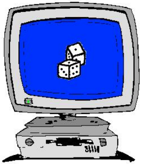
Bayesian Inference for PCFGs via Markov Chain Monte Carlo

Mark Johnson, Thomas L.Griffiths and Sharon Goldwater



Guanyi Chen

University of Edinburgh

March 18th, 2016

Overview

1. Introduction
   - Probabilistic context-free grammars (PCFGs)
   - Dirichlet Priors
   - Markov chain Monte Carlo
2. Samplers
   - A Gibbs sampler
   - A Hastings sampler
3. Application and Result

Overview

1. Introduction
   - Probabilistic context-free grammars (PCFGs)
   - Dirichlet Priors
   - Markov chain Monte Carlo
2. Samplers
   - A Gibbs sampler
   - A Hastings sampler
3. Application and Result

PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

Re-writing Rules:

S $\rightarrow$ NP VP

NP $\rightarrow$ D N | N

VP $\rightarrow$ V

N $\rightarrow$ dog | man

D $\rightarrow$ a | the | an

V $\rightarrow$ sleeps | runs

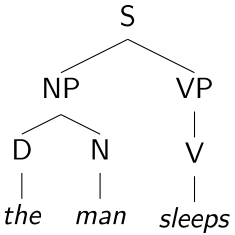
PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

Re-writing Rules:

S → NP VP
NP → D N | N
VP → V
N → dog | man
D → a | the | an
V → sleeps | runs

Parse Tree:



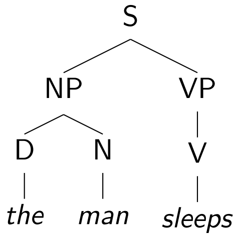
PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

Re-writing Rules:

S $\rightarrow$ NP VP
NP $\rightarrow$ D N | N
VP $\rightarrow$ V
N $\rightarrow$ dog | man
D $\rightarrow$ a | the | an
V $\rightarrow$ sleeps | runs

Parse Tree:



from Treebanks:

$$\begin{aligned} \theta_{\alpha\rightarrow\beta} &= p_{ML}(\alpha \rightarrow \beta) \\ &= \frac{C(\alpha \rightarrow \beta)}{C(\alpha)} \end{aligned}$$

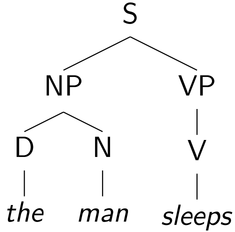
PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

Re-writing Rules:

- S $\rightarrow$ NP VP (1)
- NP $\rightarrow$ D N (0.2)| N (0.8)
- VP $\rightarrow$ V (1)
- N $\rightarrow$ dog (0.3)| man (0.7)
- D $\rightarrow$ a (0.3)| the (0.5)| an (0.3)
- V $\rightarrow$ sleeps (0.6)| runs (0.4)

Parse Tree:



from Treebanks:

$$\theta_{\alpha\rightarrow\beta} = p_{ML}(\alpha \rightarrow \beta) = \frac{C(\alpha \rightarrow \beta)}{C(\alpha)}$$

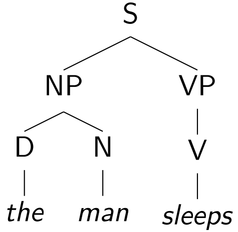
PCFGs [Collins, 2013]

$$G = (T, N, S, R)$$

Re-writing Rules:

S $\rightarrow$ NP VP (1)
NP $\rightarrow$ D N (0.2)| N (0.8)
VP $\rightarrow$ V (1)
N $\rightarrow$ dog (0.3)| man (0.7)
D $\rightarrow$ a (0.3)| the (0.5)| an (0.3)
V $\rightarrow$ sleeps (0.6)| runs (0.4)

Parse Tree:



from Treebanks:

$$\theta_{\alpha\rightarrow\beta} = p_{ML}(\alpha \rightarrow \beta) = \frac{C(\alpha \rightarrow \beta)}{C(\alpha)}$$

use CKY to maximize:

$$p_G(t|\theta) = \prod_{r\in R} \theta_r^{f_r(t)}$$

Bayesian Inference for PCFGs

Goal: Given a corpus of string (terminals) $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$, generated by known CFGs G to infer the rule probability distribution $\boldsymbol{\theta}$ that best describe the corpus.

Bayesian Inference for PCFGs

Goal: Given a corpus of string (terminals) $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$, generated by known CFGs G to infer the rule probability distribution $\boldsymbol{\theta}$ that best describe the corpus.

Maximum likelihood: Inside-Outside Algorithm (EM procedure) [Lari and Young, 1990]

Bayesian Inference for PCFGs

Goal: Given a corpus of string (terminals) $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$, generated by known CFGs G to infer the rule probability distribution $\boldsymbol{\theta}$ that best describe the corpus.

Maximum likelihood: Inside-Outside Algorithm (EM procedure) [Lari and Young, 1990]

Bayesian inference:

$$p(\boldsymbol{\theta}|\boldsymbol{w}) \propto p_G(\boldsymbol{w}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

Bayesian Inference for PCFGs

Goal: Given a corpus of string (terminals) $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)$, generated by known CFGs G to infer the rule probability distribution $\boldsymbol{\theta}$ that best describe the corpus.

Maximum likelihood: Inside-Outside Algorithm (EM procedure) [Lari and Young, 1990]

Bayesian inference:

$$p(\boldsymbol{\theta}|\boldsymbol{w}) \propto p_G(\boldsymbol{w}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

$$p(\boldsymbol{t}, \boldsymbol{\theta}|\boldsymbol{w}) \propto p(\boldsymbol{w}|\boldsymbol{t})p(\boldsymbol{t}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

Dirichlet Priors $p(\boldsymbol{\theta})$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{w})}_{\textit{posterior}} \propto \underbrace{p_G(\boldsymbol{w}|\boldsymbol{\theta})}_{\textit{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\textit{prior}}$$

Dirichlet Priors $p(\boldsymbol{\theta})$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{t})}_{\textit{posterior}} \propto \underbrace{p_G(\boldsymbol{t}|\boldsymbol{\theta})}_{\textit{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\textit{prior}}$$

Dirichlet Priors $p(\boldsymbol{\theta})$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{t})}_{\textit{posterior}} \propto \underbrace{p_G(\boldsymbol{t}|\boldsymbol{\theta})}_{\textit{likelihood}}\underbrace{p(\boldsymbol{\theta})}_{\textit{prior}}$$

Suppose A is a non-terminal at the left hand side, then all the productions $\theta_{A\to\beta}$ has a Dirichlet prior $\alpha_{A\to\beta}$:

$$p_{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{A\in N} p_{Dir}(\theta_A|\alpha_A) \propto \prod_{r\in R} \theta^{\alpha_r - 1}$$

Dirichlet Priors $p(\boldsymbol{\theta})$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{t})}_{\text{posterior}} \propto \underbrace{p_G(\boldsymbol{t}|\boldsymbol{\theta})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\text{prior}}$$

Suppose A is a non-terminal at the left hand side, then all the productions $\theta_{A\to\beta}$ has a Dirichlet prior $\alpha_{A\to\beta}$:

$$p_{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{A\in N} p_{Dir}(\theta_A|\alpha_A) \propto \prod_{r\in R} \theta^{\alpha_r - 1}$$

They are conjugate to the distribution over trees, thus the posterior is also a Dirichlet distribution:

$$p_G(\boldsymbol{\theta}|\boldsymbol{t},\boldsymbol{\alpha}) \propto \prod_{r\in R} \theta^{f_r(\boldsymbol{t})+\alpha_r-1} = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t})+\boldsymbol{\alpha})$$

Dirichlet Priors $p(\boldsymbol{\theta})$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{t})}_{\textit{posterior}} \propto \underbrace{p_G(\boldsymbol{t}|\boldsymbol{\theta})}_{\textit{likelihood}} \underbrace{p(\boldsymbol{\theta})}_{\textit{prior}}$$

Suppose A is a non-terminal at the left hand side, then all the productions $\theta_{A\to\beta}$ has a Dirichlet prior $\alpha_{A\to\beta}$:

$$p_{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \prod_{A\in N} p_{Dir}(\theta_A|\alpha_A) \propto \prod_{r\in R} \theta^{\alpha_r - 1}$$

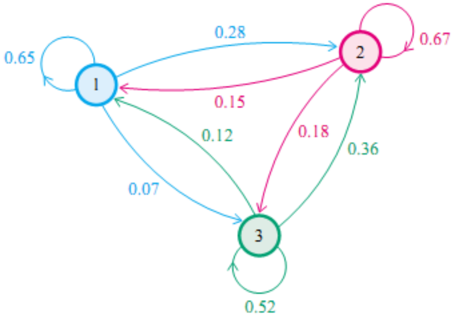
They are conjugate to the distribution over trees, thus the posterior is also a Dirichlet distribution:

$$p_G(\boldsymbol{\theta}|\boldsymbol{t},\boldsymbol{\alpha}) \propto \prod_{r\in R} \theta^{f_r(\boldsymbol{t})+\alpha_r-1} = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t})+\boldsymbol{\alpha})$$

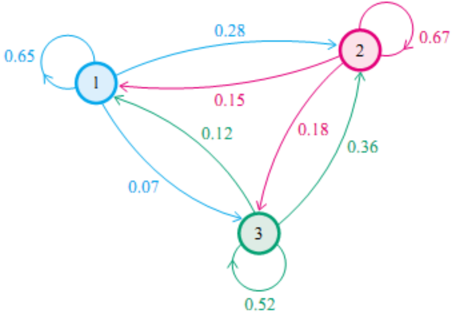
However, $\boldsymbol{t}$ is hidden, we can only observe terminal strings $\boldsymbol{w}$!!

Incredible Markov chain [rickjin, 2013]

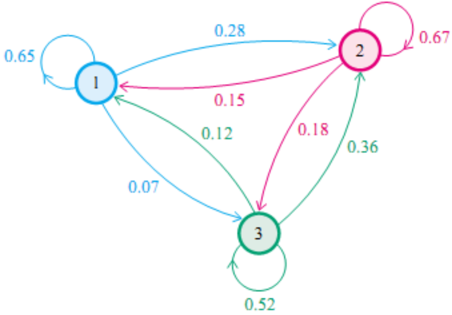
Incredible Markov chain [rickjin, 2013]



Incredible Markov chain [rickjin, 2013]

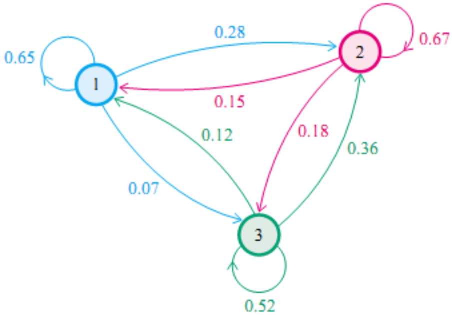


iter	s_1	s_2	s_3
π_0	0.21	0.68	0.11
π_1	0.25	0.55	0.19
π_2	0.27	0.51	0.21
π_3	0.28	0.50	0.23
π_4	0.29	0.49	0.23
π_5	0.29	0.49	0.23
π_6	0.29	0.49	0.23
π_7	0.29	0.49	0.23
...	...	...	...



iter	s_1	s_2	s_3
π_0	0.75	0.15	0.1
π_1	0.52	0.35	0.13
π_2	0.41	0.42	0.17
π_3	0.35	0.46	0.20
π_4	0.32	0.48	0.21
π_5	0.30	0.48	0.22
π_6	0.29	0.49	0.23
π_7	0.29	0.49	0.23
...	...	...	...

Incredible Markov chain [rickjin, 2013]



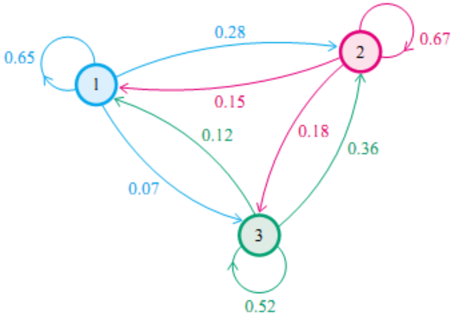
- Sampling:

$$s_{t+1} \sim q(s_t \rightarrow s_{t+1})$$

- Find a transition matrix such that the stationary distribution is the distribution we want, then sample on it
- The expectation of these samples will be the estimation

$$\mathbb{E}[\theta] \approx \frac{1}{\ell} \sum_{i=1}^{\ell} \theta_i$$

Incredible Markov chain [rickjin, 2013]



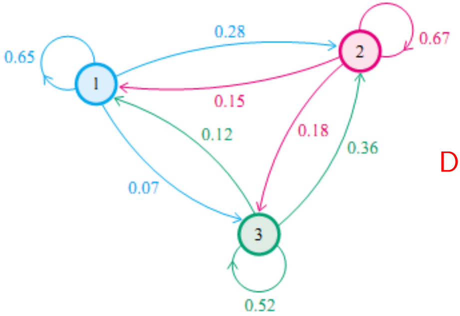
- Sampling:

$$s_{t+1} \sim q(s_t \to s_{t+1})$$

- Find a transition matrix such that the stationary distribution is the distribution we want, then sample on it
- The expectation of these samples will be the estimation

$$\mathbb{E}[\theta] \approx \frac{1}{\ell} \sum_{i=1}^{\ell} \theta_i$$

Incredible Markov chain [rickjin, 2013]



Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

[1] detailed balance condition is a sufficient but unnecessary condition

Just a test

```
>> p = [0.65 0.28 0.07; 0.15 0.67 0.18; 0.12 0.36 0.52];
>> a1 = [1, 0, 0];
>> a2 = [0.7, 0.2, 0.1];
>> b = a1 * p^100
b =
       0.2865      0.48852      0.22498
>> b = a2 * p^100
b =
       0.2865      0.48852      0.22498
```

Overview

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \rightarrow s') = \pi(s')q(s' \rightarrow s)$$

Suppose we have two points $A(x_1, y_1)$ and $B(x_1, y_2)$:

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

Suppose we have two points $A(x_1, y_1)$ and $B(x_1, y_2)$:

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$

$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

Suppose we have two points $A(x_1, y_1)$ and $B(x_1, y_2)$:

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$

$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

$$p(x_1, y_1)p(y_2|x_1) = p(x_1, y_2)p(y_1|x_1)$$

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

Suppose we have two points $A(x_1, y_1)$ and $B(x_1, y_2)$:

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$

$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

$$\pi(A)q(y_2|x_1) = \pi(B)q(y_1|x_1)$$

Gibbs Sampling 1

Detailed Balance Condition:

$$\pi(s)q(s \to s') = \pi(s')q(s' \to s)$$

Suppose we have two points $A(x_1, y_1)$ and $B(x_1, y_2)$:

$$p(x_1, y_1)p(y_2|x_1) = p(x_1)p(y_1|x_1)p(y_2|x_1)$$

$$p(x_1, y_2)p(y_1|x_1) = p(x_1)p(y_2|x_1)p(y_1|x_1)$$

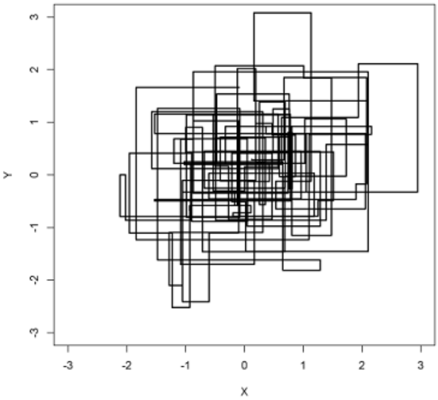
$$\pi(A)q(y_2|x_1) = \pi(B)q(y_1|x_1)$$

Gibbs Sampling 2

Sampling each component of the state conditioned on the current value of all other variables

Gibbs Sampling 2

Sampling each component of the state conditioned on the current value of all other variables



Gibbs Sampler for $(\boldsymbol{t}, \boldsymbol{\theta})$

Update each component by resampling conditioned on values for other components

Gibbs Sampler for $(\boldsymbol{t}, \boldsymbol{\theta})$

Update each component by resampling conditioned on values for other components

$$p(\boldsymbol{t}|\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} p(t_i|w_i, \theta)$$

$$p(\boldsymbol{\theta}|\boldsymbol{t}, \boldsymbol{w}, \boldsymbol{\alpha}) = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t}) + \boldsymbol{\alpha})$$

Gibbs Sampler for $(\boldsymbol{t}, \theta)$

Update each component by resampling conditioned on values for other components

$$p(\boldsymbol{t}|\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} p(t_i|w_i, \theta)$$

$$p(\boldsymbol{\theta}|\boldsymbol{t}, \boldsymbol{w}, \boldsymbol{\alpha}) = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t}) + \boldsymbol{\alpha})$$

- Gibbs sampler is highly parallelizable

Gibbs Sampler for $(\boldsymbol{t}, \theta)$

Update each component by resampling conditioned on values for other components

$$p(\boldsymbol{t}|\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} p(t_i|w_i, \theta)$$

$$p(\boldsymbol{\theta}|\boldsymbol{t}, \boldsymbol{w}, \boldsymbol{\alpha}) = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t}) + \boldsymbol{\alpha})$$

- Gibbs sampler is highly parallelizable
- Given $\boldsymbol{\theta}$, trees are independent, thus $\boldsymbol{t}$ can be sampled in parallel

Gibbs Sampler for $(\boldsymbol{t}, \theta)$

Update each component by resampling conditioned on values for other components

$$p(\boldsymbol{t}|\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} p(t_i|w_i, \theta)$$

$$p(\boldsymbol{\theta}|\boldsymbol{t}, \boldsymbol{w}, \boldsymbol{\alpha}) = p_{Dir}(\boldsymbol{\theta}|\boldsymbol{f}(\boldsymbol{t}) + \boldsymbol{\alpha})$$

- Gibbs sampler is highly parallelizable
- Given $\boldsymbol{\theta}$, trees are independent, thus $\boldsymbol{t}$ can be sampled in parallel
- More efficiently sampling from $p(t|w, \theta)$: use dynamic programming, i.e. inside and outside algorithm

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

Directly sampling on the trees:

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

Directly sampling on the trees: marginalizing out $\boldsymbol{\theta}$

$$p(\boldsymbol{t}|\alpha) = \int_{\Delta} p(\boldsymbol{t}|\theta)p(\theta|\alpha)$$

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

Directly sampling on the trees: marginalizing out $\boldsymbol{\theta}$

$$p(\boldsymbol{t}|\alpha) = \int_{\Delta} p(\boldsymbol{t}|\theta)p(\theta|\alpha)$$

Components of states are now the trees t_i:

$$p(t_i|\boldsymbol{t}_{\setminus i}, \alpha) = \frac{p(t_i|\boldsymbol{t}_i, \alpha)}{p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)}$$

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

Directly sampling on the trees: marginalizing out $\boldsymbol{\theta}$

$$p(\boldsymbol{t}|\alpha) = \int_{\Delta} p(\boldsymbol{t}|\theta)p(\theta|\alpha)$$

Components of states are now the trees t_i:

$$p(t_i|\boldsymbol{t}_{\setminus i}, \alpha) = \frac{p(t_i|\boldsymbol{t}_i, \alpha)}{p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)}$$

New Gibbs Sampler:

$$p(t_i|w_i, \boldsymbol{t}_{\setminus i}, \alpha) = \frac{p(w_i|t_i)p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)}{p(w_i|\boldsymbol{t}_{\setminus i}, \alpha)}$$

A Problem of Gibbs Sampler

Gibbs Sampler

For each sample of $\boldsymbol{\theta}$, the corpus $\boldsymbol{w}$ should be rephrasing

Directly sampling on the trees: marginalizing out $\boldsymbol{\theta}$

$$p(\boldsymbol{t}|\alpha) = \int_{\Delta} p(\boldsymbol{t}|\theta)p(\theta|\alpha)$$

Components of states are now the trees t_i:

$$p(t_i|\boldsymbol{t}_{\setminus i}, \alpha) = \frac{p(t_i|\boldsymbol{t}_i, \alpha)}{p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)}$$

New Gibbs Sampler:

$$p(t_i|w_i, \boldsymbol{t}_{\setminus i}, \alpha) = \frac{p(w_i|t_i)p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)}{p(w_i|\boldsymbol{t}_{\setminus i}, \alpha)}$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s) \neq \pi(s')Q(s; s')$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s)\alpha(s'; s) = \pi(s')Q(s; s')\alpha(s; s')$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s)\alpha(s'; s) = \pi(s')Q(s; s')\alpha(s; s')$$

where α is the acceptance rate:

$$\alpha(s'; s) = \pi(s')Q(s; s')$$

$$\alpha(s; s') = \pi(s)Q(s'; s)$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s)\alpha(s'; s) = \pi(s')Q(s; s')\alpha(s; s')$$

where α is the acceptance rate:

$$\alpha(s'; s) = \pi(s')Q(s; s')$$

$$\alpha(s; s') = \pi(s)Q(s'; s)$$

$$\pi(s)Q(s'; s) \times 0.1 = \pi(s')Q(s; s') \times 0.2$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s)\alpha(s'; s) = \pi(s')Q(s; s')\alpha(s; s')$$

where α is the acceptance rate:

$$\alpha(s'; s) = \pi(s')Q(s; s')$$

$$\alpha(s; s') = \pi(s)Q(s'; s)$$

$$\pi(s)Q(s'; s) \times 0.5 = \pi(s')Q(s; s') \times 1$$

Metropolis-Hasting

Suppose a User-Specified Proposal Distribution $Q(s'; s)$:

$$\pi(s)Q(s'; s)\alpha(s'; s) = \pi(s')Q(s; s')\alpha(s; s')$$

where α is the acceptance rate:

$$\alpha(s'; s) = \pi(s')Q(s; s')$$

$$\alpha(s; s') = \pi(s)Q(s'; s)$$

$$\alpha(s'; s) = \min\left\{1, \frac{\pi(s')Q(s; s')}{\pi(s)Q(s'; s)}\right\}$$

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

- Randomly choose index i of tree to re-sample

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

- Randomly choose index i of tree to re-sample
- Compute the PCFG Probability to be used in proposal distribution

$$\hat{\theta}_{A\rightarrow\beta} = \mathbb{E}[\theta_{A\rightarrow\beta}|\boldsymbol{t}_{\setminus i}, \alpha] = \frac{f_{A\rightarrow\beta}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\rightarrow\beta}}{\sum_{A\rightarrow\beta'\in R_A} f_{A\rightarrow\beta'}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\rightarrow\beta'}}$$

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

- Randomly choose index i of tree to re-sample
- Compute the PCFG Probability to be used in proposal distribution

$$\hat{\theta}_{A\to\beta} = \mathbb{E}[\theta_{A\to\beta}|\boldsymbol{t}_{\backslash i}, \alpha] = \frac{f_{A\to\beta}(\boldsymbol{t}_{\backslash i}) + \alpha_{A\to\beta}}{\sum_{A\to\beta'\in R_A} f_{A\to\beta'}(\boldsymbol{t}_{\backslash i}) + \alpha_{A\to\beta'}}$$

- Sample a proposal tree: $t_i' \sim p(t_i|w_i, \hat{\theta})$

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

- Randomly choose index i of tree to re-sample
- Compute the PCFG Probability to be used in proposal distribution

$$\hat{\theta}_{A\to\beta} = \mathbb{E}[\theta_{A\to\beta}|\boldsymbol{t}_{\setminus i}, \alpha] = \frac{f_{A\to\beta}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\to\beta}}{\sum_{A\to\beta' \in R_A} f_{A\to\beta'}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\to\beta'}}$$

- Sample a proposal tree: $t_i' \sim p(t_i|w_i, \hat{\theta})$
- Compute the acceptance probability:

$$A(t_i', t_i) = \min\left\{1, \frac{p(t_i'|\boldsymbol{t}_{\setminus i}, \alpha)p(t_i|w_i, \hat{\theta})}{p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)p(t_i'|w_i, \hat{\theta})}\right\}$$

A Hasting Sampler for $p(\boldsymbol{t}|\boldsymbol{w}, \alpha)$

Use the posterior $p(\boldsymbol{t}|\boldsymbol{w}, \hat{\theta})$ as the proposal distribution

- Randomly choose index i of tree to re-sample
- Compute the PCFG Probability to be used in proposal distribution

$$\hat{\theta}_{A\to\beta} = \mathbb{E}[\theta_{A\to\beta}|\boldsymbol{t}_{\setminus i}, \alpha] = \frac{f_{A\to\beta}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\to\beta}}{\sum_{A\to\beta'\in R_A} f_{A\to\beta'}(\boldsymbol{t}_{\setminus i}) + \alpha_{A\to\beta'}}$$

- Sample a proposal tree: $t_i' \sim p(t_i|w_i, \hat{\theta})$
- Compute the acceptance probability:

$$A(t_i', t_i) = \min\left\{1, \frac{p(t_i'|\boldsymbol{t}_{\setminus i}, \alpha)p(t_i|w_i, \hat{\theta})}{p(t_i|\boldsymbol{t}_{\setminus i}, \alpha)p(t_i'|w_i, \hat{\theta})}\right\}$$

- Choose a random number $x \in$ uniform$[0, 1]$, to determine whether accept or not

Overview

1. Introduction
   - Probabilistic context-free grammars (PCFGs)
   - Dirichlet Priors
   - Markov chain Monte Carlo
2. Samplers
   - A Gibbs sampler
   - A Hastings sampler
3. Application and Result

Inferring Sparse Grammar

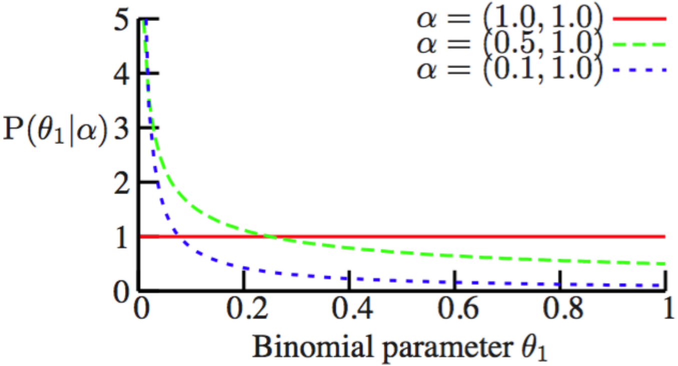
Performs poorly on inferring the PCFG as Inside-outside algorithm:

- Simple PCFGs are not accurate models of English syntactic structure
- Ignore a wide variety of lexical and syntactic dependencies in natural language

Inferring Sparse Grammar

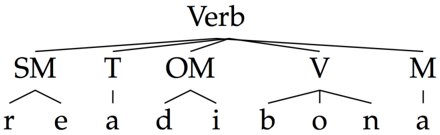
Performs poorly on inferring the PCFG as Inside-outside algorithm:

- Simple PCFGs are not accurate models of English syntactic structure
- Ignore a wide variety of lexical and syntactic dependencies in natural language



Unsupervised Morphological Analysis of Sesotho

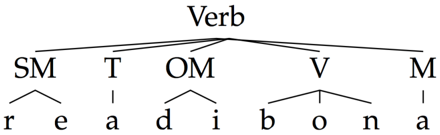
- Sesotho is a morphology rich language



Word $\rightarrow$ V
Word $\rightarrow$ V M
Word $\rightarrow$ SM V M
Word $\rightarrow$ SM T V M
Word $\rightarrow$ SM T OM V M

Unsupervised Morphological Analysis of Sesotho

- Sesotho is a morphology rich language

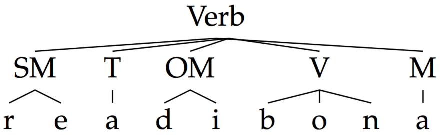


Word $\rightarrow$ V
Word $\rightarrow$ V M
Word $\rightarrow$ SM V M
Word $\rightarrow$ SM T V M
Word $\rightarrow$ SM T OM V M

- expanding the pre-terminals to each of the contiguous substrings of any verb in corpus, producing a grammar with 81,755 productions in all

Unsupervised Morphological Analysis of Sesotho

- Sesotho is a morphology rich language



Word $\rightarrow$ V
Word $\rightarrow$ V M
Word $\rightarrow$ SM V M
Word $\rightarrow$ SM T V M
Word $\rightarrow$ SM T OM V M

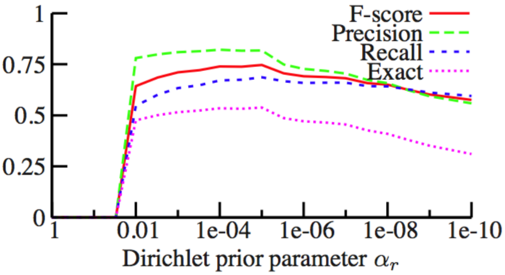
- expanding the pre-terminals to each of the contiguous substrings of any verb in corpus, producing a grammar with 81,755 productions in all
- Tested on maximum likelihood (IO), MAP (IO) and a Hasting Sampler

Results

- Maximum Likelihood learn a "Saturated" grammar: every word has its own production and $\theta_{\mathsf{Word} \to \mathsf{V}} = 1$

Results

- Maximum Likelihood learn a "Saturated" grammar: every word has its own production and $\theta_{\text{Word}\rightarrow\text{V}} = 1$
- Hasting Sampler: non-trivial structure emerges $\alpha < 0.01$



Problem of MAP and Further work

- EM Re-estimate θ in M-step

$$\theta_r^{(t+1)} \propto \mathbb{E}[f_r | \boldsymbol{w}, \theta^{(t)}]$$

Problem of MAP and Further work

- EM Re-estimate θ in M-step

$$\theta_r^{(t+1)} \propto \mathbb{E}[f_r|\boldsymbol{w}, \theta^{(t)}]$$

- Bayesian estimation for θ:

$$\theta_r^{(t+1)} \propto \max(0, \mathbb{E}[f_r|\boldsymbol{w}, \theta^{(t)}] + \alpha_r - 1)$$

Problem of MAP and Further work

- EM Re-estimate θ in M-step

$$\theta_r^{(t+1)} \propto \mathbb{E}[f_r | \boldsymbol{w}, \theta^{(t)}]$$

- Bayesian estimation for θ:

$$\theta_r^{(t+1)} \propto \max(0, \mathbb{E}[f_r | \boldsymbol{w}, \theta^{(t)}] + \alpha_r - 1)$$

- $\theta_r^{(t+1)} = 0$, then some input string failed to parse

Problem of MAP and Further work

- EM Re-estimate θ in M-step

$$\theta_r^{(t+1)} \propto \mathbb{E}[f_r|\boldsymbol{w}, \theta^{(t)}]$$

- Bayesian estimation for θ:

$$\theta_r^{(t+1)} \propto \max(0, \mathbb{E}[f_r|\boldsymbol{w}, \theta^{(t)}] + \alpha_r - 1)$$

- $\theta_r^{(t+1)} = 0$, then some input string failed to parse
- Variational Bayes may solve this [Kurihara and Sato, 2006]

Summary

- Two samplers for inferring PCFGs

Summary

- Two samplers for inferring PCFGs
- Unsupervised morphological analysis

Summary

- Two samplers for inferring PCFGs
- Unsupervised morphological analysis
- A Bayesian approach is more flexible than maximum likelihood

Summary

- Two samplers for inferring PCFGs
- Unsupervised morphological analysis
- A Bayesian approach is more flexible than maximum likelihood
- Provide essential building blocks for more complex models


Q
&
A

Supplementary Materials

1. Michael Collins' Note on PCFGs:
   http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf
2. Lecture slides of MLPR for MCMC:
   http://www.inf.ed.ac.uk/teaching/courses/mlpr/2015/slides/13_mcmc.pdf
3. Tutorial about MCMC in NIPS 2015:
   http://research.microsoft.com/apps/video/?id=259575

References I

Collins, M. (2013).
Probabilistic context-free grammars (pcfgs).

Johnson, M., Griffiths, T. L., and Goldwater, S. (2007).
Bayesian inference for pcfgs via markov chain monte carlo.
In *HLT-NAACL*, pages 139–146.

Kurihara, K. and Sato, T. (2006).
Variational bayesian grammar induction for natural language.
In *Grammatical Inference: Algorithms and Applications*, pages 84–96. Springer.

Lari, K. and Young, S. J. (1990).
The estimation of stochastic context-free grammars using the inside-outside algorithm.
Computer speech & language, 4(1):35–56.

References II

rickjin (2013).
Math gossip of lda.