A Provably Correct Learning Algorithm for Latent-Variable PCFGs

Shay Cohen\textsuperscript{1}  Michael Collins\textsuperscript{2}

\textsuperscript{1}University of Edinburgh

\textsuperscript{2}Columbia University

June 24, 2014
Previous work: spectral algorithm for L-PCFGs

 Introduced in Cohen et al. (2012), based on learning for HMMs (Hsu et al., 2009)

 An algorithm for latent-variable PCFGs

 Unlike EM, no local maxima problem

 More efficient than EM

 Experimentally, works as well as EM for parsing
Problem with previous work (Cohen et al., 2012)

Parameters are masked by an unknown linear transformation

- Negative probabilities (due to sampling error)
- Parameters cannot be easily interpreted
- Cannot improve parameters using, for example, EM
This talk in a nutshell

Like the spectral algorithm, has theoretical guarantees

Estimates are actual probabilities

More efficient than EM

Can be used to initialize EM, which converges in an iteration or two

Relies heavily on the idea of “pivot features”
  - Features that uniquely identify a latent state
  - A similar idea is used for topic modeling by Arora et al. (2013)
Outline of this talk

Latent-variable PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)

Algorithm for L-PCFG estimation

Experiments

Conclusion
L-PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)
The probability of a tree

\[ p(\text{tree}, 1 \ 3 \ 1 \ 2 \ 2 \ 4 \ 1) = \pi(S^1) \times t(S^1 \rightarrow NP^3 \ VP^2|S^1) \times t(NP^3 \rightarrow D^1 \ N^2|NP^3) \times t(VP^2 \rightarrow V^4 \ P^1|VP^2) \times q(D^1 \rightarrow \text{the}|D^1) \times q(N^2 \rightarrow \text{dog}|N^2) \times q(V^4 \rightarrow \text{saw}|V^4) \times q(P^1 \rightarrow \text{him}|P^1) \]

\[ p(\text{tree}) = \sum_{h_1 \ldots h_7} p(\text{tree}, h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7) \]
Outline of this talk

Latent-variable PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)

Algorithm for L-PCFG estimation

Experiments

Conclusion
Inside and outside trees

At node VP:

Outside tree \( o = \)

Inside tree \( t = \)

Conditionally independent given the label and the hidden state

\[ p(o, t|\text{VP}, h) = p(o|\text{VP}, h) \times p(t|\text{VP}, h) \]
Designing feature functions

Design functions $\psi$ and $\phi$:

- $\phi$ maps any inside tree to a binary vector of length $d$
- $\psi$ maps any outside tree to a binary vector of length $d'$

Outside tree $o \Rightarrow$

$$\psi(o) = [0, 1, 0, 0, \ldots, 0, 0] \in \mathbb{R}^{d'}$$

Inside tree $t \Rightarrow$

$$\phi(t) = [1, 0, 0, 0, \ldots, 0, 0] \in \mathbb{R}^{d}$$

$\psi$ and $\phi$ as multinomials $p(f)$ for $f \in [d]$ and $p(g)$ for $g \in [d']$. 
Think of \( f \) and \( g \) as representing a whole inside/outside tree

Say we had a way of getting:

- \( p(f|h, \text{VP}) \) for each \( h \) and \( f \) inside feature
- \( p(g|h, \text{VP}) \) for each \( h \) and \( g \) outside feature

Then we could run EM on a convex problem to find parameters.

How?
Binary rule estimation

Take $M$ samples of nodes with rule $\text{VP} \rightarrow \text{V} \ \text{NP}$.

At sample $i$

- $g^{(i)} = \text{outside feature at VP}$
- $f_2^{(i)} = \text{inside feature at V}$
- $f_3^{(i)} = \text{inside feature at NP}$

$$
\{\hat{t}(h_1, h_2, h_3|\text{VP} \rightarrow \text{V} \ \text{NP})|h_1, h_2, h_3\} \\
= \arg \max \sum_{i=1}^{M} \log \sum_{h_1, h_2, h_3} (\hat{t}(h_1, h_2, h_3|\text{VP} \rightarrow \text{V} \ \text{NP}) \times \\
p(g^{(i)}|h_1, \text{VP})p(f_2^{(i)}|h_2, \text{V})p(f_3^{(i)}|h_3, \text{NP})
$$
Binary rule estimation, cont’d

$$\{\hat{t}(h_1, h_2, h_3|\text{VP} \rightarrow V \text{ NP})| h_1, h_2, h_3\}$$

$$= \arg \max_{\hat{t}} \sum_{i=1}^{M} \log \sum_{h_1, h_2, h_3} (\hat{t}(h_1, h_2, h_3|\text{VP} \rightarrow V \text{ NP}) \times$$

$$p(g^{(i)}|h_1, \text{VP})p(f_2^{(i)}|h_2, V)p(f_3^{(i)}|h_3, \text{NP})$$

This objective represents the marginal probability of the corpus

It is a convex objective

Use Bayes’ rule to convert to parameters

**Main question:** how do we get the latent state distributions $p(h|f, \text{VP})$ and $p(h|g, \text{VP})$?
Vector representation of inside and outside trees

Design functions $Z$ and $Y$:

$Y$ maps any inside feature value $f \in [d']$ to a vector of length $m$.

$Z$ maps any outside feature value $g \in [d]$ to a vector of length $m$.

Convention: $m$ is the number of hidden states under the L-PCFG.

Outside tree $o \Rightarrow$

$Z(g) = [1, 0.4, -5.3, \ldots, 72] \in \mathbb{R}^m$

Inside tree $t \Rightarrow$

$Y(f) = [-3, 17, 2, \ldots, 3.5] \in \mathbb{R}^m$

$Z$ and $Y$ reduce the dimensionality of $\phi$ and $\psi$ using CCA
Identifying latent state distributions

• For each $f \in [d]$, define: $v(f) = E[Z(g)|f, \text{VP}]$

• $v(f) \in \mathbb{R}^m$ is “the expected value of an outside tree (representation) given an inside tree (feature)”
Identifying latent state distributions

- For each $f \in [d]$, define: $v(f) = E[Z(g)|f, VP]$

- $v(f) \in \mathbb{R}^m$ is “the expected value of an outside tree (representation) given an inside tree (feature)”

- By conditional independence:

$$v(f) = \sum_{h=1}^{m} p(h|f, VP)w(h)$$

where $w(h) \in \mathbb{R}^m$ and

$$w(h) = \sum_{g=1}^{d'} p(g|h, VP)Z(g) = E[Z(g)|h, VP].$$

- $w(h)$ is “the expected value of an outside tree (representation) given a latent state”
Pivot assumption

Reminder: \( v(f) = \sum_{h=1}^{m} p(h|f, \text{VP})w(h) = E[Z(g)|f, \text{VP}] \)

Need to solve with respect to \( p(h|f, \text{VP}) \)

\( v(f) \) can be estimated from data

\( w(h) \) consist of information about latent states – not observable

**Pivot assumption:** each \( h \) has \( f \) such that \( p(h|f, \text{VP}) = 1 \)
Outside representation as convex combination

Reminder: \( v(f) = \sum_{h=1}^{m} p(h|f, \text{VP}) w(h) = E[Z(g)|f, \text{VP}] \)

\[ v(101) = w(1) \]
\[ v(25) = w(2) \]
\[ v(72) = w(3) \]
\[ v(28) = w(4) \]
\[ v(33) = w(5) \]
\[ v(102) = \sum_{h=1}^{5} \alpha_h w(i) \]

Pivot assumption: each \( h \) has \( f \) such that \( p(h|f, \text{VP}) = 1 \)

Features 101, 25, 72, 28, 33 are “pivot” features for 5 states

\( \alpha \) are the latent state distributions!
Summary of algorithm

Calculate \( v(f) \) for all \( f \)

Identify \( w(h) \) for all \( h \) by finding the corners of \( \text{ConvexHull}(v(1), \ldots, v(d)) \)

Identify the distribution of latent states by solving

\[
v(f) = \sum_{h=1}^{m} p(h|f, \text{VP})w(h) = E[Z(g)|f, \text{VP}]
\]

Repeat the above for outside features \( g \)

Solve a convex marginal log-likelihood problem
Outline of this talk

Latent-variable PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)

Algorithm for L-PCFG estimation

Experiments

Conclusion
Experiments: language modeling

- Saul and Pereira (1997):

\[ p(w_2|w_1) = \sum_h p(w_2|h)p(h|w_1). \]

This model is a specific case of L-PCFG

- Experimented with bi-gram modeling for two corpora: Brown corpus and Gigaword corpus
## Results: perplexity

<table>
<thead>
<tr>
<th>m</th>
<th>Brown</th>
<th>NYT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128 256</td>
<td>test</td>
</tr>
<tr>
<td>bigram Kneser-Ney</td>
<td>408 415</td>
<td>271 279</td>
</tr>
<tr>
<td>trigram Kneser-Ney</td>
<td>386 394</td>
<td>150 158</td>
</tr>
<tr>
<td>EM iterations</td>
<td>388 365</td>
<td>284 265</td>
</tr>
<tr>
<td>pivot</td>
<td>426 597</td>
<td>782 886</td>
</tr>
<tr>
<td></td>
<td>560 715</td>
<td></td>
</tr>
</tbody>
</table>
Results: perplexity

<table>
<thead>
<tr>
<th>m</th>
<th>Brown</th>
<th>NYT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td>bigram Kneser-Ney</td>
<td>408</td>
<td>415</td>
</tr>
<tr>
<td>trigram Kneser-Ney</td>
<td>386</td>
<td>394</td>
</tr>
<tr>
<td>EM iterations</td>
<td>388</td>
<td>365</td>
</tr>
<tr>
<td>pivot</td>
<td>426</td>
<td>597</td>
</tr>
<tr>
<td>pivot+EM</td>
<td>310</td>
<td>327</td>
</tr>
</tbody>
</table>

- Initialize EM with pivot algorithm output
- EM converges in much fewer iterations
- Still consistent - called “two-step estimation” (Lehmann and Casella, 1998)
Inside features used

Consider the VP node in the following tree:

- The pairs (VP, V) and (VP, NP)
- The rule VP → V NP
- The tree fragment (VP (V saw) NP)
- The tree fragment (VP V (NP D N))
- The pair of head part-of-speech tag with VP: (VP, V)
Outside features used

Consider the D node in the following tree:

```
S
 /   
NP    VP
 /     /
D    N  V    NP
 |     /
|      D
|      |
the  cat  saw
```

The outside features consist of:

- The fragments
  
  \[
  \text{NP}, \quad \text{VP}
  \]

- The pair \((D, \text{NP})\) and triplet \((D, \text{NP}, \text{VP})\)

- The pair of head part-of-speech tag with \(D\): \((D, N)\)
Results

<table>
<thead>
<tr>
<th>m</th>
<th></th>
<th>sec. 22</th>
<th></th>
<th>sec. 23</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>32</td>
</tr>
<tr>
<td>EM iterations</td>
<td>86.69</td>
<td>88.32</td>
<td>88.35</td>
<td>88.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>30</td>
<td>30</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Spectral</td>
<td>85.60</td>
<td>87.77</td>
<td>88.53</td>
<td>88.82</td>
<td></td>
</tr>
<tr>
<td>(Cohen et al., 2013)</td>
<td>88.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pivot</td>
<td>83.56</td>
<td>86.00</td>
<td>86.87</td>
<td>86.40</td>
<td></td>
</tr>
<tr>
<td>Pivot+EM iterations</td>
<td>86.83</td>
<td>88.14</td>
<td>88.64</td>
<td>88.55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Again, EM converges in very few iterations
Conclusion

Formal guarantees:

- Statistical consistency
- No problem of local maxima

Advantages over traditional spectral methods:

- No negative probabilities
- More intuitive to understand