Hierarchical Bayesian Models of Language and Text

Yee Whye Teh
Gatsby Computational Neuroscience Unit, UCL

Joint work with Frank Wood*, Jan Gasthaus*,
Cedric Archambeau, Lancelot James

Overview

- Probabilistic Models for Language and Text Sequences
- The Sequence Memoizer
  - Hierarchical Bayesian Modelling on Context Trees
  - Modelling Power Laws with Pitman-Yor Processes
  - Non-Markov Models
  - Efficient Computation
- Conclusions

Overview

- **Probabilistic Models for Language and Text Sequences**
- The Sequence Memoizer
  - Hierarchical Bayesian Modelling on Context Trees
  - Modelling Power Laws with Pitman-Yor Processes
  - Non-Markov Models
  - Efficient Computation
- Conclusions

Sequence Models for Language and Text

- Probabilistic models for sequences of words and characters, e.g.

statistical, machine, learning

s, t, a, t, i, s, t, i, c, a, l, _, m, a, c, h, i, n, e, _, l, e, a, r, n, i, n, g

- Uses:
  - Natural language processing: speech recognition, OCR, machine translation.
  - Compression.
  - Cognitive models of language acquisition.
  - Sequence data arises in many other domains.

Probabilistic Modelling

- Set of potential outcomes/observations X.
- Set of unobserved latent variables Y.

- Joint distribution over X and Y:

$$P(x \in X, y \in Y | \theta)$$

θ parameters of the model.

- Inference:

$$P(y \in Y | x \in X, \theta) = \frac{P(y, x|\theta)}{P(x|\theta)}$$

Rev. Thomas Bayes

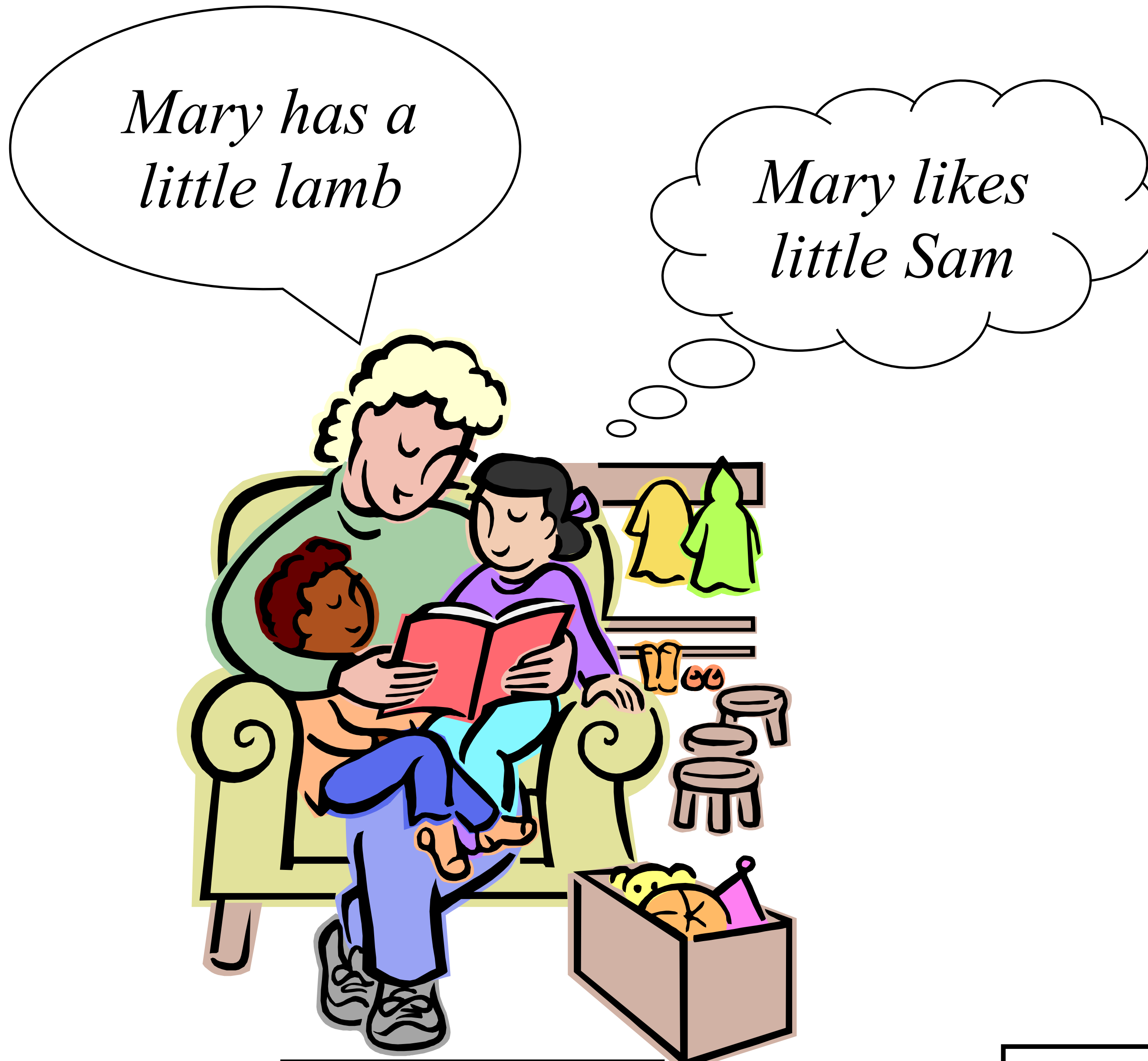
- Learning:

$$P(\text{training data}|\theta)$$

- Bayesian learning:

$$P(\theta|\text{training data}) = \frac{P(\text{training data}|\theta)P(\theta)}{Z}$$

Communication via Noisy Channel



Sentence → *Utterance* → *Reconstructed sentence*

$P(\mathrm{s})$

$P(\mathrm{u}|\mathrm{s})$

$$P(s|u) = \frac{P(s)P(u|s)}{P(u)}$$

Communication via Noisy Channel

Sentence → *foreign sentence* → *Reconstructed sentence*

$P(\mathrm{s})$

$P(\mathrm{u}|\mathrm{s})$

$$P(s|u) = \frac{P(s)P(u|s)}{P(u)}$$

Markov Models for Language and Text

- Probabilistic models for sequences of words and characters.

P(statistical machine learning) =
P(statistical)*
P(machine | statistical)*
P(learning | statistical machine)

Andrey Markov

- Usually makes a Markov assumption:

P(statistical machine learning) =
P(statistical)*
P(machine | statistical)*
P(learning | machine)

George E. P. Box

- Order of Markov model typically ranges from ~3 to > 10.

Sparsity in Markov Models

- Consider a high order Markov models:

$$P(\text{sentence}) = \prod_i P(\text{word}_i | \text{word}_{i-N+1} \ldots \text{word}_{i-1})$$

- Large vocabulary size means naïvely estimating parameters of this model from data counts is problematic for N>2.

$$P^{\text{ML}}(\text{word}_i | \text{word}_{i-N+1} \ldots \text{word}_{i-1}) = \frac{C(\text{word}_{i-N+1} \ldots \text{word}_i)}{C(\text{word}_{i-N+1} \ldots \text{word}_{i-1})}$$

- Naïve priors/regularization fail as well: most parameters have *no* associated data.
  - Smoothing.
  - Hierarchical Bayesian models.

Smoothing in Language Models

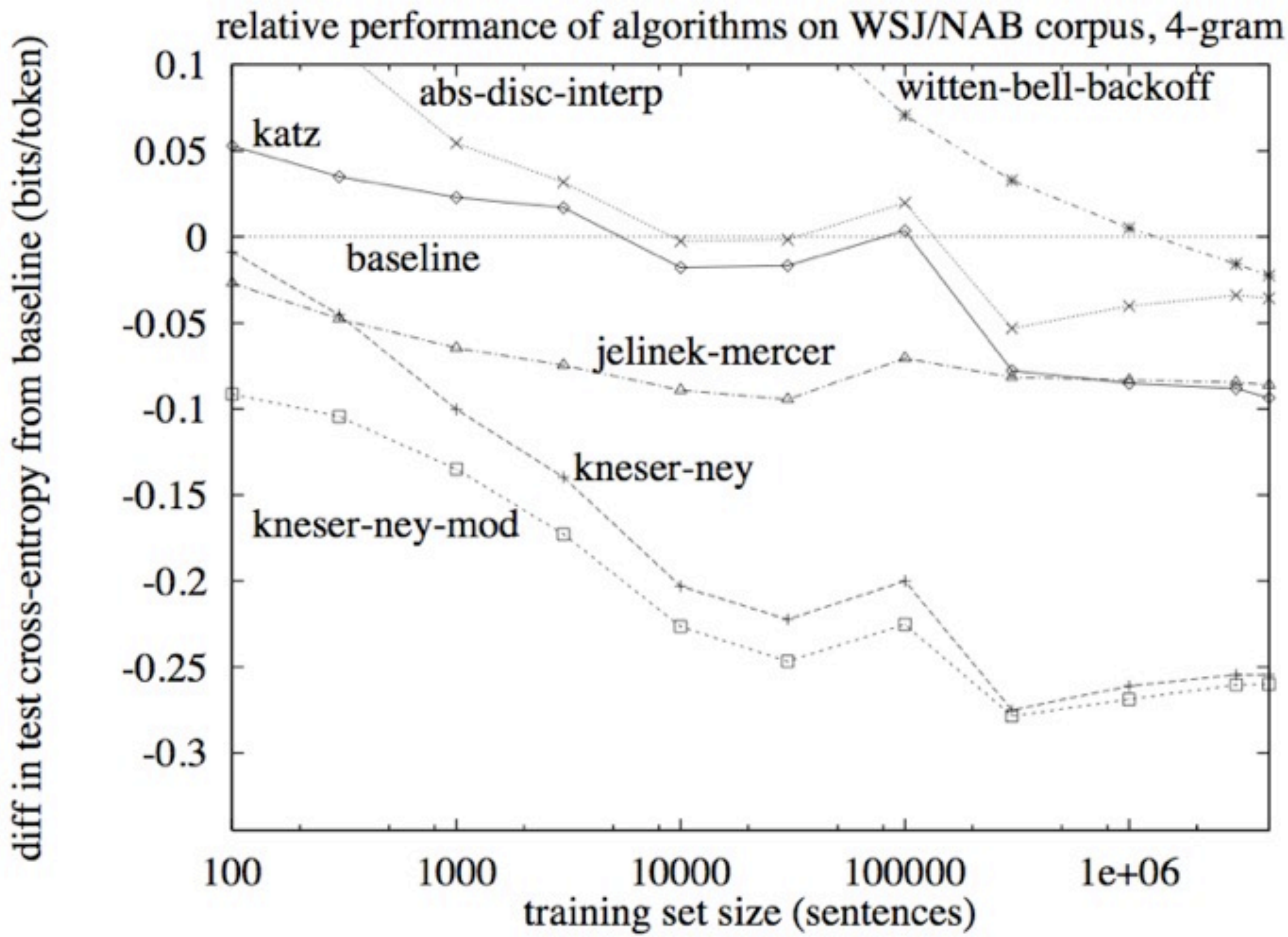
- Smoothing is a way of dealing with data sparsity by combining large and small models together.

$$P^{\text{smooth}}(\text{word}_i|\text{word}_{i-N+1}^{i-1}) = \sum_{n=1}^{N} \lambda(n) Q_n(\text{word}_i|\text{word}_{i-n+1}^{i-1})$$

- Combines expressive power of large models with better estimation of small models (cf bias-variance trade-off).

$$\begin{aligned} & P^{\text{smooth}}(\text{learning}|\text{statistical machine}) \\ = \; & \lambda(3)Q_3(\text{learning}|\text{statistical machine}) + \\ & \lambda(2)Q_2(\text{learning}|\text{machine}) + \\ & \lambda(1)Q_1(\text{learning}|\emptyset) \end{aligned}$$

Smoothing in Language Models



- [Chen and Goodman 1998] found that Interpolated and modified Kneser-Ney are best under virtually all circumstances.

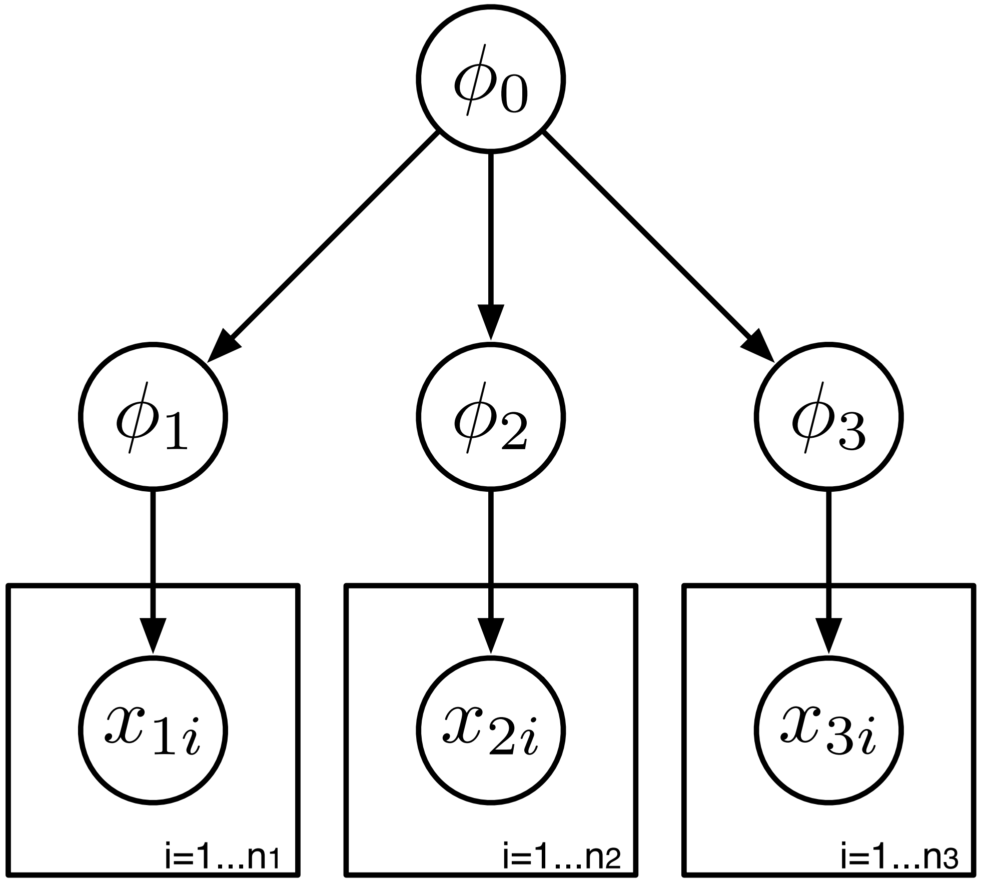
Overview

- Probabilistic Models for Language and Text Sequences
- The Sequence Memoizer
  - **Hierarchical Bayesian Modelling on Context Trees**
  - Modelling Power Laws with Pitman-Yor Processes
  - Non-Markov Models
  - Efficient Computation
- Conclusions

Hierarchical Bayesian Models

- Hierarchical modelling an important overarching theme in modern statistics [Gelman et al, 1995, James & Stein 1961].

- In machine learning, have been used for multitask learning, transfer learning, learning-to-learn and domain adaptation.

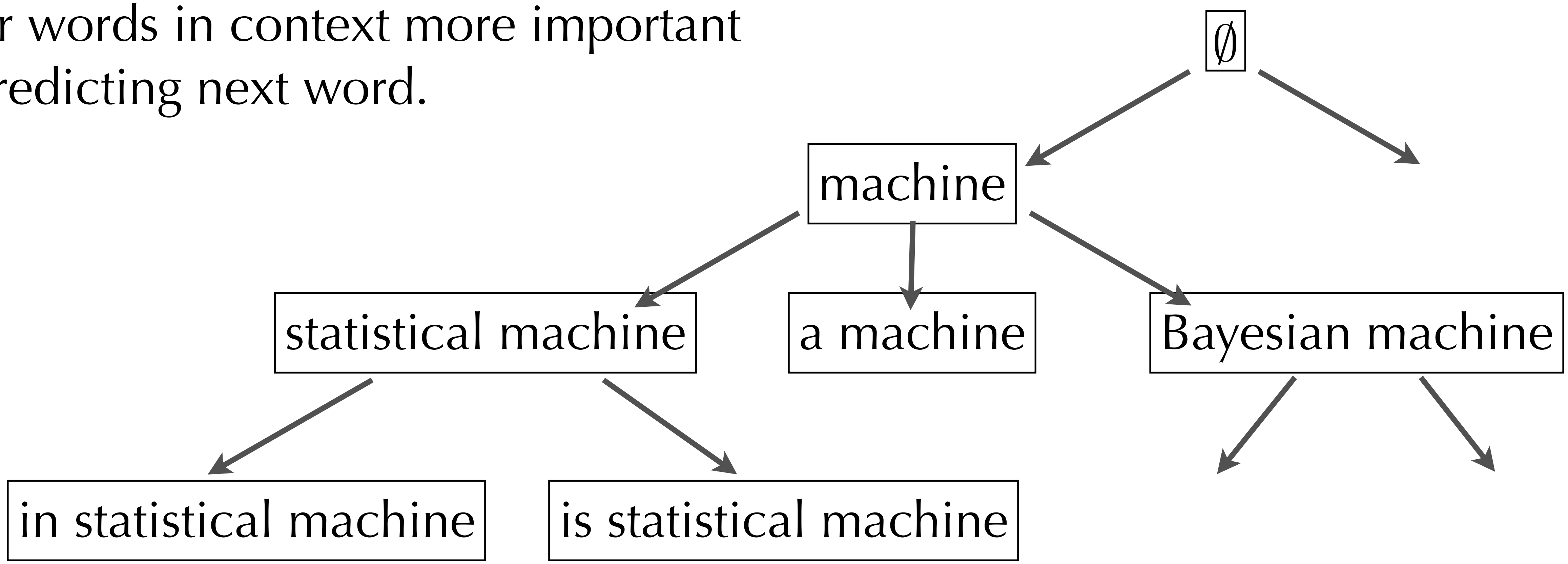


Context Tree

- *Context* of conditional probabilities naturally organized using a tree.
- Smoothing makes conditional probabilities of neighbouring contexts more similar.

$$
\begin{aligned}
&P^{\text{smooth}}(\text{learning}|\text{statistical machine}) \\
=&\lambda(3)Q_3(\text{learning}|\text{statistical machine})+ \\
&\lambda(2)Q_2(\text{learning}|\text{machine})+ \\
&\lambda(1)Q_1(\text{learning}|\emptyset)
\end{aligned}
$$

- Later words in context more important in predicting next word.



Hierarchical Bayesian Models on Context Tree

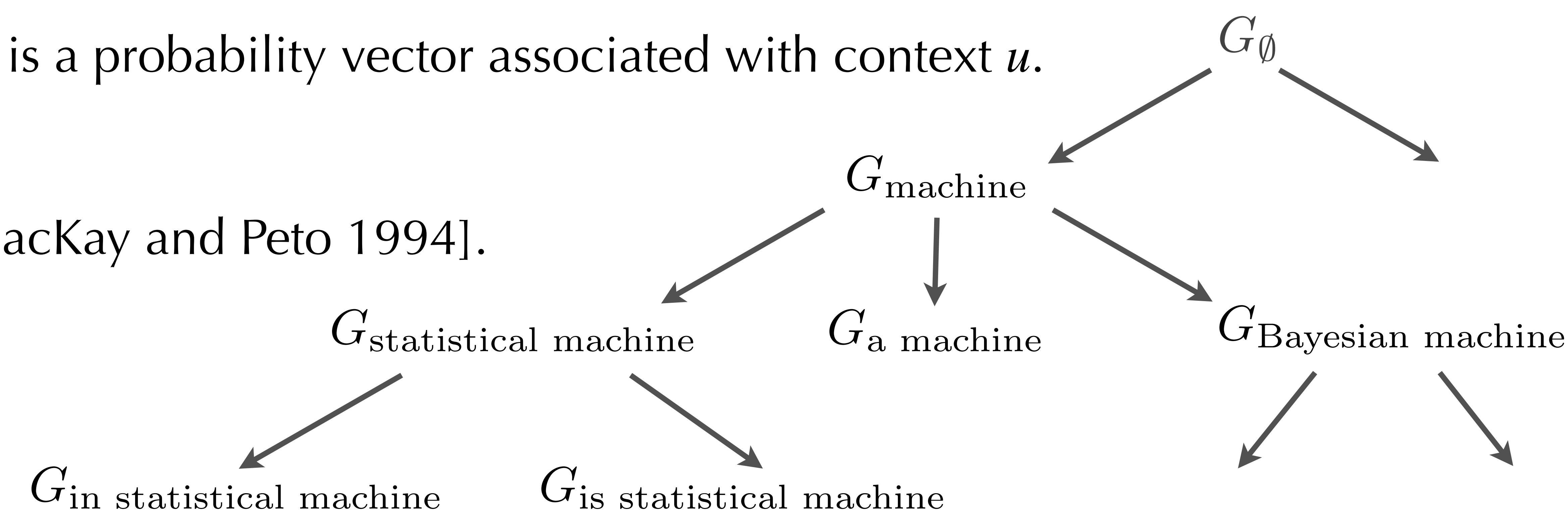
- Parametrize the conditional probabilities of Markov model:

$$P(\text{word}_i = w | \text{word}_{i-N+1}^{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in \text{vocabulary}}$$

- G_u is a probability vector associated with context u.

- [MacKay and Peto 1994].



Hierarchical Dirichlet Language Models

- What is $P(G_u | G_{\mathrm{pa}(u)})$? [MacKay and Peto 1994] proposed using the standard Dirichlet distribution over probability vectors.

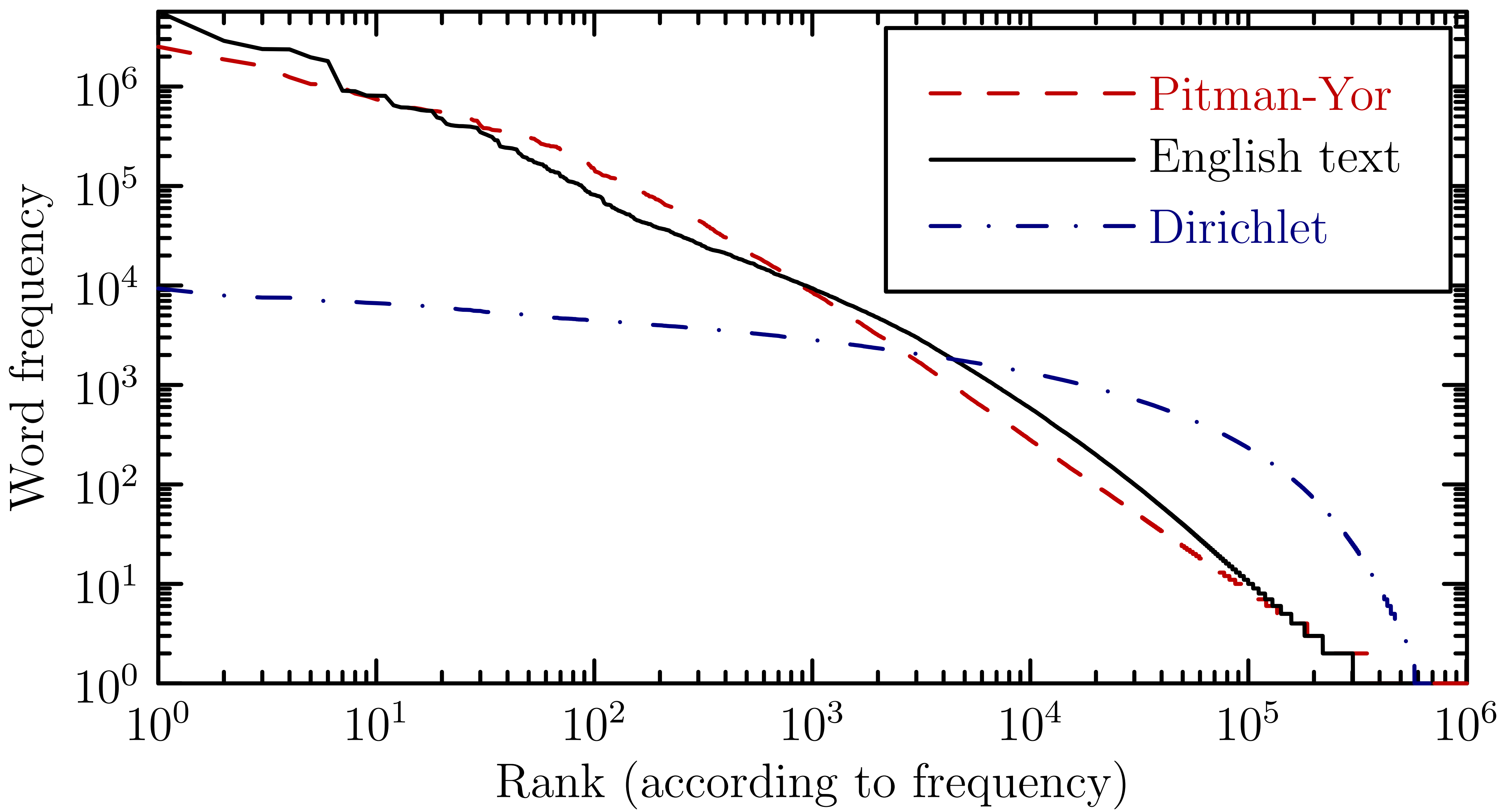
T	N-1	IKN	MKN	HDLM
2×10^6	2	148.8	144.1	191.2
4×10^6	2	137.1	132.7	172.7
6×10^6	2	130.6	126.7	162.3
8×10^6	2	125.9	122.3	154.7
10×10^6	2	122.0	118.6	148.7
12×10^6	2	119.0	115.8	144.0
14×10^6	2	116.7	113.6	140.5
14×10^6	1	169.9	169.2	180.6
14×10^6	3	106.1	102.4	136.6

- We will use Pitman-Yor processes instead [Perman, Pitman and Yor 1992], [Pitman and Yor 1997], [Ishwaran and James 2001].

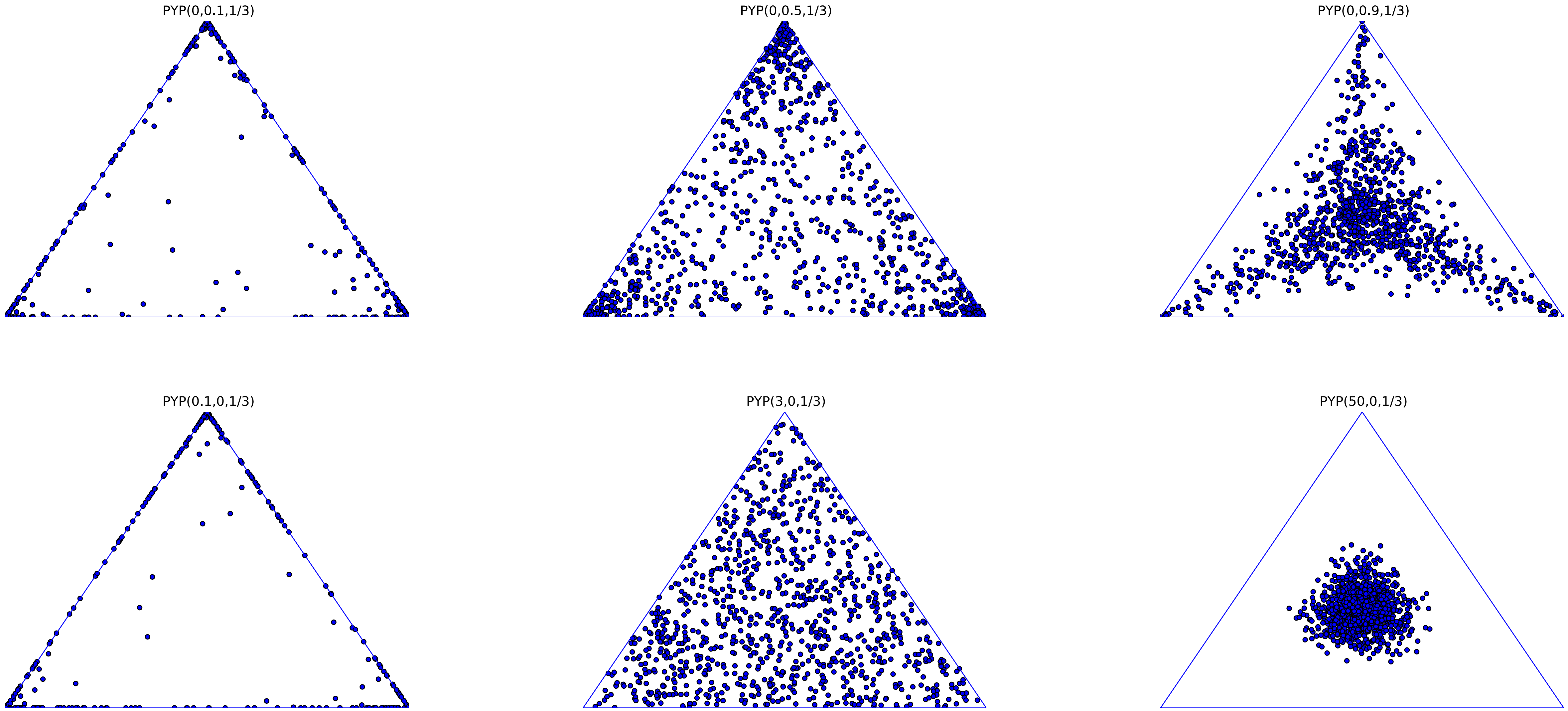
Overview

- Probabilistic Models for Language and Text Sequences
- The Sequence Memoizer
  - Hierarchical Bayesian Modelling on Context Trees
  - **Modelling Power Laws with Pitman-Yor Processes**
  - Non-Markov Models
  - Efficient Computation
- Conclusions

Pitman-Yor Processes

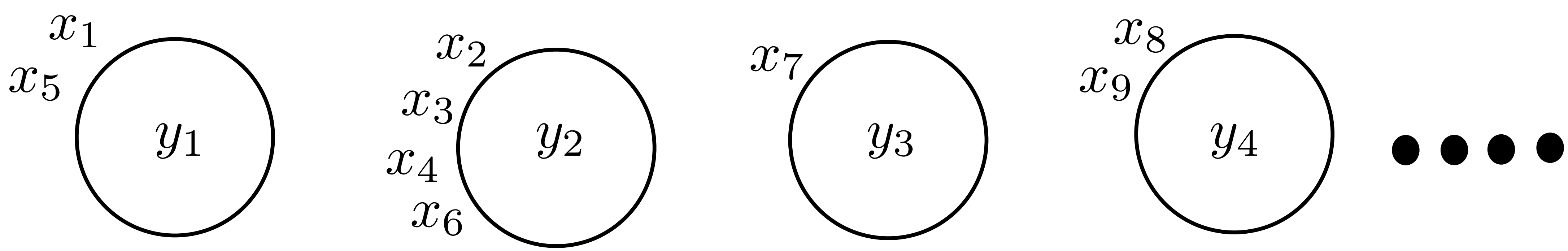


Pitman-Yor Processes



Chinese Restaurant Processes

- Easiest to understand them using Chinese restaurant processes.



$$p(\text{sit at table } k) = \frac{c_k - d}{\theta + \sum_{j=1}^{K} c_j}$$

$$p(\text{sit at new table}) = \frac{\theta + dK}{\theta + \sum_{j=1}^{K} c_j} \qquad p(\text{table serves dish } y) = H(y)$$

- Defines an exchangeable stochastic process over sequences $x_1, x_2, \ldots$
- The de Finetti measure is the Pitman-Yor process,

$$\begin{aligned} G &\sim \mathrm{PY}(\theta, d, H) \\ x_i &\sim G \qquad i = 1, 2, \ldots \end{aligned}$$

- [Perman, Pitman & Yor 1992, Pitman & Yor 1997]

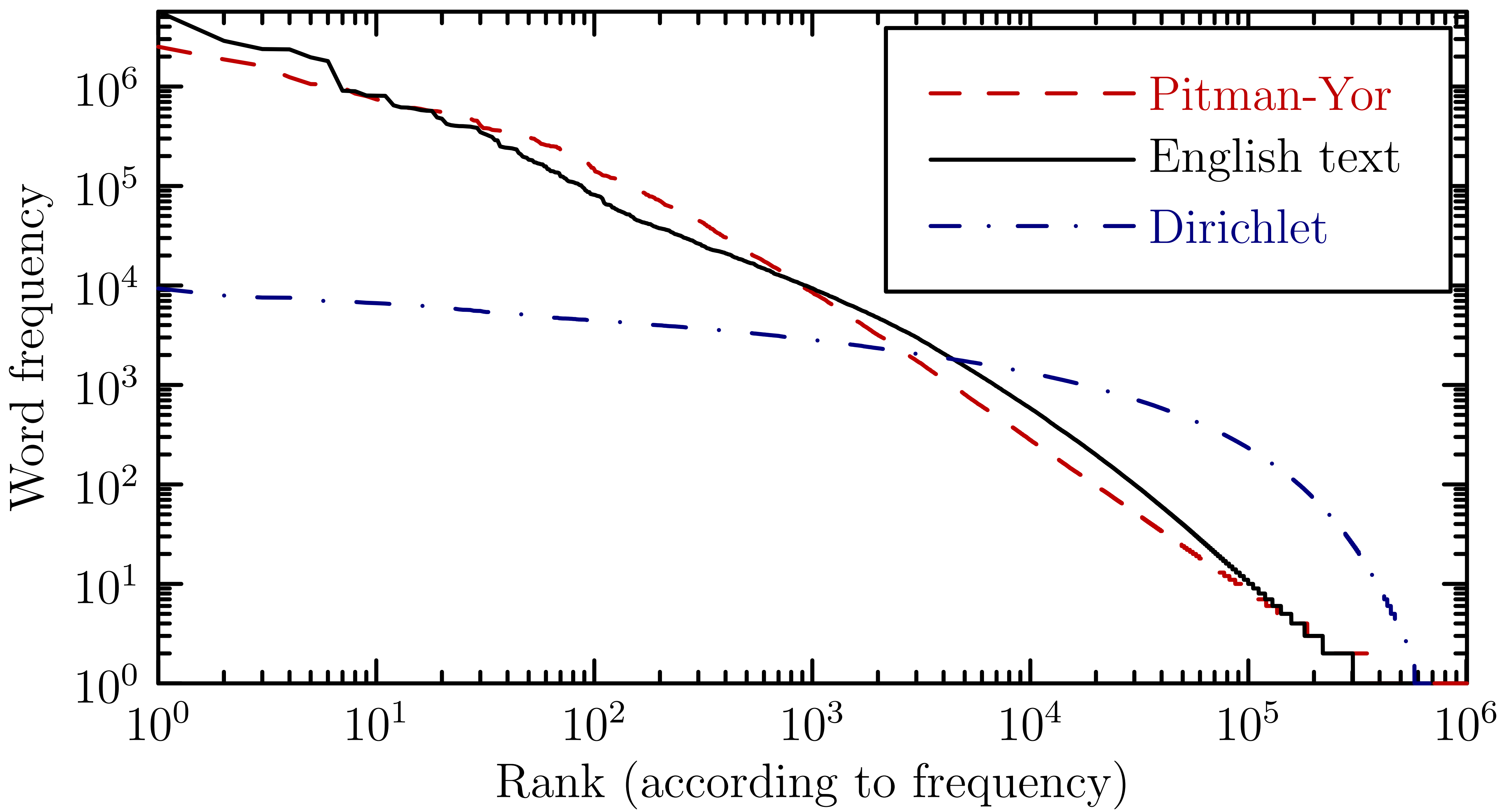
Power Law Properties of Pitman-Yor Processes

- Chinese restaurant process:

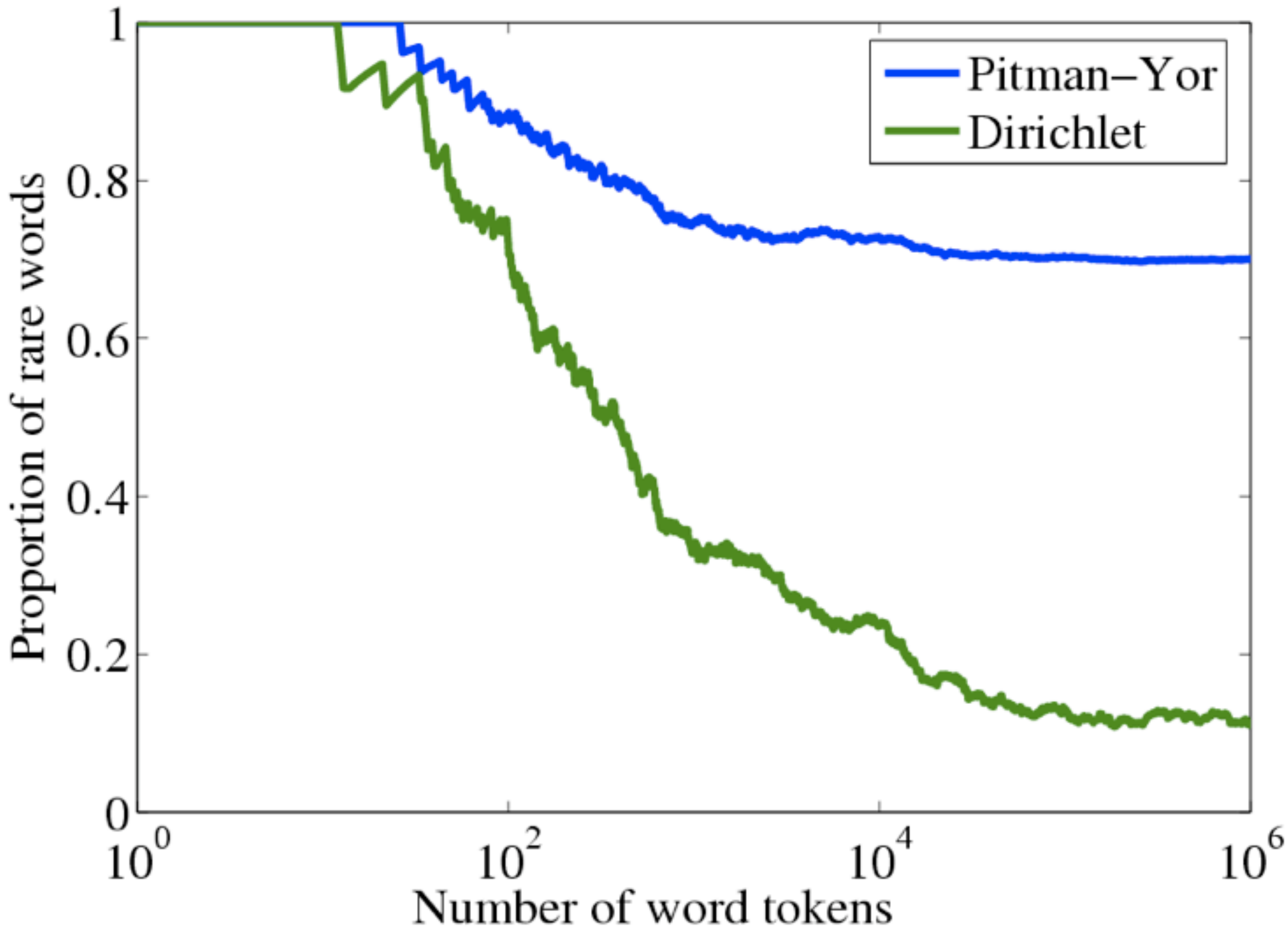
$$\begin{aligned} p(\text{sit at table } k) &\propto c_k - d \\ p(\text{sit at new table}) &\propto \theta + dK \end{aligned}$$

- Pitman-Yor processes produce distributions over words given by a power-law distribution with index $\cdot 1 + d$
  - Customers = word instances, tables = dictionary look-up;
  - Small number of common word types;
  - Large number of rare word types.
- This is more suitable for languages than Dirichlet distributions.
- [Goldwater, Griffiths and Johnson 2005] investigated the Pitman-Yor process from this perspective.

Power Law Properties of Pitman-Yor Processes



Power Law Properties of Pitman-Yor Processes



Hierarchical Pitman-Yor Language Models

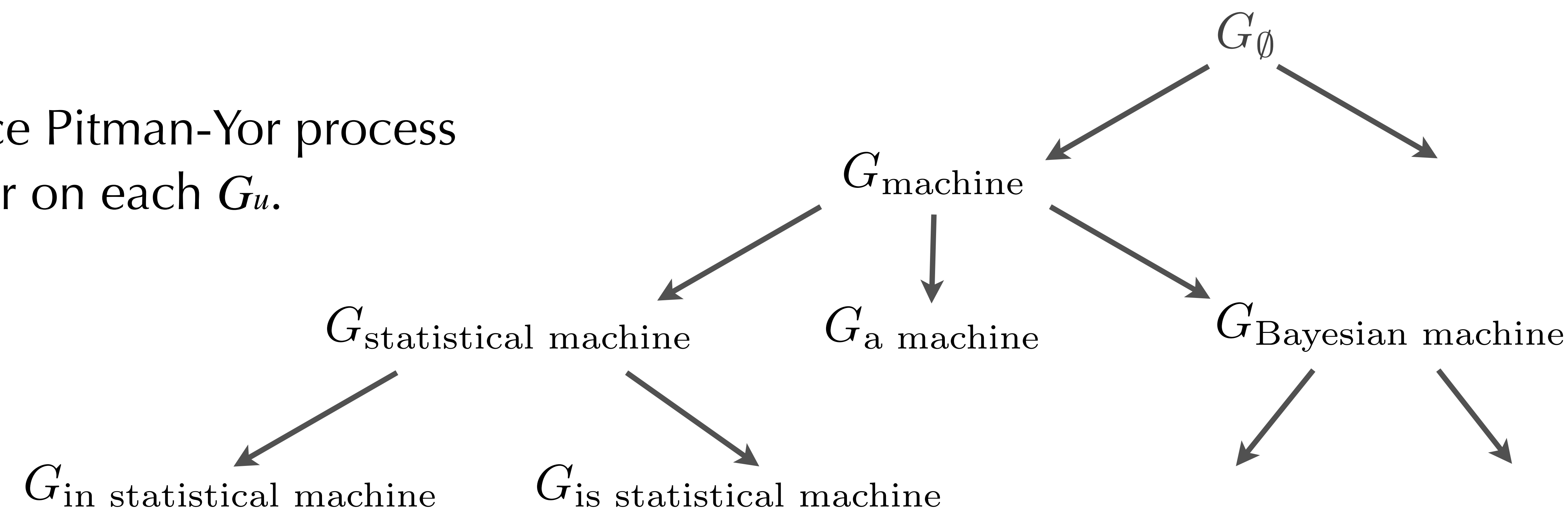
- Parametrize the conditional probabilities of Markov model:

$$P(\text{word}_i = w | \text{word}_{i-N+1}^{i-1} = u) = G_u(w)$$

$$G_u = [G_u(w)]_{w \in \text{vocabulary}}$$

- G_u is a probability vector associated with context u.

- Place Pitman-Yor process prior on each G_u.



Hierarchical Pitman-Yor Language Models

- Significantly improved on the hierarchical Dirichlet language model.
- Results better Kneser-Ney smoothing, state-of-the-art language models.

T	N-1	IKN	MKN	HDLM	HPYLM
2×10^6	2	148.8	**144.1**	191.2	144.3
4×10^6	2	137.1	**132.7**	172.7	**132.7**
6×10^6	2	130.6	126.7	162.3	**126.4**
8×10^6	2	125.9	122.3	154.7	**121.9**
10×10^6	2	122.0	118.6	148.7	**118.2**
12×10^6	2	119.0	115.8	144.0	**115.4**
14×10^6	2	116.7	113.6	140.5	**113.2**
14×10^6	1	169.9	**169.2**	180.6	169.3
14×10^6	3	106.1	102.4	136.6	**101.9**

- Similarity of perplexities not a surprise---Kneser-Ney can be derived as a particular approximate inference method.

Overview

- Probabilistic Models for Language and Text Sequences
- The Sequence Memoizer
    - Hierarchical Bayesian Modelling on Context Trees
    - Modelling Power Laws with Pitman-Yor Processes
    - **Non-Markov Models**
    - Efficient Computation
- Conclusions

Markov Models for Language and Text

- Usually makes a Markov assumption to simplify model:

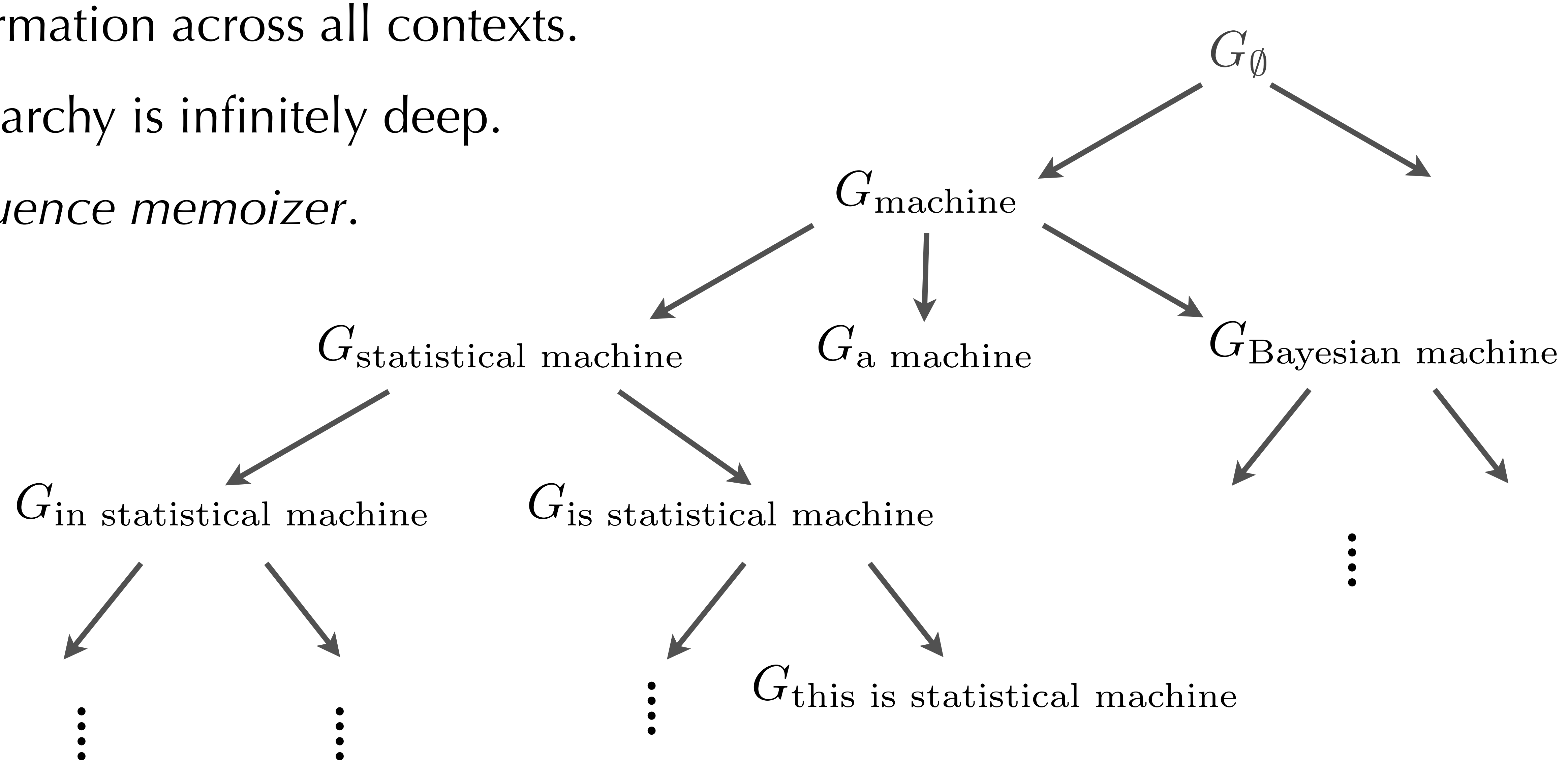
P(south parks road) ~
P(south)*
P(parks | south)*
P(road | south parks)

- Language models: usually Markov models of order 2-4 (3-5-grams).
- How do we determine the order of our Markov models?
- Is the Markov assumption a reasonable assumption?
  - Be nonparametric about Markov order...

Non-Markov Models for Language and Text

- Model the conditional probabilities of each possible word occurring after each possible context (of unbounded length).

- Use hierarchical Pitman-Yor process prior to share information across all contexts.
- Hierarchy is infinitely deep.
- *Sequence memoizer*.



Overview

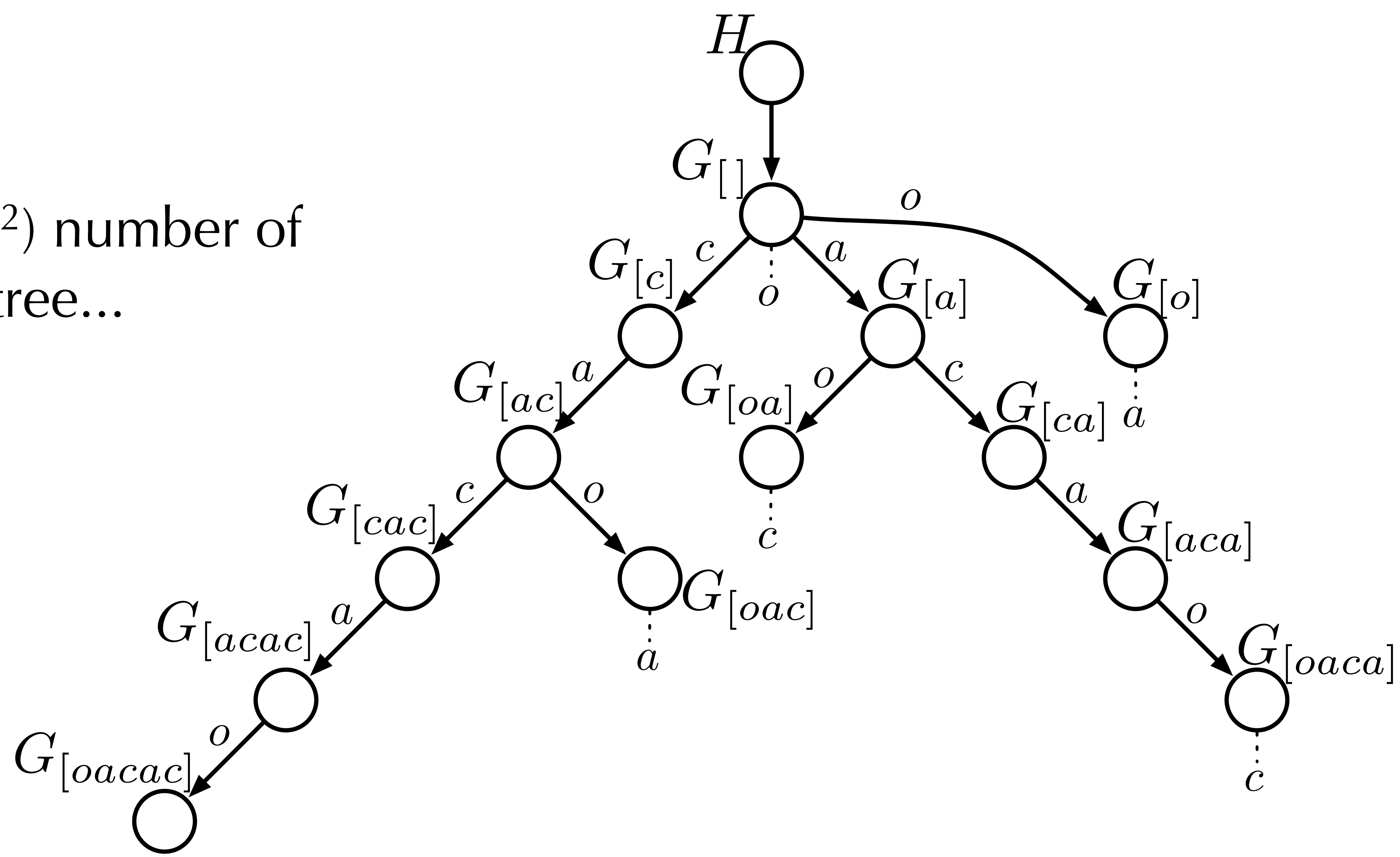
- Probabilistic Models for Language and Text Sequences
- The Sequence Memoizer
    - Hierarchical Bayesian Modelling on Context Trees
    - Modelling Power Laws with Pitman-Yor Processes
    - Non-Markov Models
    - **Efficient Computation**
- Conclusions

Model Size: Infinite -> O(T^2)

- The sequence memoizer model is very large (actually, infinite).

- Given a training sequence (e.g.: o,a,c,a,c), most of the model can be ignored (integrated out), leaving a finite number of nodes in context tree.

- But there are still O(T^2) number of nodes in the context tree...

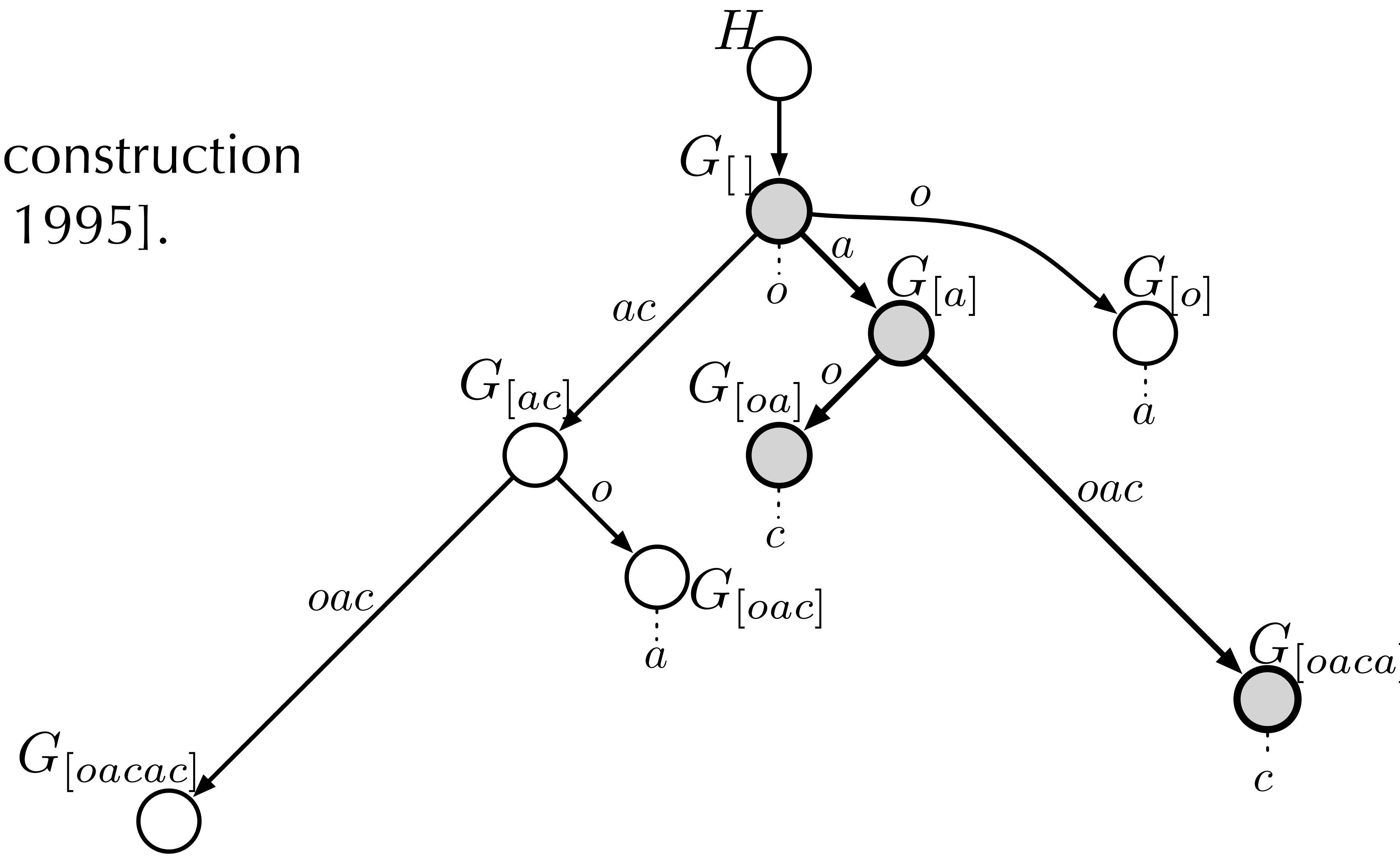


Model Size: Infinite -> O(T²) -> 2T

- Idea: integrate out non-branching, non-leaf nodes of the context tree.

- Resulting tree is related to a suffix tree data structure, and has at most 2T nodes.

- There are linear time construction algorithms [Ukkonen 1995].



Closure under Marginalization

- In marginalizing out non-branching interior nodes, need to ensure that resulting conditional distributions are still tractable.

$$
\begin{tikzcd}
G_{[a]} \arrow[d, "\mathrm{PY}(\theta_2, d_2, G_{[a]})"] \\
G_{[ca]} \arrow[d, "\mathrm{PY}(\theta_3, d_3, G_{[ca]})"] \\
G_{[aca]}
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
G_{[a]} \arrow[d, "?"] \\
G_{[aca]}
\end{tikzcd}
$$

- E.g.: If each conditional is Dirichlet, resulting conditional is not of known analytic form.

Closure under Marginalization

- In marginalizing out non-branching interior nodes, need to ensure that resulting conditional distributions are still tractable.

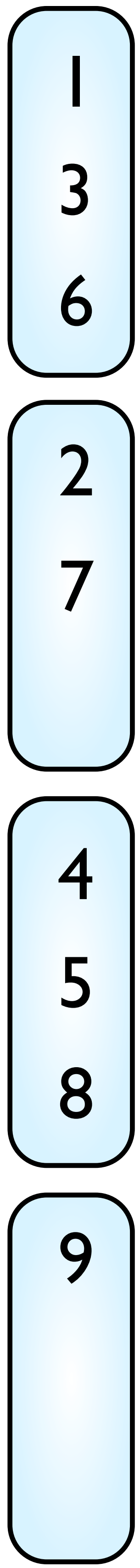
$$
\begin{tikzcd}
G_{[a]} \arrow[d, "\mathrm{PY}(\theta_2, d_2, G_{[a]})"] & G_{[a]} \arrow[dd, "\mathrm{PY}(\theta_2 d_3, d_2 d_3, G_{[a]})"] \\
G_{[ca]} \arrow[d, "\mathrm{PY}(\theta_2 d_3, d_3, G_{[ca]})"] & \\
G_{[aca]} & G_{[aca]}
\end{tikzcd}
$$

- For certain parameter settings, Pitman-Yor processes are closed under marginalization!
- [Pitman 1999, Ho, James & Lau 2006]

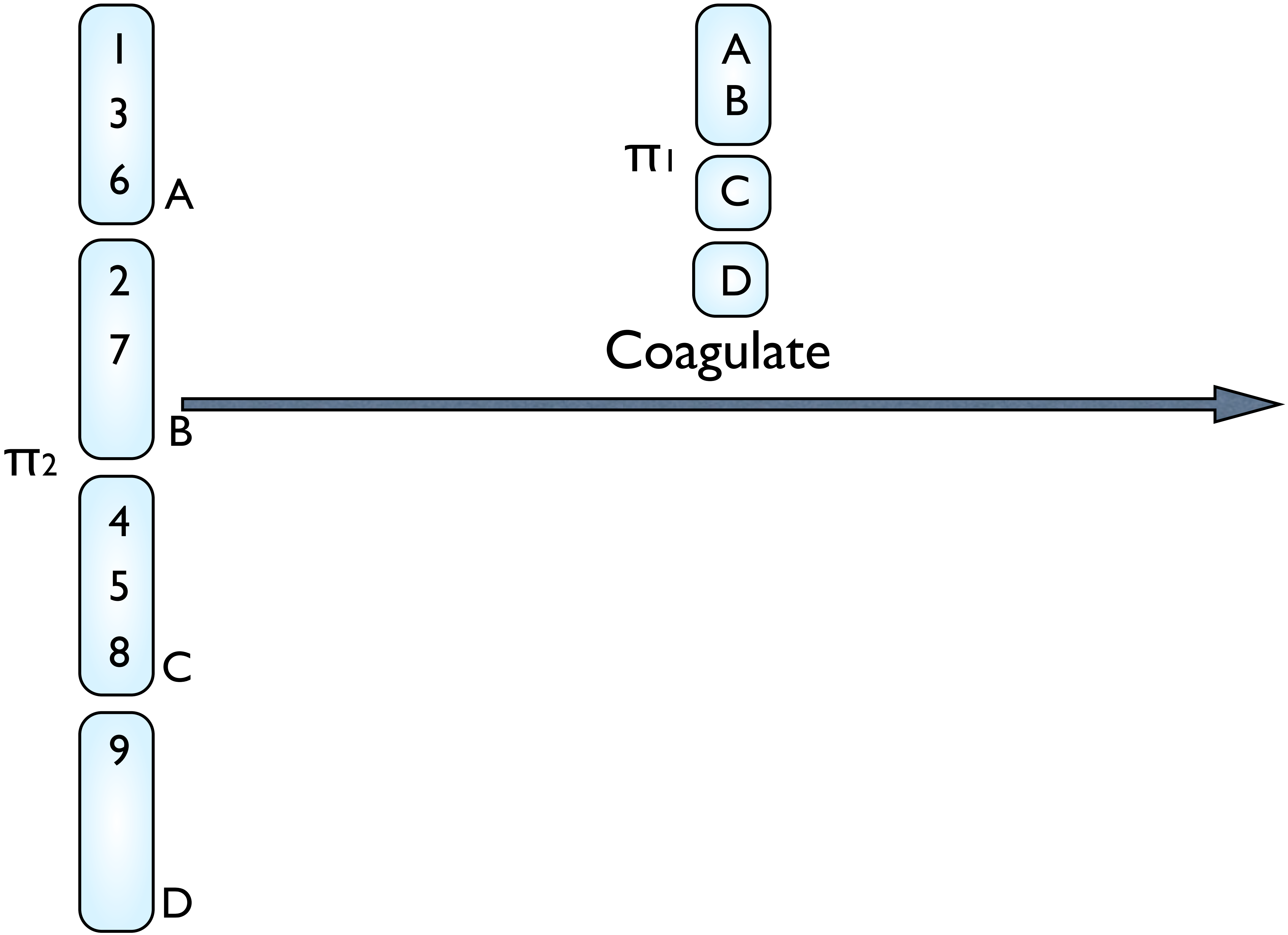
Coagulation and Fragmentation Operators

π_1

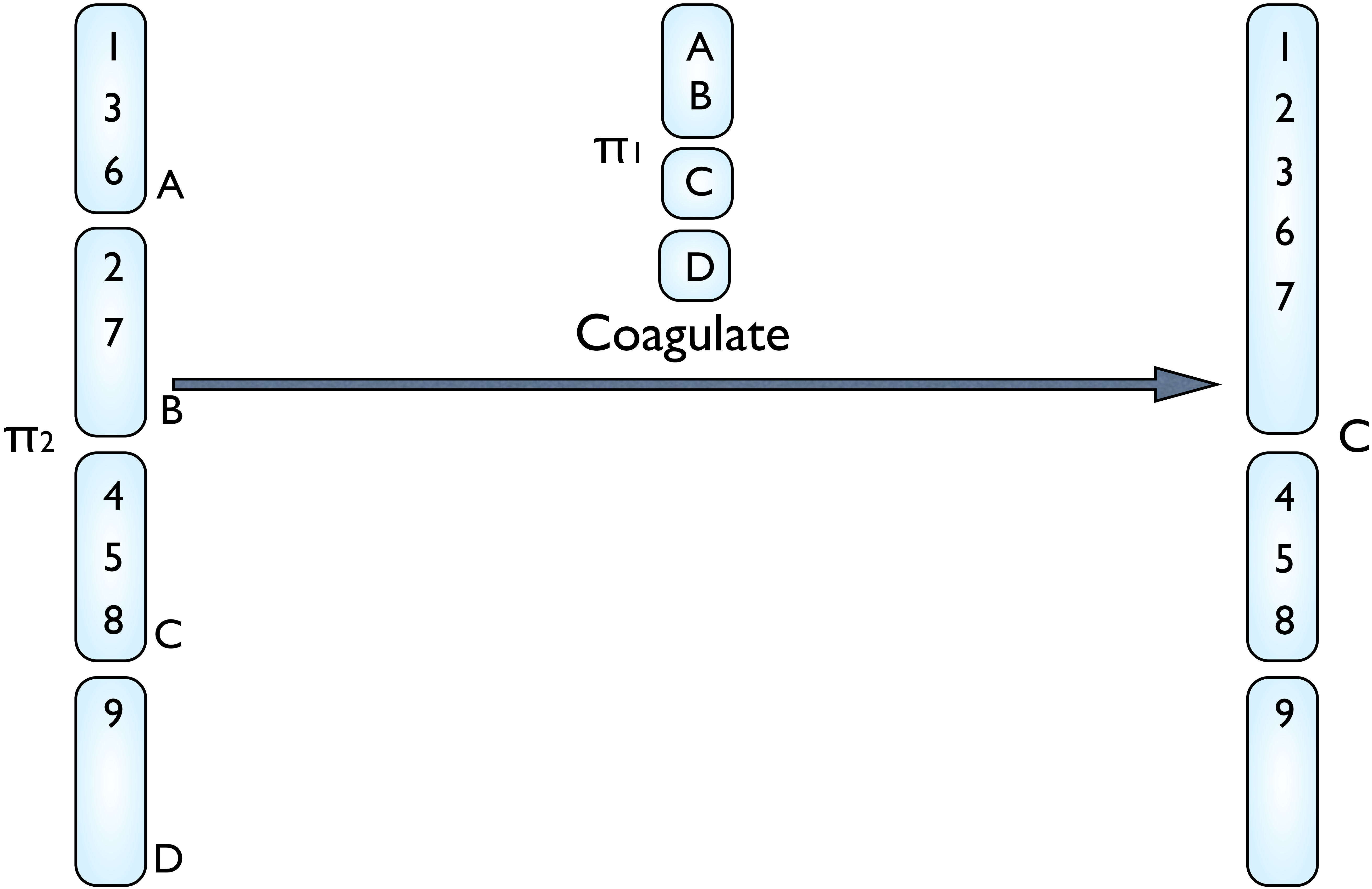
π_2



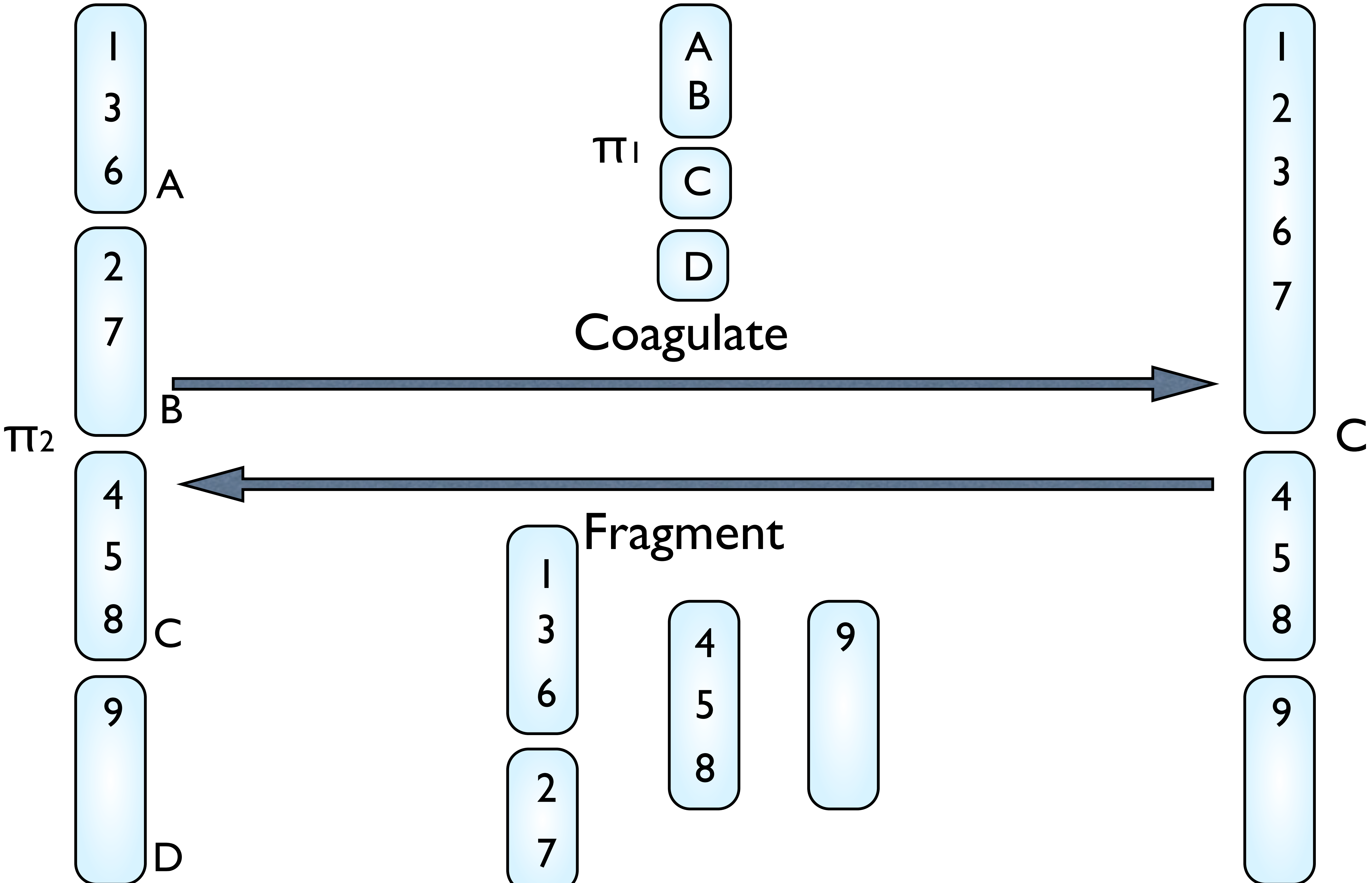
Coagulation and Fragmentation Operators



Coagulation and Fragmentation Operators



Coagulation and Fragmentation Operators

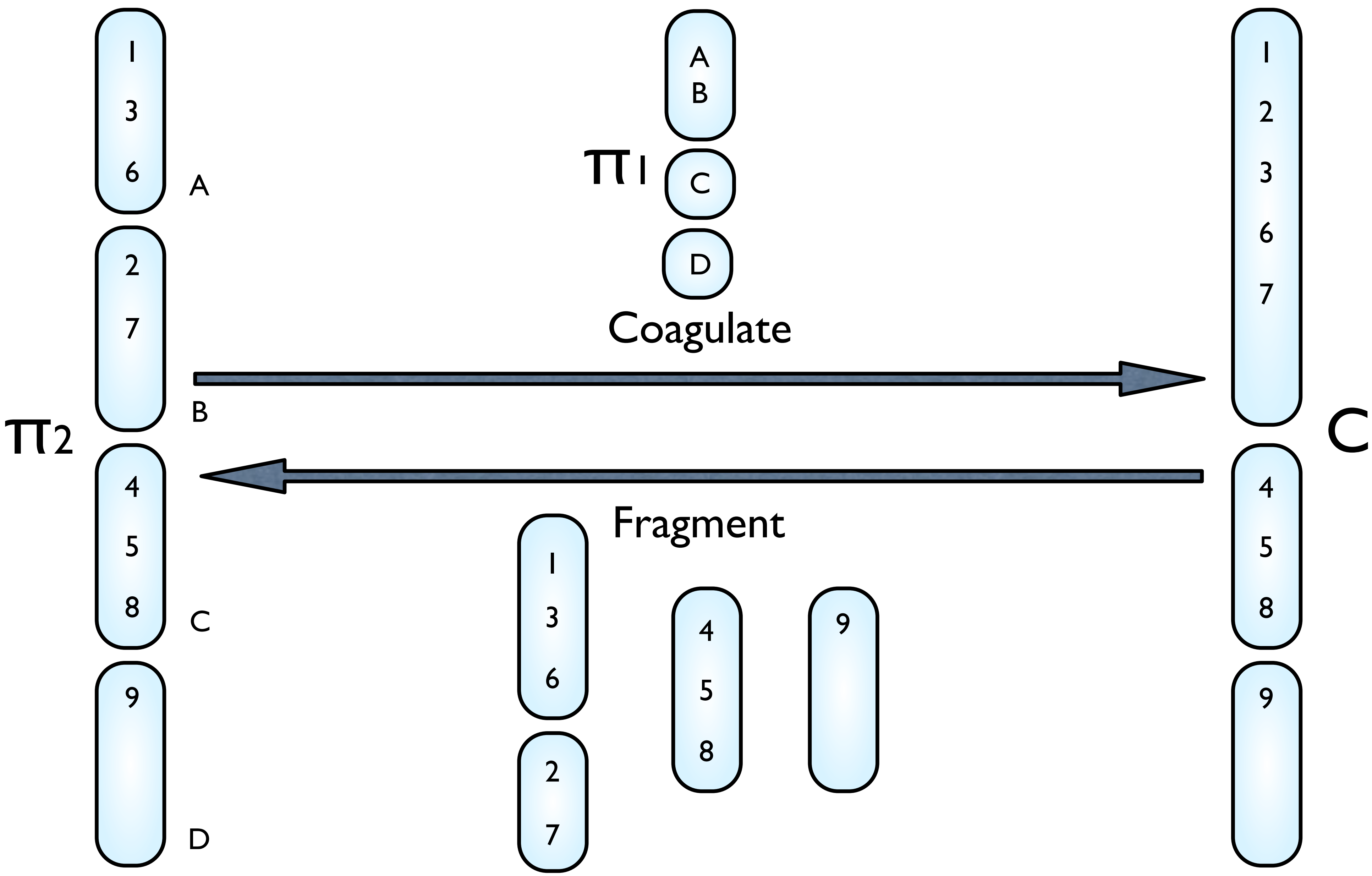


Coagulation and Fragmentation Operators

- The following statements are equivalent:

(I) $\pi_2 \sim \mathrm{CRP}_{[n]}(\alpha d_2, d_2)$ and $\pi_1 | \pi_2 \sim \mathrm{CRP}_{\pi_2}(\alpha, d_1)$

(II) $C \sim \mathrm{CRP}_{[n]}(\alpha d_2, d_1 d_2)$ and $F_a | C \sim \mathrm{CRP}_a(-d_1 d_2, d_2) \quad \forall a \in C$



Final Model Specification

Probability of sequence:

$$P(x_{1:T}) = \prod_{i=1}^{T} P(x_i | x_{1:i-1}) = \prod_{i=1}^{T} G_{x_{1:i-1}}(x_i)$$

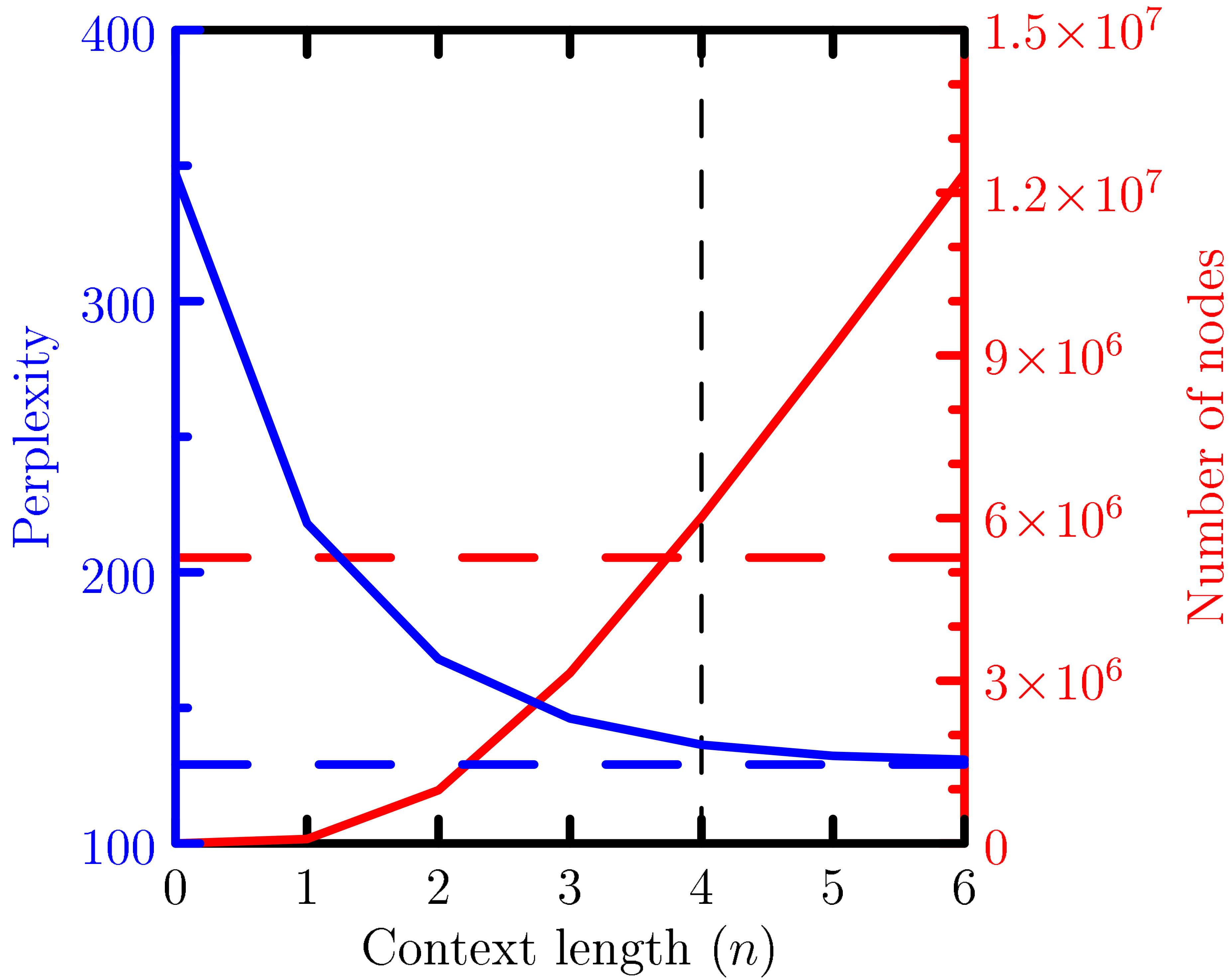
Prior over conditional probabilities:

$$G_{\emptyset} \sim \mathrm{PY}(\theta_{\emptyset}, d_{\emptyset}, H)$$
$$G_{\mathbf{u}} | G_{\sigma(\mathbf{u})} \sim \mathrm{PY}(\theta_{\mathbf{u}}, d_{\mathbf{u}}, G_{\sigma(\mathbf{u})}), \text{ for } \mathbf{u} \in \Sigma^* \backslash \{\emptyset\},$$

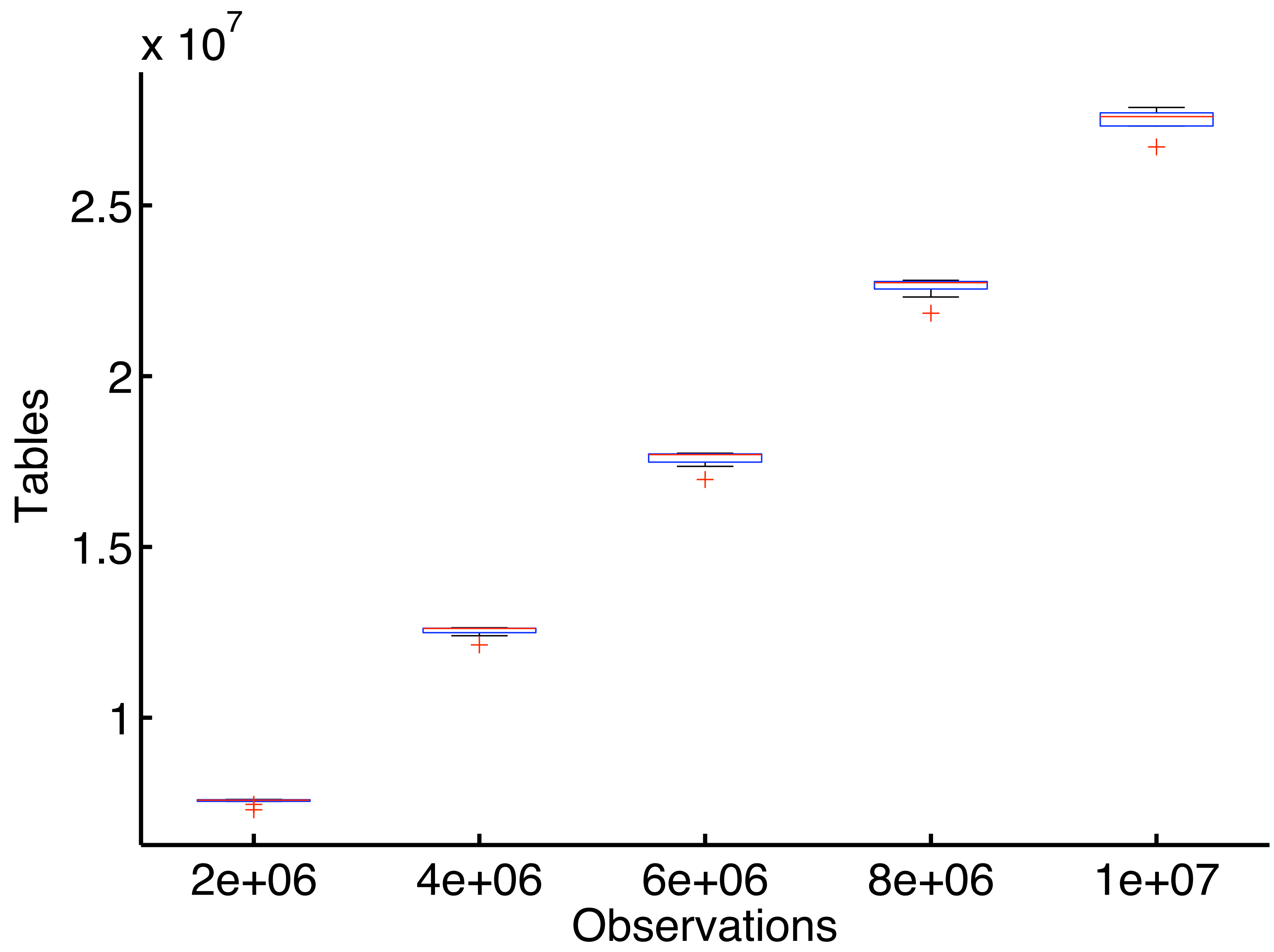
Constraint on parameters:

$$\theta_{\mathbf{u}} = \theta_{\emptyset} \prod_{v \neq \emptyset, \text{ suffix of } \mathbf{u}} d_v$$

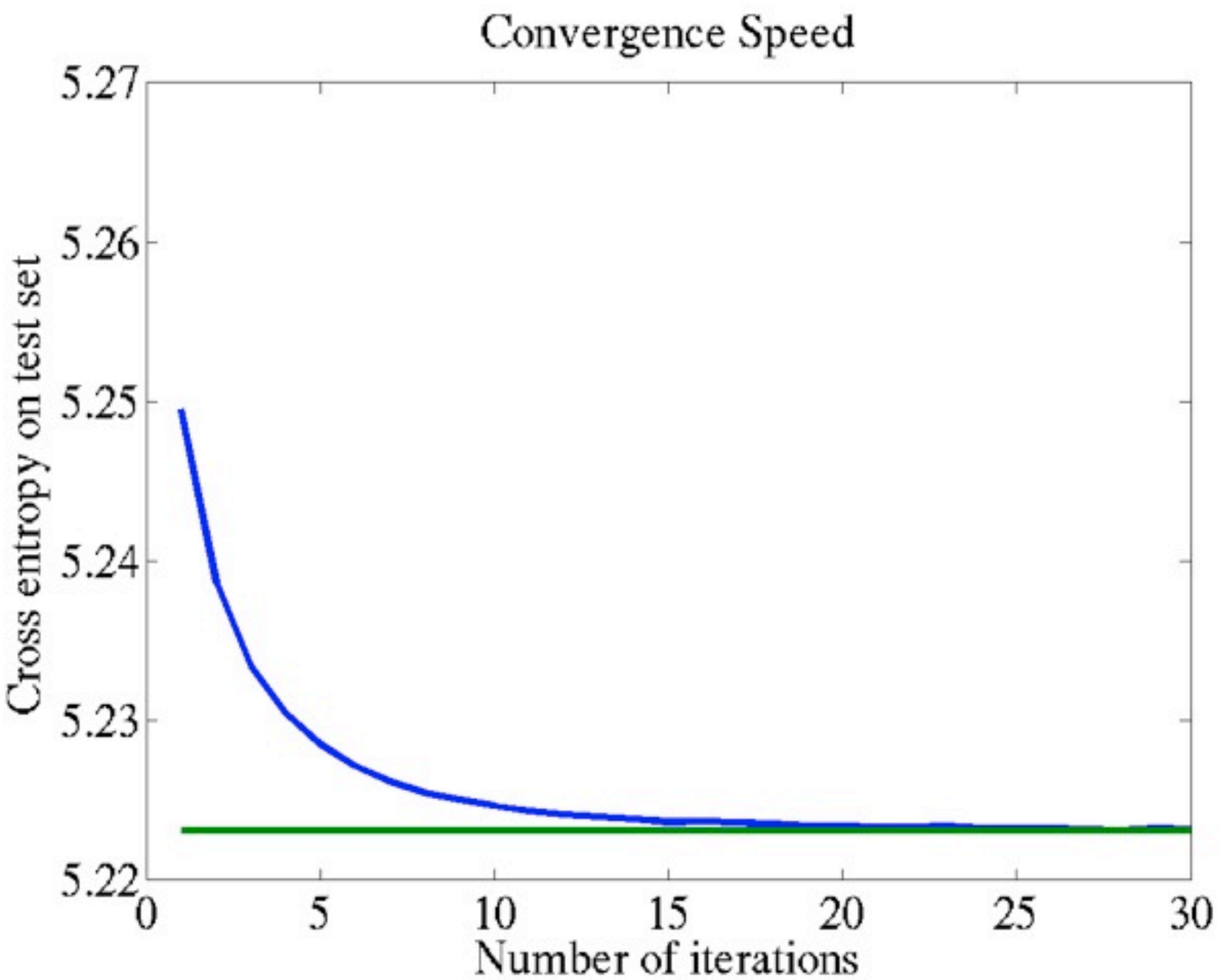
Comparison to Finite Order HPYLM



Inference using Gibbs Sampling



Inference using Gibbs Sampling



Entropic Coding for Compression

- Encoder:

$$x_i \longrightarrow \boxed{\begin{array}{c}\text{Model}\\ P(x_i|x_1\ldots x_{i-1},\theta_i)\end{array}} \longrightarrow -\log_2 P(x_i|x_1\ldots x_{i-1},\theta_i)$$

- Decoder:

$$x_i \longleftarrow \boxed{\begin{array}{c}\text{Model}\\ P(x_i|x_1\ldots x_{i-1},\theta_i)\end{array}} \longleftarrow -\log_2 P(x_i|x_1\ldots x_{i-1},\theta_i)$$

- θ_i parameter value estimated from $x_1 \ldots x_{i-1}$.

- A good probabilistic model = good compressor.

Claude Shannon

Compression Results

Model	Average bits/byte
gzip	2.61
bzip2	2.11
CTW	1.99
PPM	1.93
Sequence Memoizer	1.89

Calgary corpus
SM inference: particle filter
PPM: Prediction by Partial Matching
CTW: Context Tree Weigting
Online inference, entropic coding.

Related Works

- Infinite Markov models [Mochihashi & Sumita 2008]
- Bayesian nonparametric grammars (Goldwater, Johnson, Blunsom, Cohn etc).
- Text compression: Prediction by Partial Matching [Cleary & Witten 1984], Context Tree Weighting [Willems et al 1995]...
- Language model smoothing algorithms [Chen & Goodman 1998, Kneser & Ney 1995].
- Variable length/order/memory Markov models [Ron et al 1996, Buhlmann & Wyner 1999, Begleiter et al 2004...].
- Hierarchical Bayesian nonparametric models [Teh & Jordan 2010].

Conclusions

- Probabilistic models of sequence models without making Markov assumptions with efficient construction and inference algorithms.
- State-of-the-art text compression and language modelling results.
- Hierarchical Bayesian modelling leads to improved performance.
- Pitman-Yor processes allow us to encode prior knowledge about power-law properties, leading to improved performance.
- Hierarchical Pitman-Yor processes have been used successfully for various more linguistically motivated models.

- www.sequencememoizer.com (Java implementation)
- www.deplump.com (text compression demo)
- Jan Gasthaus' webpage (C++ implementation)

Publications

- **A Hierarchical Bayesian Language Model based on Pitman-Yor Processes**.
  Y.W. Teh. Coling/ACL 2006.
- **A Bayesian Interpretation of Interpolated Kneser-Ney**.
  Y.W. Teh. Technical Report TRA2/06, School of Computing, NUS, revised 2006.
- **A Stochastic Memoizer for Sequence Data**.
  F. Wood, C. Archambeau, J. Gasthaus, L. F. James and Y.W. Teh. ICML 2009.
- **Text Compression Using a Hierarchical Pitman-Yor Process Prior**.
  J. Gasthaus, F. Wood and Y.W. Teh. DCC 2010.
- **Forgetting Counts: Constant Memory Inference for a Dependent Hierarchical Pitman-Yor Process.**
  N. Bartlett, D. Pfau and F. Wood. ICML 2010.
- **Some Improvements to the Sequence Memoizer.**
  J. Gasthaus and Y.W. Teh. NIPS 2010.
- **The Sequence Memoizer.**
  F. Wood, J. Gasthaus, C. Archambeau, L. F. James and Y.W. Teh. CACM 2011.
- **Hierarchical Bayesian Nonparametric Models with Applications**.
  Y.W. Teh and M.I. Jordan, in Bayesian Nonparametrics. Cambridge University Press, 2010.

Thank You!

Acknowledgements:
Frank Wood, Jan Gasthaus,
Cedric Archambeau, Lancelot James

Lee Kuan Yew Foundation
Gatsby Charitable Foundation