Sampling

for unsupervised language learning

Wilker Aziz

Universiteit van Amsterdam
w.aziz@uva.nl

February 23, 2015

Content

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules
- $p : R \to [0,1]$ probability mass function over rules

$$\sum_{\alpha} p(A \to \alpha) = 1, \forall A \in V$$

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules
- $p : R \to [0, 1]$ probability mass function over rules

$$\sum_{\alpha} p(A \to \alpha) = 1, \forall A \in V$$

Let $\mathbf{w} \in \Sigma^+$ be a sentence and $\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle$ a parse tree for $\mathbf{w}$

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules
- $p : R \to [0, 1]$ probability mass function over rules

$$\sum_{\alpha} p(A \to \alpha) = 1, \forall A \in V$$

Let $\mathbf{w} \in \Sigma^+$ be a sentence and $\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle$ a parse tree for $\mathbf{w}$

- $p_G(\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle) = \prod_{A \to \alpha \in \mathbf{t}} p(A \to \alpha)$

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules
- $p : R \to [0, 1]$ probability mass function over rules

$$\sum_{\alpha} p(A \to \alpha) = 1, \forall A \in V$$

Let $\mathbf{w} \in \Sigma^+$ be a sentence and $\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle$ a parse tree for $\mathbf{w}$

- $p_G(\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle) = \prod_{A \to \alpha \in \mathbf{t}} p(A \to \alpha)$
- $p_G(\mathbf{w}) = \sum_{\mathbf{t} \in \mathcal{T}(w)} p_G(\mathbf{t})$

Recap

A PCFG $G = \langle \Sigma, N, S, R, p \rangle$

- Σ terminal vocabulary
- N nonterminal vocabulary
- $S \in N$ start symbol
- $R = \{A \to \alpha : A \in V, \alpha \in (\Sigma \cup V)^+\}$ set of rules
- $p : R \to [0, 1]$ probability mass function over rules

$$\sum_{\alpha} p(A \to \alpha) = 1, \forall A \in V$$

Let $\mathbf{w} \in \Sigma^+$ be a sentence and $\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle$ a parse tree for $\mathbf{w}$

- $p_G(\mathbf{t} = \langle S \Rightarrow^* \mathbf{w} \rangle) = \prod_{A \to \alpha \in \mathbf{t}} p(A \to \alpha)$
- $p_G(\mathbf{w}) = \sum_{\mathbf{t} \in \mathcal{T}(w)} p_G(\mathbf{t})$

from here let's assume CNF

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0, 1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

[1] Remark: this means having one parameter per rule in the grammar!

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0,1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

Maximise likelihood of data $\mathcal{W}$ given model $\boldsymbol{\theta}$

[1] Remark: this means having one parameter per rule in the grammar!

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0,1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

Maximise likelihood of data $\mathcal{W}$ given model $\boldsymbol{\theta}$

$$p_G(\mathcal{W}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} p_G(\mathbf{w}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{w})} p_G(\mathbf{t}; \boldsymbol{\theta})$$

[1] Remark: this means having one parameter per rule in the grammar!

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0,1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

Maximise likelihood of data $\mathcal{W}$ given model $\boldsymbol{\theta}$

$$p_G(\mathcal{W}|\boldsymbol{\theta}) = \prod_{\mathbf{w}\in\mathcal{W}} p_G(\mathbf{w}|\boldsymbol{\theta}) = \prod_{\mathbf{w}\in\mathcal{W}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{w})} p_G(\mathbf{t};\boldsymbol{\theta})$$

E-step conditions on current model $\boldsymbol{\theta}$
$\langle f_r \rangle_p = \sum_{\mathbf{w}} \sum_{\mathbf{t}\in\mathcal{T}(\mathbf{w})} f_r(\mathbf{t}) p(\mathbf{t};\boldsymbol{\theta})$

[1] Remark: this means having one parameter per rule in the grammar!

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0,1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

Maximise likelihood of data $\mathcal{W}$ given model $\boldsymbol{\theta}$

$$p_G(\mathcal{W}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} p_G(\mathbf{w}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{w})} p_G(\mathbf{t}; \boldsymbol{\theta})$$

E-step conditions on current model $\boldsymbol{\theta}$
$\langle f_r \rangle_p = \sum_{\mathbf{w}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{w})} f_r(\mathbf{t}) p(\mathbf{t}; \boldsymbol{\theta})$

M-step updates model maximising the likelihood of the data
$\theta_{A \rightarrow \alpha} = \frac{\langle f_{A \rightarrow \alpha} \rangle_p}{\sum_{\alpha'} \langle f_{A \rightarrow \alpha'} \rangle_p}$

[1] Remark: this means having one parameter per rule in the grammar!

Inducing PCFGs with EM

Let us define the parameters $\boldsymbol{\theta} \in [0,1]^{|R|}$ where[1]
$p(r) = \theta_r$ where $r = A \rightarrow \alpha$ is a rule in R

Maximise likelihood of data $\mathcal{W}$ given model $\boldsymbol{\theta}$

$$p_G(\mathcal{W}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} p_G(\mathbf{w}|\boldsymbol{\theta}) = \prod_{\mathbf{w} \in \mathcal{W}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{w})} p_G(\mathbf{t}; \boldsymbol{\theta})$$

E-step conditions on current model $\boldsymbol{\theta}$
$\langle f_r \rangle_p = \sum_{\mathbf{w}} \sum_{\mathbf{t} \in \mathcal{T}(\mathbf{w})} f_r(\mathbf{t}) p(\mathbf{t}; \boldsymbol{\theta})$

M-step updates model maximising the likelihood of the data
$\theta_{A \rightarrow \alpha} = \frac{\langle f_{A \rightarrow \alpha} \rangle_p}{\sum_{\alpha'} \langle f_{A \rightarrow \alpha'} \rangle_p}$

Expectations: Inside-Outside dynamic program $O(|V|^3 |\mathbf{w}|^3)$

[1] Remark: this means having one parameter per rule in the grammar!

I don't see the problem!

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$
4. What about a complex posterior?

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

unlike MLE, in MAP inference the prior is not uniform

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$
4. What about a complex posterior?

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

unlike MLE, in MAP inference the prior is not uniform

What can we do?

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$
4. What about a complex posterior?

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

unlike MLE, in MAP inference the prior is not uniform

What can we do?

- reasoning over a representative subset of the parses

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$
4. What about a complex posterior?

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

unlike MLE, in MAP inference the prior is not uniform

What can we do?

- reasoning over a representative subset of the parses
  empirical distribution

I don't see the problem!

1. What if the grammar is too large?
   Inside-Outside $O(|V|^3 n^3)$
2. What about synchronous parsing
   Inside-Outside $O(|V|^3 n^3 m^3) \approx O(|V|^3 n^6)$
3. What about a more complex model?
   joint parsing and tagging $O(|V|^3 |T|^{3k} n^3)$
4. What about a complex posterior?

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

unlike MLE, in MAP inference the prior is not uniform

What can we do?

- reasoning over a representative subset of the parses
  empirical distribution
  sampling

Tasks

1. Draw samples from a distribution

$$\{x^{(i)} \sim p(x)\}_{i=1}^{N}$$

Tasks

1. Draw samples from a distribution

$$\{x^{(i)} \sim p(x)\}_{i=1}^{N}$$

2. Compute expectations under a distribution

$$\Phi = \langle \phi(x) \rangle_{p(x)} = \int_{\mathcal{X}} \phi(x) p(x) \mathrm{d}x$$

Tasks

1. Draw samples from a distribution

$$\{x^{(i)} \sim p(x)\}_{i=1}^{N}$$

2. Compute expectations under a distribution

$$\Phi = \langle \phi(x) \rangle_{p(x)} = \int_{\mathcal{X}} \phi(x) p(x) \mathrm{d}x$$

If we could solve 1

then $\hat{\Phi} \equiv \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})$

Robert and Casella [2004]

Monte Carlo estimates

Accuracy of an MC estimate is independent of dimensionality

$$\hat{\Phi} \equiv \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i} \left(\phi(x^{(i)}) - \Phi \right)^2$$

Monte Carlo estimates

Accuracy of an MC estimate is independent of dimensionality

$$\hat{\Phi} \equiv \frac{1}{N} \sum_{i=1}^{N} \phi(x^{(i)})$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i} \left(\phi(x^{(i)}) - \Phi \right)^2$$

However, it is **very hard** to sample from high dimensional spaces!

Sampling from chart

Given a string $\mathbf{w}$, assume we can build the chart $\mathcal{T}(\mathbf{w})$

- $\langle i, A, j \rangle$ where $A \in N$ and $0 \leq i < j \leq |\mathbf{w}|$ represents a chart cell

Sampling from chart

Given a string $\mathbf{w}$, assume we can build the chart $\mathcal{T}(\mathbf{w})$

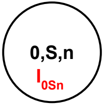
- $\langle i, A, j \rangle$ where $A \in N$ and $0 \leq i < j \leq |\mathbf{w}|$ represents a chart cell

then,

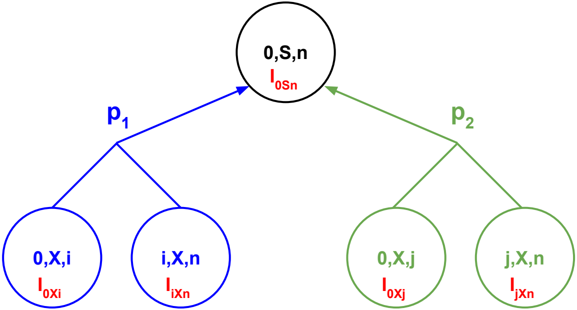
expectations trivial Inside-Outside run

sampling trivial random tree traversal from start symbol

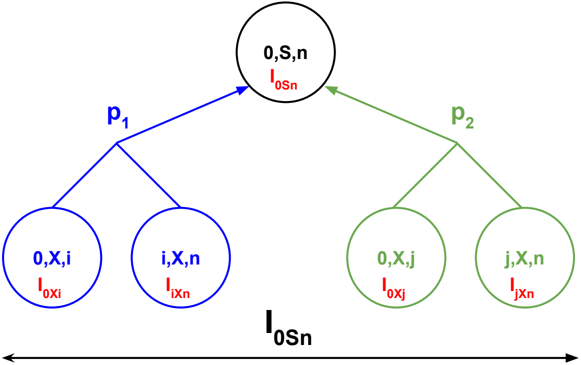
Top-down sampling illustration



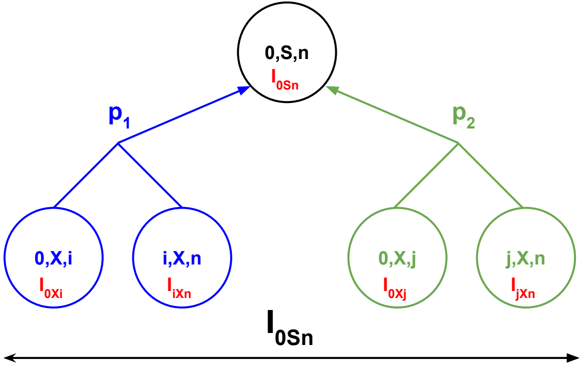
Top-down sampling illustration



Top-down sampling illustration

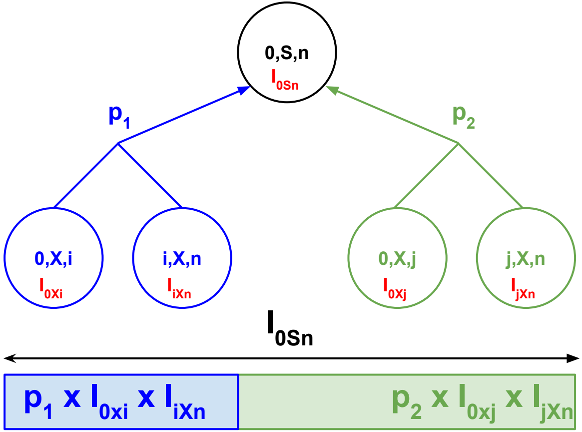


Top-down sampling illustration

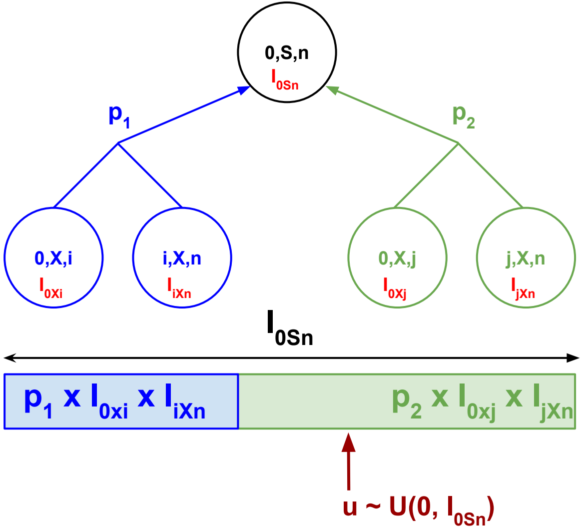


$p_1 \times I_{0xi} \times I_{iXn}$

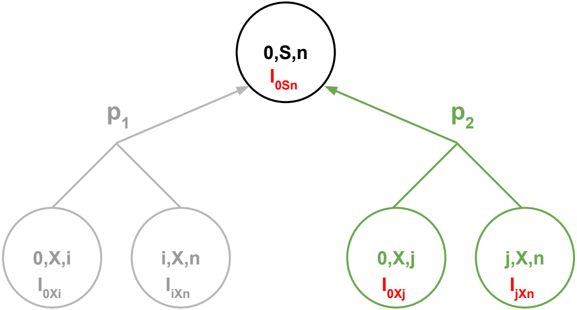
Top-down sampling illustration



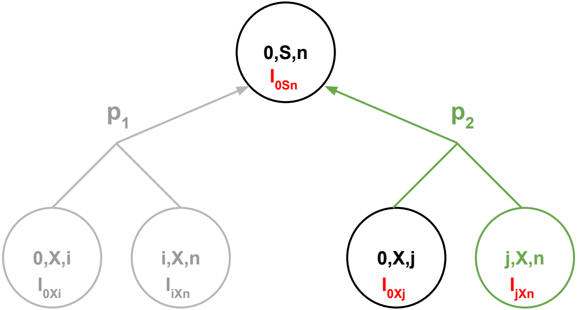
Top-down sampling illustration



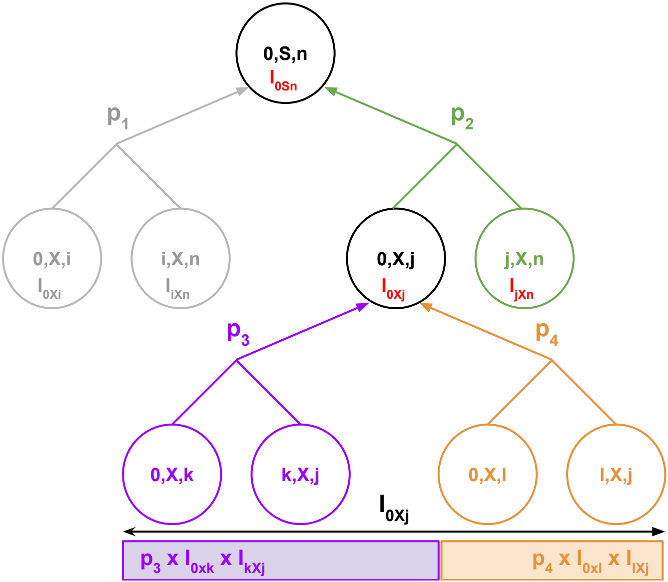
Top-down sampling illustration



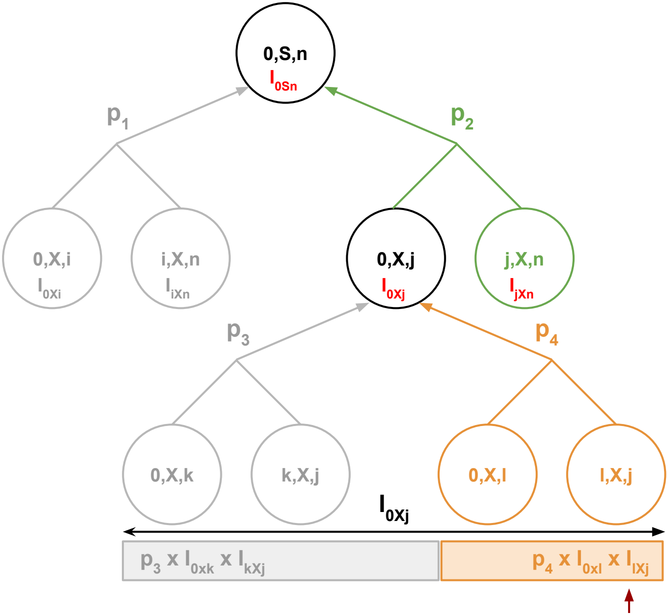
Top-down sampling illustration



Top-down sampling illustration



Top-down sampling illustration



Sampling from chart

~~Given a string $\mathbf{w}$, assume we can build the chart $\mathcal{T}(\mathbf{w})$~~

We care about the cases in which we cannot instantiate the chart!

Why is it hard to sample from high dimensional spaces?

Let's rewrite the density

$$p(x) = \frac{p^*(x)}{Z_p} = \frac{p^*(x)}{\int_{\mathcal{X}} p^*(x)\mathrm{d}x}$$

Why is it hard to sample from high dimensional spaces?

Let's rewrite the density

$$p(x) = \frac{p^*(x)}{Z_p} = \frac{p^*(x)}{\int_{\mathcal{X}} p^*(x) \mathrm{d}x}$$

The denominator is typically hard to compute

it requires summing over the entire support $\mathcal{X}$

Why is it hard to sample from high dimensional spaces?

Let's rewrite the density

$$p(x) = \frac{p^*(x)}{Z_p} = \frac{p^*(x)}{\int_{\mathcal{X}} p^*(x)\mathrm{d}x}$$

The denominator is typically hard to compute

it requires summing over the entire support $\mathcal{X}$

In parsing

it is the inside at the root of the chart

but we cannot afford building the chart!

Estimating expectations

Can we get by without expressing Z_p?

Estimating expectations

Can we get by without expressing Z_p?

We could sample uniformly directly from the support $\mathcal{X}$

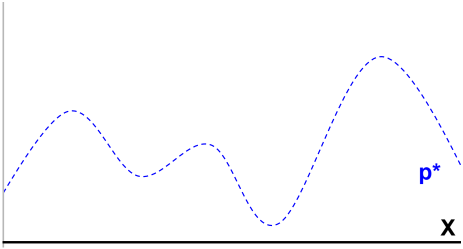
Estimating expectations

Can we get by without expressing Z_p?

We could sample uniformly directly from the support $\mathcal{X}$ approximating Z_p by how much of it we have seen

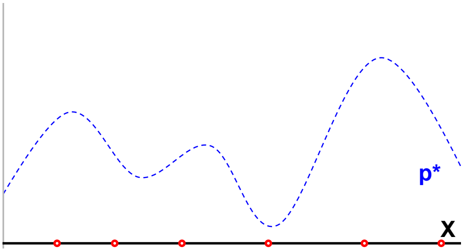
$$Z_N = \sum_{i=1}^{N} p^*(x^{(i)})$$

Uniform sampling



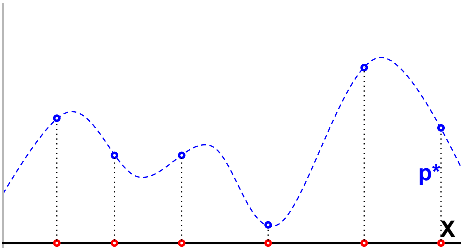
1. $p(x) = \frac{p^*(x)}{Z_p}$

Uniform sampling



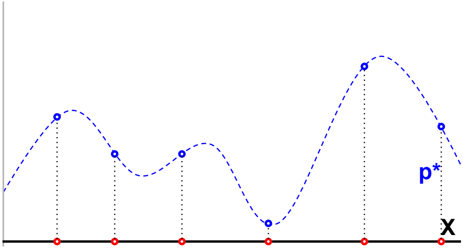
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. sample uniformly from $\mathcal{X}$

Uniform sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. sample uniformly from $\mathcal{X}$
3. $\hat{p}(x^{(i)}) = \frac{p^*(x^{(i)})}{\sum_{j=1}^{N} p^*(x^{(j)})}$

Uniform sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. sample uniformly from $\mathcal{X}$
3. $\hat{p}(x^{(i)}) = \frac{p^*(x^{(i)})}{\sum_{j=1}^{N} p^*(x^{(j)})}$

$$\hat{\Phi} = \sum_{i=1}^{N} \phi(x^{(i)}) \hat{p}(x^{(i)})$$

The trouble in high dimensional spaces

Probability mass is often concentrated in a small region

- *the typical set* T
  $|T| \approx 2^{H(x)}$ H is the Shannon-Gibbs entropy

The trouble in high dimensional spaces

Probability mass is often concentrated in a small region

- *the typical set* T
  $|T| \approx 2^{H(x)}$ $\quad$ H is the Shannon-Gibbs entropy

Suppose $\mathcal{X} \subseteq \{0,1\}^d$ a binary state space

- there are 2^d states $\quad$ think of it as derivation trees
- chance of hitting the typical set $\frac{2^{H(x)}}{2^d}$

The trouble in high dimensional spaces

Probability mass is often concentrated in a small region

- *the typical set* T
  $|T| \approx 2^{H(x)}$ H is the Shannon-Gibbs entropy

Suppose $\mathcal{X} \subseteq \{0,1\}^d$ a binary state space

- there are 2^d states think of it as derivation trees
- chance of hitting the typical set $\frac{2^{H(x)}}{2^d}$

Suppose, $H(x) = d/2$, then

- we expect 1 hit every $2^{d/2}$

The trouble in high dimensional spaces

Probability mass is often concentrated in a small region

- *the typical set* T
  $|T| \approx 2^{H(x)}$ $\qquad$ H is the Shannon-Gibbs entropy

Suppose $\mathcal{X} \subseteq \{0,1\}^d$ a binary state space

- there are 2^d states $\qquad$ think of it as derivation trees
- chance of hitting the typical set $\frac{2^{H(x)}}{2^d}$

Suppose, $H(x) = d/2$, then

- we expect 1 hit every $2^{d/2}$

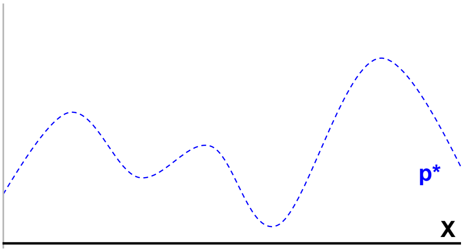
Suppose, 10^3 bits $\qquad$ think of it as rules in a chart for $|\mathbf{w}| = 10$

- $2^{500} \approx 10^{150}$ trials
  square of the number of particles in the universe
  [MacKay, 1998]

Lessons

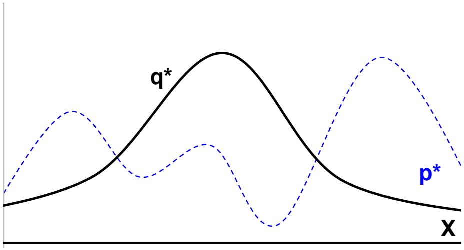
1. assessing Z_p in high dimensional spaces is hard
2. sampling is hard even when $p^*(x)$ is easy to evaluate
   (and direct access to Z_p is not required)

Importance sampling



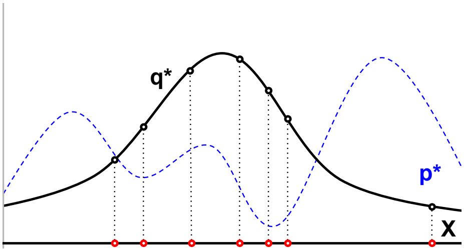
1. $p(x) = \frac{p^*(x)}{Z_p}$

Importance sampling



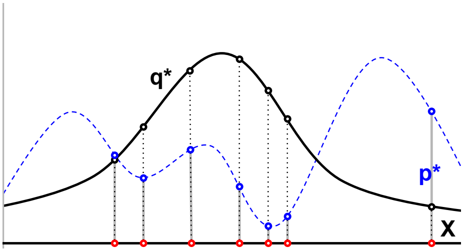
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$

Importance sampling



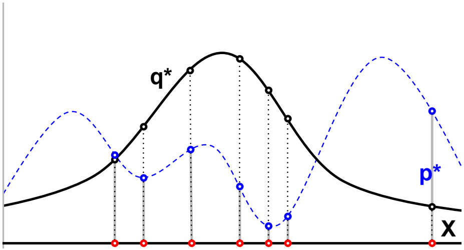
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
3. sample exactly from $q(x)$

Importance sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
3. sample exactly from $q(x)$
4. weight samples
   $w^*(x^{(i)}) = \frac{p^*(x^{(i)})}{q^*(x^{(i)})}$

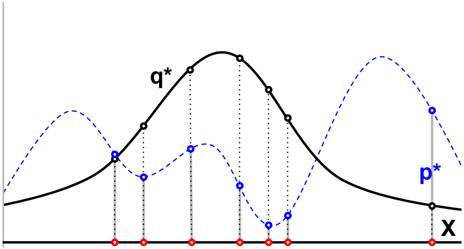
Importance sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
3. sample exactly from $q(x)$
4. weight samples
   $w^*(x^{(i)}) = \frac{p^*(x^{(i)})}{q^*(x^{(i)})}$

$$\hat{\Phi} = \frac{\sum_{i=1}^{N} \phi(x^{(i)}) w^*(x^{(i)})}{\sum_{i=1}^{N} w^*(x^{(i)})}$$

Importance sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
3. sample exactly from $q(x)$
4. weight samples
   $w^*(x^{(i)}) = \frac{p^*(x^{(i)})}{q^*(x^{(i)})}$

$$\hat{\Phi} = \sum_{i=1}^{N} \phi(x^{(i)})\,\hat{w}(x^{(i)})$$

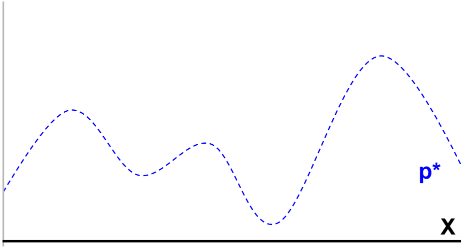
$$\hat{w}(x^{(i)}) = \frac{w^*(x^{(i)})}{\sum_{j=1}^{N} w^*(x^{(j)})}$$

Importance sampling

Introduces an instrumental distribution $q(x)$

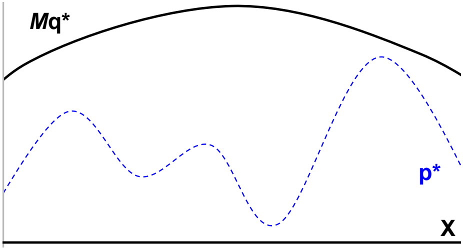
- a better guess than sampling uniformly from the state space
- $q(x)$ is such that sampling from it is trivial
- the variance of the estimate becomes a $q(x)$

Rejection sampling



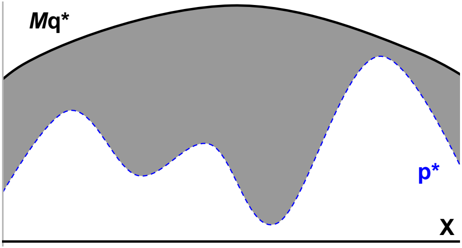
1. $p(x) = \frac{p^*(x)}{Z_p}$

Rejection sampling



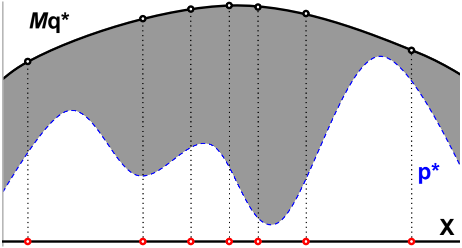
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$

Rejection sampling



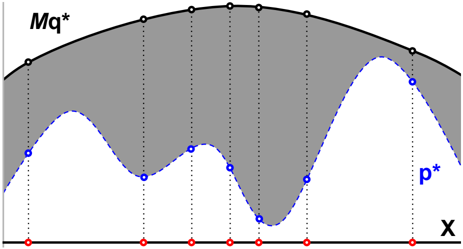
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$

Rejection sampling



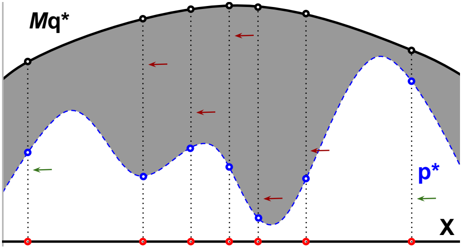
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$
3. sample exactly from $Mq(x)$

Rejection sampling



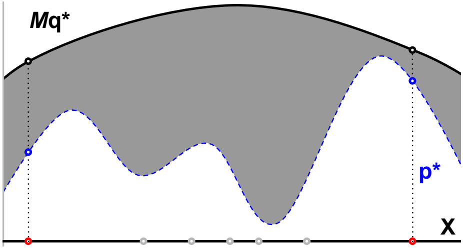
1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$
3. sample exactly from $Mq(x)$
4. acceptance test
   - $r = \frac{p^*(x')}{Mq^*(x')}$

Rejection sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$
3. sample exactly from $Mq(x)$
4. acceptance test
   - $r = \frac{p^*(x')}{Mq^*(x')}$
   - $u \sim U(0,1)$
     accept $x^{(i)} = x'$ iff $r > u$

Rejection sampling



1. $p(x) = \frac{p^*(x)}{Z_p}$
2. $q(x) = \frac{q^*(x)}{Z_q}$
   $q^*(x) = 0$ iff $p^*(x) = 0$
   $Mq^*(x) \geq p^*(x), \forall x \in \mathcal{X}$
3. sample exactly from $Mq(x)$
4. acceptance test
   - $r = \frac{p^*(x')}{Mq^*(x')}$
   - $u \sim U(0,1)$
     accept $x^{(i)} = x'$ iff $r > u$

Accepted x's make an exact sample from $p(x)$

$$\hat{\Phi} = \sum_{i=1}^{N} \phi(x^{(i)})$$

Rejection sampling

Introduces an upperbound $Mq^*(x) \geq p^*(x)$

1. sample (x, u) uniformly distributed over the $(d+1)$-dimensional surface under $Mq^*(x)$
2. retain only points uniformly distributed under $p^*(x)$

Rejection sampling

Introduces an upperbound $Mq^*(x) \geq p^*(x)$

1. sample (x, u) uniformly distributed over the $(d+1)$-dimensional surface under $Mq^*(x)$
2. retain only points uniformly distributed under $p^*(x)$

Problem

- low acceptance rate
- in high dimensional spaces, M is typically huge
  the ratio $\frac{Z_p}{MZ_q} \to 0$

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t}) p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t}) p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

- parsing $O(|V|^3 n^3)$ perhaps feasible – depending on $|V|$

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t}) p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

- parsing $O(|V|^3 n^3)$ perhaps feasible – depending on $|V|$
- tagging $O(|T|^3 n)$ where T is the set of tags feasible

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t})p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

- parsing $O(|V|^3 n^3)$ perhaps feasible – depending on $|V|$
- tagging $O(|T|^3 n)$ where T is the set of tags feasible
- together $O(|V|^3 |T|^6 n^3)$ impracticable for most tag sets!

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t}) p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

- parsing $O(|V|^3 n^3)$ perhaps feasible – depending on $|V|$
- tagging $O(|T|^3 n)$ where T is the set of tags feasible
- together $O(|V|^3 |T|^6 n^3)$ impracticable for most tag sets!

Importance sampling

the parser alone makes the instrumental distribution

$$q(\mathbf{t}) = p_G(\mathbf{t})$$

MC for parsing

Consider the integration of a parser and a 2nd order HMM tagger

$$p(\mathbf{t}) = p_G(\mathbf{t})p_{H_2}(h(\mathbf{t}))$$

where $h(\mathbf{t})$ is the sequence of tags

- parsing $O(|V|^3 n^3)$ perhaps feasible – depending on $|V|$
- tagging $O(|T|^3 n)$ where T is the set of tags feasible
- together $O(|V|^3 |T|^6 n^3)$ impracticable for most tag sets!

Importance sampling

the parser alone makes the instrumental distribution

$$q(\mathbf{t}) = p_G(\mathbf{t})$$

Rejection sampling

replace p_{H_2} by a lower-order upperbound (e.g. 0-order HMM)

$$q(\mathbf{t}) = p_G(\mathbf{t})q_{H_0}(h(\mathbf{t}))$$

Markov chain Monte Carlo

A Markov chain that leaves the desired distribution **invariant**

- unlike MC, samples are not independent
- in the limit of an infinite chain, the state of the chain converges to the target distribution
- we typically discard the beginning of the chain ($i < k$) to reduce dependency on starting conditions

Markov chain Monte Carlo

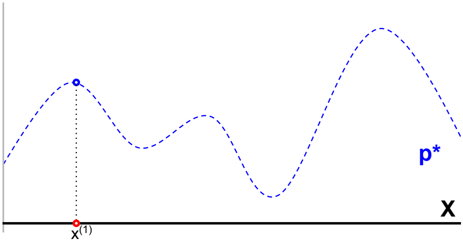
A Markov chain that leaves the desired distribution **invariant**

- unlike MC, samples are not independent
- in the limit of an infinite chain, the state of the chain converges to the target distribution
- we typically discard the beginning of the chain $(i < k)$ to reduce dependency on starting conditions

Samples and expectation

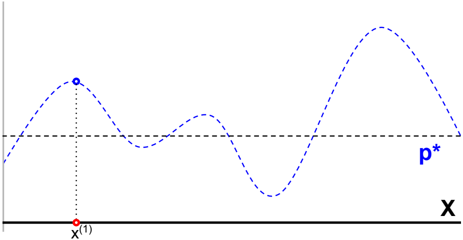
1. $\{x^{(i)}\}_{i=k}^{N}$
2. $\hat{\Phi} = \sum_{i=k}^{N} \phi(x^{(i)})$

Slice sampling



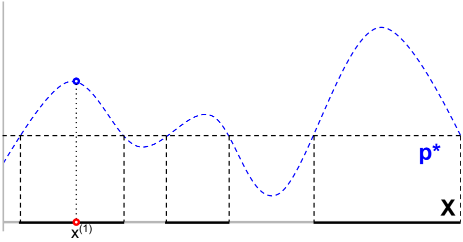
1. start with $x^{(i)}$

Slice sampling



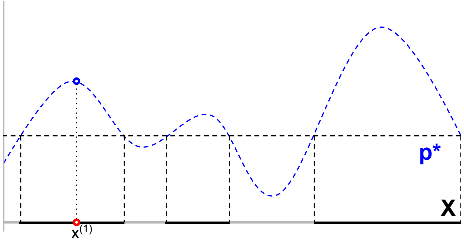
1. start with $x^{(i)}$
2. sample uniformly
   $y \sim U(0, p^*(x^{(i)}))$

Slice sampling



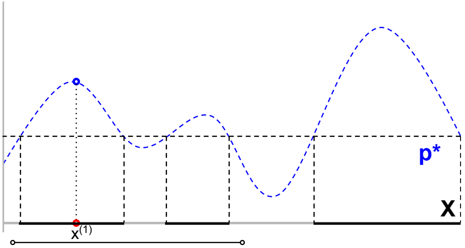
1. start with $x^{(i)}$
2. sample uniformly
   $y \sim U(0, p^*(x^{(i)}))$
3. slice the state space
   $S = \{x : p^*(x) > y\}$

Slice sampling



1. start with $x^{(i)}$
2. sample uniformly $y \sim U(0, p^*(x^{(i)}))$
3. slice the state space $S = \{x : p^*(x) > y\}$
4. sample uniformly from S
   requires a global view of p

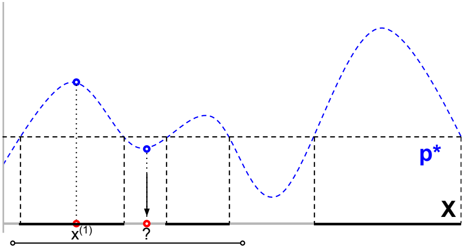
Slice sampling



1. start with $x^{(i)}$
2. sample uniformly
   $y \sim U(0, p^*(x^{(i)}))$
3. slice the state space
   $S = \{x : p^*(x) > y\}$
4. sample uniformly from S
   requires a global view of p

- find an interval I that contains $x^{(i)}$ and as much of S as feasible

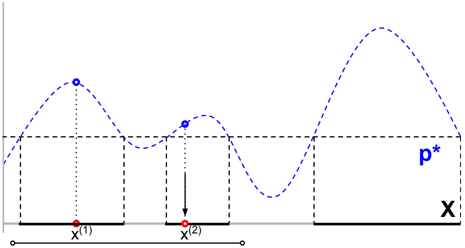
Slice sampling



1. start with $x^{(i)}$
2. sample uniformly $y \sim U(0, p^*(x^{(i)}))$
3. slice the state space $S = \{x : p^*(x) > y\}$
4. sample uniformly from S requires a global view of p

- find an interval I that contains $x^{(i)}$ and as much of S as feasible
- draw x' uniformly from I

Slice sampling



1. start with $x^{(i)}$
2. sample uniformly $y \sim U(0, p^*(x^{(i)}))$
3. slice the state space $S = \{x : p^*(x) > y\}$
4. sample uniformly from S requires a global view of p

- find an interval I that contains $x^{(i)}$ and as much of S as feasible
- draw x' uniformly from I
- make $x^{(i+1)} = x'$ if $x' \in S$, that is, $p^*(x') > y$

Slice sampling

An attempt to get a “black box” sampler

- form of auxiliary variable sampling
- no need for proxy distributions
- requires assessing p^* for a given sample and for the boundaries of an interval I
- finding I can be hard

Gibbs sampling

Task

sample from the joint $p(\mathbf{x} = x_1, \ldots, x_n)$

Gibbs sampling

Task

sample from the joint $p(\mathbf{x} = x_1, \ldots, x_n)$

Method

repeatedly sample from the conditional for each x_k

$$x_k^{(i)} \sim p(x_k | \{x_j\}_{j \neq k})$$

Gibbs sampling

Task

sample from the joint $p(\mathbf{x} = x_1, \ldots, x_n)$

Method

repeatedly sample from the conditional for each x_k

$$x_k^{(i)} \sim p(x_k | \{x_j\}_{j \neq k})$$

Conditioning greatly reduces dimensionality

Gibbs sampling

Task

sample from the joint $p(\mathbf{x} = x_1, \ldots, x_n)$

Method

repeatedly sample from the conditional for each x_k

$$x_k^{(i)} \sim p(x_k | \{x_j\}_{j \neq k})$$

Conditioning greatly reduces dimensionality

- can be done when we know how to sample from all the required conditional distributions

Gibbs sampling

Task
sample from the joint $p(\mathbf{x} = x_1, \ldots, x_n)$

Method
repeatedly sample from the conditional for each x_k

$$x_k^{(i)} \sim p(x_k | \{x_j\}_{j \neq k})$$

Conditioning greatly reduces dimensionality

- can be done when we know how to sample from all the required conditional distributions
- running the sampler for a sufficiently long time produces a samples of values for x from close to the target distribution

MCMC pros and cons

Cons

1. slow mixture (particularly Gibbs)
2. hard to diagnose convergence

Pros

1. enable inference when $p(x)$ is just too complex for dynamic programming
2. estimates can always be improved by increasing the number of samples

Synchronous PCFGs

In synchronous parsing we recognise pairs of strings $(\mathbf{x}, \mathbf{y})$

Synchronous PCFGs

In synchronous parsing we recognise pairs of strings $(\mathbf{x}, \mathbf{y})$

Consider a binary ITG (a special case of SCFG)

- start symbol S
- and a single nonterminal X which can be rewritten
  - $X \rightarrow X_1 X_2 | X_1 X_2$ direct order
  - $X \rightarrow X_1 X_2 | X_2 X_1$ inverted order

Synchronous PCFGs

In synchronous parsing we recognise pairs of strings $(\mathbf{x}, \mathbf{y})$

Consider a binary ITG (a special case of SCFG)

- start symbol S
- and a single nonterminal X which can be rewritten
  - $X \to X_1 X_2 | X_1 X_2$ direct order
  - $X \to X_1 X_2 | X_2 X_1$ inverted order

Synchronous parsing

- $\langle A, i, j, k, l \rangle$ represents a chart cell
  - $A \in V = \{S, X\}$
  - $0 \leq i < j \leq |\mathbf{x}|$ and $0 \leq k < l \leq |\mathbf{y}|$ represents a bispan

Synchronous PCFGs

In synchronous parsing we recognise pairs of strings $(\mathbf{x}, \mathbf{y})$
Consider a binary ITG (a special case of SCFG)

- start symbol S
- and a single nonterminal X which can be rewritten
  - $X \rightarrow X_1 X_2 | X_1 X_2$ direct order
  - $X \rightarrow X_1 X_2 | X_2 X_1$ inverted order

Synchronous parsing

- $\langle A, i, j, k, l \rangle$ represents a chart cell
  - $A \in V = \{S, X\}$
  - $0 \leq i < j \leq |\mathbf{x}|$ and $0 \leq k < l \leq |\mathbf{y}|$ represents a bispan

MLE via EM requires Inside-Outside $O(|\mathbf{x}|^3 |\mathbf{y}|^3)$

Synchronous PCFGs

In synchronous parsing we recognise pairs of strings $(\mathbf{x}, \mathbf{y})$
Consider a binary ITG (a special case of SCFG)

- start symbol S
- and a single nonterminal X which can be rewritten
  - $X \to X_1 X_2 | X_1 X_2$ direct order
  - $X \to X_1 X_2 | X_2 X_1$ inverted order

Synchronous parsing

- $\langle A, i, j, k, l \rangle$ represents a chart cell
  - $A \in V = \{S, X\}$
  - $0 \leq i < j \leq |\mathbf{x}|$ and $0 \leq k < l \leq |\mathbf{y}|$ represents a bispan

MLE via EM requires Inside-Outside $O(|\mathbf{x}|^3 |\mathbf{y}|^3)$ prohibitive!

Slice sampling for synchronous parsing

We introduce an auxiliary variable per chart cell

- chart
  $S = \{\langle A, i, j, k, l\rangle : 0 \le i < j \le |\mathbf{x}|, 0 \le k < l \le |\mathbf{y}|, A \in V\}$
- slice variables
  $\mathbf{u} = \{u_s \in [0, 1] : s \in S\}$

Slice sampling for synchronous parsing

We introduce an auxiliary variable per chart cell

- chart
  $S = \{\langle A, i, j, k, l \rangle : 0 \leq i < j \leq |\mathbf{x}|, 0 \leq k < l \leq |\mathbf{y}|, A \in V\}$
- slice variables
  $\mathbf{u} = \{u_s \in [0, 1] : s \in S\}$

The slice variables act as cutoff on the probabilities of the rules considered in each cell

Slice sampling for synchronous parsing

We introduce an auxiliary variable per chart cell

- chart
  $S = \{\langle A, i, j, k, l \rangle : 0 \leq i < j \leq |\mathbf{x}|, 0 \leq k < l \leq |\mathbf{y}|, A \in V\}$
- slice variables
  $\mathbf{u} = \{u_s \in [0, 1] : s \in S\}$

The slice variables act as cutoff on the probabilities of the rules considered in each cell

- rule applications r_s with $\theta_{r_s} \leq u_s$ are pruned from the dynamic program

Slice sampling for synchronous parsing

Sampling $p(\mathbf{u}|\mathbf{t})$

- u_s are conditionally independent

$$u_s \sim p(u_s|\mathbf{t}) = \begin{cases} U(u_s; 0, \theta_{r_s}) & \text{if } r_s \in \mathbf{t} \\ \beta(u_s; a, 1) & \text{otherwise} \end{cases}$$

Slice sampling for synchronous parsing

Sampling $p(\mathbf{u}|\mathbf{t})$

- u_s are conditionally independent

$$u_s \sim p(u_s|\mathbf{t}) = \begin{cases} U(u_s; 0, \theta_{r_s}) & \text{if } r_s \in \mathbf{t} \\ \beta(u_s; a, 1) & \text{otherwise} \end{cases}$$

Sampling $p(\mathbf{t}|\mathbf{u})$

- requires computing Inside weights for a truncated chart

$$\mathbf{t} \sim p(\mathbf{t}|\mathbf{u}) \propto \prod_{u_s : r_s \in \mathbf{t}} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\beta(u_s; a, 1)}$$

Slice sampling for synchronous parsing

Sampling $p(\mathbf{u}|\mathbf{t})$

- u_s are conditionally independent

$$u_s \sim p(u_s|\mathbf{t}) = \begin{cases} U(u_s; 0, \theta_{r_s}) & \text{if } r_s \in \mathbf{t} \\ \beta(u_s; a, 1) & \text{otherwise} \end{cases}$$

Sampling $p(\mathbf{t}|\mathbf{u})$

- requires computing Inside weights for a truncated chart

$$\mathbf{t} \sim p(\mathbf{t}|\mathbf{u}) \propto \prod_{u_s : r_s \in \mathbf{t}} \frac{\mathbb{I}(u_s < \theta_{r_s})}{\beta(u_s; a, 1)}$$

The hyperparameter a controls the degree of pruning

Blunsom and Cohn [2010]

Bayesian inference

Bayesian MAP inference

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

- $\mathcal{W}$ data (set of strings)
- $p_G(\mathcal{W}|\boldsymbol{\theta})$ likelihood of data given model $\boldsymbol{\theta}$
- $p(\boldsymbol{\theta})$ prior (if uniform we get MLE)

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Each $\theta_{A\to\beta}$ has a corresponding Dirichlet parameter $\alpha_{A\to\beta}$

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Each $\theta_{A\to\beta}$ has a corresponding Dirichlet parameter $\alpha_{A\to\beta}$

- R_A are the productions with LHS A
- $\boldsymbol{\theta}_A$ and $\boldsymbol{\alpha}_A$ are parameters associated with rules in R_A

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Each $\theta_{A \to \beta}$ has a corresponding Dirichlet parameter $\alpha_{A \to \beta}$

- R_A are the productions with LHS A
- $\boldsymbol{\theta}_A$ and $\boldsymbol{\alpha}_A$ are parameters associated with rules in R_A

The Dirichlet prior is

- $p_D(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \prod_{A \in N} p_D(\boldsymbol{\theta}_A; \boldsymbol{\alpha}_A)$

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Each $\theta_{A\rightarrow\beta}$ has a corresponding Dirichlet parameter $\alpha_{A\rightarrow\beta}$

- R_A are the productions with LHS A
- $\boldsymbol{\theta}_A$ and $\boldsymbol{\alpha}_A$ are parameters associated with rules in R_A

The Dirichlet prior is

- $p_D(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \prod_{A\in N} p_D(\boldsymbol{\theta}_A; \boldsymbol{\alpha}_A)$
- $p_D(\boldsymbol{\theta}_A; \boldsymbol{\alpha}_A) \propto \prod_{r\in R_A} \theta_r^{\alpha_r - 1}$

Dirichlet prior

We make an assumption about $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim p_D(\boldsymbol{\theta}; \boldsymbol{\alpha})$$

Each $\theta_{A \to \beta}$ has a corresponding Dirichlet parameter $\alpha_{A \to \beta}$

- R_A are the productions with LHS A
- $\boldsymbol{\theta}_A$ and $\boldsymbol{\alpha}_A$ are parameters associated with rules in R_A

The Dirichlet prior is

- $p_D(\boldsymbol{\theta}; \boldsymbol{\alpha}) = \prod_{A \in N} p_D(\boldsymbol{\theta}_A; \boldsymbol{\alpha}_A)$
- $p_D(\boldsymbol{\theta}_A; \boldsymbol{\alpha}_A) \propto \prod_{r \in R_A} \theta_r^{\alpha_r - 1}$

Posterior is also a Dirichlet

- $p_D(\boldsymbol{\theta} | \mathcal{T}; \boldsymbol{\alpha}) = p_D(\boldsymbol{\theta} | \mathbf{f}(\mathcal{T}) + \boldsymbol{\alpha})$

"updates the prior conditioning on evidence"

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

In Gibbs sampling, we can sample from the joint $p(\mathcal{T},\boldsymbol{\theta})$ by sampling from the conditionals

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

In Gibbs sampling, we can sample from the joint $p(\mathcal{T},\boldsymbol{\theta})$ by sampling from the conditionals

- $p(\mathcal{T}|\boldsymbol{\theta},\mathcal{W};\boldsymbol{\alpha}) = \prod_{\mathbf{w}\in\mathcal{W}} p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

In Gibbs sampling, we can sample from the joint $p(\mathcal{T},\boldsymbol{\theta})$ by sampling from the conditionals

- $p(\mathcal{T}|\boldsymbol{\theta},\mathcal{W};\boldsymbol{\alpha}) = \prod_{\mathbf{w}\in\mathcal{W}} p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$

$$\mathbf{t} \sim p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$$

requires Inside weights

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

In Gibbs sampling, we can sample from the joint $p(\mathcal{T},\boldsymbol{\theta})$ by sampling from the conditionals

- $p(\mathcal{T}|\boldsymbol{\theta},\mathcal{W};\boldsymbol{\alpha}) = \prod_{\mathbf{w}\in\mathcal{W}} p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$

$$\mathbf{t} \sim p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$$

requires Inside weights

- $p(\boldsymbol{\theta}|\mathcal{T},\mathcal{W};\boldsymbol{\alpha}) = p_D(\boldsymbol{\theta}|\mathbf{f}(\mathcal{T})+\boldsymbol{\alpha}) = \prod_{A\in N} p_D(\boldsymbol{\theta}_A|\mathbf{f}_A(\mathcal{T})+\boldsymbol{\alpha})$

Gibbs sampling

Let's rewrite the posterior in terms of a joint distribution

$$p(\boldsymbol{\theta}|\mathcal{W};\boldsymbol{\alpha}) = \sum_{\mathcal{T}\in G(\mathcal{W})} p(\mathcal{T},\boldsymbol{\theta}|\mathcal{W};\alpha)$$

In Gibbs sampling, we can sample from the joint $p(\mathcal{T},\boldsymbol{\theta})$ by sampling from the conditionals

- $p(\mathcal{T}|\boldsymbol{\theta},\mathcal{W};\boldsymbol{\alpha}) = \prod_{\mathbf{w}\in\mathcal{W}} p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$

$$\mathbf{t} \sim p(\mathbf{t}|\mathbf{w},\boldsymbol{\theta};\boldsymbol{\alpha})$$

requires Inside weights

- $p(\boldsymbol{\theta}|\mathcal{T},\mathcal{W};\boldsymbol{\alpha}) = p_D(\boldsymbol{\theta}|\mathbf{f}(\mathcal{T})+\boldsymbol{\alpha}) = \prod_{A\in N} p_D(\boldsymbol{\theta}_A|\mathbf{f}_A(\mathcal{T})+\boldsymbol{\alpha})$

$$\boldsymbol{\theta}_A \sim p_D(\boldsymbol{\theta}_A|\mathbf{f}_A(\mathcal{T})+\boldsymbol{\alpha})$$

there exists efficient techniques to sample from a Dirichlet

Johnson et al. [2007]

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

People tend to associate MC/MCMC with Bayesian inference

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

People tend to associate MC/MCMC with Bayesian inference

- they are **orthogonal**

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

People tend to associate MC/MCMC with Bayesian inference

- they are **orthogonal**

One can sample in order to

- compute expectations (e.g. when performing MLE)

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

People tend to associate MC/MCMC with Bayesian inference

- they are **orthogonal**

One can sample in order to

- compute expectations (e.g. when performing MLE)
- marginalise latent variables

Sampling

Sampling provides a powerful inference framework

- when the distribution is **too complex** to be represented
- also for Bayesian inference
  where complex posteriors **do not factorise** conveniently
  as the likelihood typically does

People tend to associate MC/MCMC with Bayesian inference

- they are **orthogonal**

One can sample in order to

- compute expectations (e.g. when performing MLE)
- marginalise latent variables
- approximate distributions in general

Questions?

References I

Phil Blunsom and Trevor Cohn. Inducing synchronous grammars with slice sampling. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 238–241, Los Angeles, California, June 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N10-1028`.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York, April 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N07/N07-1018`.

References II

David J. C. MacKay. Introduction to monte carlo methods, 1998.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004. ISBN 0387212396.