Spectral Learning of Latent-Variable PCFGs

Karl Stratos[1]

Joint work with Shay Cohen[1], Michael Collins[1], Dean Foster[2], and Lyle Ungar[2]

[1]Columbia University

[2]University of Pennsylvania

Latent-Variable Models for NLP and Speech

- Latent-variable models are of huge importance.
  - Speech recognition with HMMs
  - Gaussian mixture models
  - Machine translation with alignments as hidden variables
  - Latent-variable PCFGs (Matsuzaki et al., Petrov et al.)
  - Many many others
- The EM algorithm is remarkably successful. **But**:
  - No guarantee of reaching the global maximum of the likelihood function
  - Theoretical problem: parameter estimates not consistent
  - Practical problems: local optima difficult to deal with

There is Hope

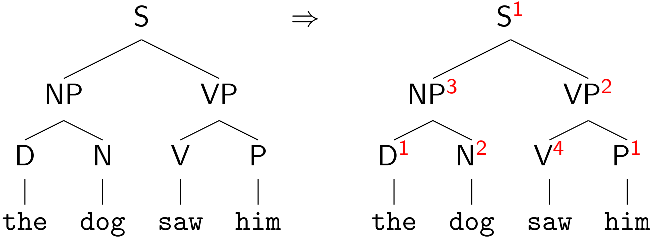
- Dasgupta (1999): Under separation conditions, it is possible to learn GMMs.
- Moitra and Valiant (2010): Arbitrary GMMs can be learned in polynomial time and sample complexity.
- Hsu, Kakade, and Zhang (2009): Under rank conditions, it is possible to learn HMMs efficiently and consistently.
- Kakade and Foster (2007): Under a wide class of models, CCA projections yield an optimal space for predicting hidden variables.

This Work

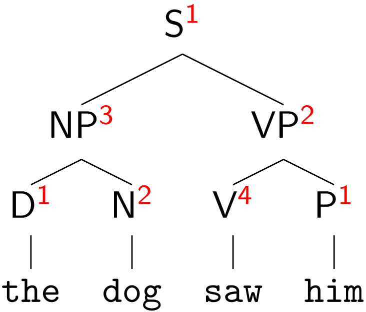
- A spectral algorithm for learning latent-variable PCFGs
  L-PCFGs: Strong parsing performance (Petrov et al., 2006)
- Guaranteed to give consistent parameter estimates under assumptions on singular values
- Simple and efficient (SVD and matrix operations)

Overview

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



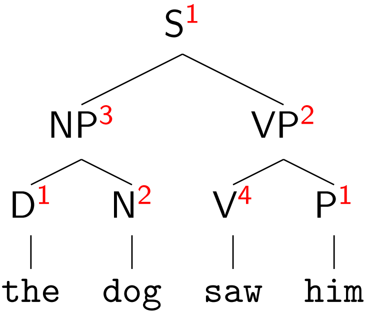
The Probability of a Tree



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

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

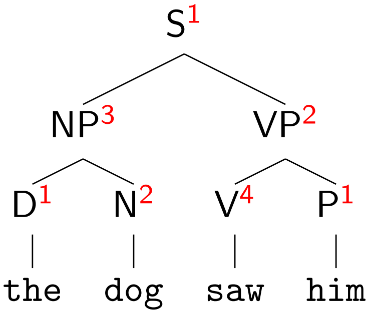
The Probability of a Tree



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

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

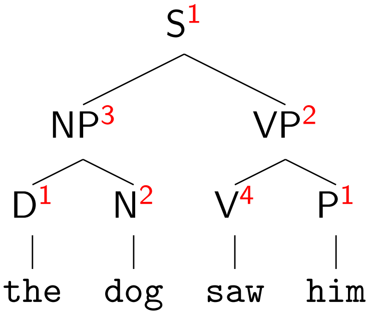
The Probability of a Tree



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

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

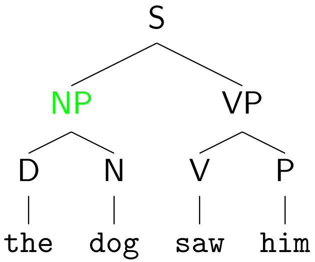
The Probability of a Tree



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

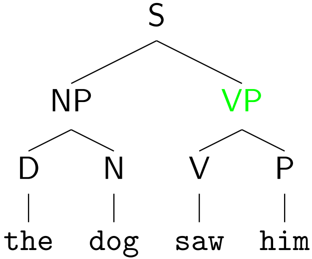
$$
p(\text{tree}) = \sum_{h_1 \ldots h_7} p(\mathit{tree}, h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7)
$$

Calculating Tree Probability with Dynamic Programming



$$b_h^1 = \sum_{h_2, h_3} t(\mathrm{NP}^h \rightarrow \mathrm{D}^{h_2} \ \mathrm{N}^{h_3} | \mathrm{NP}^h) \times q(\mathrm{D}^{h_2} \rightarrow \texttt{the} | \mathrm{D}^{h_2}) \times q(\mathrm{N}^{h_3} \rightarrow \texttt{dog} | \mathrm{N}^{h_3})$$

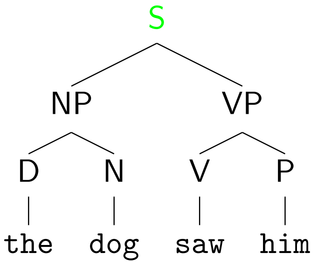
Calculating Tree Probability with Dynamic Programming



$$b_h^1 = \sum_{h_2, h_3} t(\mathrm{NP}^h \rightarrow \mathrm{D}^{h_2}\ \ \mathrm{N}^{h_3} | \mathrm{NP}^h) \times q(\mathrm{D}^{h_2} \rightarrow \texttt{the} | \mathrm{D}^{h_2}) \times q(\mathrm{N}^{h_3} \rightarrow \texttt{dog} | \mathrm{N}^{h_3})$$

$$b_h^2 = \sum_{h_2, h_3} t(\mathrm{VP}^h \rightarrow \mathrm{V}^{h_2}\ \ \mathrm{P}^{h_3} | \mathrm{VP}^h) \times q(\mathrm{V}^{h_2} \rightarrow \texttt{saw} | \mathrm{V}^{h_2}) \times q(\mathrm{P}^{h_3} \rightarrow \texttt{him} | \mathrm{P}^{h_3})$$

Calculating Tree Probability with Dynamic Programming

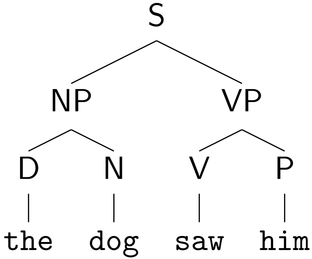


$$b_h^1 = \sum_{h_2, h_3} t(\text{NP}^h \rightarrow \text{D}^{h_2} \ \ \text{N}^{h_3} | \text{NP}^h) \times q(\text{D}^{h_2} \rightarrow \texttt{the} | \text{D}^{h_2}) \times q(\text{N}^{h_3} \rightarrow \texttt{dog} | \text{N}^{h_3})$$

$$b_h^2 = \sum_{h_2, h_3} t(\text{VP}^h \rightarrow \text{V}^{h_2} \ \ \text{P}^{h_3} | \text{VP}^h) \times q(\text{V}^{h_2} \rightarrow \texttt{saw} | \text{V}^{h_2}) \times q(\text{P}^{h_3} \rightarrow \texttt{him} | \text{P}^{h_3})$$

$$b_h^3 = \sum_{h_2, h_3} t(\text{S}^h \rightarrow \text{NP}^{h_2} \ \ \text{VP}^{h_3} | \text{S}^h) \times b_{h_2}^1 \times b_{h_3}^2$$

Calculating Tree Probability with Dynamic Programming



$$b_h^1 = \sum_{h_2, h_3} t(\text{NP}^h \rightarrow \text{D}^{h_2} \ \ \text{N}^{h_3} | \text{NP}^h) \times q(\text{D}^{h_2} \rightarrow \texttt{the} | \text{D}^{h_2}) \times q(\text{N}^{h_3} \rightarrow \texttt{dog} | \text{N}^{h_3})$$

$$b_h^2 = \sum_{h_2, h_3} t(\text{VP}^h \rightarrow \text{V}^{h_2} \ \ \text{P}^{h_3} | \text{VP}^h) \times q(\text{V}^{h_2} \rightarrow \texttt{saw} | \text{V}^{h_2}) \times q(\text{P}^{h_3} \rightarrow \texttt{him} | \text{P}^{h_3})$$

$$b_h^3 = \sum_{h_2, h_3} t(\text{S}^h \rightarrow \text{NP}^{h_2} \ \ \text{VP}^{h_3} | \text{S}^h) \times b_{h_2}^1 \times b_{h_3}^2$$

$$p(\text{tree}) = \sum_h \pi(S^h) \times b_h^3$$

Marginals of a Sentence

- Given a sentence x, a marginal is defined as

$$\mu(a, i, j) = \sum_{t \in \tau(x) : (a,i,j) \in t} p(t)$$

for all (a, i, j) tuples.

- These marginals can be computed using a variant of the inside-outside algorithm.
- A dynamic programming algorithm (Goodman, 1996) can be used to find the optimal parse defined as

$$t^* = \arg\max_{t \in \tau(x)} \sum_{(a,i,j) \in t} \mu(a, i, j)$$

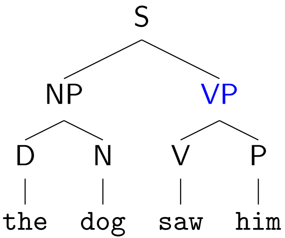
Parameter Estimation

- So this is a **parameter estimation** problem.
  - Given only skeletal trees, can we estimate π, t and q?
- Past work used EM (Matsuzaki et al., 2005, Petrov et al. 2006).
  - No guarantee of converging to the correct distribution
  - Prone to local optima
- We present a spectral estimation method.
  - Under assumptions on singular values, gives consistent parameter estimates
  - Relatively simple, efficient

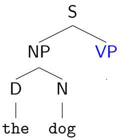
Overview

Inside and Outside Trees

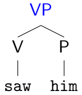
At node VP:



Outside tree o =



Inside tree t =



Conditionally independent given the label and the hidden state

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

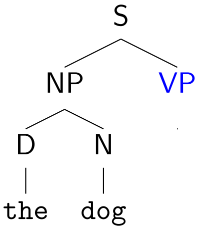
Vector Representation of Inside and Outside Trees

Assume functions Z and Y:

Z maps any outside tree to a vector of length m.

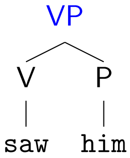
Y maps any inside tree to a vector of length m.

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



Outside tree $o \Rightarrow$

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

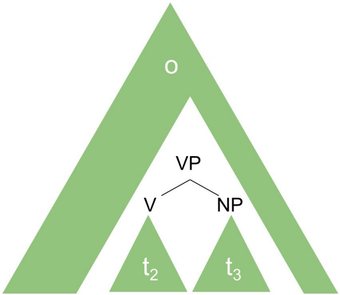


Inside tree $t \Rightarrow$

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

Parameter Estimation for Binary Rules

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



At sample i

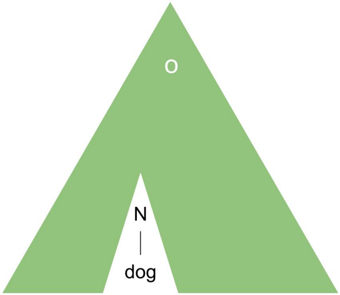
- $o^{(i)}$ = outside tree at VP
- $t_2^{(i)}$ = inside tree at V
- $t_3^{(i)}$ = inside tree at NP

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

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

Parameter Estimation for Unary Rules

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



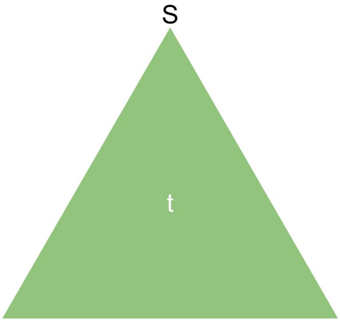
At sample i

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

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

Parameter Estimation for the Root

Take M samples of the root S.

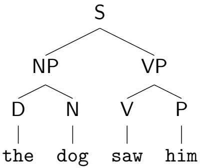


At sample i

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

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

Calculating Tree Probability with Dynamic Programming: Revisited



$$\hat{b}^1_h = \sum_{h_2, h_3} \hat{t}(\text{NP}^h \rightarrow \text{D}^{h_2} \ \ \text{N}^{h_3} | \text{NP}^h) \times \hat{q}(\text{D}^{h_2} \rightarrow \texttt{the} | \text{D}^{h_2}) \times \hat{q}(\text{N}^{h_3} \rightarrow \texttt{dog} | \text{N}^{h_3})$$

$$\hat{b}^2_h = \sum_{h_2, h_3} \hat{t}(\text{VP}^h \rightarrow \text{V}^{h_2} \ \ \text{P}^{h_3} | \text{VP}^h) \times \hat{q}(\text{V}^{h_2} \rightarrow \texttt{saw} | \text{V}^{h_2}) \times \hat{q}(\text{P}^{h_3} \rightarrow \texttt{him} | \text{P}^{h_3})$$

$$\hat{b}^3_h = \sum_{h_2, h_3} \hat{t}(\text{S}^h \rightarrow \text{NP}^{h_2} \ \ \text{VP}^{h_3} | \text{S}^h) \times \hat{b}^1_{h_2} \times \hat{b}^2_{h_3}$$

$$p(\text{tree}) = \sum_{h} \hat{\pi}(S^h) \times \hat{b}^3_h$$

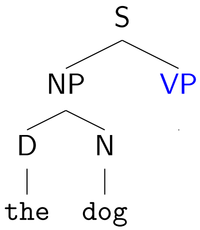
Overview

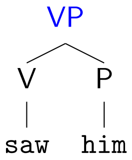
Deriving Z and Y

Design functions ψ and ϕ:

ψ maps any outside tree to a vector of length d'

ϕ 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 $t \Rightarrow$

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

Z and Y will be reduced dimensional representations of ψ and ϕ.

Reducing Dimensions via a Singular Value Decomposition

Have M samples of a node with non-terminal a. At sample i, $o^{(i)}$ is the outside tree rooted at a and $t^{(i)}$ is the inside tree rooted at a.

- Compute a matrix $\hat{\Omega}^a \in \mathbb{R}^{d \times d'}$ with entries

$$[\hat{\Omega}^a]_{j,k} = \frac{1}{M} \sum_{i=1}^{M} \phi_j(t^{(i)}) \psi_k(o^{(i)})$$

Reducing Dimensions via a Singular Value Decomposition

Have M samples of a node with non-terminal a. At sample i, $o^{(i)}$ is the outside tree rooted at a and $t^{(i)}$ is the inside tree rooted at a.

- Compute a matrix $\hat{\Omega}^a \in \mathbb{R}^{d \times d'}$ with entries

$$[\hat{\Omega}^a]_{j,k} = \frac{1}{M} \sum_{i=1}^{M} \phi_j(t^{(i)}) \psi_k(o^{(i)})$$

- An SVD:

$$\underbrace{\hat{\Omega}^a}_{d \times d'} \approx \underbrace{U^a}_{d \times m} \underbrace{\Sigma^a}_{m \times m} \underbrace{(V^a)^T}_{m \times d'}$$

Reducing Dimensions via a Singular Value Decomposition

Have M samples of a node with non-terminal a. At sample i, $o^{(i)}$ is the outside tree rooted at a and $t^{(i)}$ is the inside tree rooted at a.

- Compute a matrix $\hat{\Omega}^a \in \mathbb{R}^{d \times d'}$ with entries

$$[\hat{\Omega}^a]_{j,k} = \frac{1}{M} \sum_{i=1}^{M} \phi_j(t^{(i)}) \psi_k(o^{(i)})$$

- An SVD:

$$\underbrace{\hat{\Omega}^a}_{d \times d'} \approx \underbrace{U^a}_{d \times m} \underbrace{\Sigma^a}_{m \times m} \underbrace{(V^a)^T}_{m \times d'}$$

- Projection:

$$Y(t^{(i)}) = \underbrace{(U^a)^T}_{m \times d} \underbrace{\phi(t^{(i)})}_{d \times 1} \in \mathbb{R}^m$$

$$Z(o^{(i)}) = \underbrace{(\Sigma^a)^{-1}}_{m \times m} \underbrace{(V^a)^T}_{m \times d'} \underbrace{\psi(o^{(i)})}_{d' \times 1} \in \mathbb{R}^m$$

Consistency and Sample Complexity

If the $d \times d'$ matrix

$$\Omega^a = \mathbf{E}[\phi(T)\psi(O)^T | \text{label} = a]$$

has rank m, these projections yield consistent parameter estimates with high probability. The required number of samples grows polynomially in

- m: the number of hidden states
- $\log R$: where R is the number of rules
- Spectral properties of the grammar (e.g., $\max \frac{1}{\sigma^a}$ where σ^a is the m^{th} largest singular value of Ω^a)

A Summary of the Algorithm

1. Design feature functions ϕ and ψ for inside and outside trees.
2. Use SVD to compute vectors

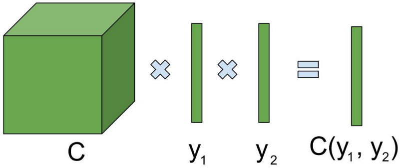
   $Y(t) \in \mathbb{R}^m$ for inside trees
   $Z(o) \in \mathbb{R}^m$ for outside trees
3. Estimate the parameters $\hat{t}$, $\hat{q}$, and $\hat{\pi}$ from the training data.
4. Parse a new sentence by computing its marginals with these parameters.

Overview

Tensor Definition

A third-order tensor $C \in \mathbb{R}^{m \times m \times m}$ is a set of m^3 values $[C]_{j,k,l}$. It can be viewed as a function $C : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ that takes two vectors $y_1, y_2 \in \mathbb{R}^m$ as input and returns a vector $C(y_1, y_2) \in \mathbb{R}^m$ as output. The output vector has entries

$$[C(y_1, y_2)]_h = \sum_{h_2, h_3} \left([C]_{h,h_2,h_3} \times [y_1]_{h_2} \times [y_2]_{h_3} \right)$$



Tensor Form of the Parameters

For each non-terminal a, define a vector $\pi^a \in \mathbb{R}^m$ with entries

$$[\pi^a]_h = \pi(a^h)$$

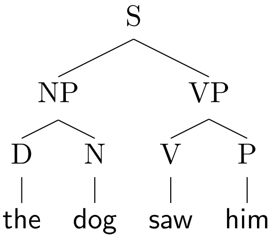
For each rule $a \to x$, define a vector $q_{a\to x} \in \mathbb{R}^m$ with entries

$$[q_{a\to x}]_h = q_{a\to x}(a^h \to x|a^h)$$

For each rule $a \to b\ c$, define a tensor $T^{a\to b\ c} \in \mathbb{R}^{m\times m\times m}$ with entries

$$[T^{a\to b\ c}]_{h_1,h_2,h_3} = t(a^{h_1} \to b^{h_2}\ c^{h_3}|a^{h_1})$$

Dynamic Programming in Tensor Form



$$T^{\mathrm{S}\to\mathrm{NP}\,\mathrm{VP}}(T^{\mathrm{NP}\to\mathrm{D}\,\mathrm{N}}(q_{\mathrm{D}\to\mathrm{the}}, q_{\mathrm{N}\to\mathrm{dog}}), T^{\mathrm{VP}\to\mathrm{V}\,\mathrm{P}}(q_{\mathrm{V}\to\mathrm{saw}}, q_{\mathrm{P}\to\mathrm{him}}))\ \pi^{\mathrm{S}}$$

$$|||$$

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

Thought Experiment

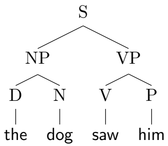
- We want the parameters (in tensor form)

$$\pi^a \in \mathbb{R}^m$$
$$q_{a \to x} \in \mathbb{R}^m$$
$$T^{a \to b\ c}(y_2, y_3) \in \mathbb{R}^m$$

- What if we had an invertible matrix $G^a \in \mathbb{R}^{m \times m}$ for every non-terminal a?
- And what if we had instead

$$c^a = G^a \pi^a$$
$$c_{a \to x} = q_{a \to x}(G^a)^{-1}$$
$$C^{a \to b\ c}(y_2, y_3) = T^{a \to b\ c}(y_2 G^b, y_3 G^c)(G^a)^{-1}$$

Cancellation of the Linear Operators



$$C^{\mathrm{S}\to\mathrm{NP\,VP}}(C^{\mathrm{NP}\to\mathrm{D\,N}}(c_{\mathrm{D}\to\mathrm{the}}, c_{\mathrm{N}\to\mathrm{dog}}), C^{\mathrm{VP}\to\mathrm{V\,P}}(c_{\mathrm{V}\to\mathrm{saw}}, c_{\mathrm{P}\to\mathrm{him}}))\ c^{\mathrm{S}}$$

$$|||$$

$$T^{\mathrm{S}\to\mathrm{NP\,VP}}(T^{\mathrm{NP}\to\mathrm{D\,N}}(q_{\mathrm{D}\to\mathrm{the}}(G^{\mathrm{D}})^{-1}G^{\mathrm{D}}, q_{\mathrm{N}\to\mathrm{dog}}(G^{\mathrm{N}})^{-1}G^{\mathrm{N}})(G^{\mathrm{NP}})^{-1}G^{\mathrm{NP}},$$
$$T^{\mathrm{VP}\to\mathrm{V\,P}}(q_{\mathrm{V}\to\mathrm{saw}}(G^{\mathrm{V}})^{-1}G^{\mathrm{V}}, q_{\mathrm{P}\to\mathrm{him}}(G^{\mathrm{P}})^{-1}G^{\mathrm{P}})(G^{\mathrm{VP}})^{-1}G^{\mathrm{VP}})(G^{\mathrm{S}})^{-1}G^{\mathrm{S}}\pi^{\mathrm{S}}$$

$$|||$$

$$T^{\mathrm{S}\to\mathrm{NP\,VP}}(T^{\mathrm{NP}\to\mathrm{D\,N}}(q_{\mathrm{D}\to\mathrm{the}}, q_{\mathrm{N}\to\mathrm{dog}}), T^{\mathrm{VP}\to\mathrm{V\,P}}(q_{\mathrm{V}\to\mathrm{saw}}, q_{\mathrm{P}\to\mathrm{him}}))\ \pi^{\mathrm{S}}$$

$$|||$$

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

Estimation Guarantees

- Basic argument: If Ω^a has rank m, parameters $\hat{C}^{a\rightarrow b\ c}$, $\hat{c}_{a\rightarrow x}$, and $\hat{c}^a$ converge to

$$
\begin{aligned}
C^{a\rightarrow b\ c}(y_2, y_3) &= T^{a\rightarrow b\ c}(y_2 G^b, y_3 G^c)(G^a)^{-1} \\
c_{a\rightarrow x} &= q_{a\rightarrow x}(G^a)^{-1} \\
c^a &= G^a \pi^a
\end{aligned}
$$

  for some G^a that is invertible.
- Because the parameters converge, the estimated distribution $\hat{p}(\text{tree})$ converges to the true distribution $p(\text{tree})$, and the estimated marginal $\hat{\mu}(a, i, j)$ converges to the true marginal $\mu(a, i, j)$.

Preliminary Experiments

The algorithm is much faster than EM.

- SVD: modern algorithms are very efficient
- Parameter calculation: takes less time than a single iteration of EM

A straightforward implementation lags behind EM by about 1-2% in F1 measure.

Current work: experiments focused on understanding the method and improving performance

Summary

We presented a spectral algorithm that yields a consistent estimator for L-PCFGs

- Simple and efficient: SVD and standard matrix operations

Future work includes

- Pushing the empirical side of the algorithm
- Deriving spectral algorithms for other latent-variable models in NLP