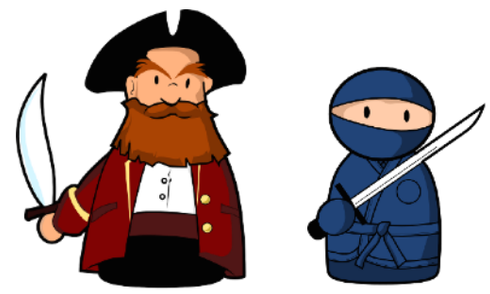
Non-Parametric Bayesian Models

Computational Linguistics

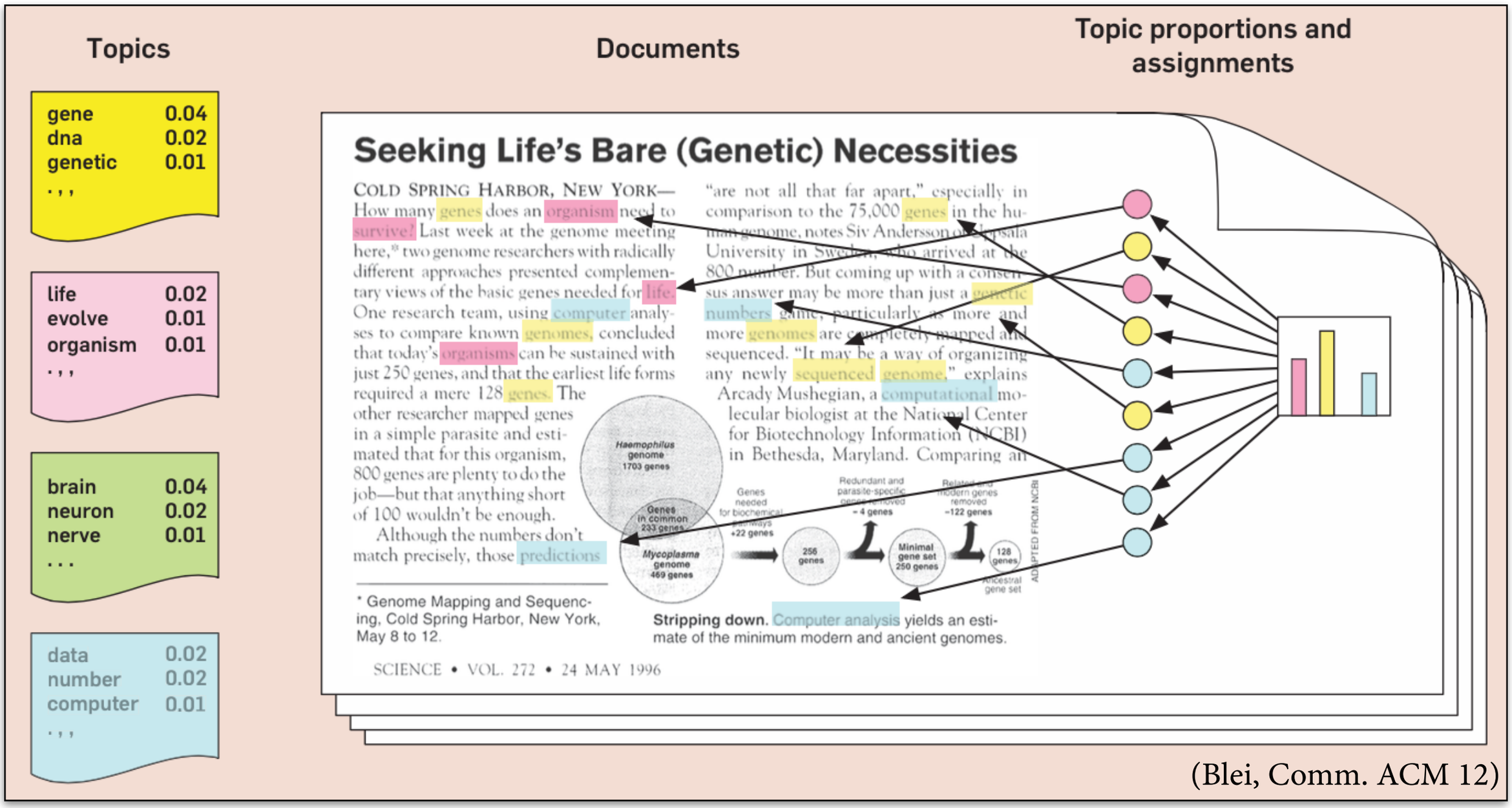
Alexander Koller

21 January 2020

with help from Christoph Teichmann
and illustrations by Martín Villalba



Topic models

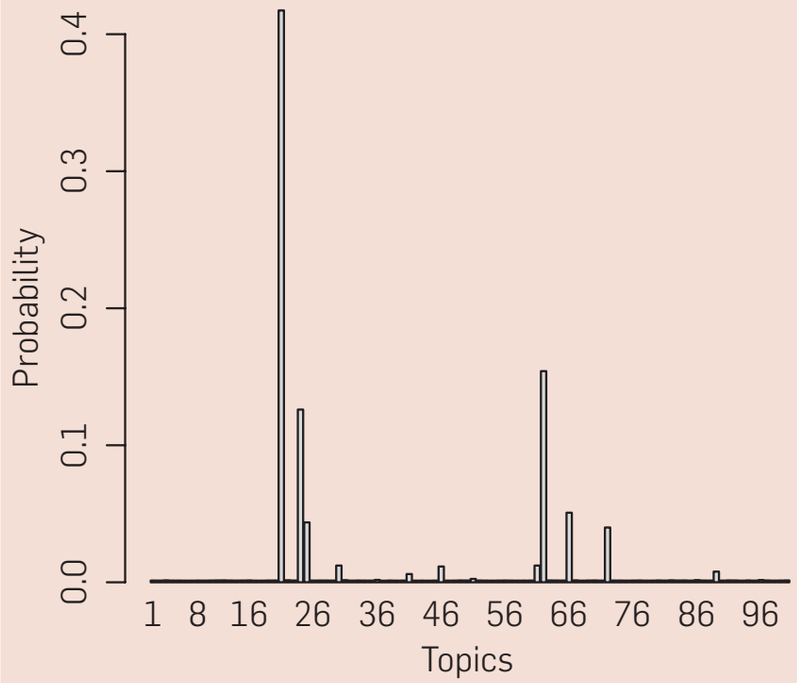


(Blei, Comm. ACM 12)

learn: word probs. for (abstract) *topics* ← given: raw documents → learn: topic mixture in each document

Examples

(Blei 2012)



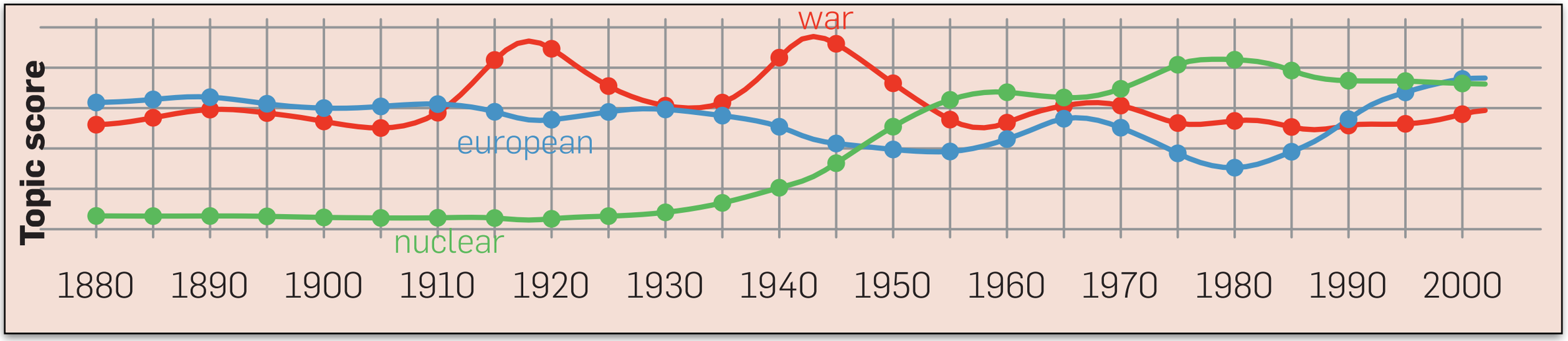
“Genetics”	**“Evolution”**	**“Disease”**	**“Computers”**
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

topic mixture for
one article in *Science*

15 words with highest $\phi_{k,w}$
for each topic over whole corpus
(with made-up topic label)

Examples

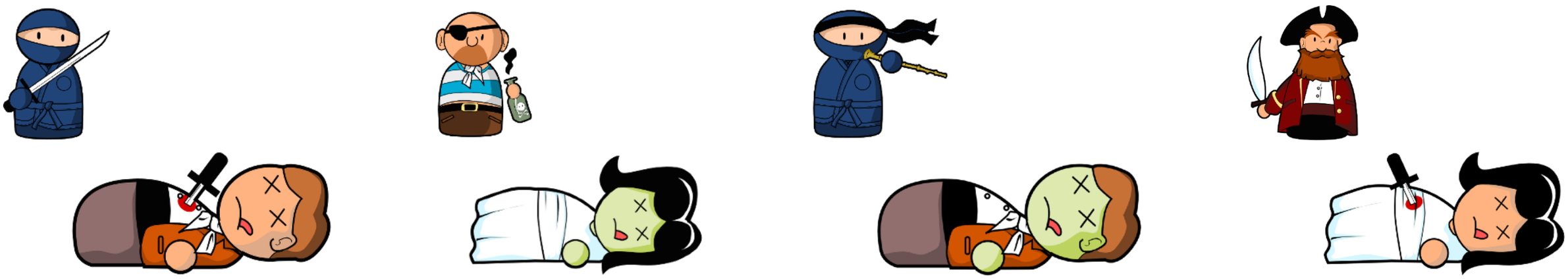
development of topics from *Science* over time (1880-2002)



(Blei 2012)

Last time

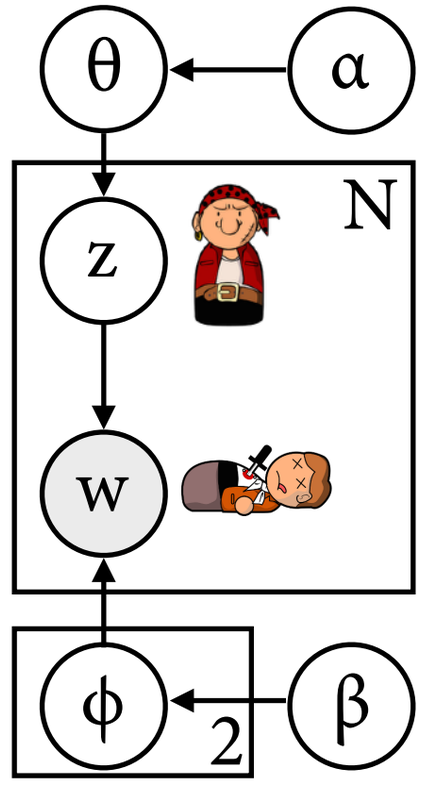
Say you come across some people who have been stabbed or poisoned.
You know that each of them was killed by a pirate or a ninja.
You can tell how each person died, but not by whom they were killed.



Generative story

- We assume deaths are generated as follows:

$(\theta_{pi}, \theta_{ni}) \sim \text{Dir}(\alpha, \alpha)$
$(\phi_{st|pi}, \phi_{po|pi}), (\phi_{st|ni}, \phi_{po|ni}) \sim \text{Dir}(\beta, \beta)$
$z_1, \ldots, z_K \sim \text{Categorical}(\theta)$
$w_i \sim \text{Categorical}(\phi_{zi})$

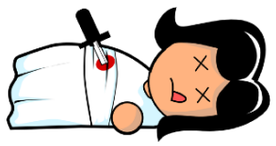


- That is:
  - $P(z_i = pi) = \theta_{pi}$, $P(z_i = ni) = \theta_{ni}$
  - if z_i came out as "pi", then $P(w_i = st) = \phi_{st|pi}$

I abbreviate $\theta = (\theta_{pi}, \theta_{ni})$, $\phi_{pi} = (\phi_{st|pi}, \phi_{po|pi})$, $\phi_{ni} = (\phi_{st|ni}, \phi_{po|ni})$.
α, β are assumed given and are called *hyperparameters*.

Supervised learning

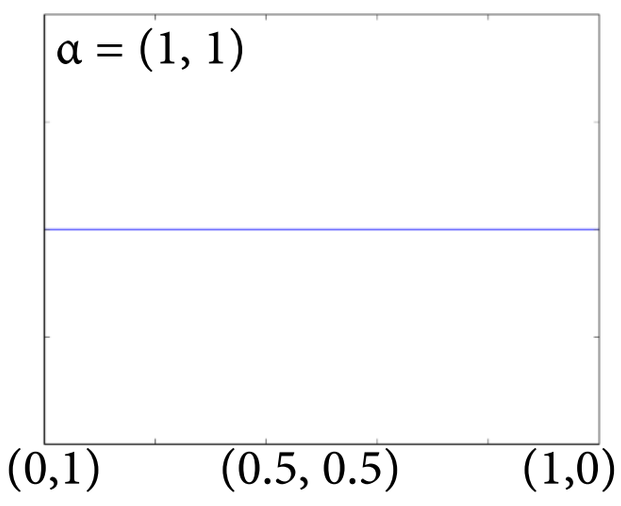
If all killers are known, P(M | D) is easy to compute.

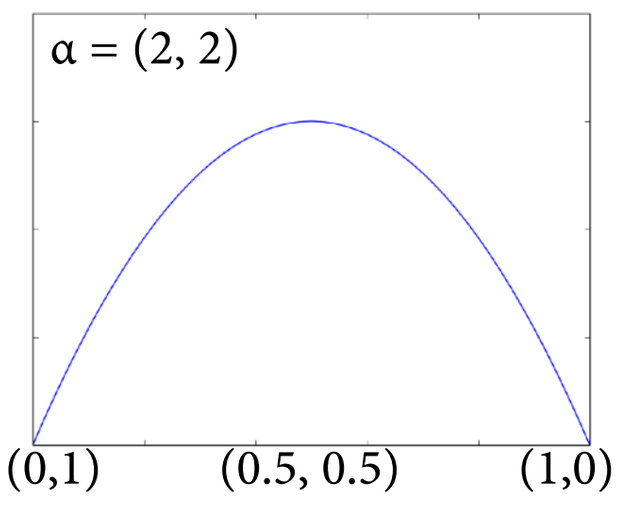
i	z_i	w_i
1		
2		

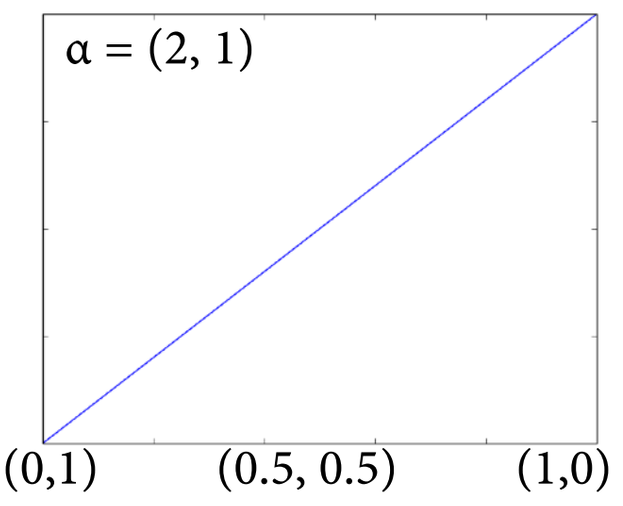
$$P(M) = \mathrm{Dir}_{\alpha,\alpha}(\theta) \cdot \mathrm{Dir}_{\beta,\beta}(\phi_{\mathrm{pi}}) \cdot \mathrm{Dir}_{\beta,\beta}(\phi_{\mathrm{ni}})$$
$$\propto \theta_{\mathrm{pi}}^{\alpha-1} \cdot \theta_{\mathrm{ni}}^{\alpha-1} \cdot \phi_{\mathrm{st|pi}}^{\beta-1} \cdot \phi_{\mathrm{po|pi}}^{\beta-1} \cdot \phi_{\mathrm{st|ni}}^{\beta-1} \cdot \phi_{\mathrm{po|ni}}^{\beta-1}$$

$$P(D \mid M) = P(z_1 = \mathrm{pi}, w_1 = \mathrm{st}, z_2 = \mathrm{ni}, w_2 = \mathrm{po})$$
$$= \theta_{\mathrm{pi}} \cdot \phi_{\mathrm{st|pi}} \cdot \theta_{\mathrm{ni}} \cdot \phi_{\mathrm{po|ni}}$$

$$P(M \mid D) \propto P(D \mid M) \cdot P(M)$$
$$\propto \theta_{\mathrm{pi}}^{\alpha} \cdot \theta_{\mathrm{ni}}^{\alpha} \cdot \phi_{\mathrm{st|pi}}^{\beta} \cdot \phi_{\mathrm{po|pi}}^{\beta-1} \cdot \phi_{\mathrm{st|ni}}^{\beta-1} \cdot \phi_{\mathrm{po|ni}}^{\beta}$$
$$\propto \mathrm{Dir}_{\alpha+1,\alpha+1}(\theta) \cdot \mathrm{Dir}_{\beta+1,\beta}(\phi_{\mathrm{pi}}) \cdot \mathrm{Dir}_{\beta,\beta+1}(\phi_{\mathrm{ni}})$$

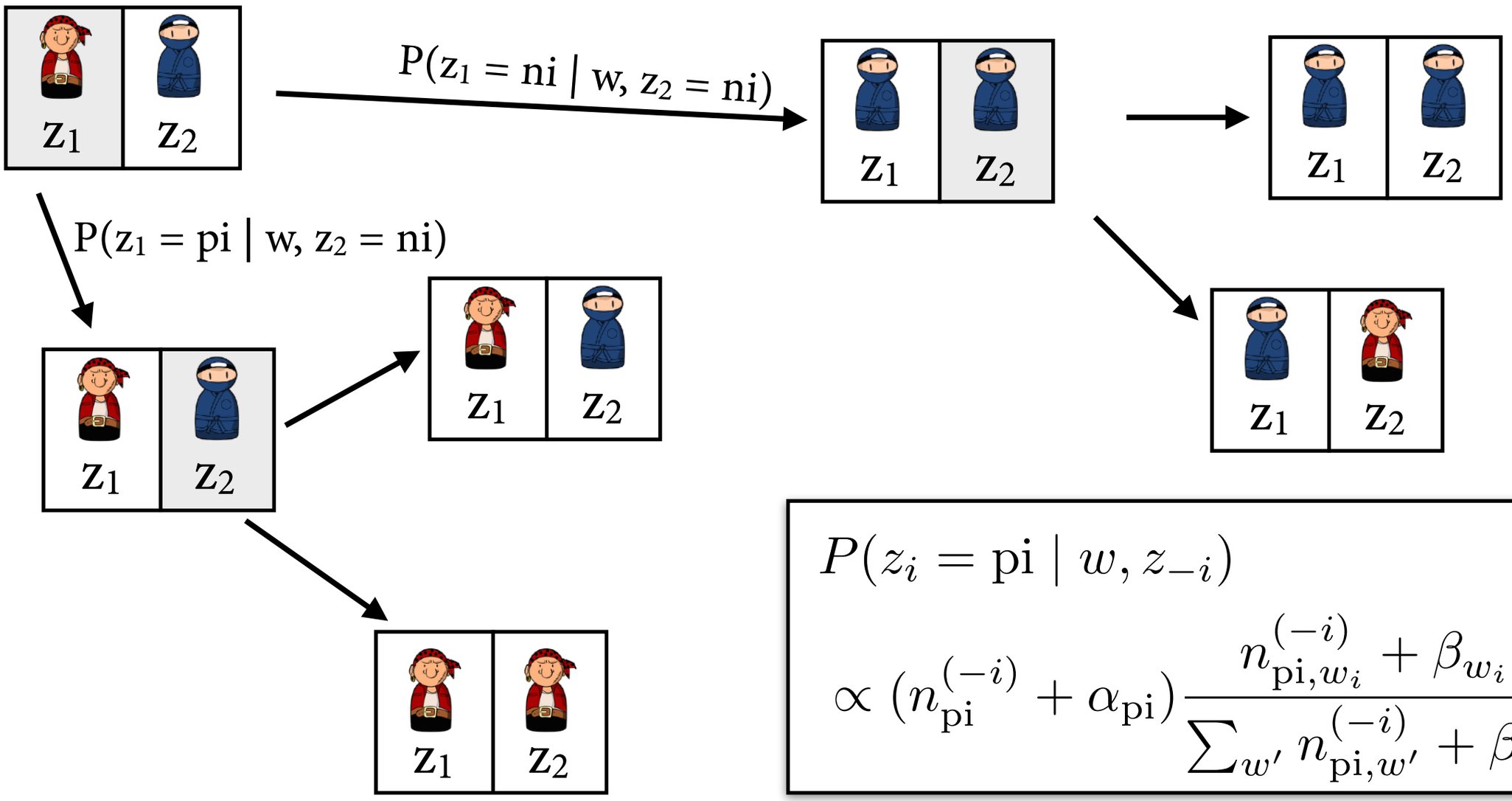






Gibbs Sampling

- Gibbs sampling is MCMC method for computing expected values under posterior distribution.



$$P(z_i = \text{pi} \mid w, z_{-i})$$

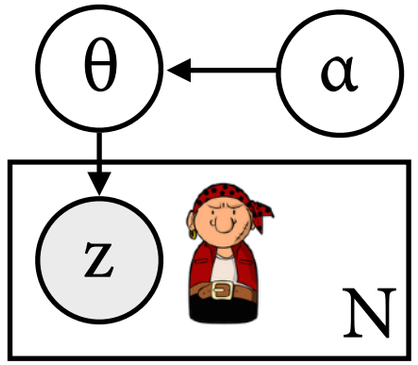
$$\propto (n_{\text{pi}}^{(-i)} + \alpha_{\text{pi}}) \frac{n_{\text{pi},w_i}^{(-i)} + \beta_{w_i|\text{pi}}}{\sum_{w'} n_{\text{pi},w'}^{(-i)} + \beta_{w'|\text{pi}}}$$

Let's simplify

- To bring out today's point more clearly, consider a parade of observable pirates and ninjas (unlike last time, where they were latent):

$\theta = (\theta_{pi}, \theta_{ni}) \sim \mathrm{Dir}(\alpha_{pi}, \alpha_{ni})$

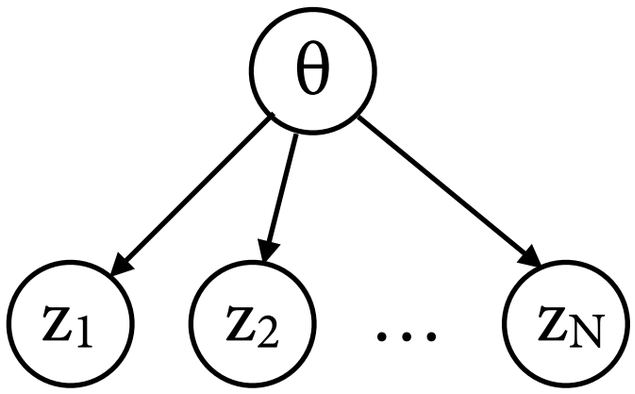
$z_1, \ldots, z_N \sim \mathrm{Categorical}(\theta)$



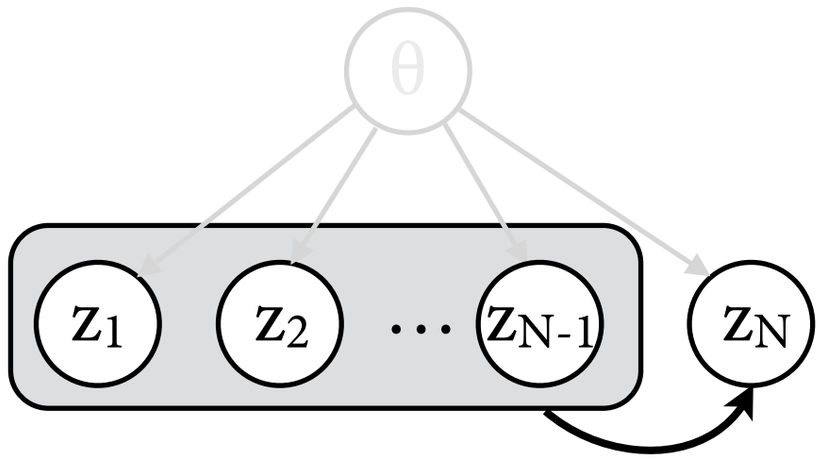
- Posterior after observations $D = z_1, \ldots, z_N$:

$$P(\theta \mid D) = \mathrm{Dir}_{\alpha_{\mathrm{pi}}+n_{\mathrm{pi}},\alpha_{\mathrm{ni}}+n_{\mathrm{ni}}}(\theta_{\mathrm{pi}}, \theta_{\mathrm{ni}})$$

Predictive distributions



posterior
$P(\theta \mid z_1, \ldots, z_N)$



predictive probability
$P(z_N \mid z_1, \ldots, z_{N-1})$

Predictive distribution

- We can determine the predictive distribution by marginalizing over the model:

$$P(z_i = \text{pi} \mid z_1, \ldots, z_{i-1}) = \int P(z_i = \text{pi} \mid M, z_1, \ldots, z_{i-1}) \cdot P(M \mid z_1, \ldots, z_{i-1}) \, \mathrm{d}M$$

$$= \int \theta_{\text{pi}} \cdot P(M \mid z_1, \ldots, z_{i-1}) \, \mathrm{d}M$$

$$= \ldots$$

$$= \frac{\alpha_{\text{pi}} + n_{\text{pi}}^{-i}}{\alpha_{\text{pi}} + \alpha_{\text{ni}} + (i-1)}$$

pirates in 1, …, i-1

- Or equivalently, with $\alpha = \alpha_{\text{pi}} + \alpha_{\text{ni}}$ and $\pi_{\text{pi}} = \alpha_{\text{pi}} / \alpha$:

$$P(z_i = \text{pi} \mid z_1, \ldots, z_{i-1}) = \frac{n_{\text{pi}}^{-i}}{\alpha + (i-1)} + \frac{\alpha \cdot \pi_{\text{pi}}}{\alpha + (i-1)}$$

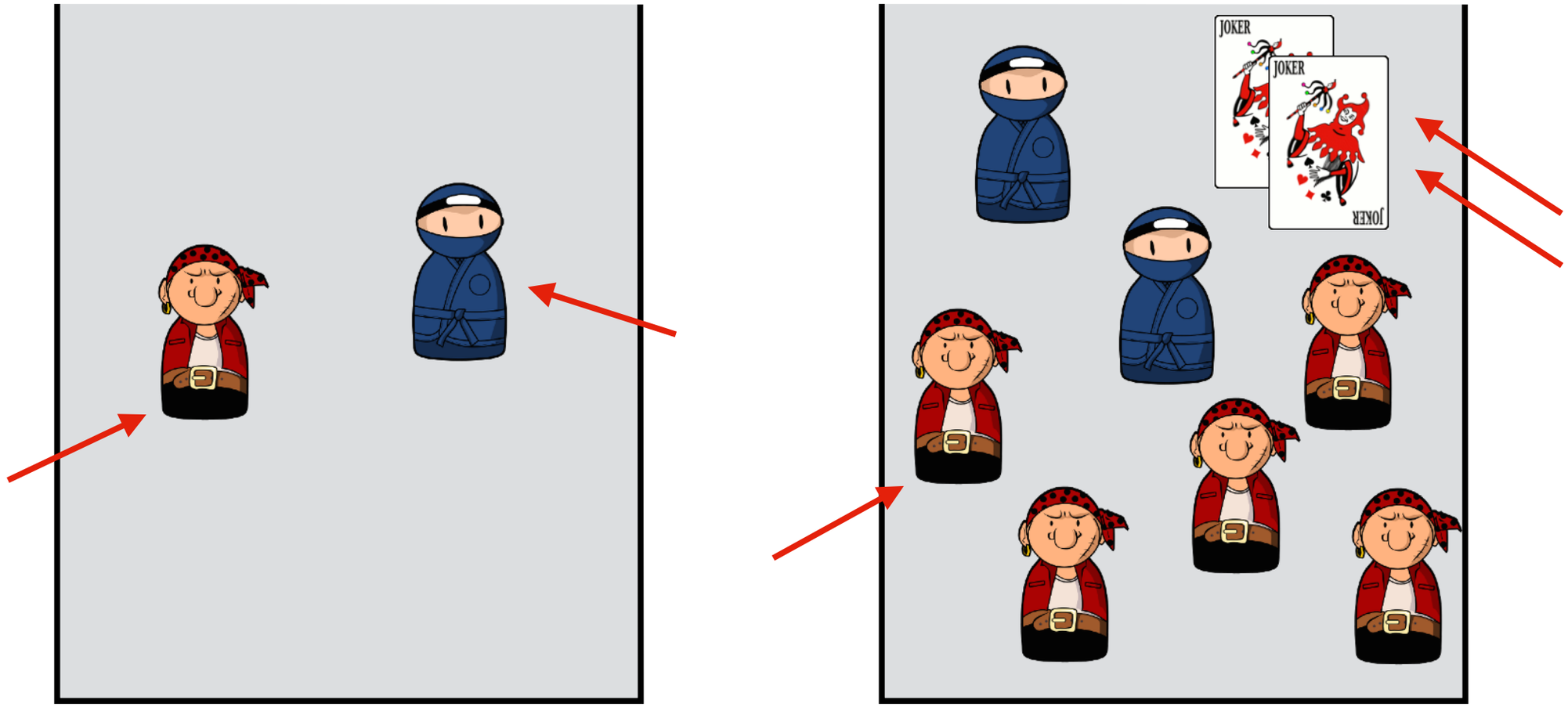
Teichmann Pocket Process

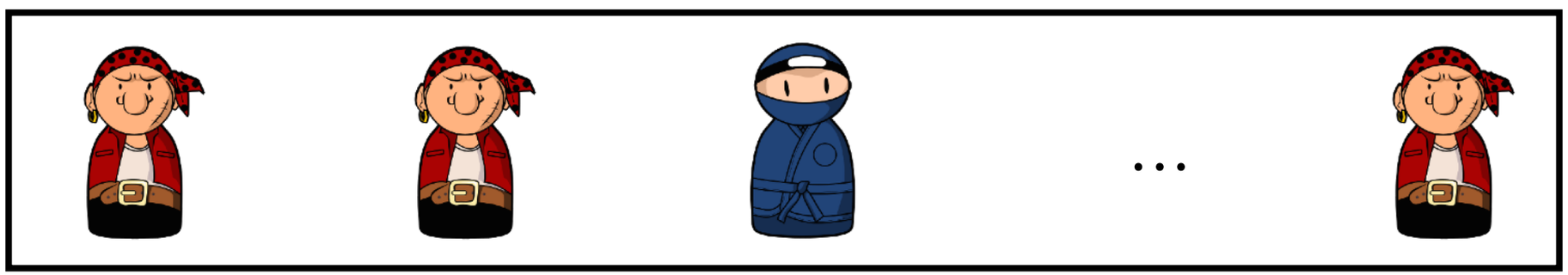
- Illustrate this distribution as follows:
  - left pocket contains α_{pi} pirates, α_{ni} ninjas
  - right pocket contains α jokers
  - randomly draw a card from right pocket
  - if it is pirate or ninja, output that guy and put him *and a clone of him* in right pocket
  - if it is joker, randomly draw a guy X from left pocket; output X, put him back in left pocket, and put clone in right

- Officially called "Polya urn scheme", but I prefer Christoph Teichmann's pocket metaphor.

Teichmann Pocket Process

$$P(z_i = \text{pi} \mid z_1, \ldots, z_{i-1}) = \frac{n_{\text{pi}}^{-i}}{\alpha + (i-1)} + \frac{\alpha}{\alpha + (i-1)} \cdot \pi_{\text{pi}}$$





z_1 z_2 z_3 ... z_7

Non-parametric models

- Key limitation of models so far: must specify number K of topics / of killer types.

- Will now generalize this to a class of Bayesian models that automatically pick as many killer types as needed to fit the data.

- Called *non-parametric models* because number of parameters not fixed in advance like in $(\theta_1, \ldots, \theta_K)$.

Non-parametric models

- Idea: prob dist over infinite space of events.
  - assume some *base distribution* G over these events
  - add Polya-urn-style caching model on top of it

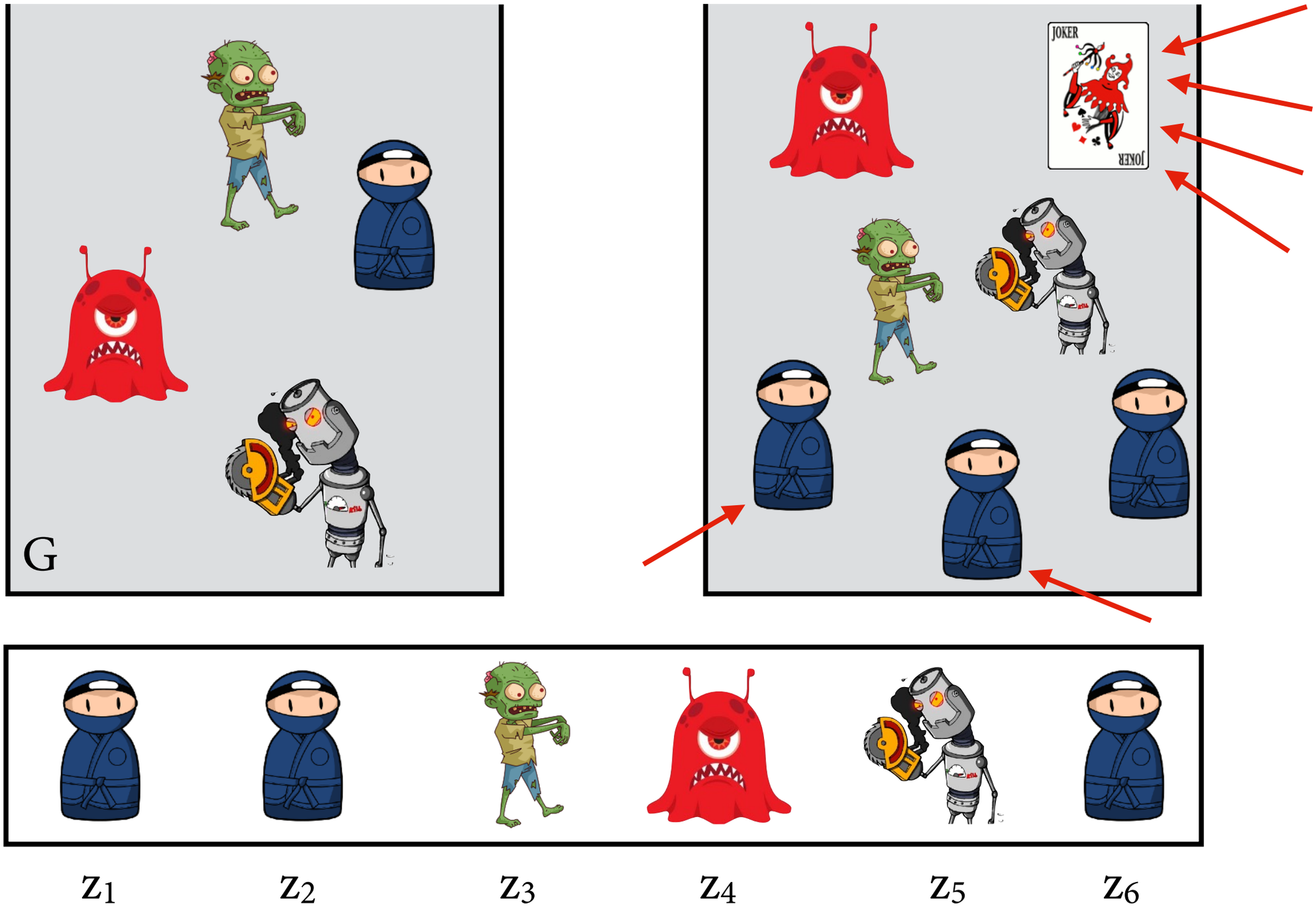
- Can simply adapt predictive distribution:

$$P(z_i = k \mid z_1, \ldots, z_{i-1}) = \frac{n_k^{-i}}{\alpha + (i-1)} + \frac{\alpha}{\alpha + (i-1)} \cdot G(k)$$

- Earlier pirate-ninja distribution:
  G(pi) = π_{pi}, G(ni) = π_{ni} with $\pi_{pi} + \pi_{ni} = 1$.

Teichmann Pocket Process

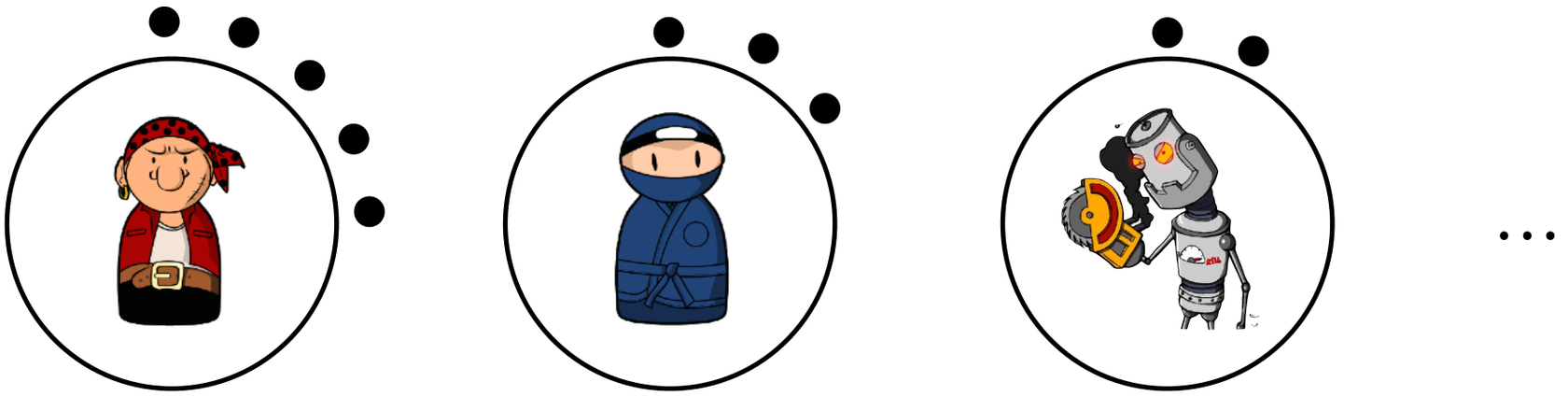
$$P(z_i = k \mid z_1, \ldots, z_{i-1}) = \frac{n_k^{-i}}{\alpha + (i-1)} + \frac{\alpha}{\alpha + (i-1)} \cdot G(k)$$



z_1 z_2 z_3 z_4 z_5 z_6

Chinese Restaurant Process

- Alternative illustration (very popular in literature):
  - Chinese restaurant with infinite sequence of tables, each of which has infinite seating capacity.
  - With probability $\frac{n_k^{-i}}{\alpha + (i-1)}$, customer i chooses to sit at table k (which has n_k^{-i} other people sitting at it).
  - With probability $\frac{\alpha}{\alpha + (i-1)}$ customer opens up new table, and label L for new table drawn at random from G(L).

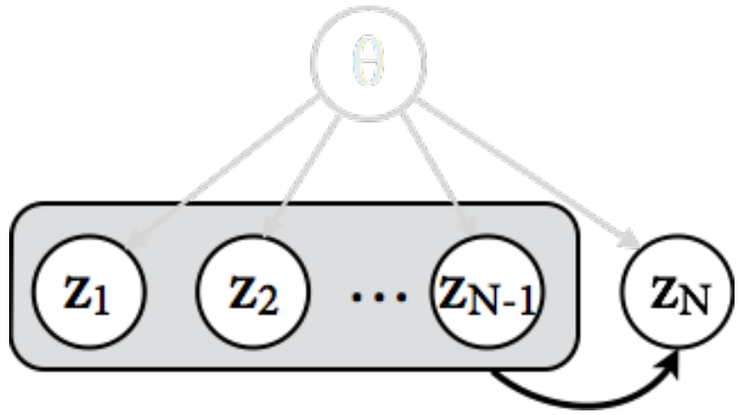


Exchangeability

- The CRP is *exchangeable:* when computing $P(z_i \mid z^{-i})$, can pretend that z_i is last event (and tables have all customers from z^{-i} on them).

- Can therefore use predictive prob in two ways:
  - as predictive prob, to predict next unseen event
  - in Gibbs sampling, to resample z_i based on the others

- De Finetti's theorem: Exchangeable observations are independent given some latent variables.
  - for CRP, distribution over latent variables is *Dirichlet process*



Grammar induction

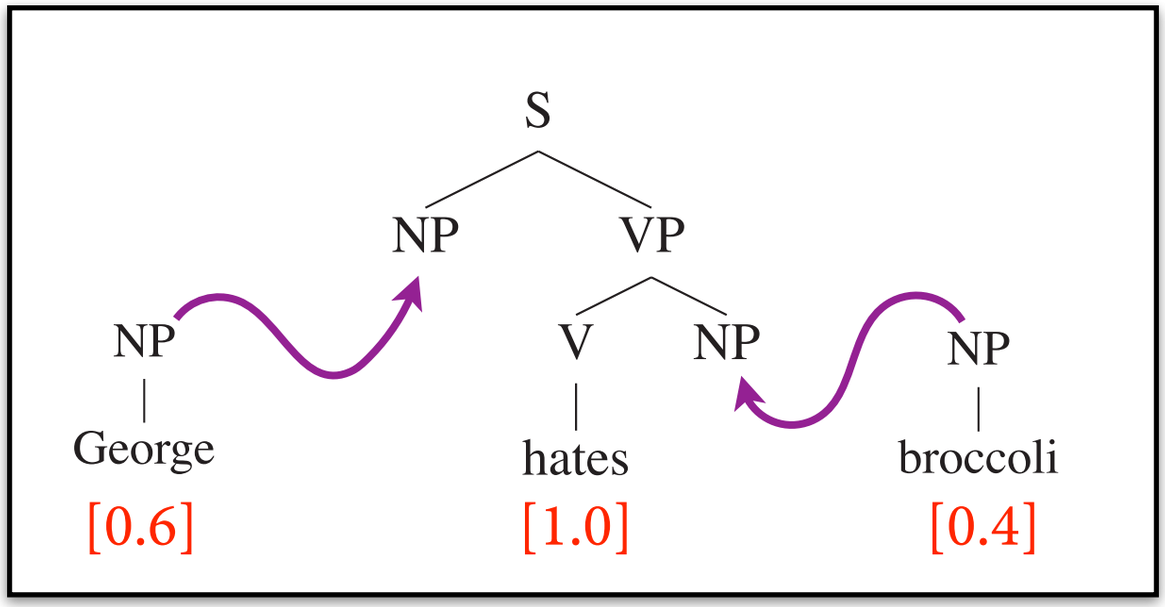
- Tree substitution grammar (TSG) is a grammar formalism in which *elementary trees* are combined using the *substitution* operation.

- In the Penn Treebank, we can only observe the derived trees.
  - Unclear how they were constructed from elementary trees.

- How can we induce a (probabilistic) TSG grammar from the Penn Treebank trees?

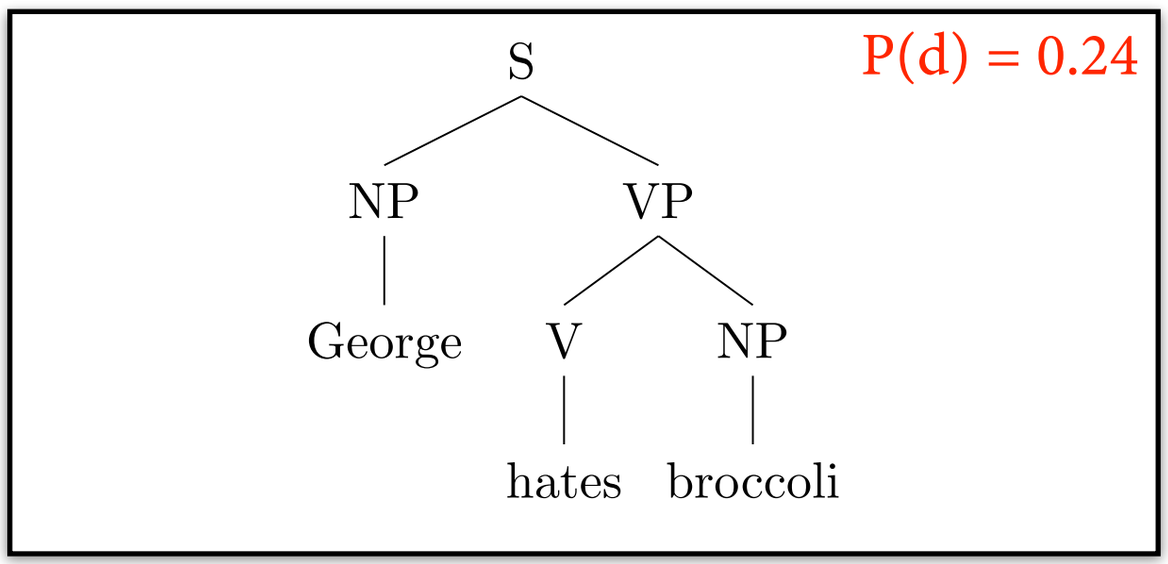
PTSG: Example

grammar with three elementary trees



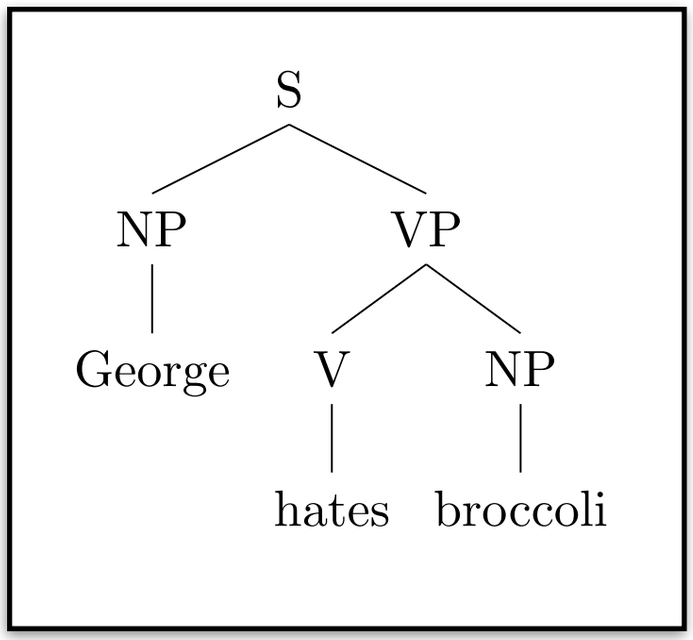
substitution into nonterminal leaves

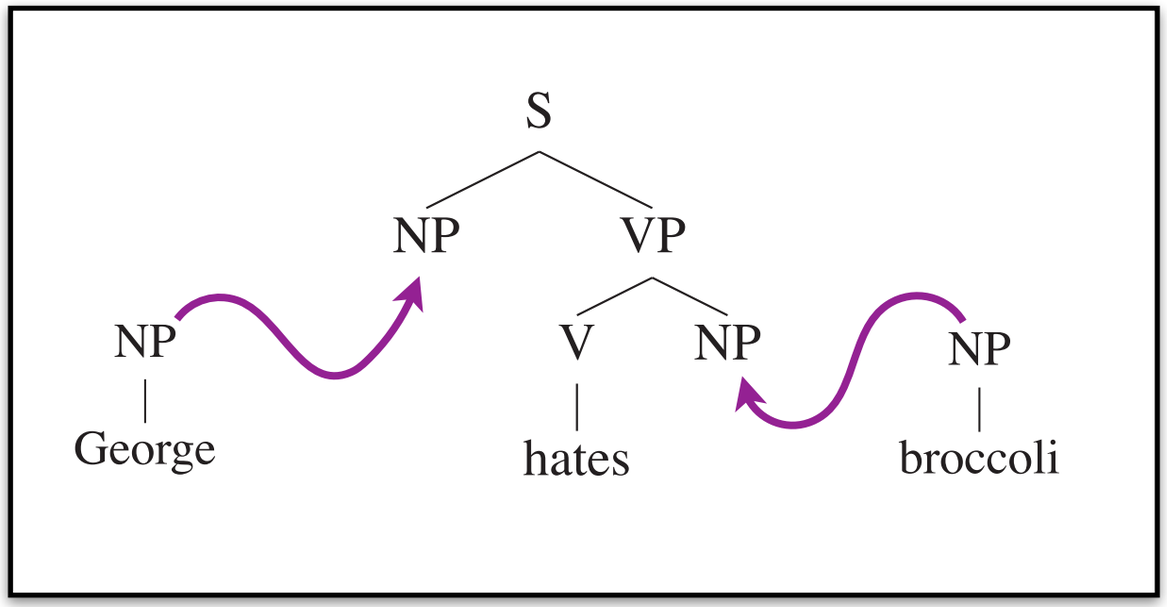
e-tree probabilities

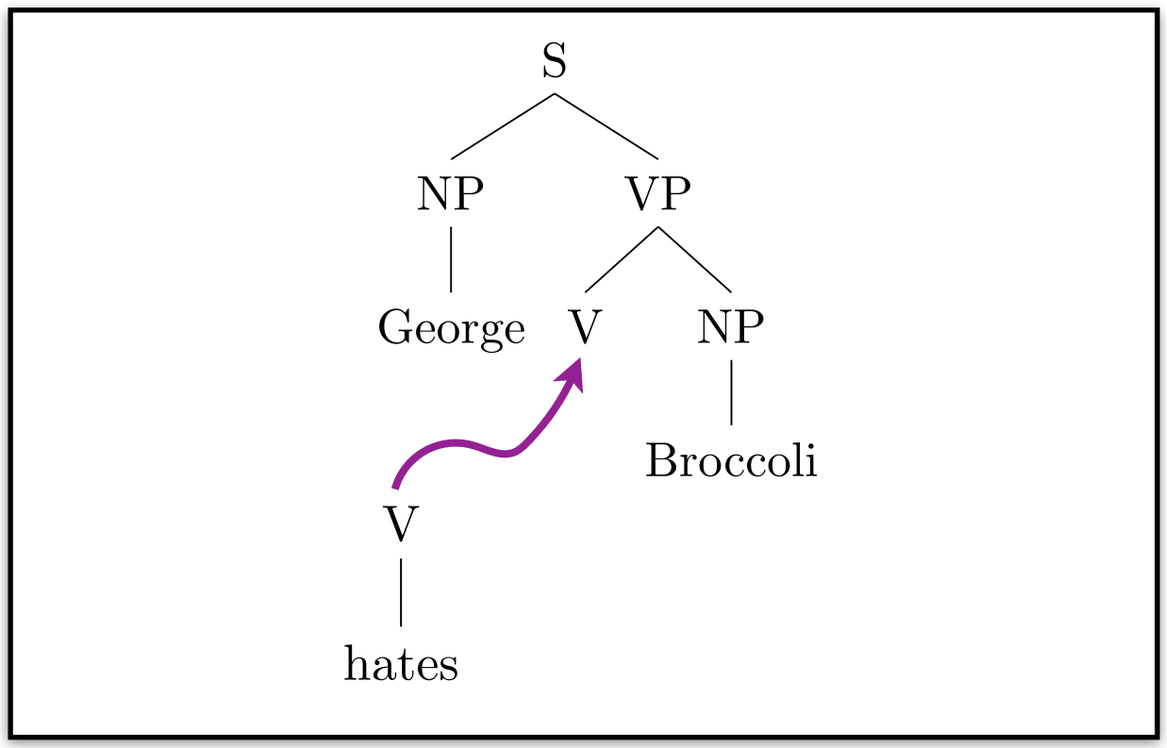


derivation tree

TSG induction







Why not EM?

- Our default method for learning with latent variables L so far: Expectation Maximization (EM).

- EM tries to find maximum-likelihood estimate:

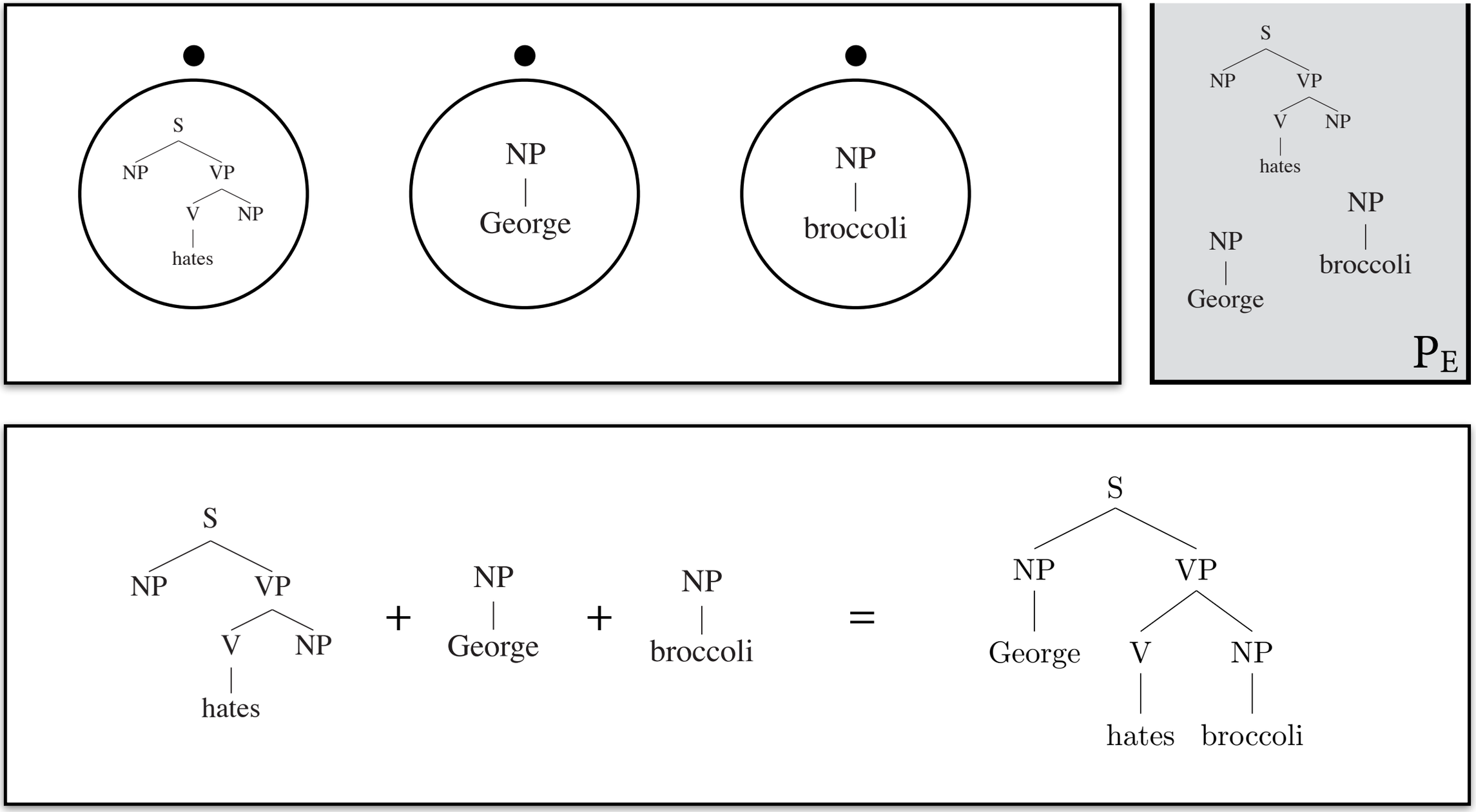
$$\begin{aligned}\max_p P(D \mid p) \\ = \max_p \sum_L P(L) \cdot P(D \mid L, p)\end{aligned}$$

- This does not work for grammar induction: Max-likelihood estimate makes each derived tree a single elementary tree.
  - Need prior on L to avoid this: What does a “reasonable” grammar look like?

Generative story

reuse k: $P(z_i = k \mid z_1, \ldots, z_{i-1}) = \frac{n_k^-}{\alpha + (i-1)}$

new table: $P(z_i = K^- + 1 \mid z_1, \ldots, z_{i-1}) = \frac{\alpha}{\alpha + (i-1)}$

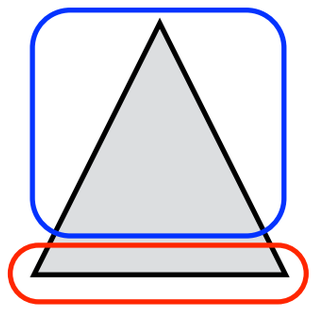


In full model (Cohn et al. 2010), there is a separate restaurant for each root symbol.
Also uses extension of CRP called the *Pitman-Yor Process*.

Base distribution for e-trees

- Prob dist over infinite set of e-trees; prob decays exponentially with size of e-tree (hence sum to 1).

$$P_{\mathrm{E}}(e|c) = \prod_{i \in I(e)} (1 - s_{c_i}) \prod_{f \in F(e)} s_{c_f} \prod_{c' \to \alpha \in e} P_{\mathrm{C}}(\alpha|c') \ ,$$



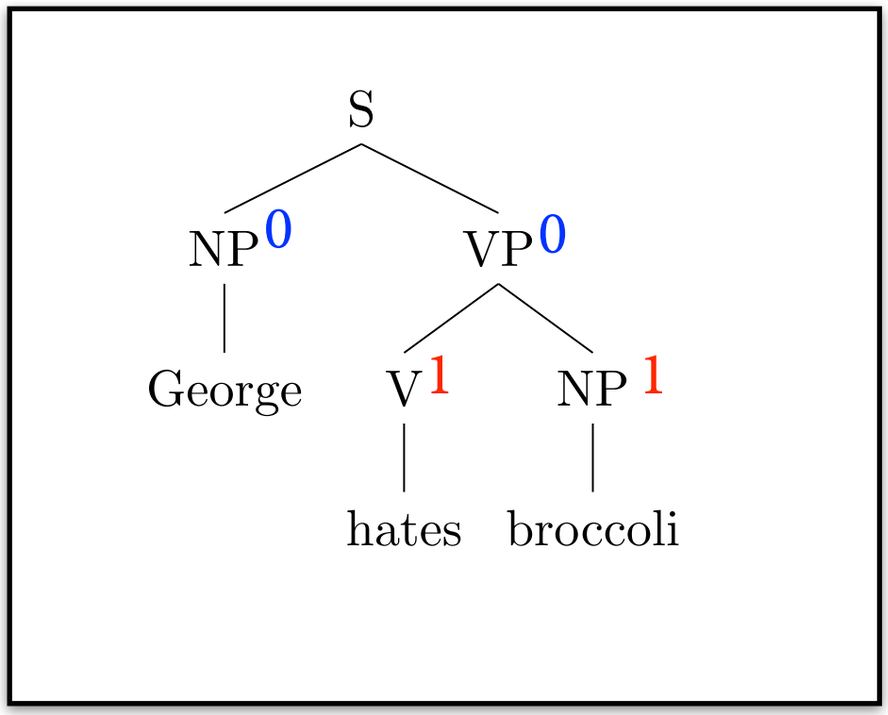
prob that e-tree creation stops with category c_i of i

base distribution over context-free rules for c'

- Also need to specify distribution P_C over cf. rules.

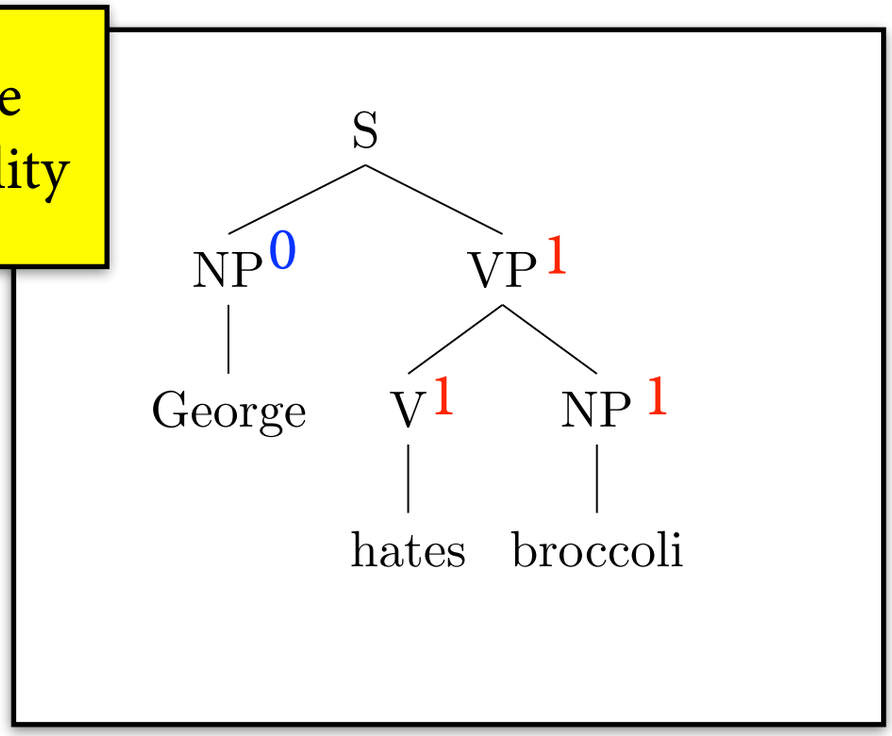
Gibbs Sampling for TSGs

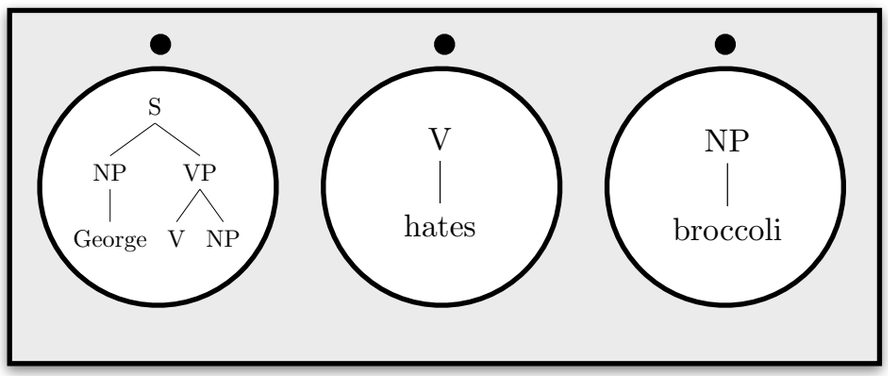
Gibbs state: nodes marked as 1 (substitution site) or 0 (inside e-tree)

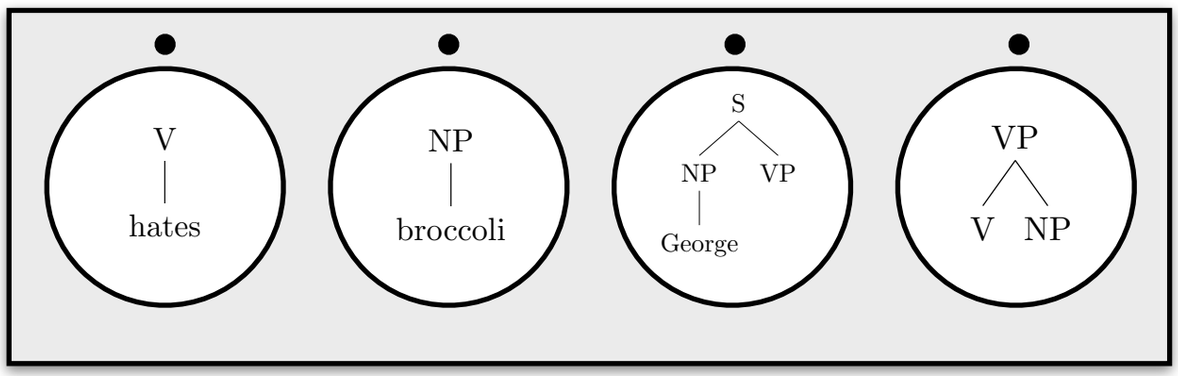


this is easy to compute because of exchangeability

P(VP 0 → 1 | all others stay)







Grammar induction

- Using Gibbs sampler, we can sample from posterior, given PTB as observations.

- We want to learn how to parse new strings with PTSG. Can do this in various ways, e.g.:
  - estimate expected values of PTSG parameters θ with Gibbs
  - include sentence to be parsed in Gibbs sampling and return most frequent tree (MPT, MPD in Cohn et al.)

Results

Parser	≤ 40		all	
	F1	**EX**	**F1**	**EX**
MLE PCFG	64.2	7.2	63.1	6.7
TSG PYP Viterbi	83.6	24.6	82.7	22.9
TSG PYP MPD	84.2	27.2	83.3	25.4
TSG PYP MPT	84.7	28.0	83.8	26.2
TSG PYP MER	85.4	27.2	84.7	25.8
DOP (Zuidema, 2007)			83.8	26.9
Berkeley parser (Petrov and Klein, 2007)	90.6		90.0	
Berkeley parser (restricted)	87.3	31.0	86.6	29.0
Reranking parser (Charniak and Johnson, 2005)	92.0		91.4	
Shindo et al., 2012 (single)	91.6		91.1	
Shindo et al., 2012 (multiple)	92.9		92.4	

Conclusion

- Predictive probabilities:
  - integrate posterior distribution over models
  - yields intuitive stochastic processes (Polya)
- Extend to non-parametric models:
  - distributions over infinite domains with caching
  - Chinese Restaurant Process, Pitman-Yor
- Apply to grammar induction, e.g. for TSGs.