
<th>ae</th>
<th>ag</th>
<th>an</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1257</td>
<td>1285</td>
<td>1302</td>
<td>1294</td>
<td>1356</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ae</th>
<th>am</th>
<th>an</th>
<th>au</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1336</td>
<td>1306</td>
<td>1390</td>
<td>1346</td>
<td>1306</td>
<td></td>
</tr>
</tbody>
</table>
Example OCR Cascade

<table>
<thead>
<tr>
<th>b</th>
<th>h</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1440</td>
<td>1558</td>
<td>1575</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ba</th>
<th>ha</th>
<th>ka</th>
<th>be</th>
<th>he</th>
</tr>
</thead>
<tbody>
<tr>
<td>1440</td>
<td>1558</td>
<td>1575</td>
<td>1413</td>
<td>1553</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ab</th>
<th>ad</th>
<th>ag</th>
<th>ea</th>
</tr>
</thead>
<tbody>
<tr>
<td>1400</td>
<td>1480</td>
<td>1575</td>
<td>1397</td>
<td>1393</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ac</th>
<th>ae</th>
<th>ag</th>
<th>an</th>
</tr>
</thead>
<tbody>
<tr>
<td>1257</td>
<td>1285</td>
<td>1302</td>
<td>1294</td>
<td>1356</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>aa</th>
<th>ae</th>
<th>am</th>
<th>an</th>
<th>au</th>
</tr>
</thead>
<tbody>
<tr>
<td>1336</td>
<td>1306</td>
<td>1390</td>
<td>1346</td>
<td>1306</td>
</tr>
</tbody>
</table>

Mean $\approx 1412$
Max $= 1575$
Threshold $\approx 1502$

$\alpha = 0.44$
Example OCR Cascade

- h
- k
- k

Mean $\approx 1412$
Max $= 1575$
Threshold $\approx 1502$

- ha
- ka
- he
- ke
- kn

- ad
- ed
- nd

- do

- ow
- om

$\alpha = 0.44$
Example OCR Cascade
Example: Pictorial Structure Cascade

H
URA
T
ULA
LRA
LLA

k ≈ 320,000
k^2 ≈ 100mil
Quantifying Loss

• Filter loss

$$L_f(y, \theta_x) = 1 \left[ \theta_x(y) \leq t_x(\alpha) \right]$$

- If \text{score(truth)} > threshold, all true states are safe

• Efficiency loss

$$L_e(y, \theta_x) = \frac{1}{m} \sum_{y_c} 1 \left[ \theta_x^*(y_c) > t_x(\alpha) \right]$$

- Proportion of unfiltered clique assignments
Learning One Cascade Level

- Fix $\alpha$, solve convex problem for $\theta$

\[
\min_{\theta, \xi \geq 0} \frac{\lambda}{2} ||\theta||^2 + \frac{1}{n} \sum_i \xi_i \\
\text{s.t.} \quad \theta^T f(x^i, y^i) \geq t_{x^i}(\alpha) + 1 - \xi_i
\]

No filter mistakes          Margin w/ slack

\[
\min_{\theta} \frac{\lambda}{2} ||\theta||^2 + \frac{1}{n} \sum_i [t_{x^i}(\alpha) + 1 - \theta^T f(x^i, y^i)]_+
\]

Minimize filter mistakes at efficiency level $\alpha$
An Online Algorithm

- Stochastic sub-gradient update:

\[ \theta \leftarrow (1 - \lambda)\theta + \eta f(x^i, y^i) - \eta \partial t_{x^i}(\alpha) \]

**Features of truth**

\[ \partial t_{x^i}(\alpha) = \alpha f(x^i, y^*) + (1 - \alpha) \frac{1}{m} \sum_{y_c} f(x_i, y^*(y_c)) \]

**Convex combination:** Features of best guess + Average features of max marginal “witnesses”
Generalization Bounds

- W.h.p. $(1-\delta)$, filtering and efficiency loss observed on training set generalize to new data.

\[
\mathbb{E} \mathcal{L}_f(Y, \theta_X) \leq \mathbb{E} \phi_f(Y, \theta_X) + O \left( \frac{m \sqrt{\ell B}}{\gamma \sqrt{n}} \right) + \sqrt{\frac{8 \ln(2/\delta)}{n}},
\]

Expected loss on true distribution

\[
\mathcal{L}_f(y, \theta_x) = 1 \left[ \theta_x(y) \leq t_x(\alpha) \right]
\]
Generalization Bounds

- W.h.p. \((1-\delta)\), filtering and efficiency loss observed on training set \textbf{generalize} to new data:

\[
\mathbb{E} \mathcal{L}_f(Y, \theta_X) \leq \hat{\mathbb{E}} \phi_f(Y, \theta_X) + O \left( \frac{m \sqrt{\ell B}}{\gamma \sqrt{n}} \right) + \sqrt{\frac{8 \ln(2/\delta)}{n}},
\]

\textbf{Empirical }\gamma\text{-ramp upper bound}

\[\gamma\text{-ramp}(\theta_x(y) - t_x(\alpha))\]
Generalization Bounds

- W.h.p. $(1-\delta)$, filtering and efficiency loss observed on training set generalize to new data.

\[
\mathbb{E} L_f(Y, \theta_X) \leq \mathbb{E} \phi_f(Y, \theta_X) + O \left( \frac{m \sqrt{\ell B}}{\gamma \sqrt{n}} \right) + \sqrt{\frac{8 \ln(2/\delta)}{n}},
\]

- number of examples
- number of clique assignments
- number of cliques
- $|\theta|$
Generalization Bounds

• W.h.p. $(1-\delta)$, filtering and efficiency loss observed on training set generalize to new data:

\[
\mathbb{E} \mathcal{L}_f(Y, \theta_X) \leq \mathbb{E} \phi_f(Y, \theta_X) + O \left( \frac{m \sqrt{\ell B}}{\gamma \sqrt{n}} \right) + \sqrt{\frac{8 \ln(2/\delta)}{n}},
\]

Similar bound holds for $\mathcal{L}_e$ and all $\alpha \in [0,1]$. 
OCR Experiments

Dataset: http://www.cis.upenn.edu/~taskar/ocr
Efficiency Experiments

• **POS Tagging** (WSJ + CONLL datasets)
  – English, Bulgarian, Portuguese

• **Compare 2\textsuperscript{nd} order model filters**
  – Structured perceptron (max-sum)
  – SCP w/ $\alpha \in [0, 0.2, 0.4, 0.6, 0.8]$ (max-sum)
  – CRF log-likelihood (sum-product marginals)

• **Tightly controlled**
  – Use random 40% of training set
  – Use development set to fit regularization parameter $\lambda$ and $\alpha$ for all methods
POS Tagging Results

English

- Max-sum (SCP)
- Max-sum (0-1)
- Sum-product (CRF)
POS Tagging Results

Portuguese

- Max-sum (SCP)
- Max-sum (0-1)
- Sum-product (CRF)
POS Tagging Results

Bulgarian

Error Cap (%) on Development Set

Efficiency Loss (Test)

- Blue: Max-sum (SCP)
- Orange: Max-sum (0-1)
- Green: Sum-product (CRF)
English POS Cascade (2$^{nd}$ order)

<table>
<thead>
<tr>
<th></th>
<th>Full</th>
<th>SCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>96.83</td>
<td>96.82</td>
</tr>
<tr>
<td>Filter Loss (%)</td>
<td>0</td>
<td>0.12</td>
</tr>
<tr>
<td>Test Time (ms)</td>
<td>173.28</td>
<td><strong>1.56</strong></td>
</tr>
<tr>
<td>Avg. Num States</td>
<td>1935.7</td>
<td><strong>3.93</strong></td>
</tr>
</tbody>
</table>
## Pose Estimation (Buffy Dataset)

![Image showing pose estimation results](image)

<table>
<thead>
<tr>
<th>method</th>
<th>torso</th>
<th>head</th>
<th>upper arms</th>
<th>lower arms</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ferrari et al 08</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>61.4%</td>
</tr>
<tr>
<td>Ferrari et al. 09</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>74.5%</td>
</tr>
<tr>
<td>Andriluka et al. 09</td>
<td>98.3%</td>
<td>95.7%</td>
<td>86.8%</td>
<td>51.7%</td>
<td>78.8%</td>
</tr>
<tr>
<td>Eichner et al. 09</td>
<td>98.72%</td>
<td>97.87%</td>
<td>92.8%</td>
<td>59.79%</td>
<td>80.1%</td>
</tr>
<tr>
<td>CPS (ours)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

"Ferrari score:" percent of parts correct within radius
Pose Estimation

PCP

Percent of Correctly Matched Parts vs Correctly Estimated Segment Threshold (default = 0.5)

- CPS
- Ferrari et al. CVPR09
- Ferrari et al. CVPR08
- Eichner et al. BMVC09
Cascade Efficiency vs. Accuracy

<table>
<thead>
<tr>
<th>cascade level</th>
<th>max state size ( l_x \times l_y \times l_\theta )</th>
<th>average states per part</th>
<th>search space reduction</th>
<th>RMSE of oracle ( \times 10^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 \times 10 \times 12</td>
<td>31626</td>
<td>92.8%</td>
<td>5.14</td>
</tr>
<tr>
<td>2</td>
<td>10 \times 10 \times 24</td>
<td>14776</td>
<td>95.5%</td>
<td>5.66</td>
</tr>
<tr>
<td>3</td>
<td>40 \times 40 \times 24</td>
<td>4191</td>
<td>98.7%</td>
<td>6.41</td>
</tr>
<tr>
<td>4</td>
<td>80 \times 80 \times 24</td>
<td>882</td>
<td>99.7%</td>
<td>6.92</td>
</tr>
<tr>
<td>5</td>
<td>110 \times 122 \times 24</td>
<td>703</td>
<td>99.8%</td>
<td>7.24</td>
</tr>
</tbody>
</table>
Features: Shape/Segmentation Match
Features: Contour Continuation
Conclusions

• Novel loss significantly improves efficiency

• Principled learning of accurate cascades

• Deep cascades focus structured inference and allow rich models

• Open questions:
  – How to learn cascade structure
  – Dealing with intractable models
  – Other applications: factored dynamic models, grammars
Thanks!

Structured Prediction Cascades, Weiss & Taskar, AISTATS10

Cascaded Models for Articulated Pose Estimation, Sapp, Toshev& Taskar, ECCV10