Experiments with Spectral Learning of Latent-Variable PCFGs

Shay Cohen

Department of Computer Science
Columbia University

Joint work with Karl Stratos\textsuperscript{1}, Michael Collins\textsuperscript{1}, Dean P. Foster\textsuperscript{2} and Lyle Ungar\textsuperscript{2}

\textsuperscript{1}Columbia University
\textsuperscript{2}University of Pennsylvania

June 10, 2013
Spectral algorithms

Broadly construed:
Algorithms that make use of spectral decomposition

Recent work:
Spectral algorithms with latent-variables (statistically consistent):

- Gaussian mixtures (Vempala and Wang, 2004)
- Hidden Markov models (Hsu et al., 2009; Siddiqi et al., 2010)
- Latent-variable models (Kakade and Foster, 2007)
- Grammars (Bailly et al., 2010; Luque et al., 2012; Cohen et al., 2012; Dhillon et al., 2012)

Prior work: mostly theoretical
This talk in a nutshell

Experiments on spectral estimation of latent-variable PCFGs

Accuracy is the same as EM, but order of magnitude more efficient

The algorithm has PAC-style guarantees
Outline of this talk

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

Spectral algorithm for L-PCFGs (Cohen et al., 2012)

Experiments

Conclusion
L-PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)

NP  VP
D  N  V  P

the  dog  saw  him

NP  VP
D  N  V  P

the  dog  saw  him
The probability of a tree

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

\[ p(\text{tree}) = \sum_{h_1...h_7} p(\text{tree}, h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7) \]
The EM algorithm

Goal: estimate $\pi$, $t$ and $q$ from labeled data

EM is a remarkable algorithm for learning from incomplete data

It has been used for L-PCFG parsing, among other things

It has two flaws:

- Requires careful initialization
- Does not give consistent parameter estimates

More generally, it {	extit{locally}} maximizes the objective function
Outline of this talk

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

Spectral algorithm for L-PCFGs (Cohen et al., 2012)

Experiments

Conclusion
Inside and outside trees

At node $\textit{VP}$:

Outside tree $o =$

```
S
   /\  \\
NP / \ VP
  /   \   \\
D  N   V  P
the dog saw him
```

Inside tree $t =$

```
S
   /\  \\
NP / \ VP
  /   \   \\
    D  N
     the dog

VP
   /\  \\
V  P
   saw him
```

Conditionally independent given the label and the hidden state

$$p(o, t|\textit{VP}, h) = p(o|\textit{VP}, h) \times p(t|\textit{VP}, h)$$
Spectral algorithm

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

$\psi$ maps any outside tree to a vector of length $d'$

$\phi$ maps any inside tree to a vector of length $d$

Outside tree $o \Rightarrow$

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

Inside tree $t \Rightarrow$

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

Project the feature vectors to $m$-dimensional space ($m << d$)

- Use singular value decomposition

The result of the projection is two functions $Z$ and $Y$:

- $Z$ maps any outside tree to a vector of length $m$
- $Y$ maps any inside tree to a vector of length $m$

**Outside tree $o \Rightarrow$**

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

**Inside tree $t \Rightarrow$**

$$Y(t) = [-3, 17, 2, \ldots, 3.5] \in \mathbb{R}^m$$
Parameter estimation for binary rules

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

At sample $i$

- $o^{(i)} = \text{outside tree at } VP$
- $t_2^{(i)} = \text{inside tree at } V$
- $t_3^{(i)} = \text{inside tree at } NP$

$$
\hat{i}(VP^{h_1} \rightarrow V^{h_2} \ NP^{h_3} | VP^{h_1}) = \frac{\text{count}(VP \rightarrow V \ NP)}{\text{count}(VP)} \times \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_2^{(i)}) \times Y_{h_3}(t_3^{(i)}) \right)
$$
Parameter estimation for unary rules

Take $M$ samples of nodes with rule $N \rightarrow \text{dog}$.

At sample $i$

- $o^{(i)} = \text{outside tree at } N$

\[
\hat{q}(N^h \rightarrow \text{dog}|N^h) = \frac{\text{count}(N \rightarrow \text{dog})}{\text{count}(N)} \times \frac{1}{M} \sum_{i=1}^{M} Z_h(o^{(i)})
\]
Parameter estimation for the root

Take $M$ samples of the root $S$.

At sample $i$

- $t^{(i)} = \text{inside tree at } S$

\[
\hat{\pi}(S^h) = \frac{\text{count}(\text{root}=S)}{\text{count}(\text{root})} \times \frac{1}{M} \sum_{i=1}^{M} Y_h(t^{(i)})
\]
Outline of this talk

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

Spectral algorithm for L-PCFGs (Cohen et al., 2012)

Experiments

Conclusion
Performance with expectation-maximization ($m = 32$): 88.56%

Vanilla PCFG maximum likelihood estimation performance: 68.62%

For the rest of the talk, we will focus on $m = 32$
Key ingredients for accurate spectral learning

Feature functions

Handling negative marginals

Scaling of features

Smoothing
Inside features used

Consider the VP node in the following tree:

```
S
  NP  VP
    D       
    N       
  the    saw
       
NP
  D       
  N
    the   dog
```

The inside features consist of:

- The pairs \((\text{VP}, \text{V})\) and \((\text{VP}, \text{NP})\)
- The rule \(\text{VP} \rightarrow \text{V} \ \text{NP}\)
- The tree fragment \((\text{VP} \ (\text{V} \ \text{saw}) \ \text{NP})\)
- The tree fragment \((\text{VP} \ \text{V} \ (\text{NP} \ \text{D} \ \text{N}))\)
- The pair of head part-of-speech tag with VP: \((\text{VP}, \text{V})\)
- The width of the subtree spanned by VP: \((\text{VP}, 2)\)
Outside features used

Consider the D node in the following tree:

```
S
   NP          VP
     D          V  NP
        N   the  saw  D
       cat  the  N  dog
```

The outside features consist of:

- The fragments

```
  NP
   D*
   N
```

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

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

- The widths of the spans left and right to D: \((D, 3)\) and \((D, 1)\)
Accuracy (section 22 of the Penn treebank)

The accuracy out-of-the-box with these features is:

55.09%

EM’s accuracy: 88.56%
Negative marginals

Sampling error can lead to negative marginals

Signs of marginals are flipped

On certain sentences, this gives the world’s worst parser:

$$t^* = \arg\max_t -\text{score}(t) = \arg\min_t \text{score}(t)$$

Taking the absolute value of the marginals fixes it

Likely to be caused by sampling error
Accuracy (section 22 of the Penn treebank)

The accuracy with absolute-value marginals is:

80.23%

EM’s accuracy: 88.56%
Scaling of features by inverse variance

Features are mostly binary. Replace $\phi_i(t)$ by

$$\phi_i(t) \times \sqrt{\frac{1}{\text{count}(i) + \kappa}}$$

where $\kappa = 5$

This is an approximation to replacing $\phi(t)$ by

$$(C)^{-1/2}\phi(t)$$

where $C = E[\phi\phi^T]$  

Closely related to canonical correlation analysis (e.g. Dhillon et al., 2011)
Accuracy (section 22 of the Penn treebank)

The accuracy with scaling is:

86.47%

EM’s accuracy: 88.56%
Smoothing

Estimates required:

\[
\hat{E}(VP^{h_1} \rightarrow V^{h_2} \ NP^{h_3} | VP^{h_1}) = \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_{2}^{(i)}) \times Y_{h_3}(t_{3}^{(i)}) \right)
\]

Smooth using “backed-off” estimates, e.g.:

\[
\lambda \hat{E}(VP^{h_1} \rightarrow V^{h_2} \ NP^{h_3} | VP^{h_1}) + (1 - \lambda) \hat{F}(VP^{h_1} \rightarrow V^{h_2} \ NP^{h_3} | VP^{h_1})
\]

where

\[
\hat{F}(VP^{h_1} \rightarrow V^{h_2} \ NP^{h_3} | VP^{h_1})
\]

\[
= \left( \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_{2}^{(i)}) \right) \right) \times \left( \frac{1}{M} \sum_{i=1}^{M} Y_{h_3}(t_{3}^{(i)}) \right)
\]
The accuracy with smoothing is: 

88.82%

EM's accuracy: 88.56%
Final results

Final results on the Penn treebank

<table>
<thead>
<tr>
<th></th>
<th>section 22</th>
<th></th>
<th>section 23</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EM</td>
<td>spectral</td>
<td>EM</td>
<td>spectral</td>
</tr>
<tr>
<td>$m = 8$</td>
<td>86.87</td>
<td>85.60</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>$m = 16$</td>
<td>88.32</td>
<td>87.77</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>$m = 24$</td>
<td>88.35</td>
<td>88.53</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>$m = 32$</td>
<td>88.56</td>
<td>88.82</td>
<td>87.76</td>
<td>88.05</td>
</tr>
</tbody>
</table>
Simple feature functions

Use rule above (for outside) and rule below (for inside)

Corresponds to parent annotation and sibling annotation

Accuracy:

88.07%

Accuracy of parent and sibling annotation: 82.59%

The spectral algorithm distills latent states

Avoids overfitting caused by Markovization
Training time ($m = 32$)

EM runs for 9 hours and 21 minutes per iteration

Spectral algorithm runs for less than 10 hours beginning to end

EM requires about 20 iterations to converge (187h12m)
Outline of this talk

Latent-variable PCFGs \textit{(Matsuzaki et al., 2005; Petrov et al., 2006)}

Spectral algorithm for L-PCFGs \textit{(Cohen et al., 2012)}

Experiments

Conclusion
Conclusion

Presented spectral algorithms as a method for estimating latent-variable models

Formal guarantees:
- Statistical consistency
- No problem of local maxima

Complexity:
- Most time is spent on aggregating statistics
- Much faster than EM (20x faster)

Future work:
- Promising direction for hybrid EM-spectral algorithm (89.85%)