The Infinite PCFG using Hierarchical Dirichlet Processes

EMNLP 2007 Prague, Czech Republic

June 29, 2007

Percy Liang Slav Petrov
Michael I. Jordan Dan Klein

University of California
CALNLP
Berkeley

S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise

S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise

How do we choose the grammar complexity?

Grammar induction:

How many grammar symbols (NP, VP, etc.)?

?

She heard the noise

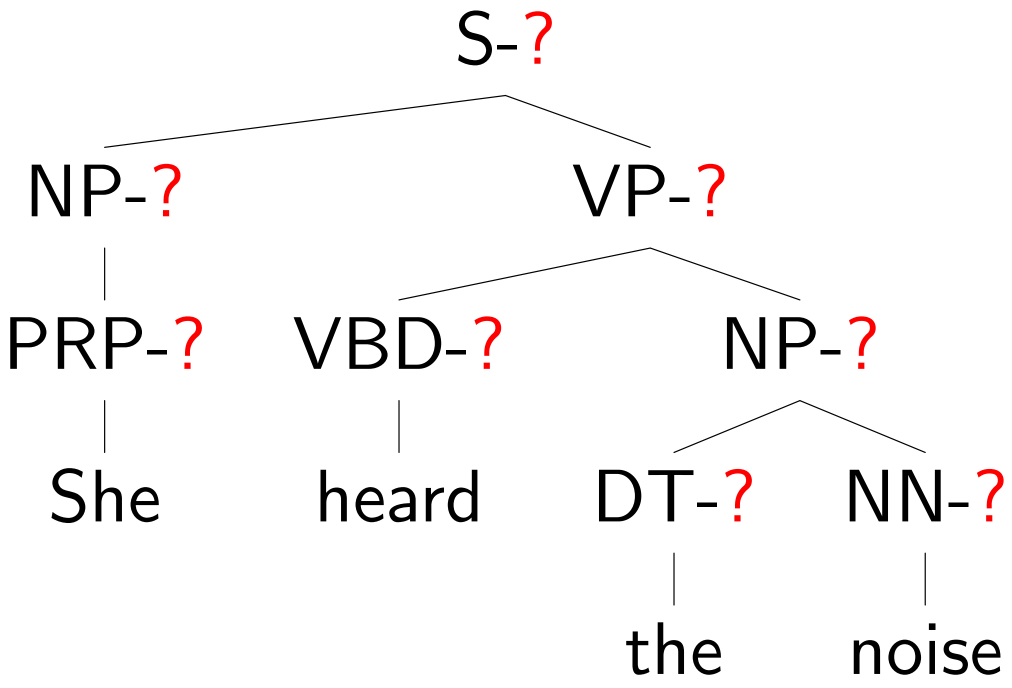
How do we choose the grammar complexity?

Grammar induction:

How many grammar symbols (NP, VP, etc.)?

Grammar refinement:

How many grammar subsymbols (NP-loc, NP-subj, etc.)?



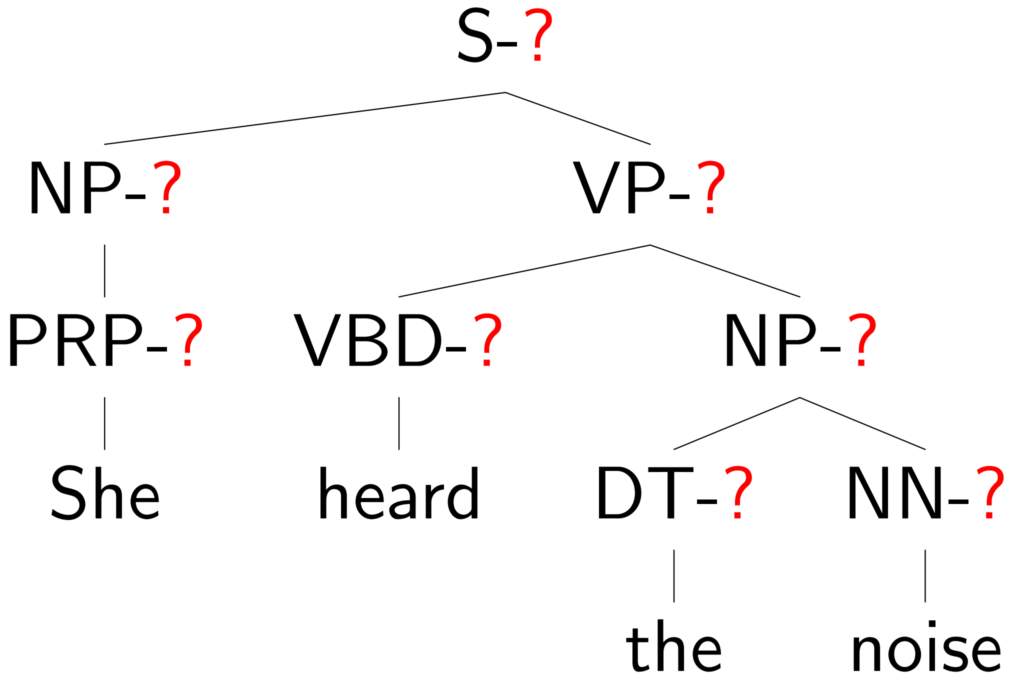
How do we choose the grammar complexity?

Grammar induction:

How many grammar symbols (NP, VP, etc.)?

Grammar refinement:

How many grammar subsymbols (NP-loc, NP-subj, etc.)?



Our solution: the HDP-PCFG allows number of (sub)symbols to adapt to data

A motivating example

True grammar:

$\mathsf{S} \rightarrow \mathsf{A}\,\mathsf{A} \mid \mathsf{B}\,\mathsf{B} \mid \mathsf{C}\,\mathsf{C} \mid \mathsf{D}\,\mathsf{D}$

$\mathsf{A} \rightarrow \mathsf{a}_1 \mid \mathsf{a}_2 \mid \mathsf{a}_3$

$\mathsf{B} \rightarrow \mathsf{b}_1 \mid \mathsf{b}_2 \mid \mathsf{b}_3$

$\mathsf{C} \rightarrow \mathsf{c}_1 \mid \mathsf{c}_2 \mid \mathsf{c}_3$

$\mathsf{D} \rightarrow \mathsf{d}_1 \mid \mathsf{d}_2 \mid \mathsf{d}_3$

A motivating example

True grammar:

$\mathsf{S} \rightarrow \mathsf{A}\,\mathsf{A} \mid \mathsf{B}\,\mathsf{B} \mid \mathsf{C}\,\mathsf{C} \mid \mathsf{D}\,\mathsf{D}$

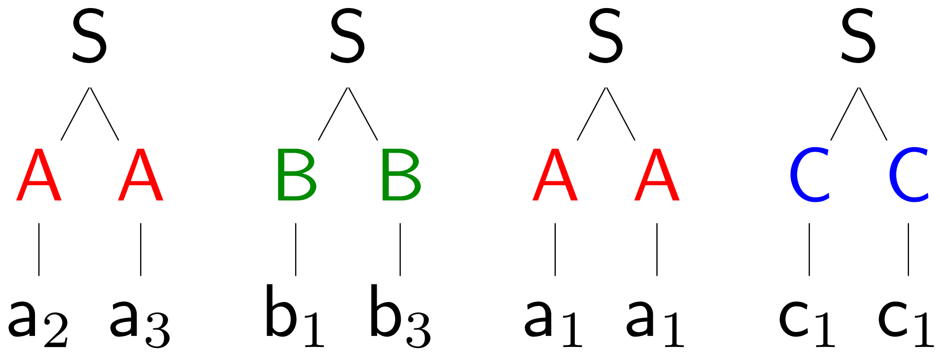
$\mathsf{A} \rightarrow \mathsf{a}_1 \mid \mathsf{a}_2 \mid \mathsf{a}_3$

$\mathsf{B} \rightarrow \mathsf{b}_1 \mid \mathsf{b}_2 \mid \mathsf{b}_3$

$\mathsf{C} \rightarrow \mathsf{c}_1 \mid \mathsf{c}_2 \mid \mathsf{c}_3$

$\mathsf{D} \rightarrow \mathsf{d}_1 \mid \mathsf{d}_2 \mid \mathsf{d}_3$

Generate examples:



A motivating example

True grammar:

$S \rightarrow A\,A \mid B\,B \mid C\,C \mid D\,D$

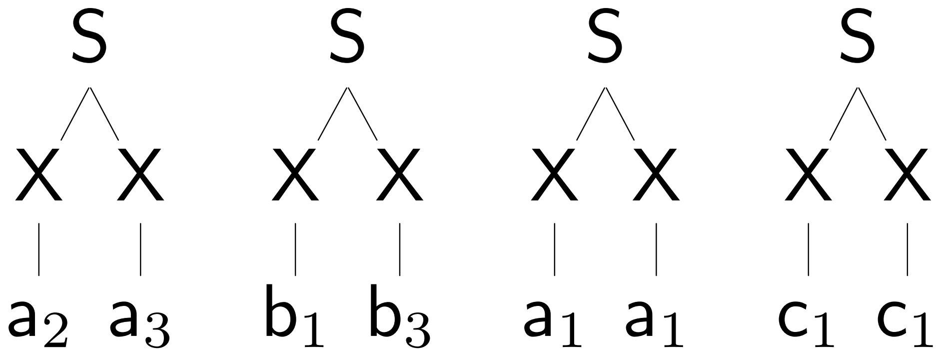
$A \rightarrow a_1 \mid a_2 \mid a_3$

$B \rightarrow b_1 \mid b_2 \mid b_3$

$C \rightarrow c_1 \mid c_2 \mid c_3$

$D \rightarrow d_1 \mid d_2 \mid d_3$

Collapse A,B,C,D ⇒ X:



A motivating example

True grammar:

$\mathrm{S} \rightarrow \mathrm{A}\,\mathrm{A} \mid \mathrm{B}\,\mathrm{B} \mid \mathrm{C}\,\mathrm{C} \mid \mathrm{D}\,\mathrm{D}$

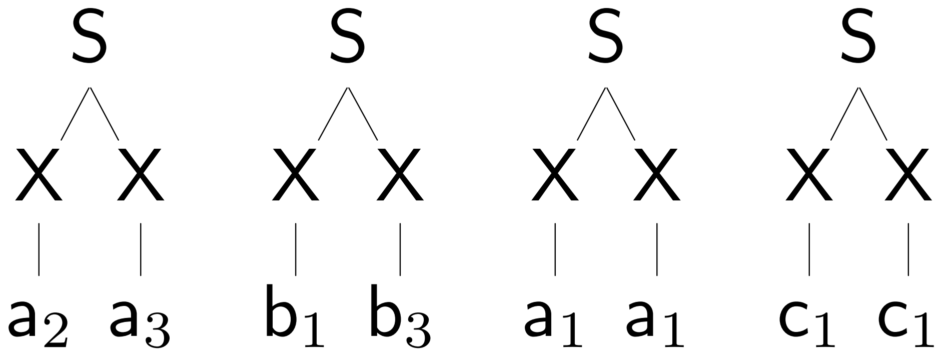
$\mathrm{A} \rightarrow a_1 \mid a_2 \mid a_3$

$\mathrm{B} \rightarrow b_1 \mid b_2 \mid b_3$

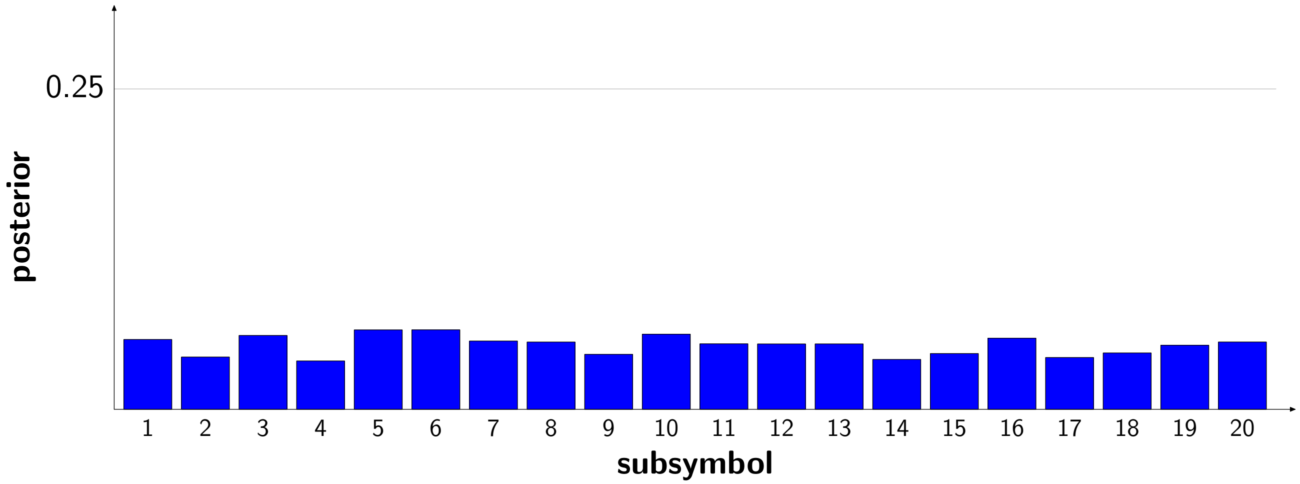
$\mathrm{C} \rightarrow c_1 \mid c_2 \mid c_3$

$\mathrm{D} \rightarrow d_1 \mid d_2 \mid d_3$

Collapse A,B,C,D $\Rightarrow$ X:



Results:



standard PCFG

A motivating example

True grammar:

$S \rightarrow A\,A \mid B\,B \mid C\,C \mid D\,D$

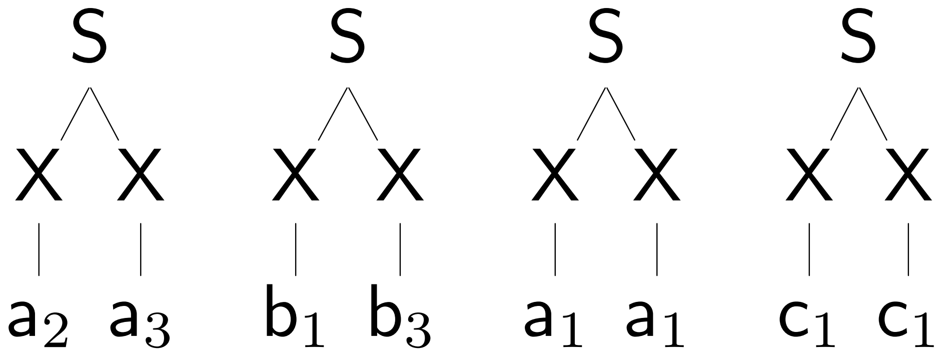
$A \rightarrow a_1 \mid a_2 \mid a_3$

$B \rightarrow b_1 \mid b_2 \mid b_3$

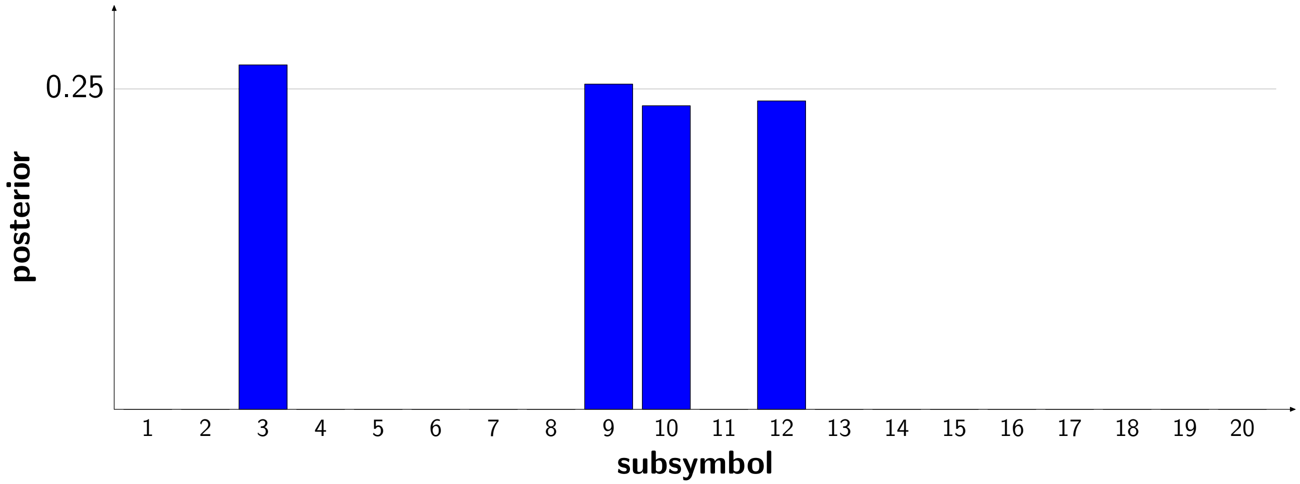
$C \rightarrow c_1 \mid c_2 \mid c_3$

$D \rightarrow d_1 \mid d_2 \mid d_3$

Collapse A,B,C,D ⇒ X:



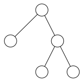
Results:



HDP-PCFG

The meeting of two fields

Grammar learning



Lexicalized

[Charniak, 1996]

[Collins, 1999]

Manual refinement

[Johnson, 1998]

[Klein, Manning, 2003]

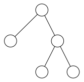
Automatic refinement

[Matsuzaki, et al., 2005]

[Petrov, et al., 2006]

The meeting of two fields

Grammar learning



Lexicalized

[Charniak, 1996]

[Collins, 1999]

Manual refinement

[Johnson, 1998]

[Klein, Manning, 2003]

Automatic refinement

[Matsuzaki, et al., 2005]

[Petrov, et al., 2006]

Bayesian nonparametrics ○ ○ ○ ○ ⋯

Basic theory

[Ferguson, 1973]

[Antoniak, 1974]

[Sethuraman, 1994]

[Escobar, West, 1995]

[Neal, 2000]

More complex models

[Teh, et al., 2006]

[Beal, et al., 2002]

[Goldwater, et al., 2006]

[Sohn, Xing, 2007]

The meeting of two fields

Grammar learning

Lexicalized

[Charniak, 1996]
[Collins, 1999]

Manual refinement

[Johnson, 1998]
[Klein, Manning, 2003]

Automatic refinement

[Matsuzaki, et al., 2005]
[Petrov, et al., 2006]

Bayesian nonparametrics ○ ○ ○ ○ ···

Basic theory

[Ferguson, 1973]
[Antoniak, 1974]
[Sethuraman, 1994]
[Escobar, West, 1995]
[Neal, 2000]

More complex models

[Teh, et al., 2006]
[Beal, et al., 2002]
[Goldwater, et al., 2006]
[Sohn, Xing, 2007]

Nonparametric grammars

[Johnson, et al., 2006]
[Finkel, et al., 2007]
[Liang, et al., 2007]

The meeting of two fields

Grammar learning

Lexicalized

[Charniak, 1996]

[Collins, 1999]

Ma

Auto

Bayesian nonparametrics ○○○○ ⋯

Basic theory

[Ferguson, 1973]

[Antoniak, 1974]

Our contribution

- Definition of the HDP-PCFG
- Simple and efficient variational inference algorithm
- Empirical comparison with finite models on a full-scale parsing task

Nonparametric grammars

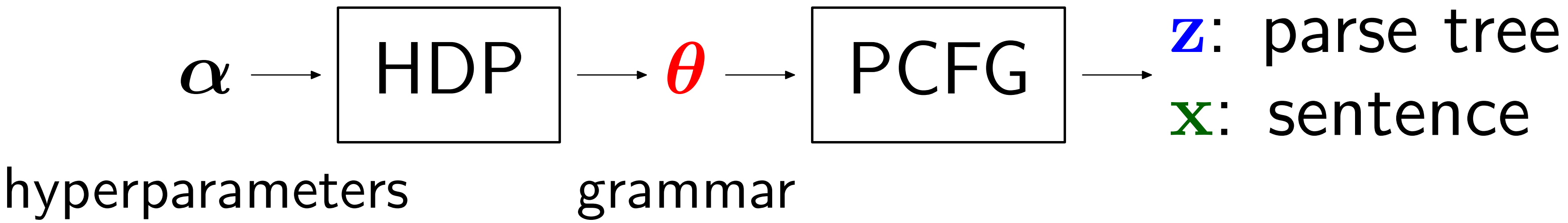
[Johnson, et al., 2006]

[Finkel, et al., 2007]

[Liang, et al., 2007]

Bayesian paradigm

Generative model:



$\boldsymbol{\alpha} \longrightarrow \boxed{\text{HDP}} \longrightarrow \boldsymbol{\theta} \longrightarrow \boxed{\text{PCFG}} \longrightarrow$

$\mathbf{z}$: parse tree

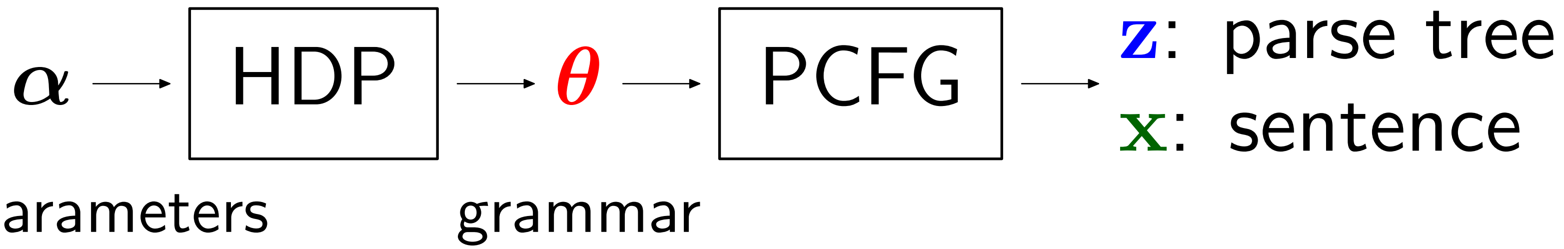
$\mathbf{x}$: sentence

hyperparameters

grammar

Bayesian paradigm

Generative model:



$\boldsymbol{\alpha}$ → HDP → $\boldsymbol{\theta}$ → PCFG →

$\mathbf{z}$: parse tree

$\mathbf{x}$: sentence

hyperparameters

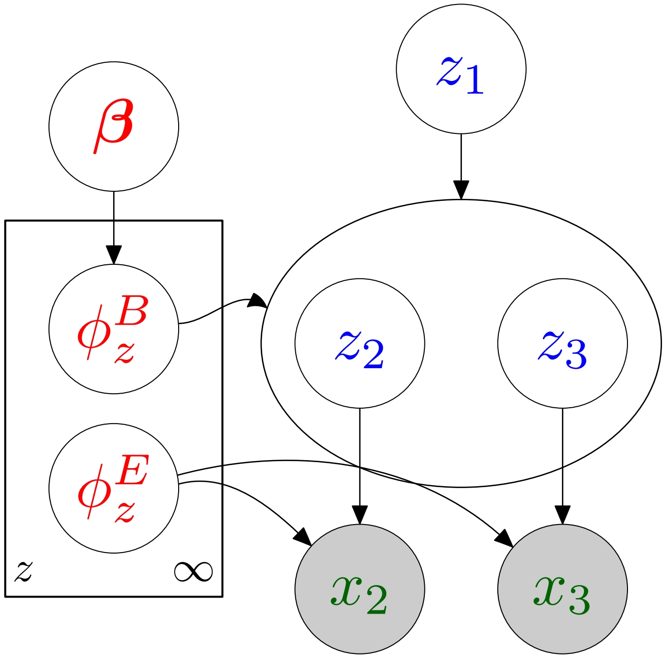
grammar

Bayesian posterior inference:

Observe $\mathbf{x}$.

What's $\boldsymbol{\theta}$ and $\mathbf{z}$?

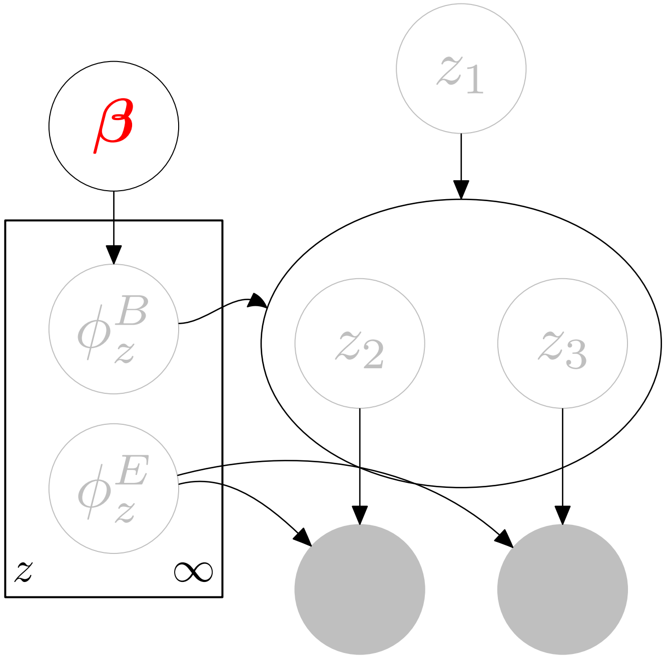
HDP probabilistic context-free grammars



HDP-PCFG

$\boldsymbol{\beta} \sim$ GEM(α) [generate distribution over symbols]
For each symbol $z \in \{1, 2, \dots\}$:
  $\phi_z^E \sim$ Dirichlet(α^E) [generate emission probs]
  $\phi_z^B \sim$ DP$(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$ [binary production probs]
For each nonterminal node i:
  $(z_{L(i)}, z_{R(i)}) \sim$ Multinomial$(\phi_{z_i}^B)$ [child symbols]
For each preterminal node i:
  $x_i \sim$ Multinomial$(\phi_{z_i}^E)$ [terminal symbol]

HDP probabilistic context-free grammars



HDP-PCFG

$\boldsymbol{\beta} \sim$ GEM(α) [generate distribution over symbols]

For each symbol $z \in \{1, 2, \dots\}$:

$\phi_z^E \sim$ Dirichlet(α^E) [generate emission probs]

$\phi_z^B \sim$ DP$(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$ [binary production probs]

For each nonterminal node i:

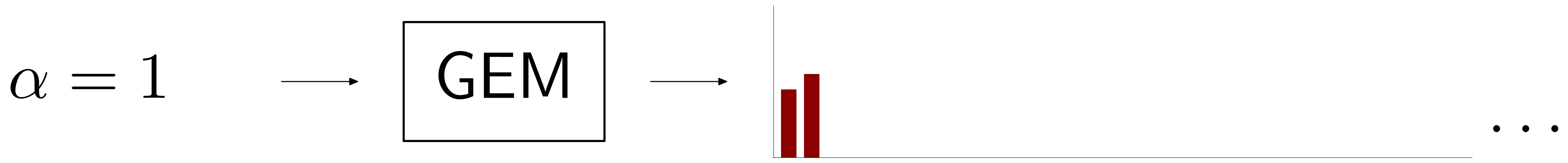
$(z_{L(i)}, z_{R(i)}) \sim$ Multinomial$(\phi_{z_i}^B)$ [child symbols]

For each preterminal node i:

$x_i \sim$ Multinomial$(\phi_{z_i}^E)$ [terminal symbol]

HDP-PCFG: prior over symbols

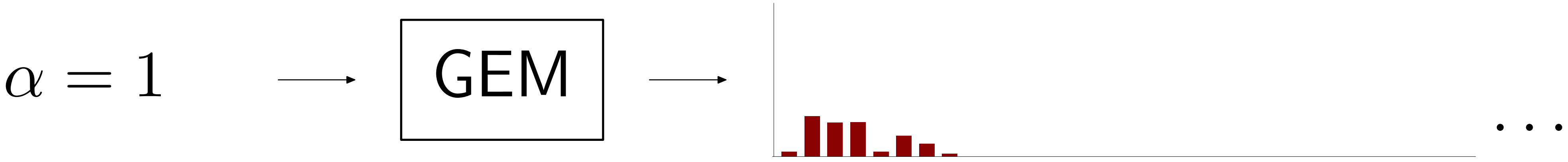
$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



$$\alpha = 1 \longrightarrow \boxed{\mathsf{GEM}} \longrightarrow \quad \cdots$$

HDP-PCFG: prior over symbols

$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$

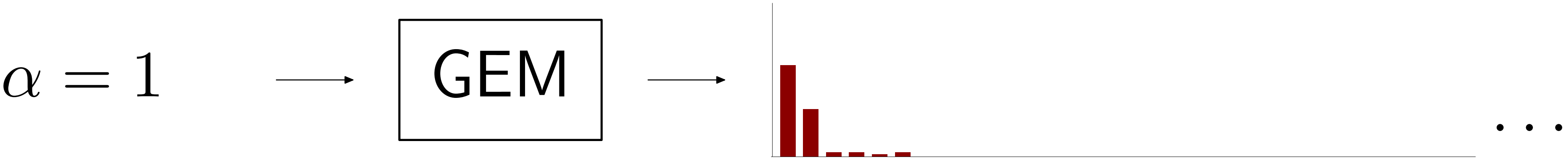


$\alpha = 1$

GEM

HDP-PCFG: prior over symbols

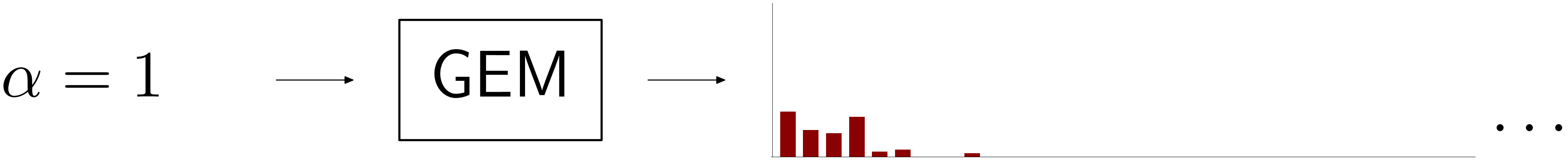
$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



$$\alpha = 1 \longrightarrow \boxed{\mathsf{GEM}} \longrightarrow \cdots$$

HDP-PCFG: prior over symbols

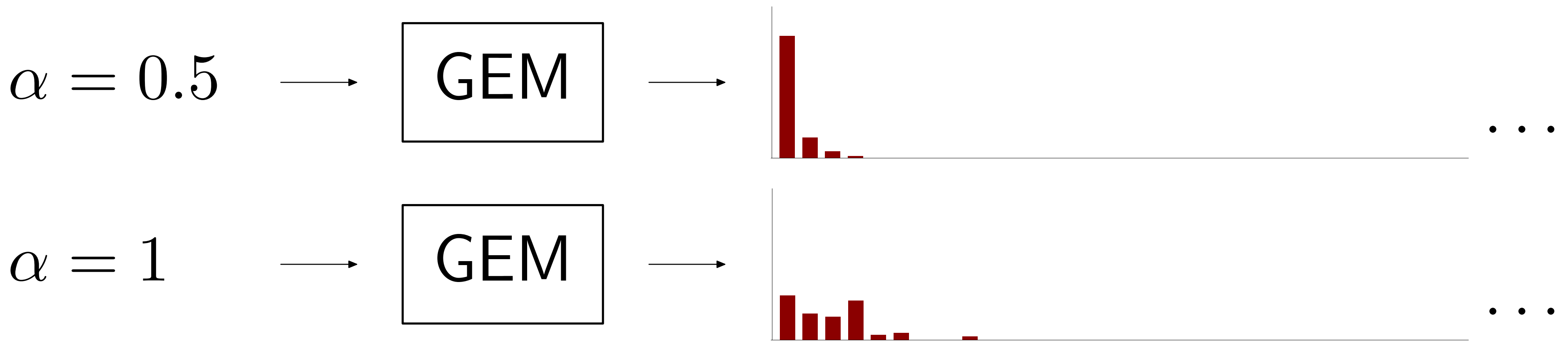
$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



$$\alpha = 1 \longrightarrow \boxed{\mathsf{GEM}} \longrightarrow \cdots$$

HDP-PCFG: prior over symbols

$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$

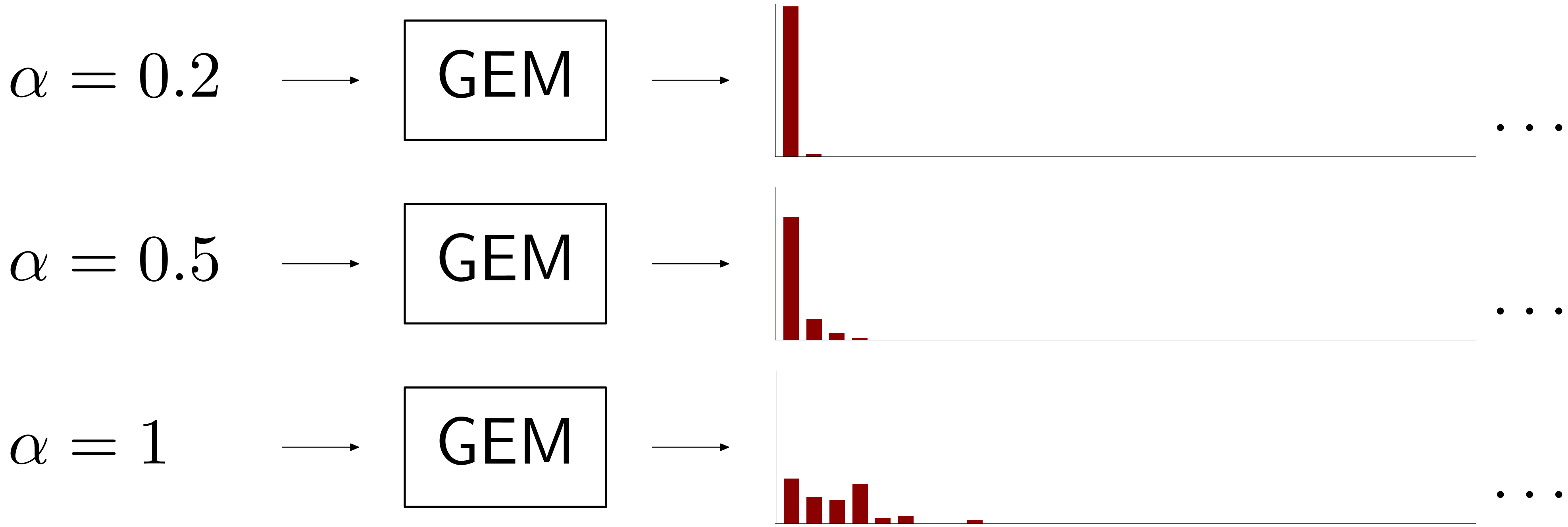


$$\alpha = 0.5 \longrightarrow \boxed{\mathsf{GEM}} \longrightarrow \cdots$$

$$\alpha = 1 \longrightarrow \boxed{\mathsf{GEM}} \longrightarrow \cdots$$

HDP-PCFG: prior over symbols

$$\boldsymbol{\beta} \sim \text{GEM}(\alpha)$$



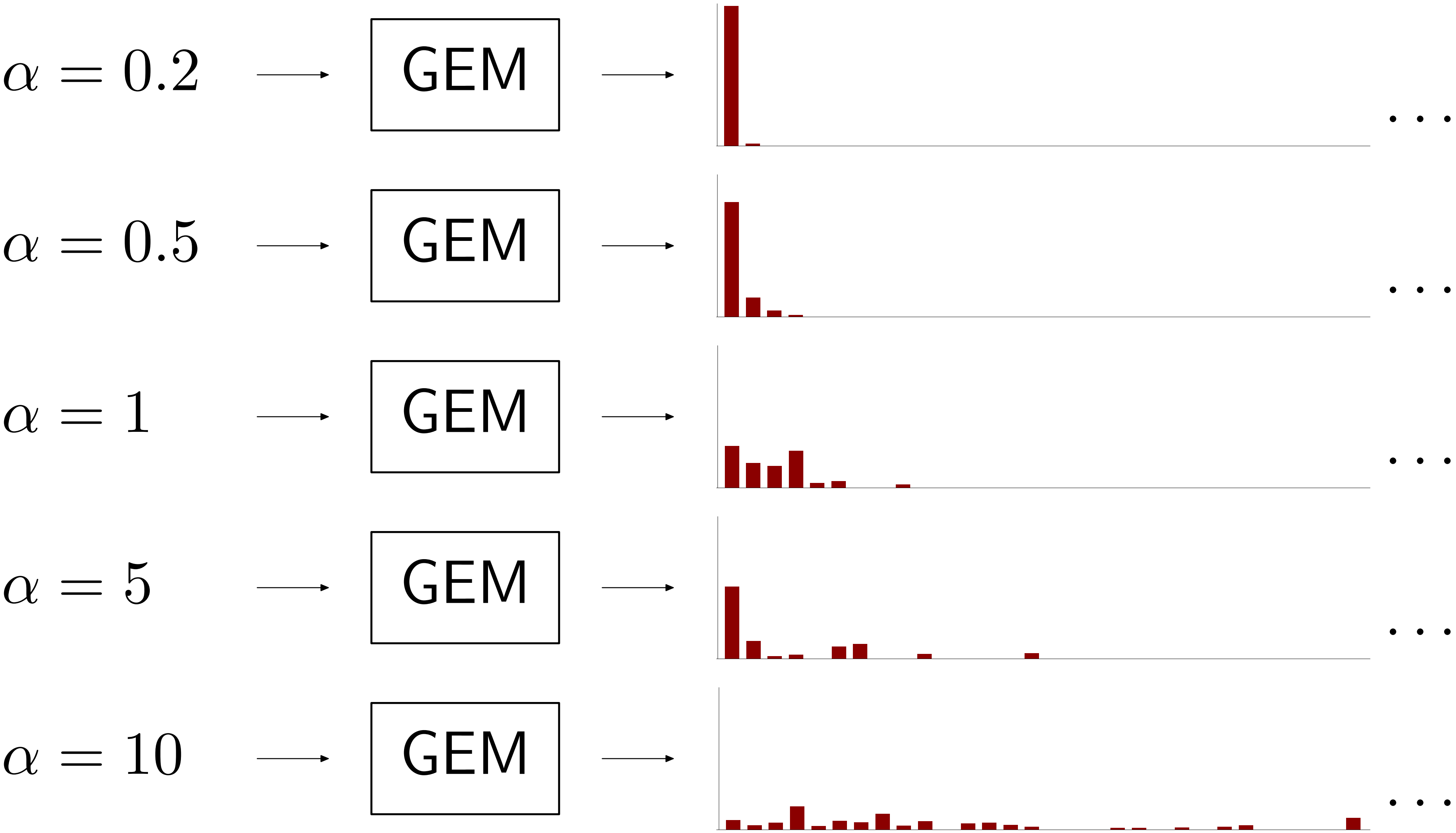
$\alpha = 0.2 \longrightarrow \boxed{\text{GEM}} \longrightarrow$

$\alpha = 0.5 \longrightarrow \boxed{\text{GEM}} \longrightarrow$

$\alpha = 1 \longrightarrow \boxed{\text{GEM}} \longrightarrow$

HDP-PCFG: prior over symbols

$$\boldsymbol{\beta} \sim \text{GEM}(\alpha)$$



$$\alpha = 0.2 \longrightarrow \boxed{\text{GEM}} \longrightarrow \cdots$$

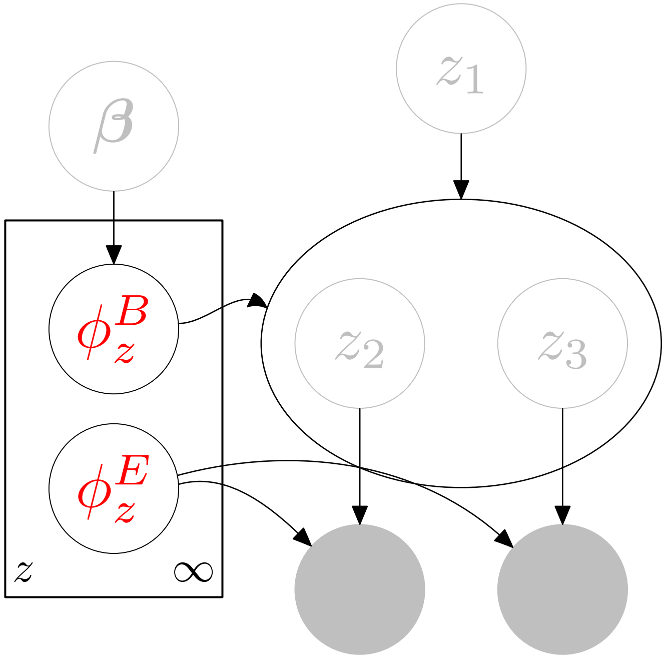
$$\alpha = 0.5 \longrightarrow \boxed{\text{GEM}} \longrightarrow \cdots$$

$$\alpha = 1 \longrightarrow \boxed{\text{GEM}} \longrightarrow \cdots$$

$$\alpha = 5 \longrightarrow \boxed{\text{GEM}} \longrightarrow \cdots$$

$$\alpha = 10 \longrightarrow \boxed{\text{GEM}} \longrightarrow \cdots$$

HDP probabilistic context-free grammars



HDP-PCFG

$\boldsymbol{\beta} \sim$ GEM(α) [generate distribution over symbols]

For each symbol $z \in \{1, 2, \dots\}$:

$\phi_z^E \sim$ Dirichlet(α^E) [generate emission probs]

$\phi_z^B \sim$ DP($\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T$) [binary production probs]

For each nonterminal node i:

$(z_{L(i)}, z_{R(i)}) \sim$ Multinomial($\phi_{z_i}^B$) [child symbols]

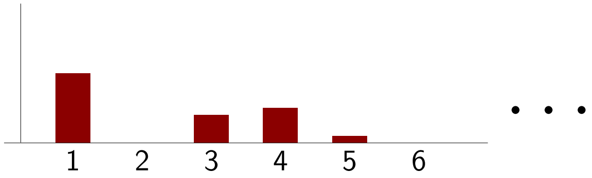
For each preterminal node i:

$x_i \sim$ Multinomial($\phi_{z_i}^E$) [terminal symbol]

HDP-PCFG: binary productions

Distribution over symbols (top-level):

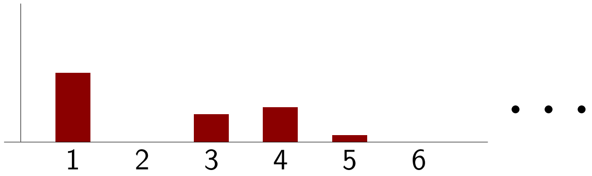
$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



HDP-PCFG: binary productions

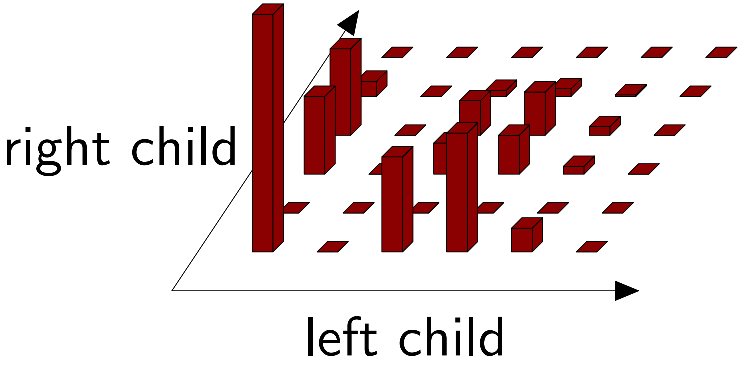
Distribution over symbols (top-level):

$$\boldsymbol{\beta} \sim \mathrm{GEM}(\alpha)$$



Mean distribution over child symbols:

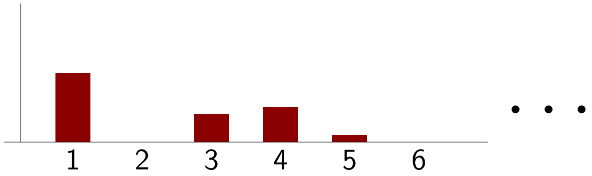
$$\boldsymbol{\beta}\boldsymbol{\beta}^T$$



HDP-PCFG: binary productions

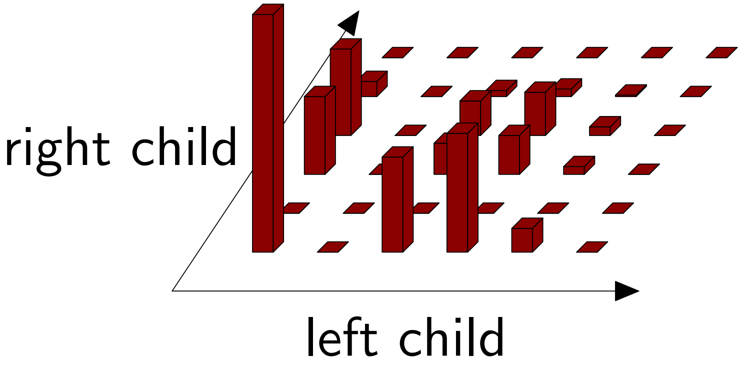
Distribution over symbols (top-level):

$$\boldsymbol{\beta} \sim \mathrm{GEM}(\alpha)$$



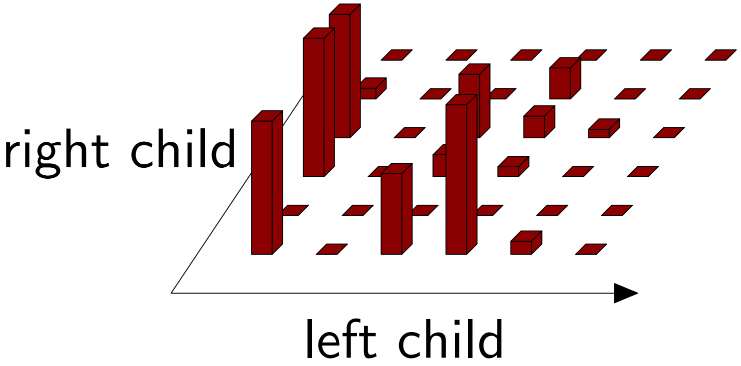
Mean distribution over child symbols:

$$\boldsymbol{\beta}\boldsymbol{\beta}^T$$



Distribution over child symbols (per-state):

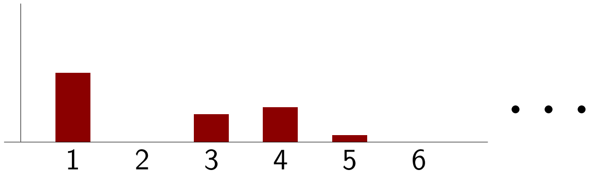
$$\phi_z^B \sim \mathrm{DP}(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$$



HDP-PCFG: binary productions

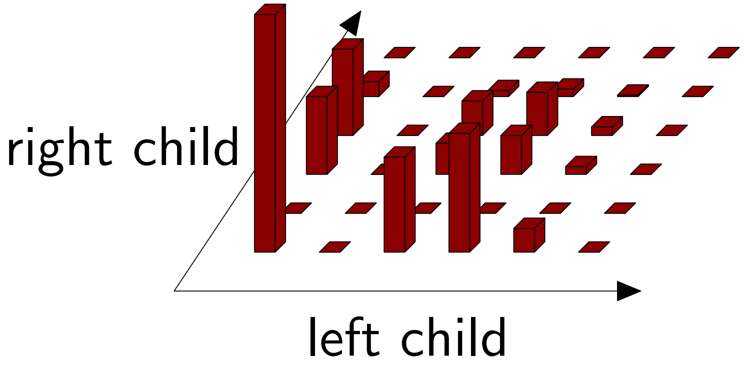
Distribution over symbols (top-level):

$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



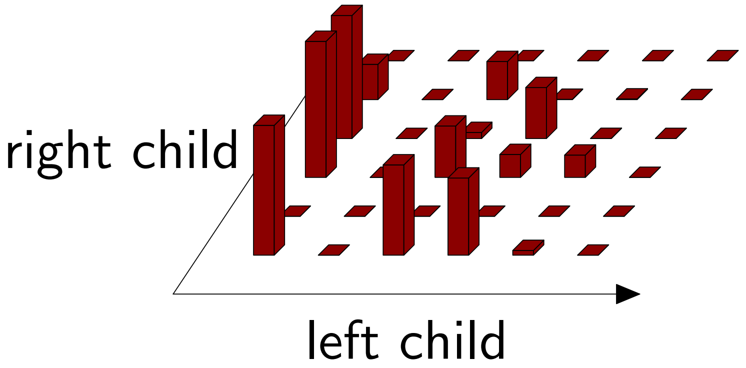
Mean distribution over child symbols:

$$\boldsymbol{\beta}\boldsymbol{\beta}^T$$



Distribution over child symbols (per-state):

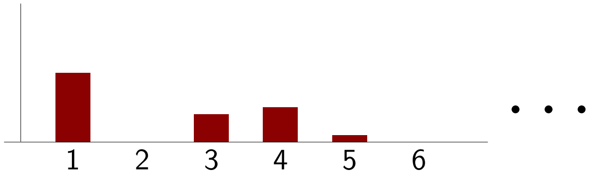
$$\phi_z^B \sim \mathsf{DP}(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$$



HDP-PCFG: binary productions

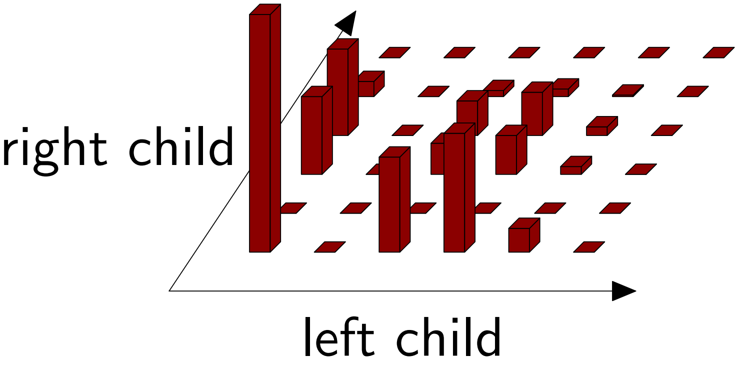
Distribution over symbols (top-level):

$$\boldsymbol{\beta} \sim \mathrm{GEM}(\alpha)$$



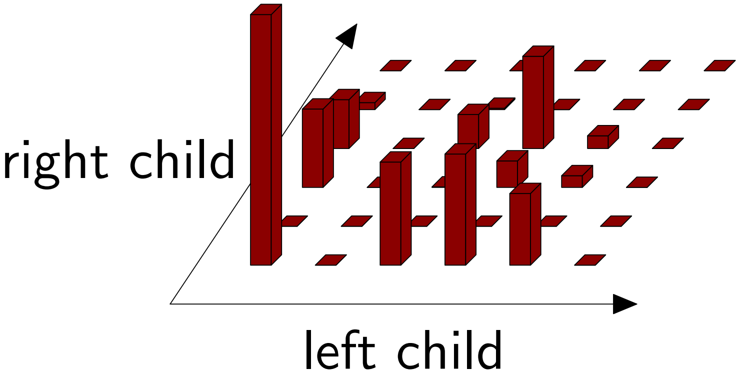
Mean distribution over child symbols:

$$\boldsymbol{\beta}\boldsymbol{\beta}^T$$



Distribution over child symbols (per-state):

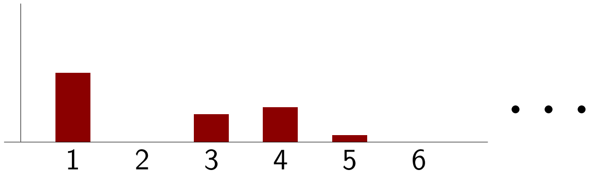
$$\phi_z^B \sim \mathrm{DP}(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$$



HDP-PCFG: binary productions

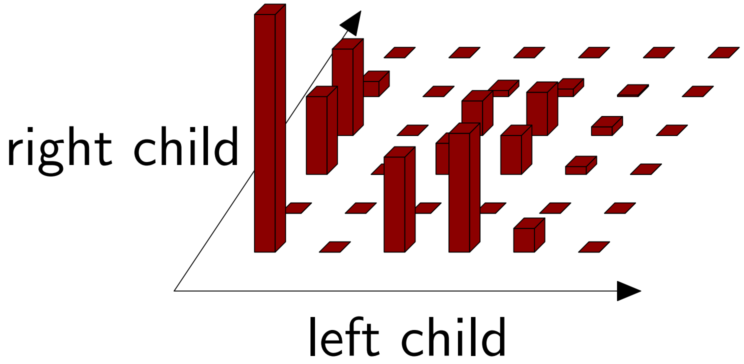
Distribution over symbols (top-level):

$$\boldsymbol{\beta} \sim \mathsf{GEM}(\alpha)$$



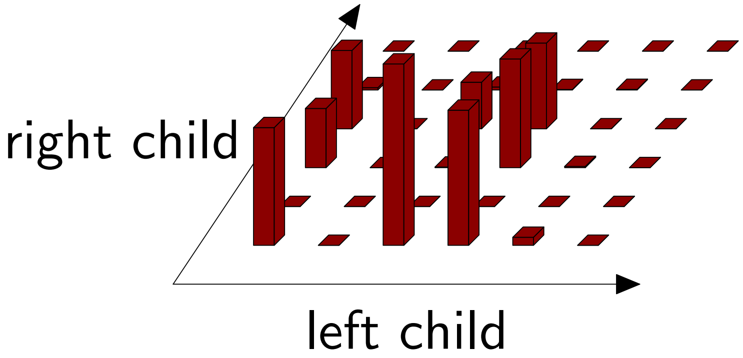
Mean distribution over child symbols:

$$\boldsymbol{\beta}\boldsymbol{\beta}^T$$

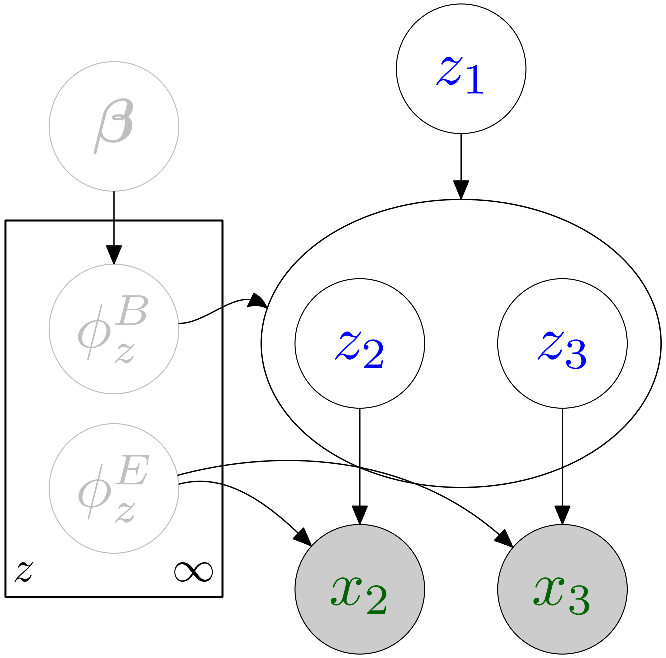


Distribution over child symbols (per-state):

$$\phi_z^B \sim \mathsf{DP}(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$$



HDP probabilistic context-free grammars



HDP-PCFG

$\beta \sim$ GEM(α) [generate distribution over symbols]

For each symbol $z \in \{1, 2, \dots\}$:

$\phi_z^E \sim$ Dirichlet(α^E) [generate emission probs]

$\phi_z^B \sim$ DP$(\alpha^B, \beta\beta^T)$ [binary production probs]

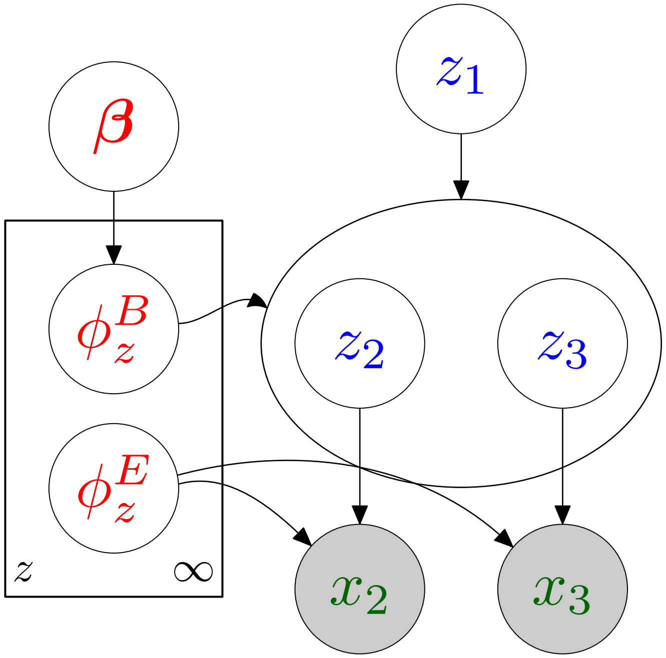
For each nonterminal node i:

$(z_{L(i)}, z_{R(i)}) \sim$ Multinomial$(\phi_{z_i}^B)$ [child symbols]

For each preterminal node i:

$x_i \sim$ Multinomial$(\phi_{z_i}^E)$ [terminal symbol]

HDP probabilistic context-free grammars



HDP-PCFG

$\boldsymbol{\beta} \sim$ GEM(α) [generate distribution over symbols]

For each symbol $z \in \{1, 2, \dots\}$:

- $\phi_z^E \sim$ Dirichlet(α^E) [generate emission probs]
- $\phi_z^B \sim$ DP$(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$ [binary production probs]

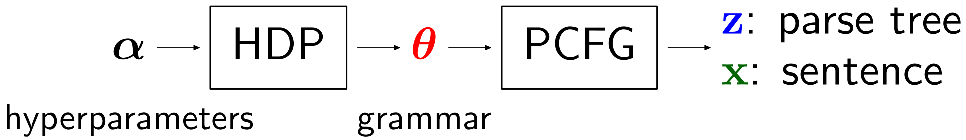
For each nonterminal node i:

- $(z_{L(i)}, z_{R(i)}) \sim$ Multinomial$(\phi_{z_i}^B)$ [child symbols]

For each preterminal node i:

- $x_i \sim$ Multinomial$(\phi_{z_i}^E)$ [terminal symbol]

Variational Bayesian inference



$\boldsymbol{\alpha} \rightarrow \boxed{\text{HDP}} \rightarrow \boldsymbol{\theta} \rightarrow \boxed{\text{PCFG}} \rightarrow$

$\mathbf{z}$: parse tree

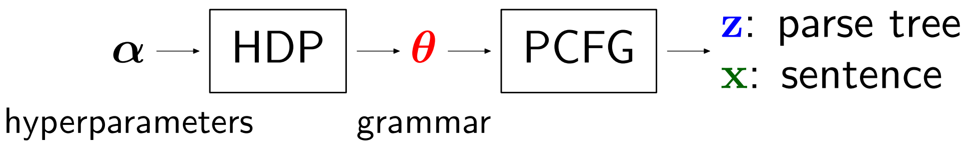
$\mathbf{x}$: sentence

hyperparameters

grammar

Goal: compute posterior $p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{x})$

Variational Bayesian inference



$\boldsymbol{\alpha} \longrightarrow$ HDP $\longrightarrow \boldsymbol{\theta} \longrightarrow$ PCFG $\longrightarrow$

$\mathbf{z}$: parse tree

$\mathbf{x}$: sentence

hyperparameters

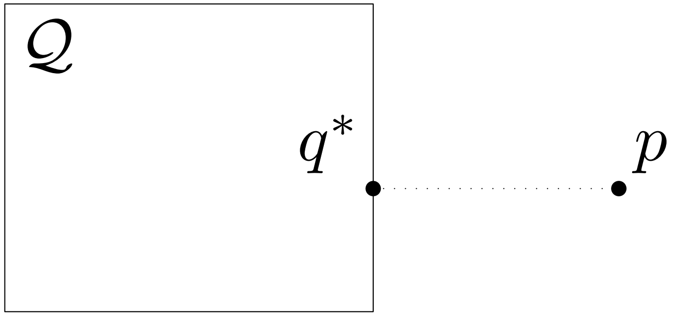
grammar

Goal: compute posterior $p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{x})$

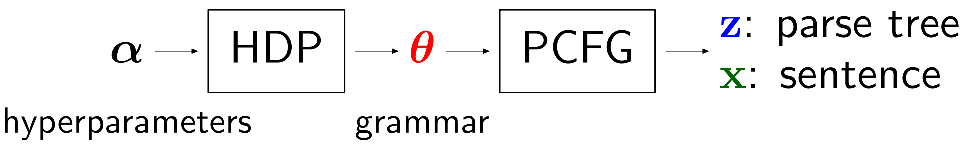
Variational inference:

approximate posterior with best from a set of tractable distributions $\mathcal{Q}$:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathrm{KL}(q \,||\, p)$$



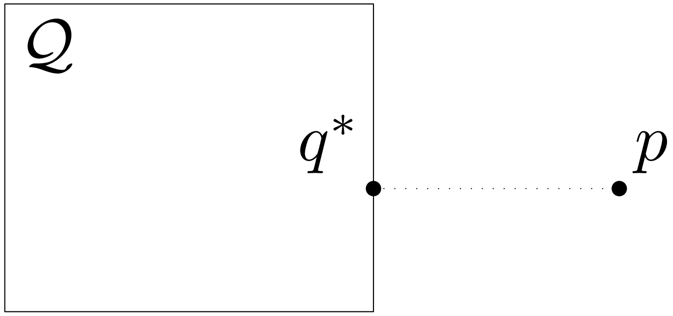
Variational Bayesian inference



Goal: compute posterior $p(\boldsymbol{\theta}, \mathbf{z} \mid \mathbf{x})$

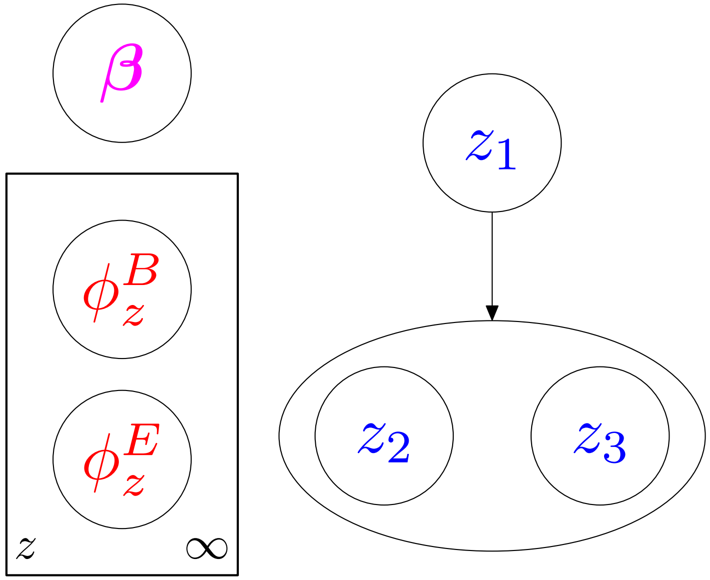
Variational inference: approximate posterior with best from a set of tractable distributions $\mathcal{Q}$:

$$q^* = \operatorname*{argmin}_{q \in \mathcal{Q}} \mathrm{KL}(q \,||\, p)$$



Mean-field approximation:

$$\mathcal{Q} = \left\{ q : q = q(\mathbf{z}) q(\boldsymbol{\beta}) q(\boldsymbol{\phi}) \right\}$$



Coordinate-wise descent algorithm

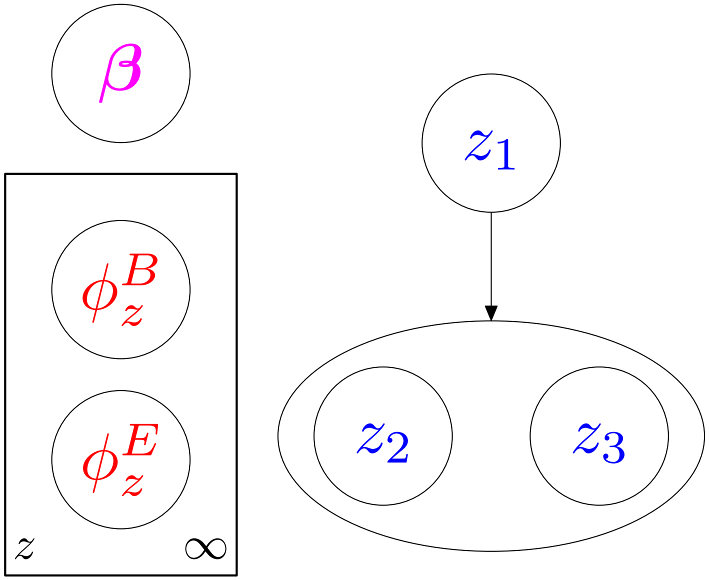
Goal: $\operatorname*{argmin}_{q \in \mathcal{Q}} \mathsf{KL}(q \,||\, p)$

$$q = q(\mathbf{z})q(\boldsymbol{\phi})q(\boldsymbol{\beta})$$

$\mathbf{z}$ = parse tree

$\boldsymbol{\phi}$ = rule probabilities

$\boldsymbol{\beta}$ = inventory of symbols



Coordinate-wise descent algorithm

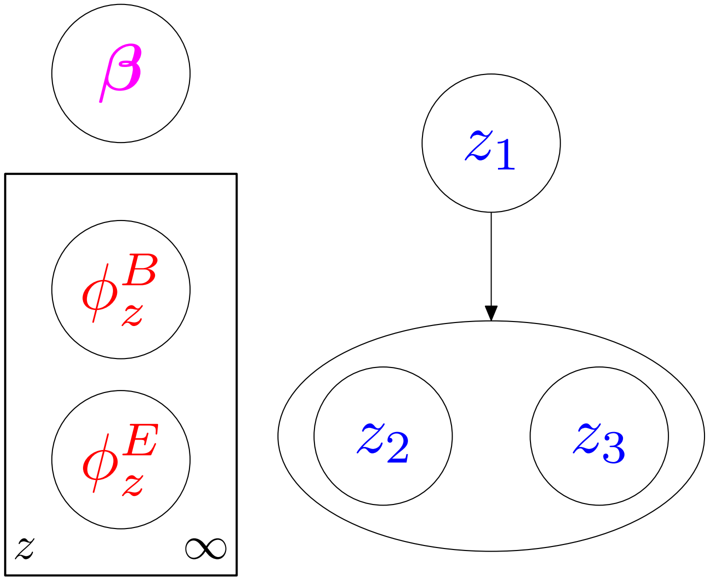
Goal: $\operatorname*{argmin}_{q \in \mathcal{Q}} \mathsf{KL}(q \,||\, p)$

$$q = q(\mathbf{z}) q(\boldsymbol{\phi}) q(\boldsymbol{\beta})$$

$\mathbf{z}$ = parse tree

$\boldsymbol{\phi}$ = rule probabilities

$\boldsymbol{\beta}$ = inventory of symbols



Iterate:

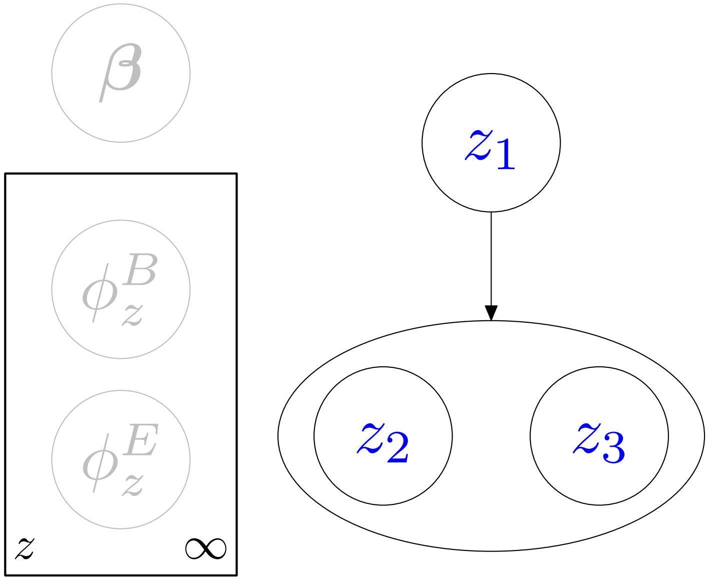
- Optimize $q(\mathbf{z})$ (E-step):
- Optimize $q(\boldsymbol{\phi})$ (M-step):
- Optimize $q(\boldsymbol{\beta})$ (no equivalent in EM):

Coordinate-wise descent algorithm

Goal: $\underset{q \in \mathcal{Q}}{\operatorname{argmin}} \, \mathsf{KL}(q \,||\, p)$

$$q = q(\mathbf{z})q(\boldsymbol{\phi})q(\boldsymbol{\beta})$$

$\mathbf{z}$ = parse tree
$\boldsymbol{\phi}$ = rule probabilities
$\boldsymbol{\beta}$ = inventory of symbols



Iterate:

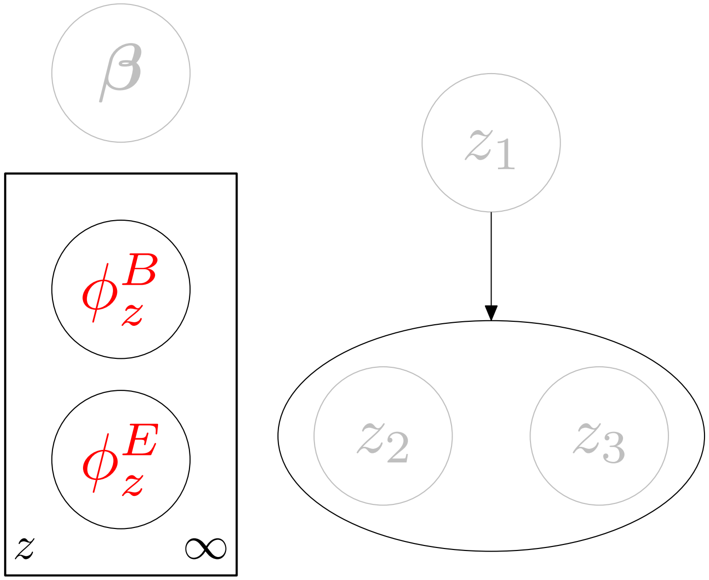
- Optimize $q(\mathbf{z})$ (E-step):
  - Inside-outside with rule weights $W(r)$
  - Gather expected rule counts $C(r)$
- Optimize $q(\boldsymbol{\phi})$ (M-step):
- Optimize $q(\boldsymbol{\beta})$ (no equivalent in EM):

Coordinate-wise descent algorithm

Goal: $\operatorname*{argmin}_{q \in \mathcal{Q}} \mathrm{KL}(q \,||\, p)$

$$q = q(\mathbf{z}) q(\boldsymbol{\phi}) q(\boldsymbol{\beta})$$

$\mathbf{z}$ = parse tree
$\boldsymbol{\phi}$ = rule probabilities
$\boldsymbol{\beta}$ = inventory of symbols



Iterate:

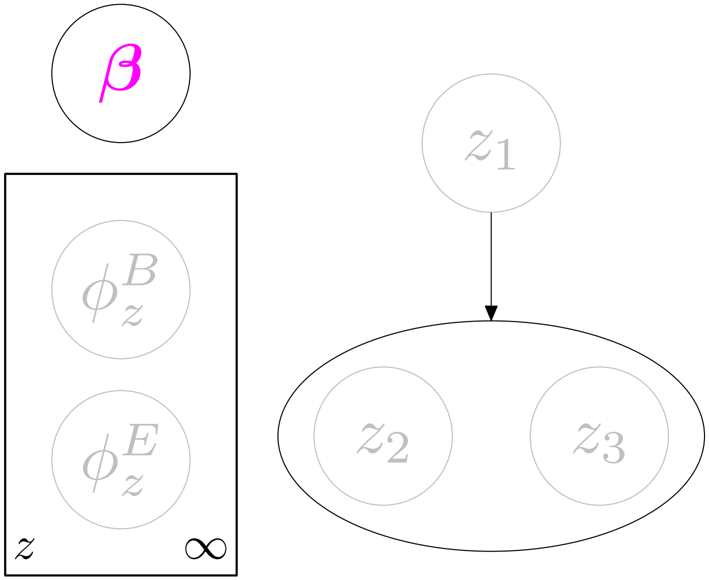
- Optimize $q(\mathbf{z})$ (E-step):
  - Inside-outside with rule weights $W(r)$
  - Gather expected rule counts $C(r)$
- Optimize $q(\boldsymbol{\phi})$ (M-step):
  - Update Dirichlet posteriors (expected rule counts + pseudocounts)
  - Compute rule weights $W(r)$
- Optimize $q(\boldsymbol{\beta})$ (no equivalent in EM):

Coordinate-wise descent algorithm

Goal: $\operatorname{argmin}_{q \in \mathcal{Q}} \mathrm{KL}(q \,||\, p)$

$$q = q(\mathbf{z})q(\boldsymbol{\phi})q(\boldsymbol{\beta})$$

$\mathbf{z}$ = parse tree
$\boldsymbol{\phi}$ = rule probabilities
$\boldsymbol{\beta}$ = inventory of symbols



Iterate:

- Optimize $q(\mathbf{z})$ (E-step):
  - Inside-outside with rule weights $W(r)$
  - Gather expected rule counts $C(r)$
- Optimize $q(\boldsymbol{\phi})$ (M-step):
  - Update Dirichlet posteriors (expected rule counts + pseudocounts)
  - Compute rule weights $W(r)$
- Optimize $q(\boldsymbol{\beta})$ (no equivalent in EM):
  - Truncate at level K (set the maximum number of symbols)
  - Use projected gradient to adapt number of symbols

Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized $\Rightarrow$ extra degree of freedom

Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized $\Rightarrow$ extra degree of freedom

EM (maximum likelihood):

$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized $\Rightarrow$ extra degree of freedom

EM (maximum likelihood):

$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

EM (maximum a posteriori):

$$W(r) = \frac{\text{prior}(r) - 1 + C(r)}{\sum_{r'} \text{prior}(r') - 1 + C(r')}$$

Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized ⇒ extra degree of freedom

EM (maximum likelihood):

$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

EM (maximum a posteriori):

$$W(r) = \frac{\text{prior}(r) - 1 + C(r)}{\sum_{r'} \text{prior}(r') - 1 + C(r')}$$

Mean-field (with DP prior):

$$W(r) = \frac{\exp \Psi(\text{prior}(r) + C(r))}{\exp \Psi(\sum_{r'} \text{prior}(r') + C(r'))}$$

Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized ⇒ extra degree of freedom

EM (maximum likelihood):

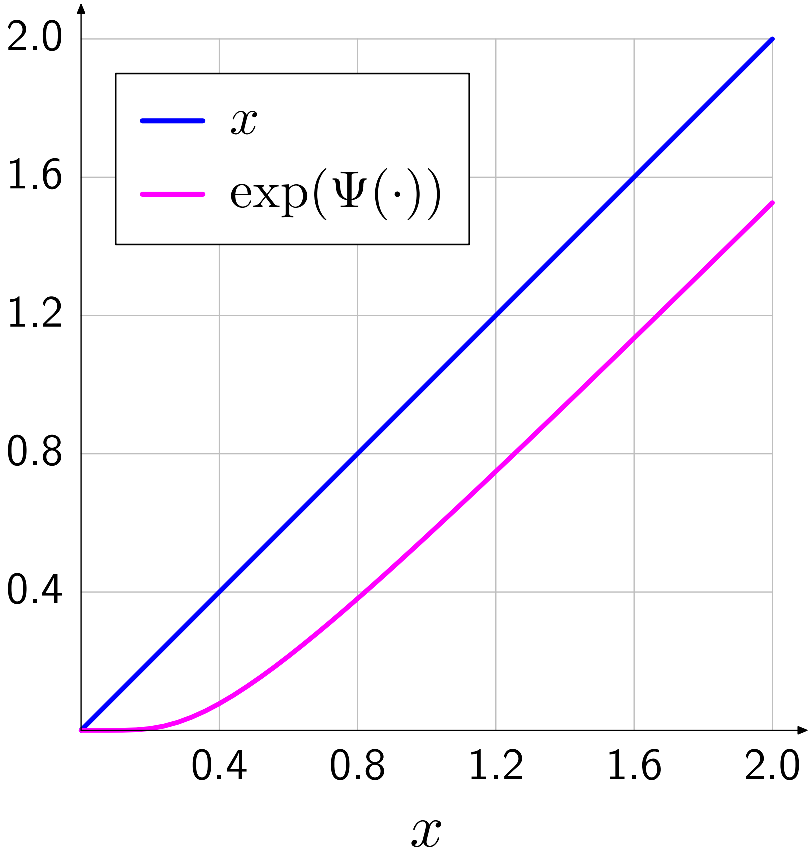
$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

EM (maximum a posteriori):

$$W(r) = \frac{\text{prior}(r) - 1 + C(r)}{\sum_{r'} \text{prior}(r') - 1 + C(r')}$$

Mean-field (with DP prior):

$$W(r) = \frac{\exp \Psi(\text{prior}(r) + C(r))}{\exp \Psi(\sum_{r'} \text{prior}(r') + C(r'))}$$



Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized ⇒ extra degree of freedom

EM (maximum likelihood):

$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

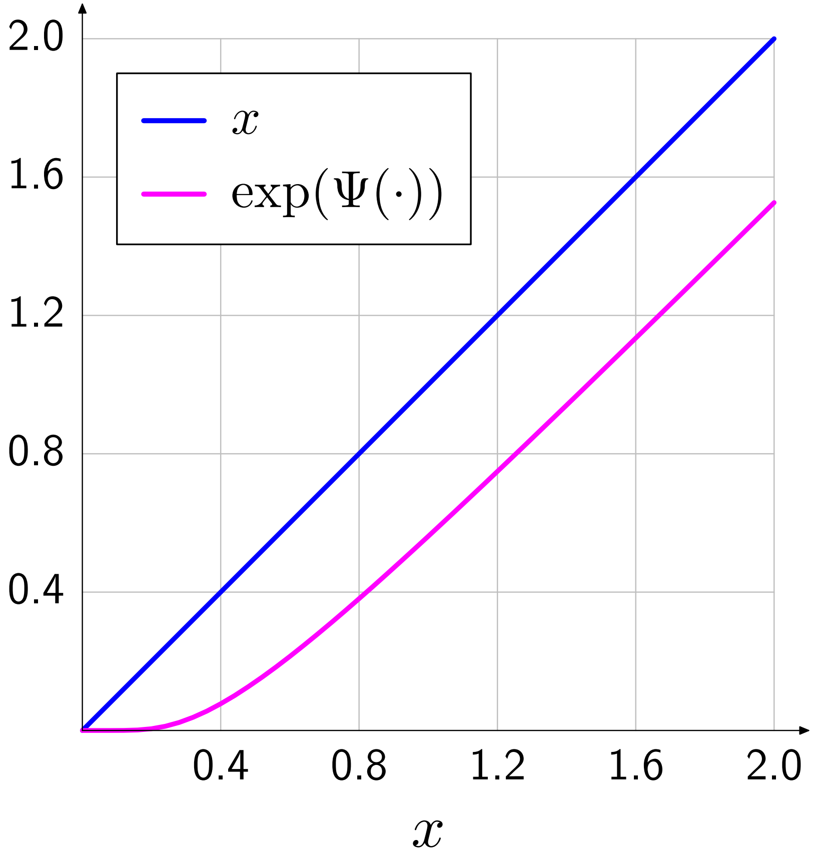
EM (maximum a posteriori):

$$W(r) = \frac{\text{prior}(r)-1+C(r)}{\sum_{r'} \text{prior}(r')-1+C(r')}$$

Mean-field (with DP prior):

$$W(r) = \frac{\exp \Psi(\text{prior}(r)+C(r))}{\exp \Psi(\sum_{r'} \text{prior}(r')+C(r'))}$$

$$\approx \frac{C(r)-0.5}{\sum_{r'} C(r')-0.5}$$



Rule weights

- Weight $W(r)$ of rule r similar to probability $p(r)$
- $W(r)$ unnormalized ⇒ extra degree of freedom

EM (maximum likelihood):

$$W(r) = \frac{C(r)}{\sum_{r'} C(r')}$$

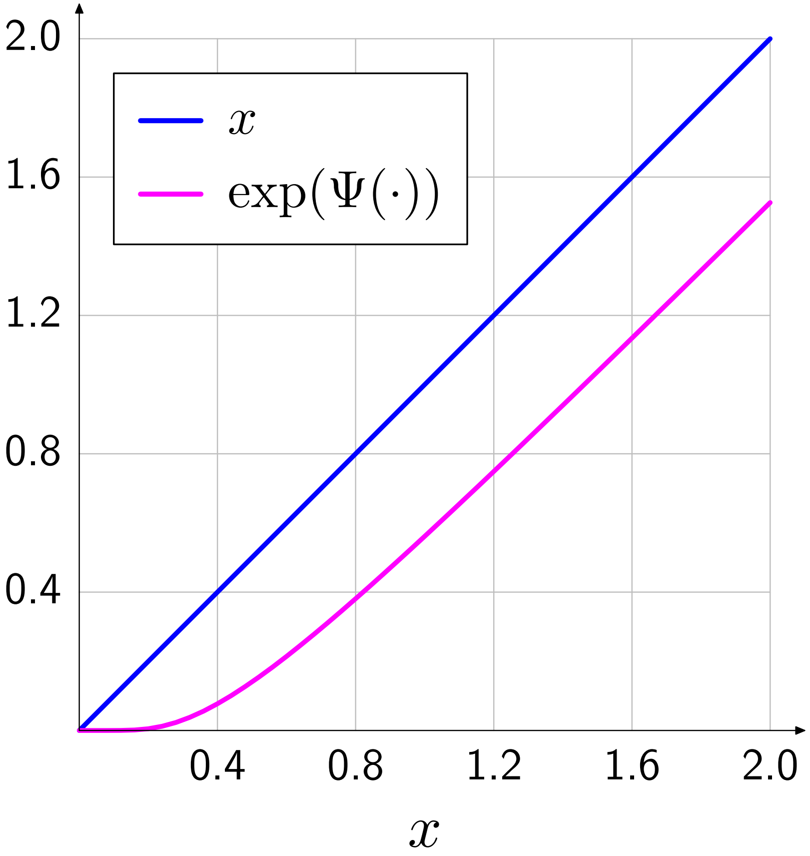
EM (maximum a posteriori):

$$W(r) = \frac{\text{prior}(r) - 1 + C(r)}{\sum_{r'} \text{prior}(r') - 1 + C(r')}$$

Mean-field (with DP prior):

$$W(r) = \frac{\exp \Psi(\text{prior}(r) + C(r))}{\exp \Psi(\sum_{r'} \text{prior}(r') + C(r'))}$$

$$\approxeq \frac{C(r) - 0.5}{\sum_{r'} C(r') - 0.5}$$



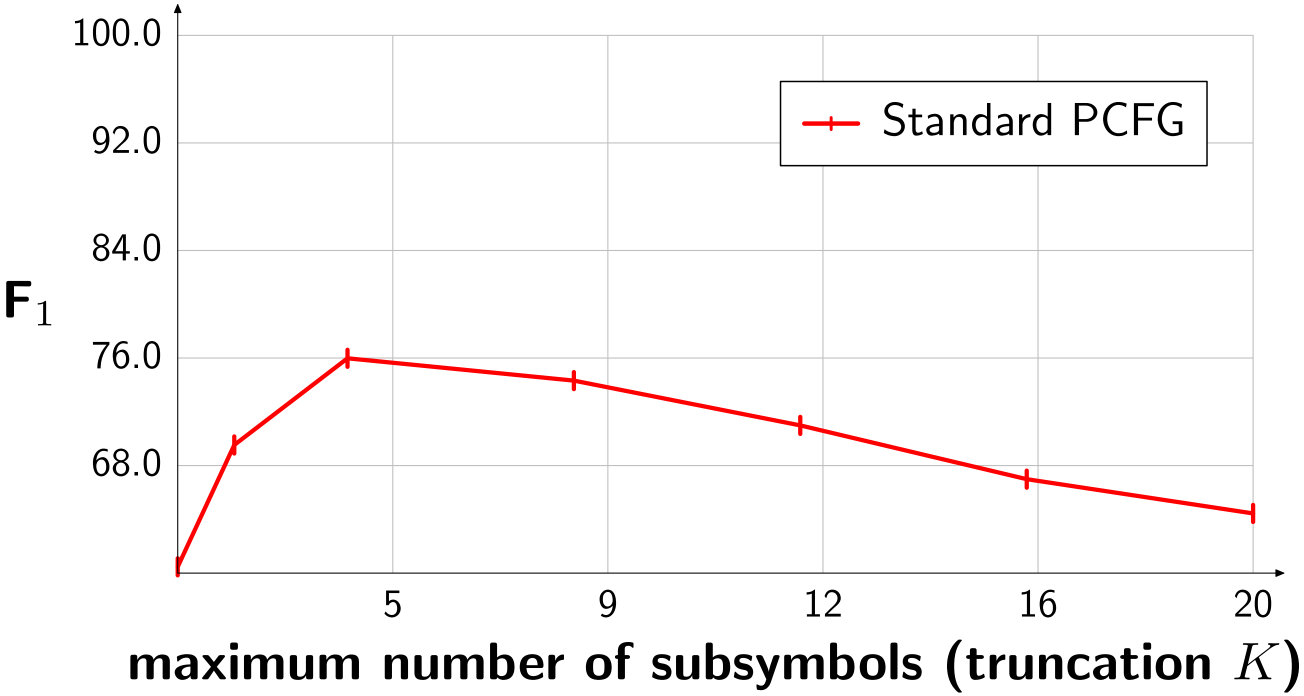
Subtract 0.5 ⇒ small counts hurt more than large counts

⇒ rich gets richer ⇒ controls number of symbols

Parsing the WSJ Penn Treebank

Setup: grammar refinement (split symbols into subsymbols)

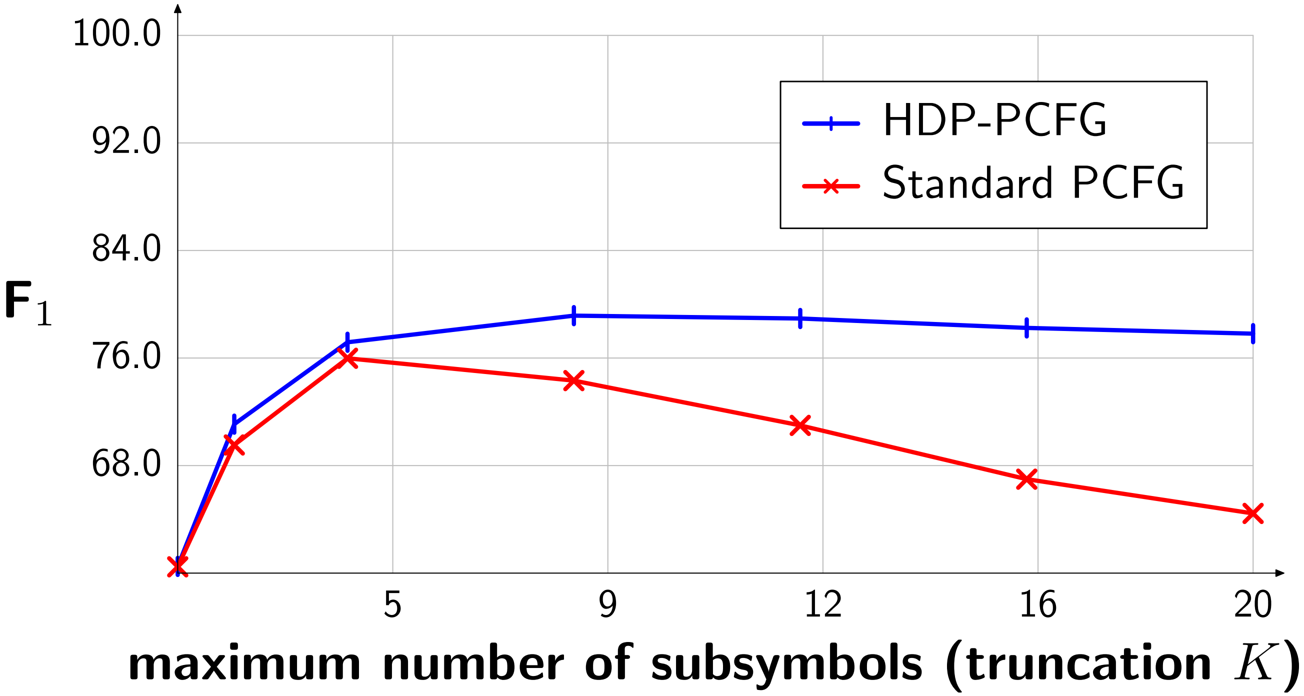
Training on one section:



Parsing the WSJ Penn Treebank

Setup: grammar refinement (split symbols into subsymbols)

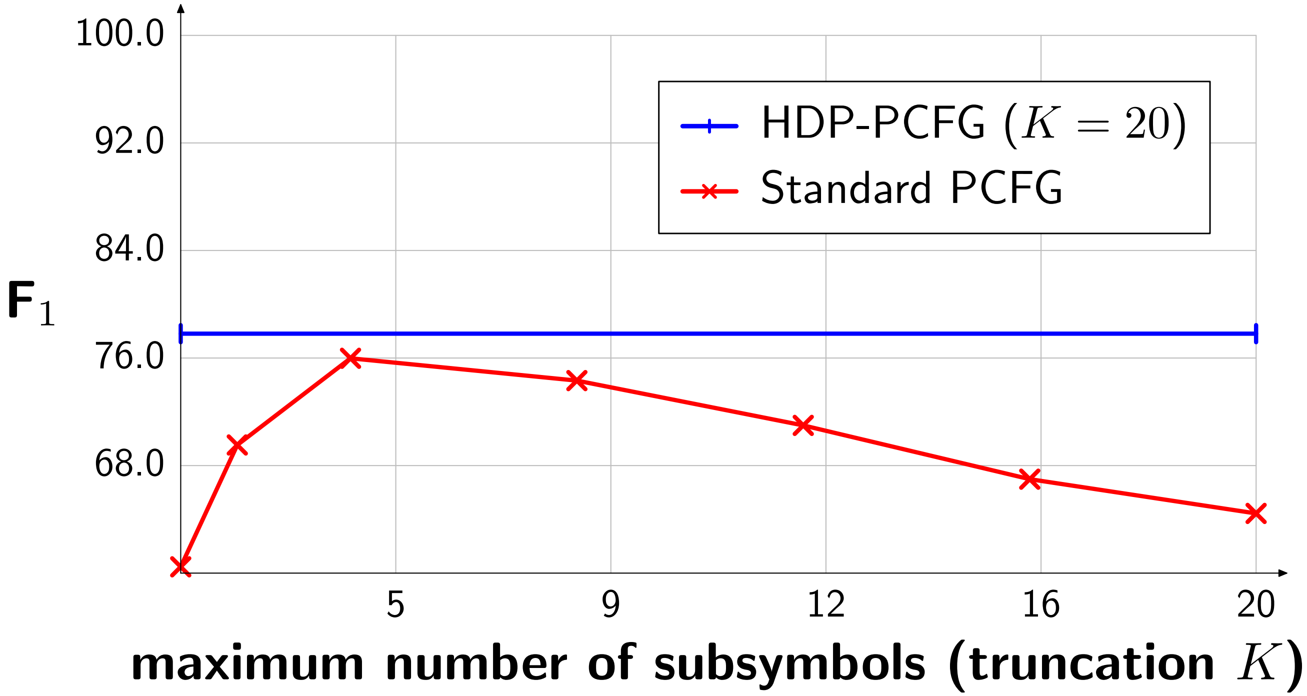
Training on one section:



Parsing the WSJ Penn Treebank

Setup: grammar refinement (split symbols into subsymbols)

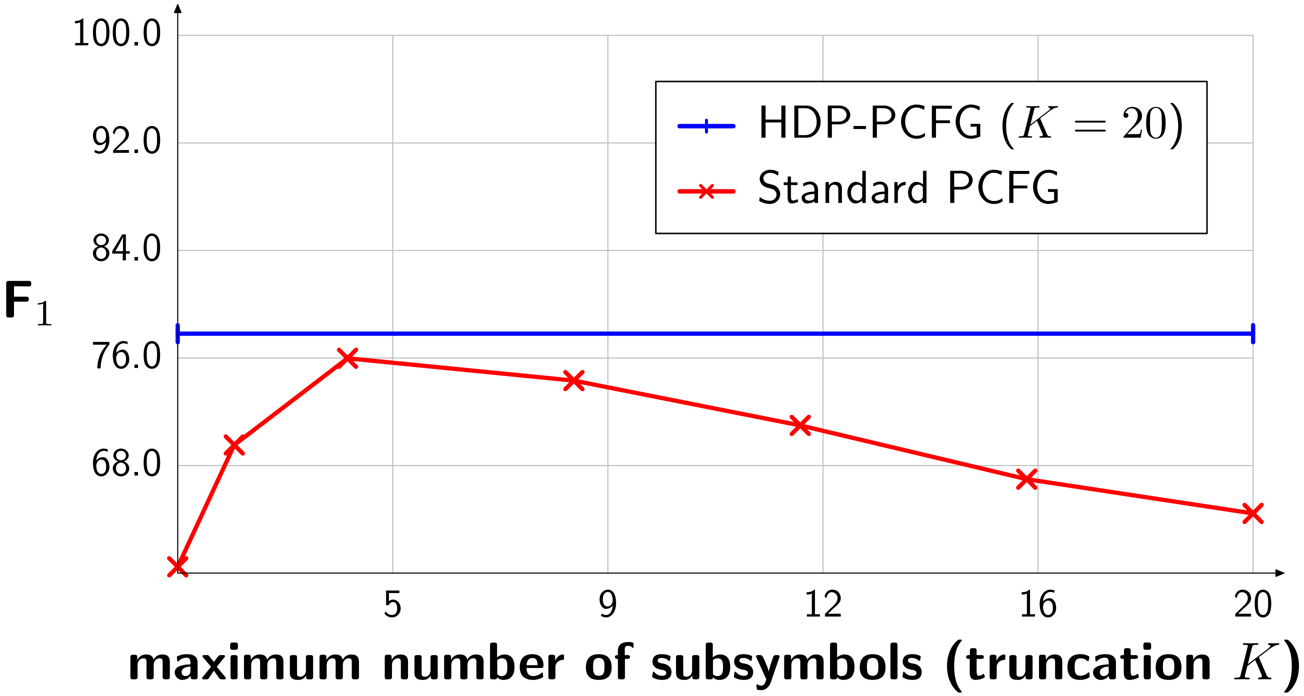
Training on one section:



Parsing the WSJ Penn Treebank

Setup: grammar refinement (split symbols into subsymbols)

Training on one section:



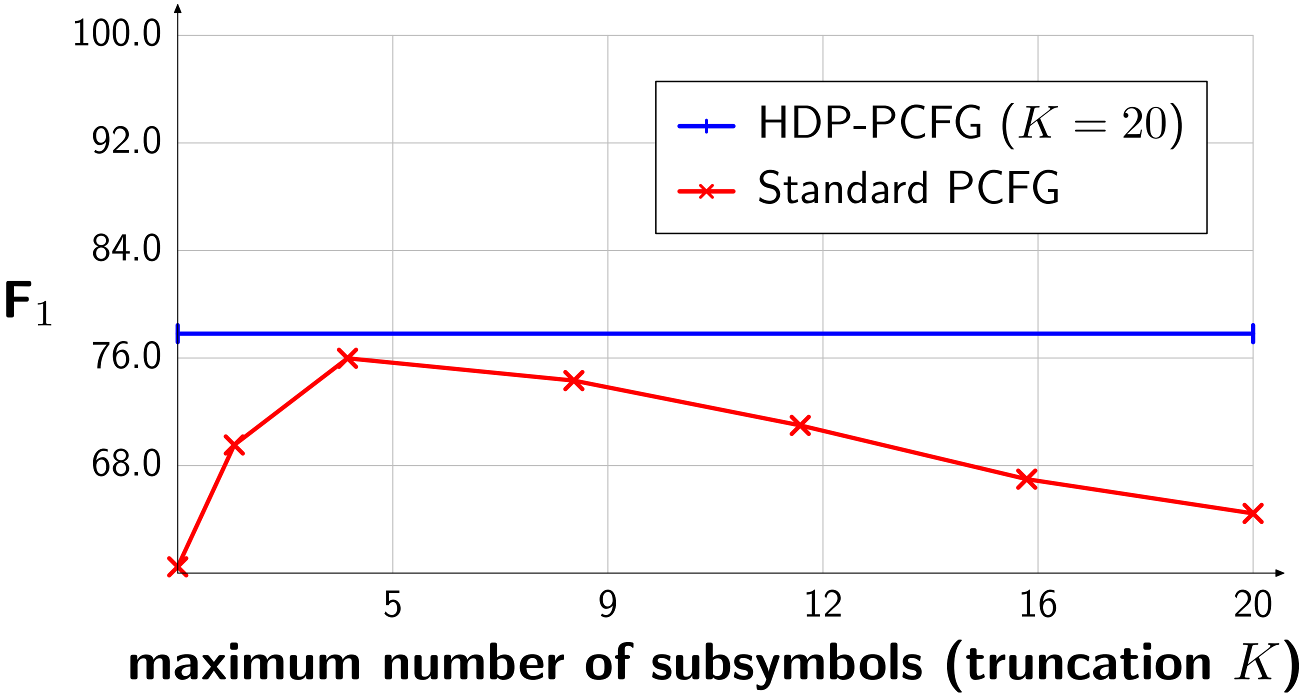
Training on 20 sections: $(K = 16)$

Standard PCFG: 86.23 HDP-PCFG: 87.08

Parsing the WSJ Penn Treebank

Setup: grammar refinement (split symbols into subsymbols)

Training on one section:



Training on 20 sections: $(K = 16)$

Standard PCFG: 86.23 HDP-PCFG: 87.08

Results:

- HDP-PCFG overfits less than standard PCFG
- If have large amounts of data, HDP-PCFG $\approxeq$ standard PCFG

Conclusions

- What? HDP-PCFG model
  allows number of grammar symbols to adapt to data
- How? Mean-field algorithm (variational inference)
  simple, efficient, similar to EM
- When? Have small amounts of data
  overfits less than standard PCFG
- Why? Declarative framework
  Grammar complexity specified declaratively in model
  rather than in learning procedure

S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise
S → NP VP
NP → PRP
VP → VBD NP
NP → DT NN
PRP → she
VBD → heard
DT → the
NN → noise