# Experiments with Spectral Learning of Latent-Variable PCFGs

Shay Cohen

Department of Computer Science
Columbia University

Joint work with Karl Stratos[1], Michael Collins[1], Dean P. Foster[2] and Lyle Ungar[2]

[1]Columbia University

[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

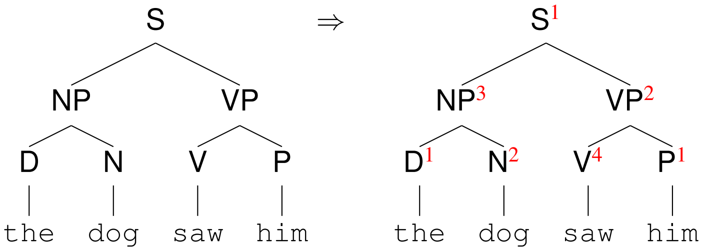
Latent-variable PCFGs (Matsuzaki et al., 2005; Petrov et al., 2006)

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

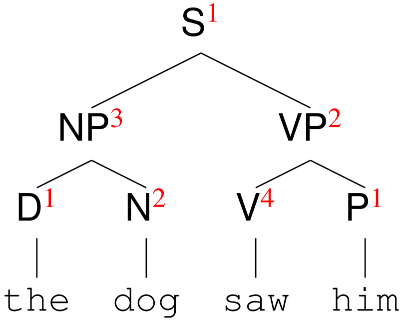
Experiments

Conclusion

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



# The probability of a tree




$$
\begin{aligned}
& p(\text{tree}, 1\ 3\ 1\ 2\ 2\ 4\ 1) \\
&= \pi(\mathsf{S}^1)\times \\
&\quad t(\mathsf{S}^1 \rightarrow \mathsf{NP}^3\ \ \mathsf{VP}^2 | \mathsf{S}^1)\times \\
&\quad t(\mathsf{NP}^3 \rightarrow \mathsf{D}^1\ \ \mathsf{N}^2 | \mathsf{NP}^3)\times \\
&\quad t(\mathsf{VP}^2 \rightarrow \mathsf{V}^4\ \ \mathsf{P}^1 | \mathsf{VP}^2)\times \\
&\quad q(\mathsf{D}^1 \rightarrow \texttt{the} | \mathsf{D}^1)\times \\
&\quad q(\mathsf{N}^2 \rightarrow \texttt{dog} | \mathsf{N}^2)\times \\
&\quad q(\mathsf{V}^4 \rightarrow \texttt{saw} | \mathsf{V}^4)\times \\
&\quad q(\mathsf{P}^1 \rightarrow \texttt{him} | \mathsf{P}^1)
\end{aligned}
$$

$$
p(\text{tree}) = \sum_{h_1 \ldots h_7} p(tree, h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7)
$$

# 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 **locally** maximizes the objective function

# Outline of this talk

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

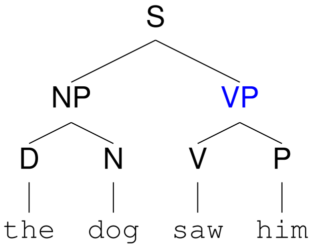
Spectral algorithm for L-PCFGs (Cohen et al., 2012)

Experiments

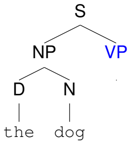
Conclusion

# Inside and outside trees

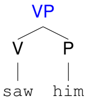
At node VP:




Outside tree $o =$




Inside tree $t =$




Conditionally independent given the label and the hidden state

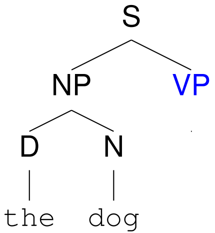
$$p(o, t|\text{VP}, h) = p(o|\text{VP}, h) \times p(t|\text{VP}, h)$$

# Spectral algorithm

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

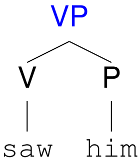
$\psi$ maps any outside tree to a vector of length $d'$

$\phi$ 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 $t \Rightarrow$

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

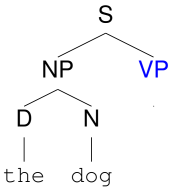
# Spectral algorithm

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

- Use singular value decomposition

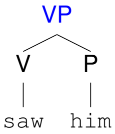
The result of the projection is two functions $Z$ and $Y$:

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




Outside tree $o \Rightarrow$

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

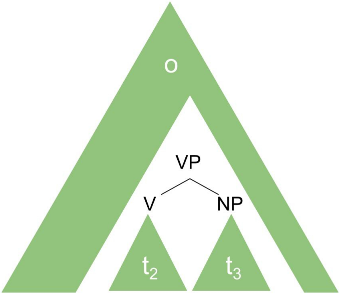



Inside tree $t \Rightarrow$

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

# Parameter estimation for binary rules

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




At sample $i$

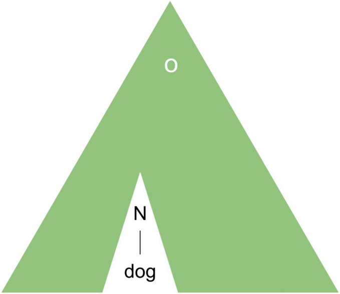
- $o^{(i)}$ = outside tree at VP
- $t_2^{(i)}$ = inside tree at V
- $t_3^{(i)}$ = inside tree at NP

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

$$= \frac{\textbf{count}(\text{VP} \rightarrow \text{V} \ \ \text{NP})}{\textbf{count}(\text{VP})} \times \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_2^{(i)}) \times Y_{h_3}(t_3^{(i)}) \right)$$

# Parameter estimation for unary rules

Take $M$ samples of nodes with rule N $\rightarrow$ `dog`.



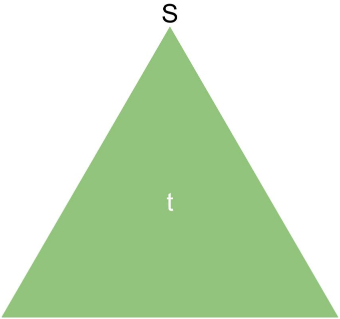

At sample $i$

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

$$\hat{q}(\mathsf{N}^{h} \rightarrow \texttt{dog}|\mathsf{N}^{h}) = \frac{\textbf{count}(\mathsf{N} \rightarrow \texttt{dog})}{\textbf{count}(\mathsf{N})} \times \frac{1}{M}\sum_{i=1}^{M} Z_{h}(o^{(i)})$$

# Parameter estimation for the root

Take $M$ samples of the root S.




At sample $i$

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

$$\hat{\pi}(\mathsf{S}^{h}) = \frac{\textbf{count}(\text{root=S})}{\textbf{count}(\text{root})} \times \frac{1}{M}\sum_{i=1}^{M} Y_{h}(t^{(i)})$$

# 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

# Results with EM (section 22 of Penn treebank)

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

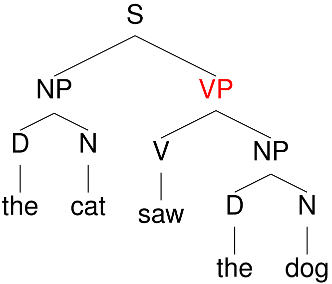
Handling negative marginals

Scaling of features

Smoothing

# Inside features used

Consider the VP node in the following tree:

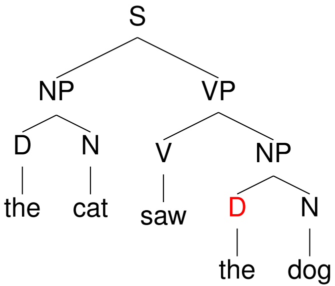



The inside features consist of:

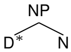
- The pairs `(VP, V)` and `(VP, NP)`
- The rule `VP` $\rightarrow$ `V NP`
- The tree fragment `(VP (V saw) NP)`
- The tree fragment `(VP V (NP D N))`
- The pair of head part-of-speech tag with VP: `(VP, V)`
- The width of the subtree spanned by VP: `(VP, 2)`

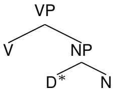
# Outside features used

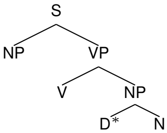
Consider the D node in the following tree:




The outside features consist of:

- The fragments 

, 

and 

- The pair `(D, NP)` and triplet `(D, NP, VP)`
- The pair of head part-of-speech tag with D: `(D, 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:

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

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 $C = E[\phi\phi^\top]$

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:

$$\hat{E}(\mathsf{VP}^{h_1} \rightarrow \mathsf{V}^{h_2}\ \ \mathsf{NP}^{h_3} | \mathsf{VP}^{h_1}) = \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_2^{(i)}) \times Y_{h_3}(t_3^{(i)}) \right)$$

Smooth using "backed-off" estimates, e.g.:

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

where

$$\begin{aligned} &\hat{F}(\mathsf{VP}^{h_1} \rightarrow \mathsf{V}^{h_2}\ \ \mathsf{NP}^{h_3} | \mathsf{VP}^{h_1}) \\ &= \left( \frac{1}{M} \sum_{i=1}^{M} \left( Z_{h_1}(o^{(i)}) \times Y_{h_2}(t_2^{(i)}) \right) \right) \times \left( \frac{1}{M} \sum_{i=1}^{M} Y_{h_3}(t_3^{(i)}) \right) \end{aligned}$$

# Accuracy (section 22 of the Penn treebank)

The accuracy with smoothing is:

$$88.82\%$$

EM's accuracy: $88.56\%$

# Final results

Final results on the Penn treebank

| | section 22 | | section 23 | |
|---|---|---|---|---|
| | EM | spectral | EM | spectral |
| $m = 8$ | 86.87 | 85.60 | — | — |
| $m = 16$ | 88.32 | 87.77 | — | — |
| $m = 24$ | 88.35 | 88.53 | — | — |
| $m = 32$ | 88.56 | 88.82 | 87.76 | 88.05 |

# 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 (Matsuzaki et al., 2005; Petrov et al., 2006)

Spectral algorithm for L-PCFGs (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%)