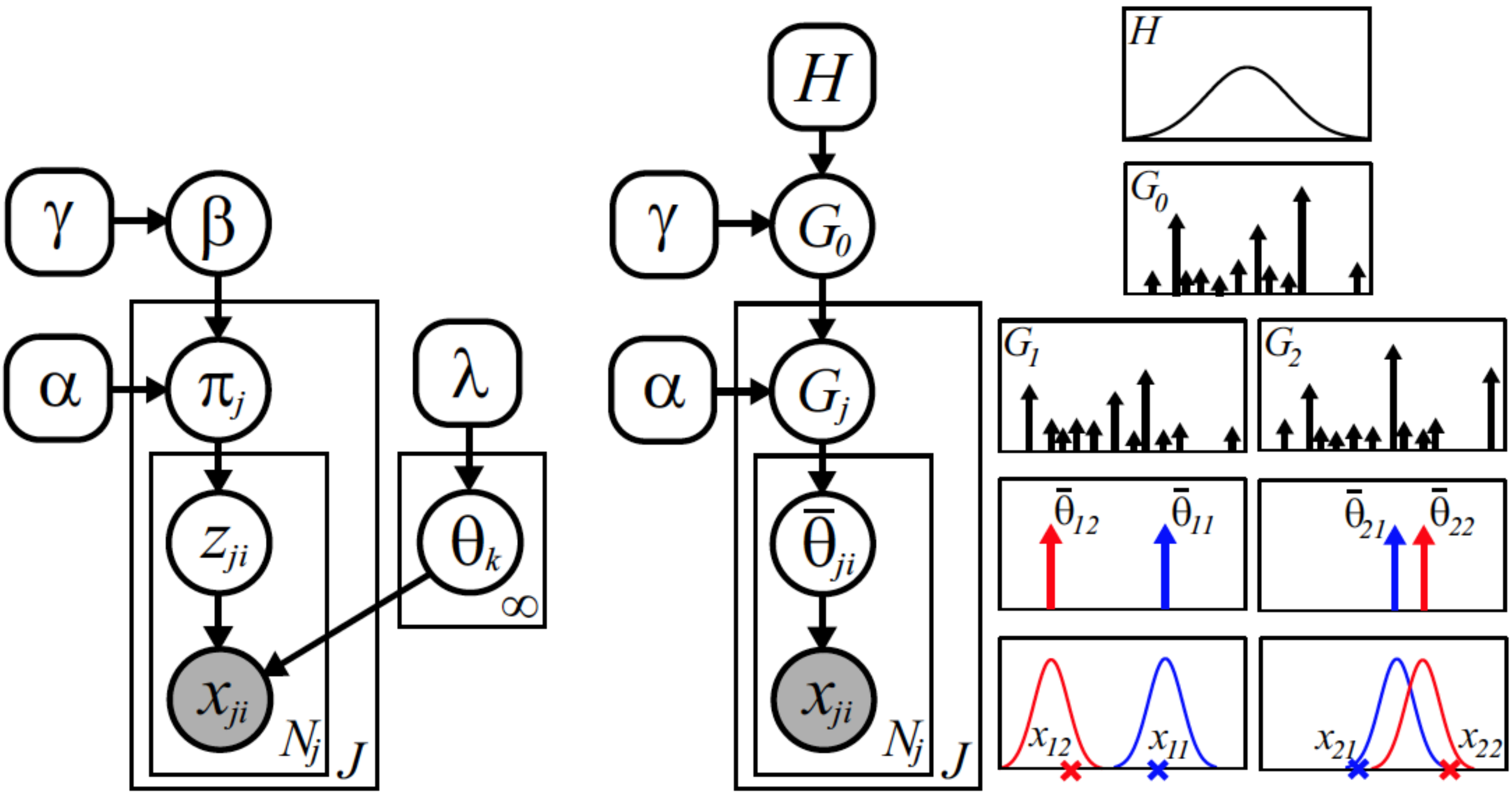
Applied Bayesian Nonparametrics

Special Topics in Machine Learning
Brown University CSCI 2950-P, Fall 2011

October 27: Pitman-Yor Processes, Infinite Markov Models

Hierarchical Dirichlet Process



Hierarchical Dirichlet Process

$$G_0(\theta) = \sum_{k=1}^{\infty} \beta_k \delta(\theta, \theta_k)$$

$$G_j(\theta) = \sum_{t=1}^{\infty} \widetilde{\pi}_{jt} \delta(\theta, \widetilde{\theta}_{jt})$$

$$G_j(\theta) = \sum_{k=1}^{\infty} \pi_{jk} \delta(\theta, \theta_k)$$

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

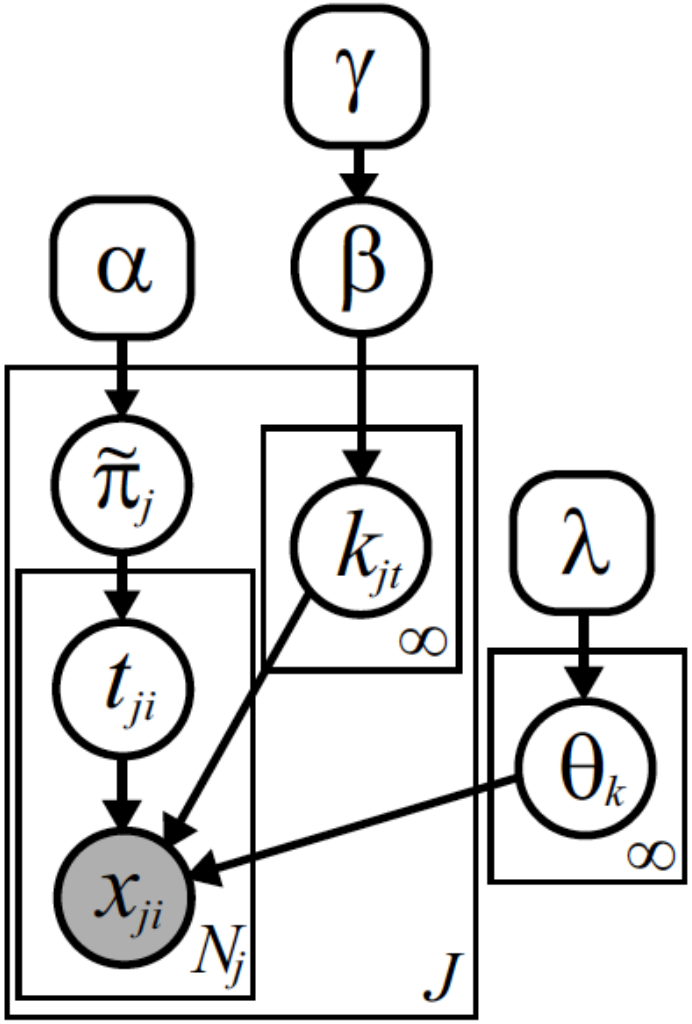
$$\theta_k \sim H(\lambda) \qquad k = 1, 2, \ldots$$

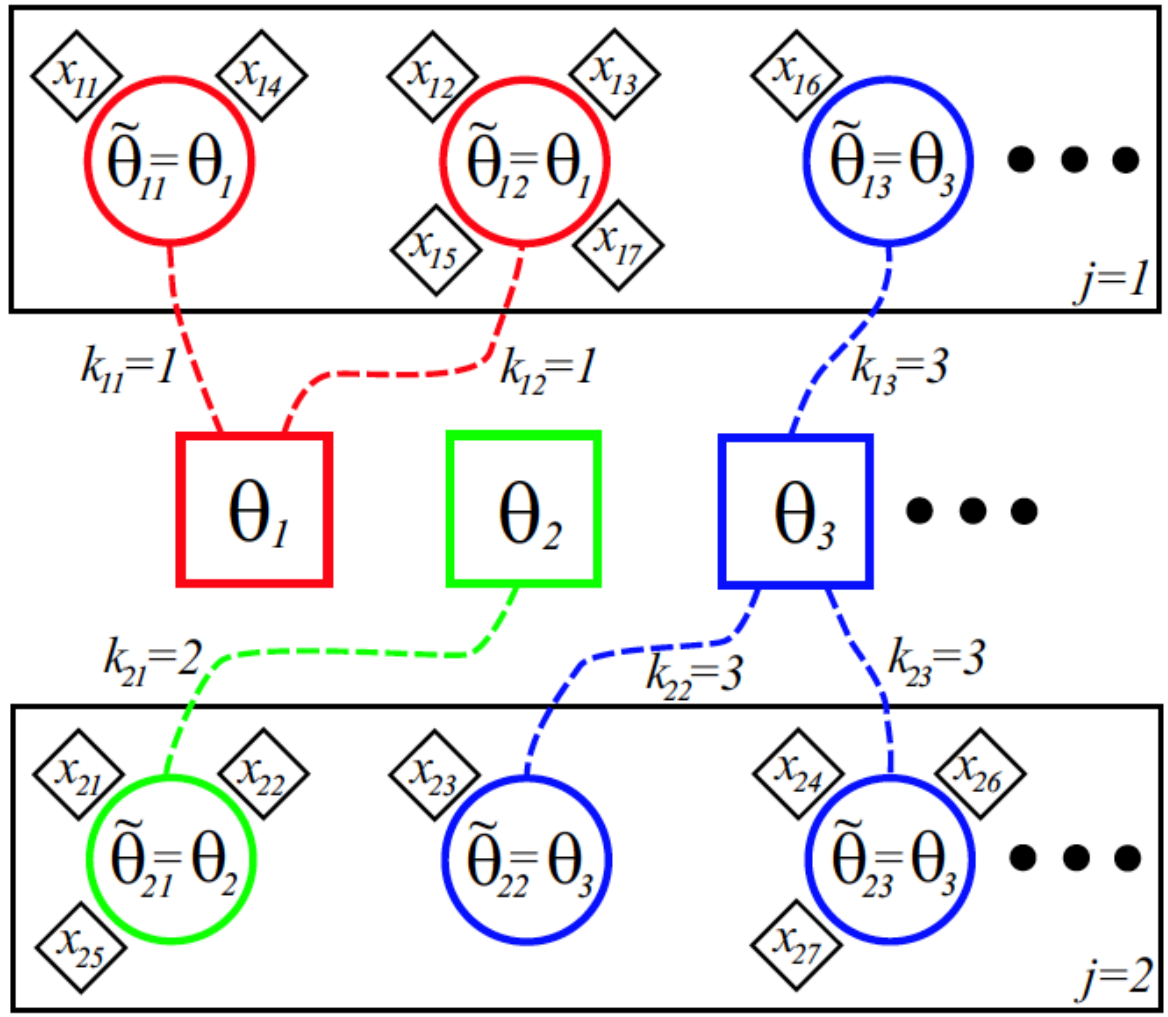
$$\widetilde{\boldsymbol{\pi}}_j \sim \mathrm{GEM}(\alpha)$$

$$\widetilde{\theta}_{jt} \sim G_0 \qquad t = 1, 2, \ldots$$

$$\pi_{jk} = \sum_{t|k_{jt}=k} \widetilde{\pi}_{jt}$$

Chinese Restaurant Franchise





$$p(t_{ji} \mid t_{j1}, \ldots, t_{ji-1}, \alpha) \propto \sum_{t} N_{jt}\delta(t_{ji}, t) + \alpha\delta(t_{ji}, \bar{t})$$

$$p(k_{jt} \mid \mathbf{k}_1, \ldots, \mathbf{k}_{j-1}, k_{j1}, \ldots, k_{jt-1}, \gamma) \propto \sum_{k} M_k\delta(k_{jt}, k) + \gamma\delta(k_{jt}, \bar{k})$$

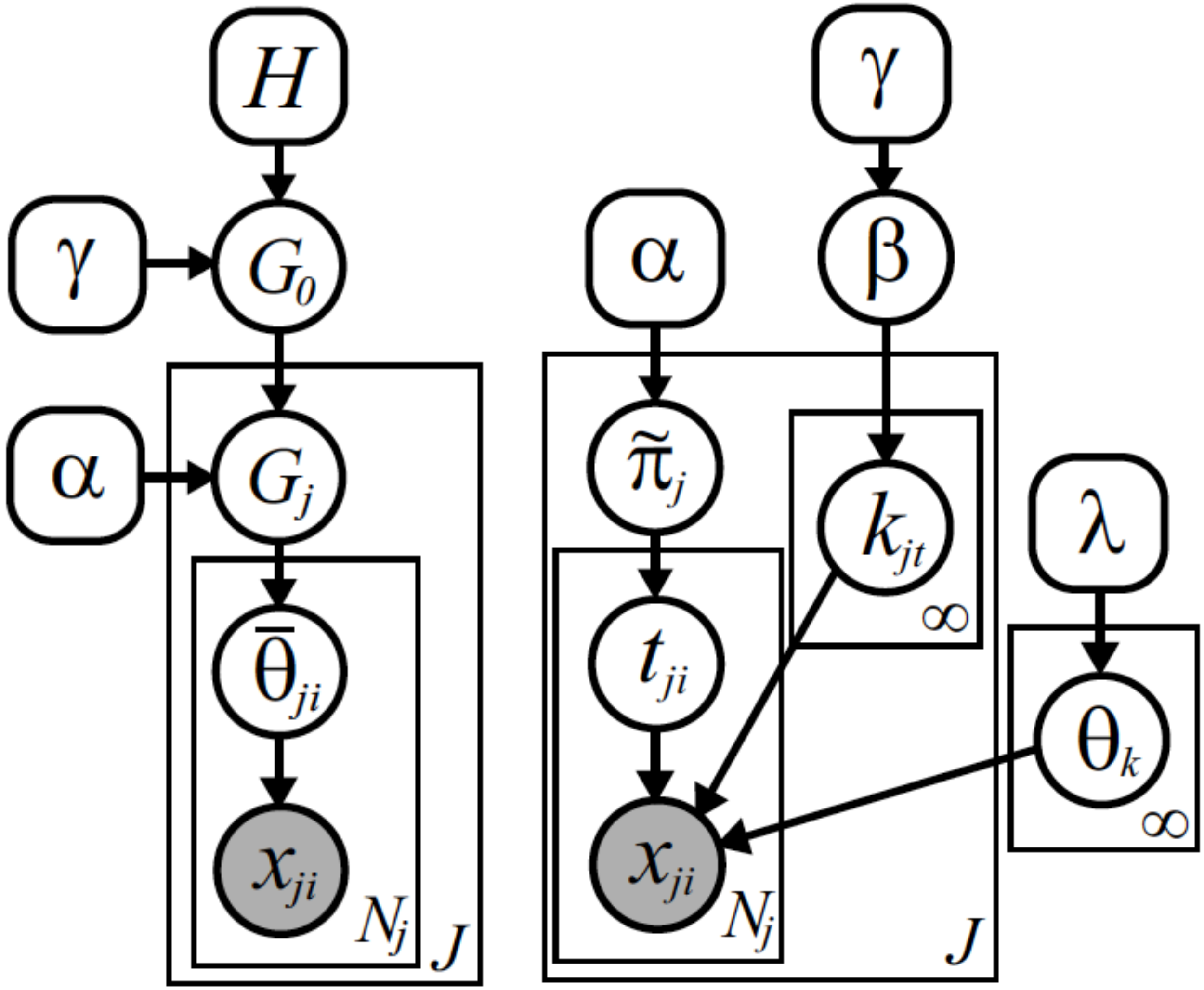
Views of Hierarchical Dirichlet

Construction repeated at each level of hierarchy:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$$

$$\beta_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell)$$

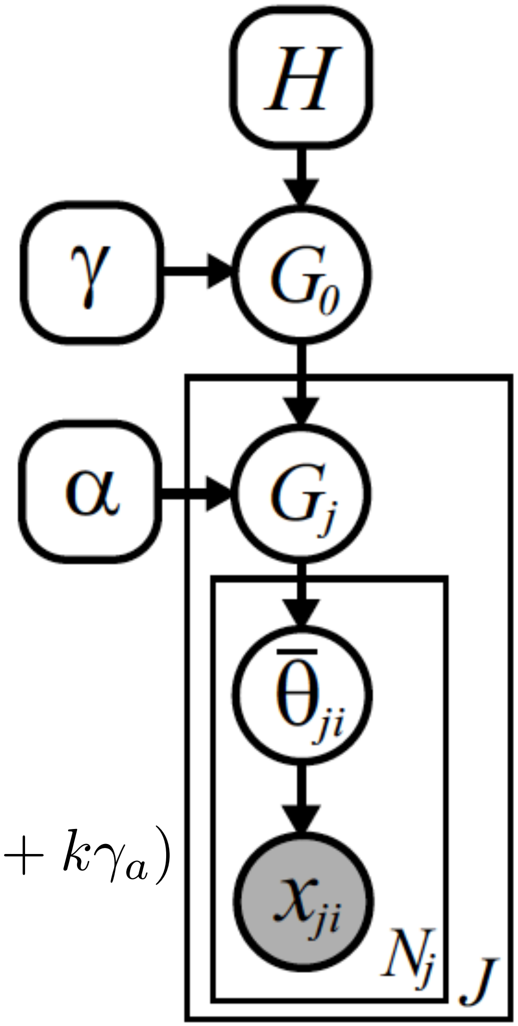
$$v_k \sim \mathrm{Beta}(1, \gamma)$$

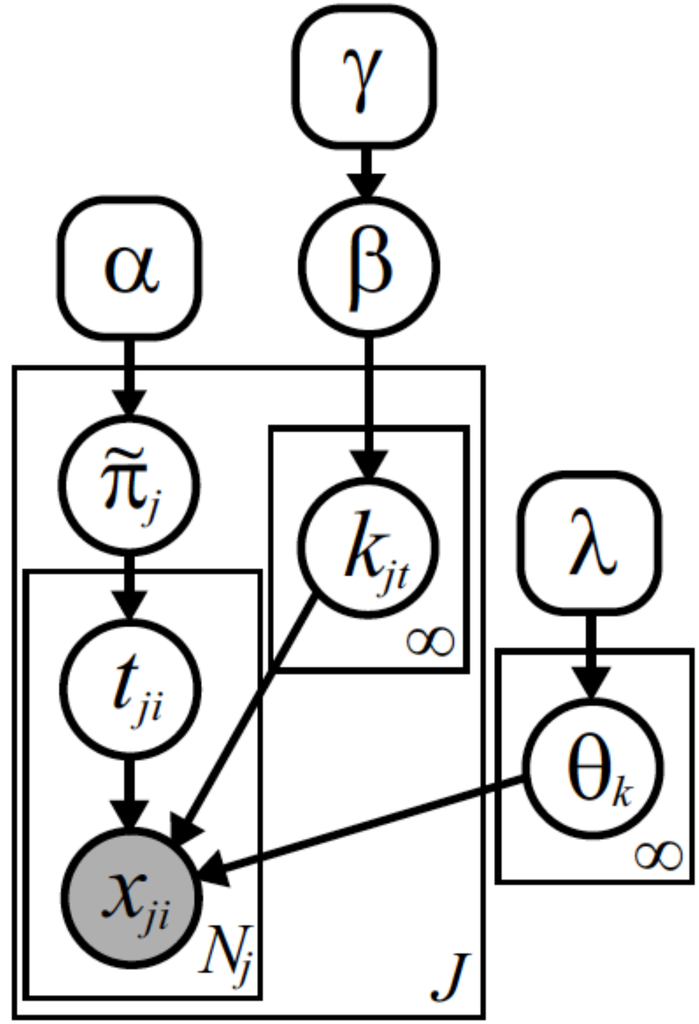


For some ordering of all tables in all groups:

$$p(k_{N+1} \mid k_N, \ldots, k_1, \gamma) = \frac{1}{N+\gamma}\left[\sum_{k=1}^{K} M_k \delta(k_{N+1}, k) + \gamma\delta(k_{N+1}, K+1)\right]$$

Views of Hierarchical Pitman-Yor





Construction repeated at each level of hierarchy:

$$G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}$$

$$\beta_k = v_k \prod_{\ell=1}^{k-1} (1 - v_\ell)$$

$$v_k \sim \mathrm{Beta}(1 - \gamma_a, \gamma_b + k\gamma_a)$$

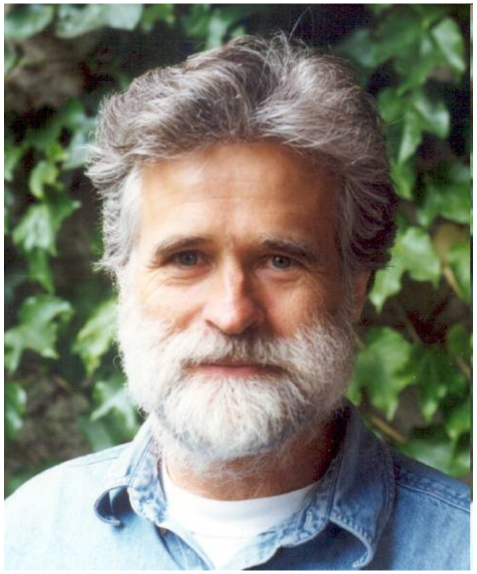
For some ordering of all tables in all groups:

$$p(k_{N+1} \mid k_N, \ldots, k_1, \gamma) = \frac{1}{N + \gamma_b} \left[\sum_{k=1}^{K} (M_k - \gamma_a) \delta(k_{N+1}, k) + (\gamma_b + K\gamma_a) \delta(k_{N+1}, K+1) \right]$$

Why Pitman-Yor?

Generalizing the Dirichlet Process

- Distribution on partitions leads to a generalized *Chinese restaurant process*
- Special cases arise as excursion lengths for Markov chains, Brownian motions, …



Jim Pitman

Power Law Distributions

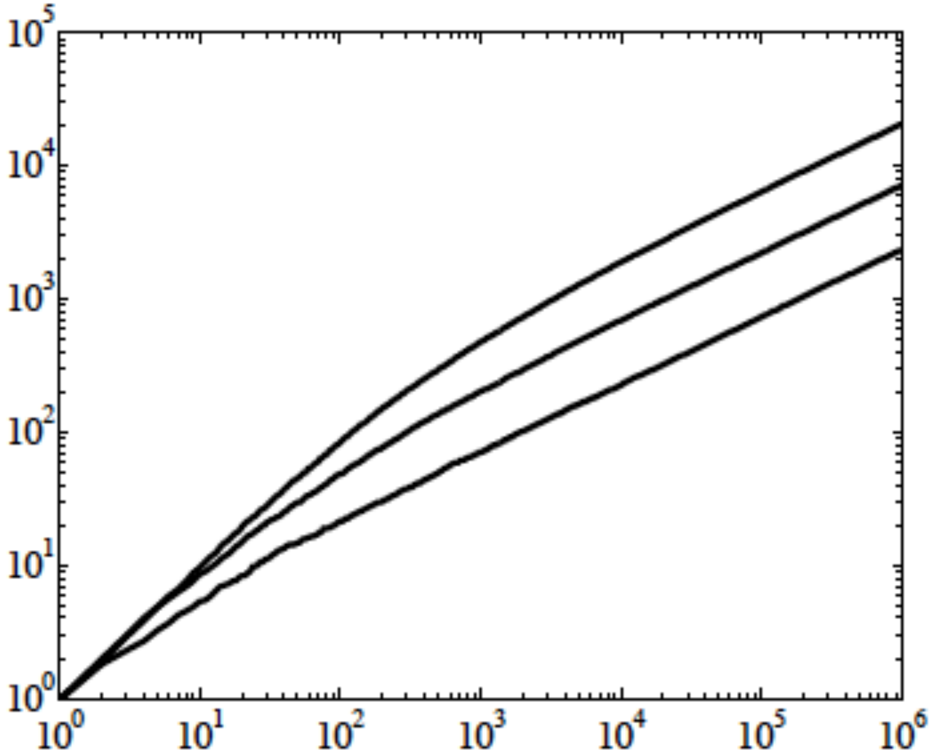
	DP	**PY**
Number of unique clusters in N observations	$\mathcal{O}(b \log N)$	***Heaps' Law:*** $\mathcal{O}(bN^a)$
Size of sorted cluster weight k	$\mathcal{O}\left(\alpha_b \left(\frac{1+b}{b}\right)^{-k}\right)$	***Zipf's Law:*** $\mathcal{O}\left(\alpha_{ab} k^{-\frac{1}{a}}\right)$

Marc Yor

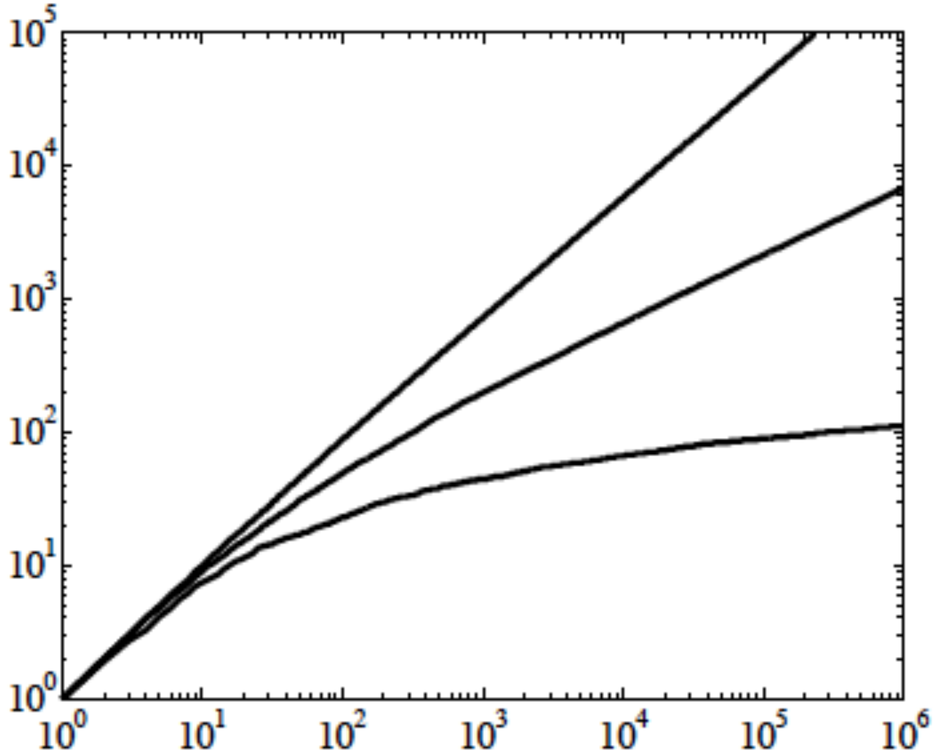
Natural Language Statistics

Goldwater, Griffiths, & Johnson, 2005
Teh, 2006

Number of Unique Words

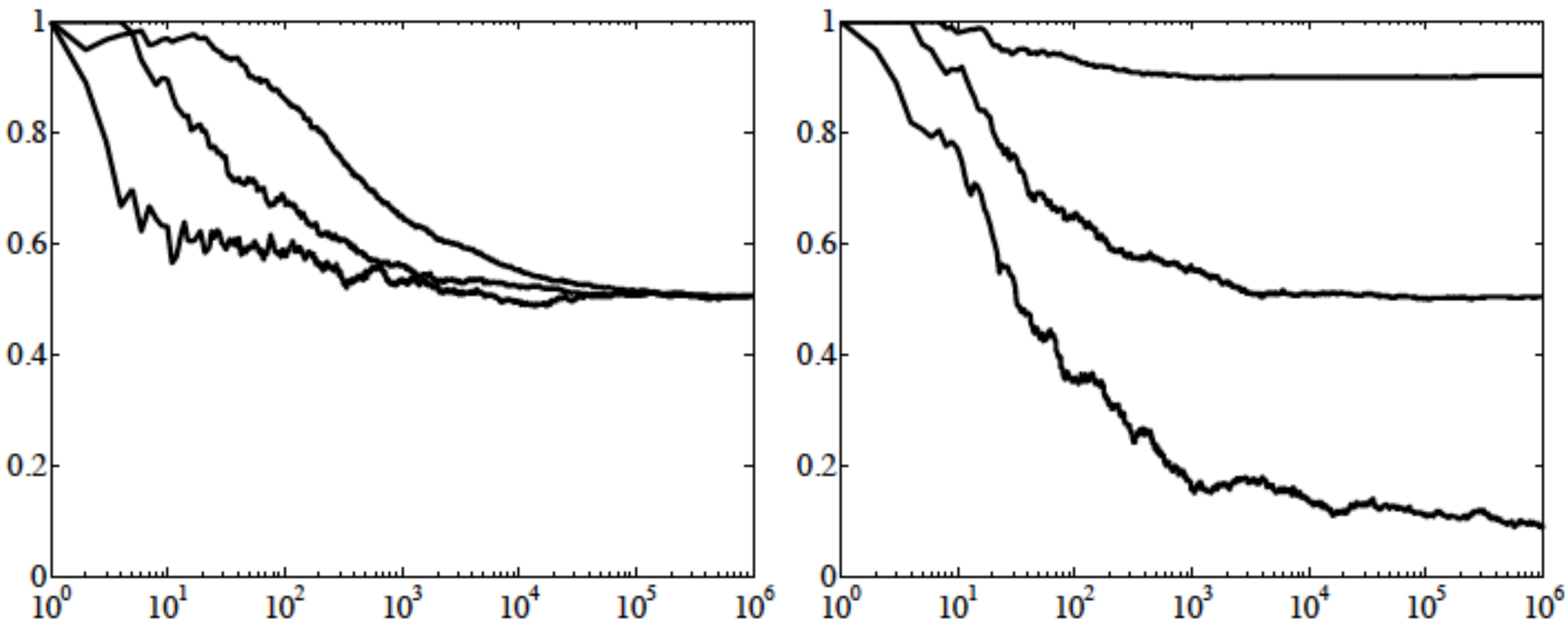


Fixed discount parameter



Fixed concentration parameter

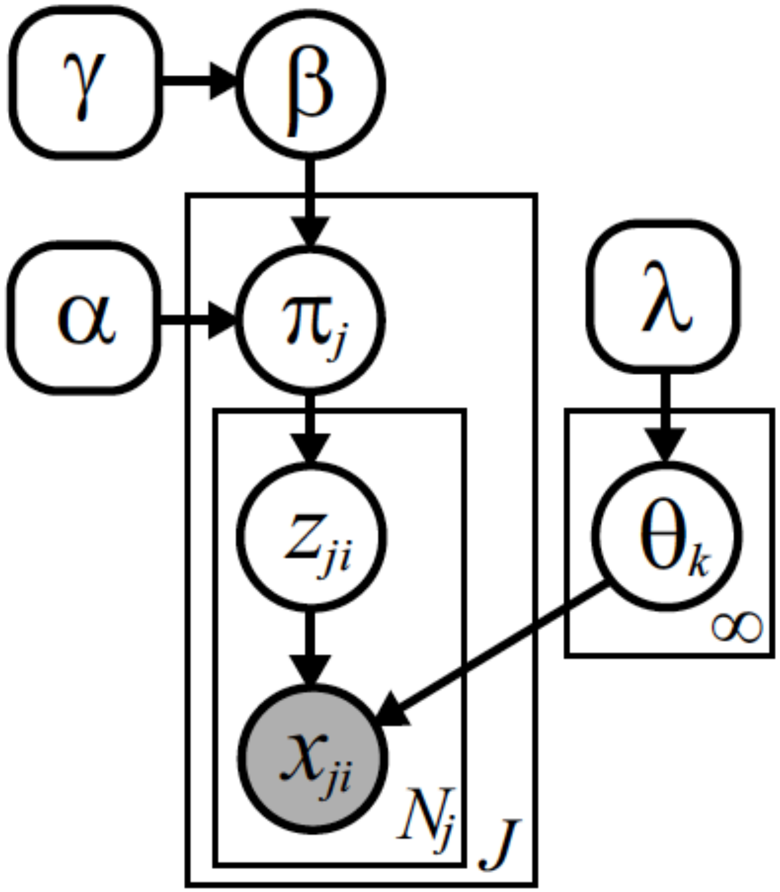
Fraction of Words Appearing Once



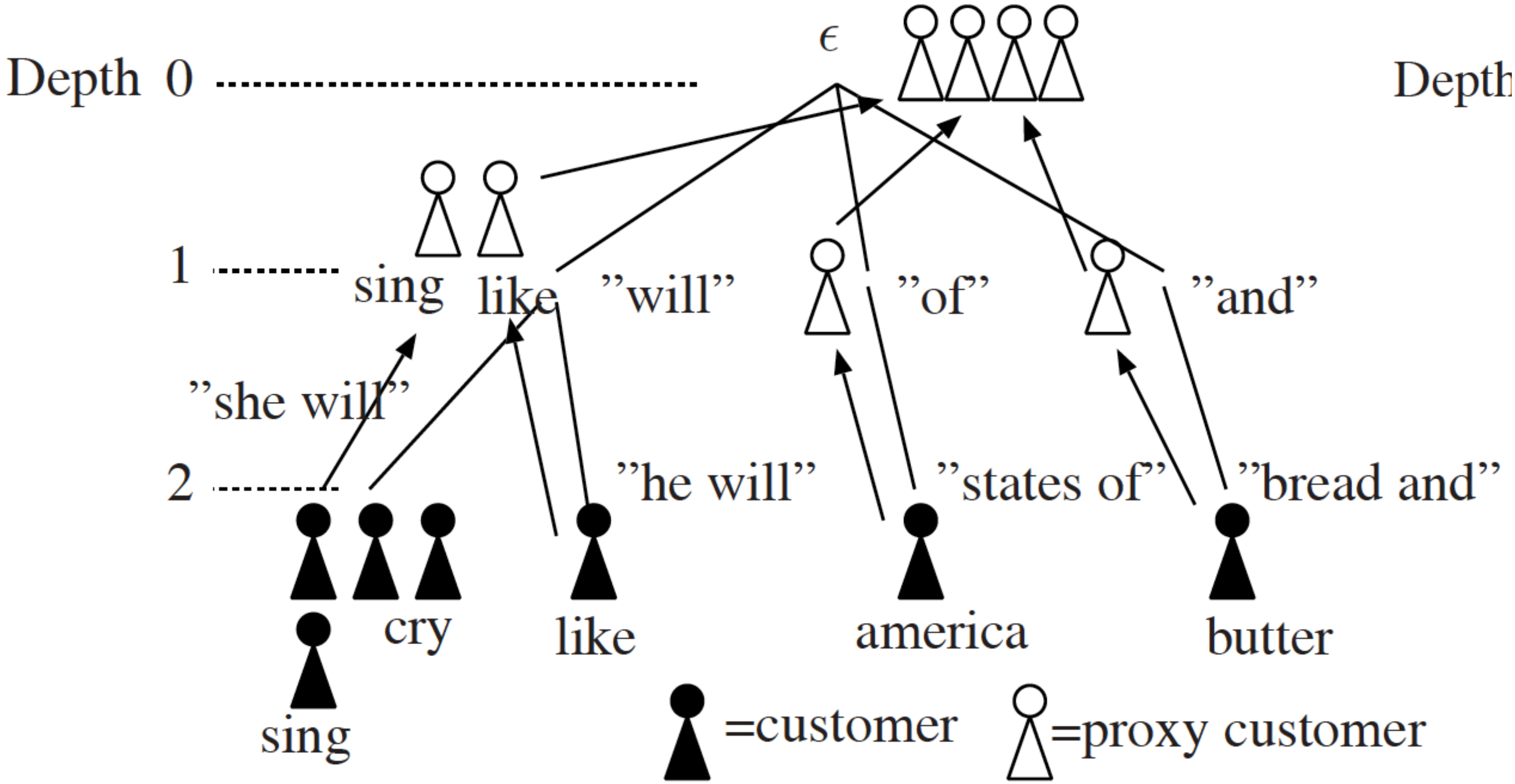
Fixed discount parameter

Fixed concentration parameter

A Third View: Tractable for PY?

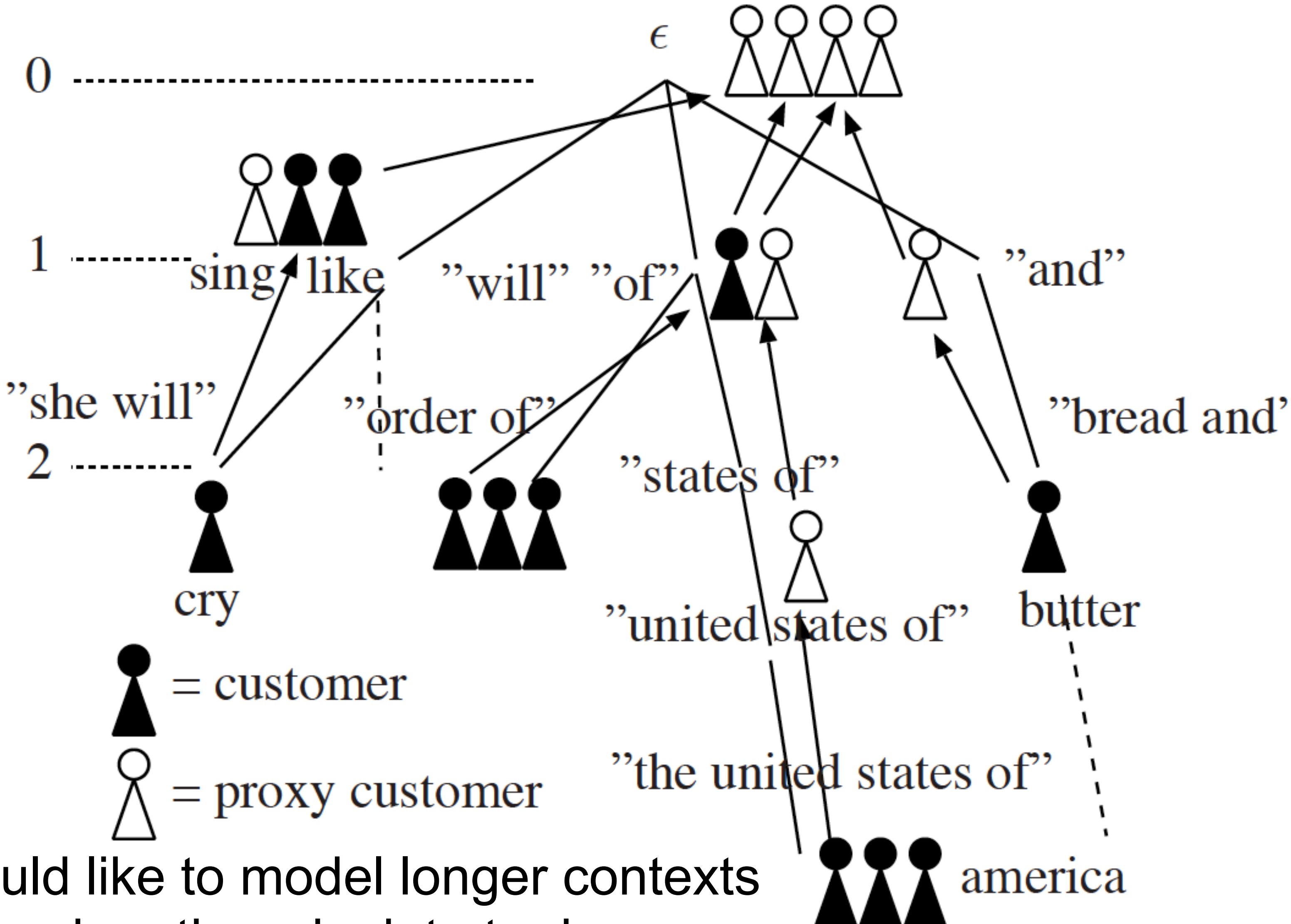


Fixed-Order Suffix Tree



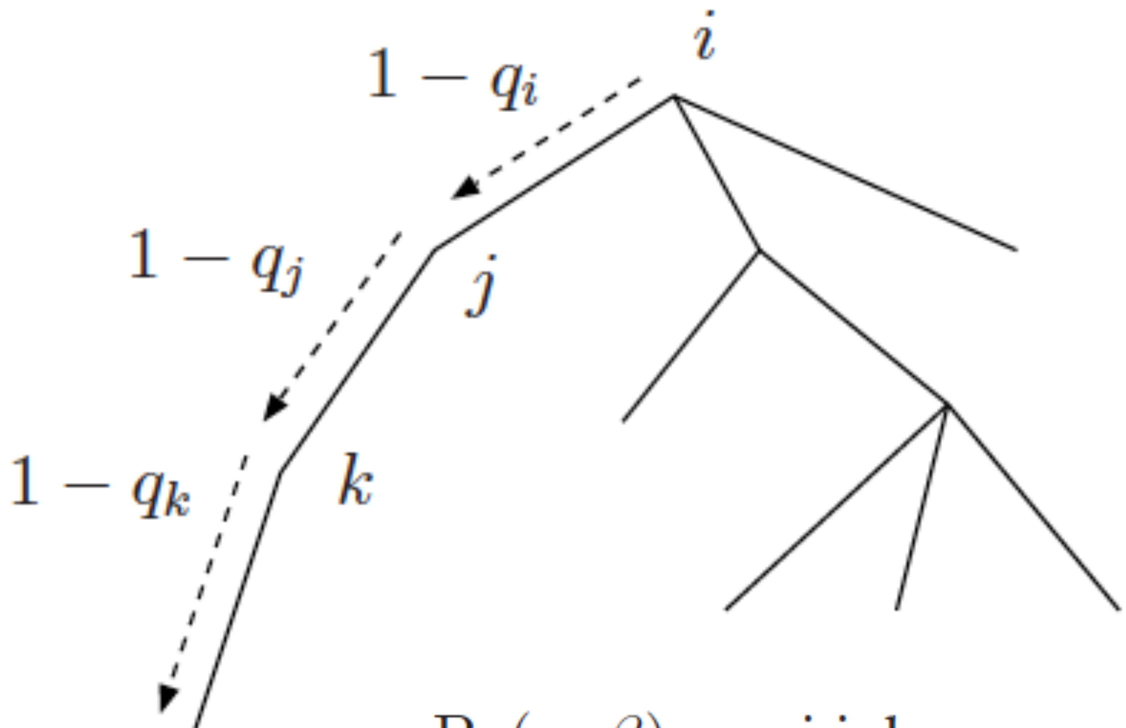
- Fixed branching structure determined by vocabulary
- Depth is a free parameter which has a major impact on computational cost, storage, accuracy, etc.

Variable-Order Suffix Tree



- Would like to model longer contexts only when there is data to do so
- How can we make this dynamic?

Making Order Random Under Prior



$$q_i \sim \mathrm{Be}(\alpha, \beta) \qquad \text{i.i.d.}$$

$$p(n = l | h) = q_l \prod_{i=0}^{l-1} (1 - q_i)$$

$$p(n_t = l | \mathbf{s}_{-t}, \mathbf{z}_{-t}, \mathbf{n}_{-t}) = \frac{a_l + \alpha}{a_l + b_l + \alpha + \beta} \prod_{i=0}^{l-1} \frac{b_i + \beta}{a_i + b_i + \alpha + \beta}$$

Example: Character Model

> ‘how queershaped little children drawling-desks, which would get through that dormouse!’ said alice; ‘let us all for anything the secondly, but it to have and another question, but i shalled out, ‘you are old,’ said the you’re trying to far out to sea.

(a) Random walk generation from a character model.

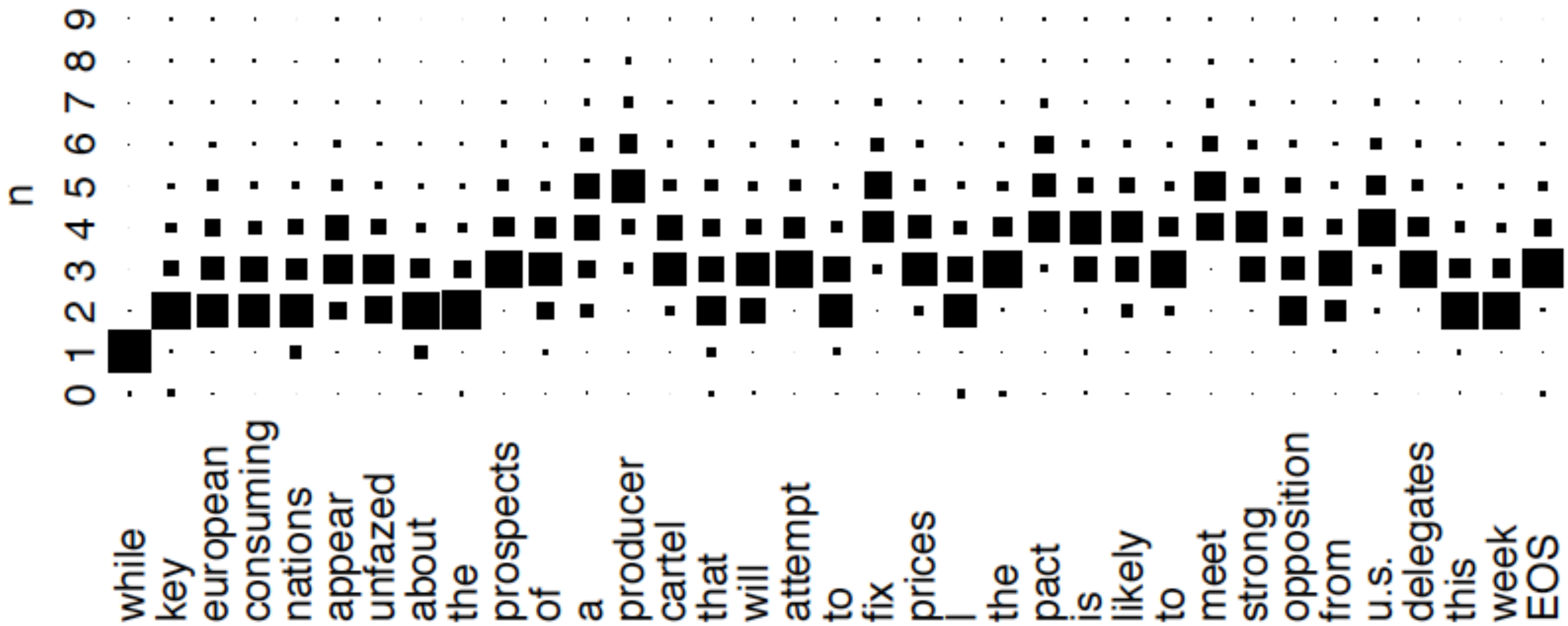
Character	s a i d ␣ a l i c e ; ␣ ‘ l e t ␣ u s ␣ a l l ␣ f o r ␣ a n y t h i n g ␣ t h e ␣ s e c o n d l y , ␣ ···
Markov order	56547106543714824465544556456777533459116489894447343 ···

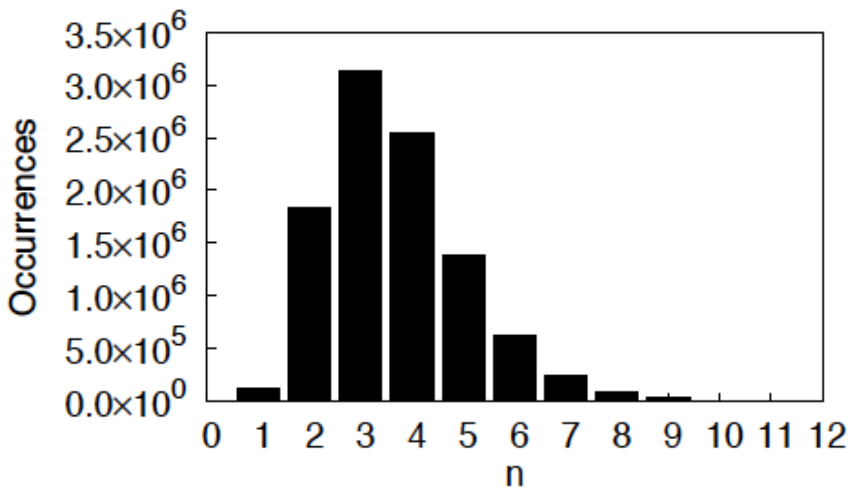
(b) Markov orders used to generate each character above.

Figure 5: Character-based infinite Markov model trained on “Alice in Wonderland.”

Max. order	Perplexity
$n = 3$	6.048
$n = 5$	3.803
$n = 10$	3.519
$n = \infty$	**3.502**

Example: Word Model





n	HPYLM	VPYLM	Nodes(H)	Nodes(V)
3	113.60	113.74	1,417K	**1,344K**
5	101.08	101.69	12,699K	**7,466K**
7	N/A	**100.68**	*27,193K*	**10,182K**
8	N/A	**100.58**	*34,459K*	**10,434K**
∞	—	**100.36**	—	**10,629K**

Modeling versus Learning

- Mochihashi & Sumita would like to learn model orders of higher order where they have more data, lower order where they have less
- They do this by making the true order finite but random under the generative model
- The Sequence Memoizer's arguably cleaner approach: Make the order infinite for all samples, but find a way to make this well regularized and computationally tractable