TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel

Spring 2019

Lecture 15:
Finish Inference in Bayesian NLP, Start Bayesian Nonparametrics

Roadmap

- intro (1 lecture)
- deep learning for NLP (5 lectures)
- structured prediction (4.5 lectures)
- generative models, latent variables, unsupervised learning, variational autoencoders (1.5 lectures)
- **Bayesian methods in NLP (2 lectures)**
- Bayesian nonparametrics in NLP (1.5 lectures)
- research tips & other topics (0.5 lectures)

Assignments

- questions about Assignment 4?

Generative Story Template

1: Draw a set of parameters θ from $p(\Theta)$

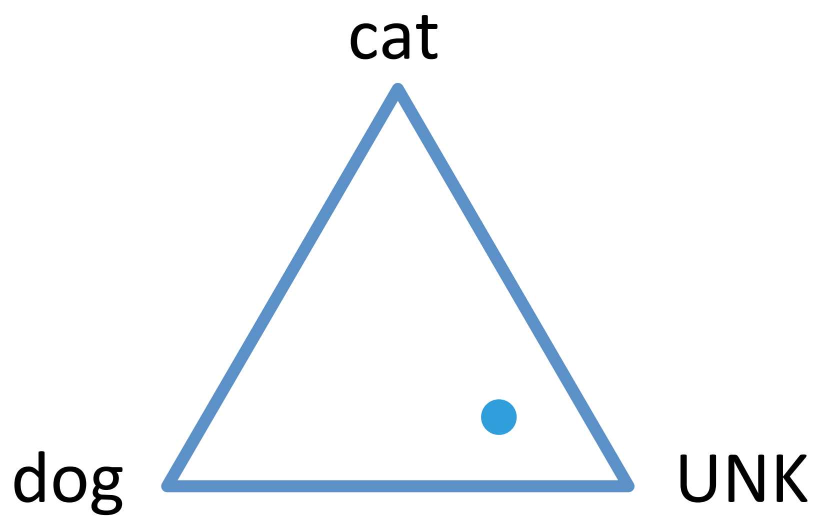
2: Draw a latent structure z from $p(Z \mid \theta)$

3: Draw the observed data x from $p(X \mid z, \theta)$

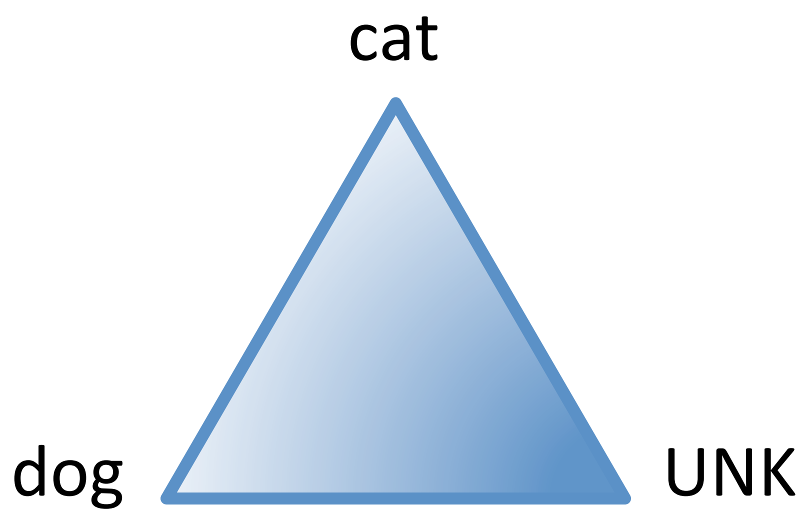
the above generative story implies the following factorization of the joint distribution:

$$p(x, z, \theta) = p(\theta)p(z \mid \theta)p(x \mid z, \theta)$$

- categorical = point on the simplex



- Dirichlet = distribution over the simplex



Posterior over Categorical Parameters?

$$p(Y = y \mid \theta) = \prod_{i=1}^{K} \theta_i^{y_i}$$

$$p(\Theta = \theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

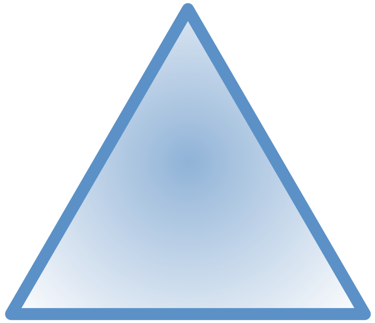
posterior (given a single observation *y*):

$$p(\theta \mid y, \alpha) \propto p(\theta \mid \alpha) p(y \mid \theta) \propto \left(\prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \right) \times \left(\prod_{i=1}^{K} \theta_i^{y_i} \right)$$

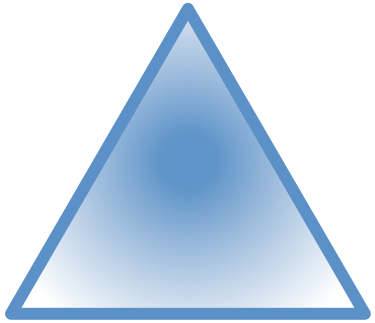
$$= \prod_{i=1}^{K} \theta_i^{\alpha_i + y_i - 1}$$

Posterior over Categorical Parameters?

prior: $p(\Theta = \theta \mid \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$



posterior: $p(\theta \mid y, \alpha) \propto \prod_{i=1}^{K} \theta_i^{\alpha_i + y_i - 1}$



posterior has form of another Dirichlet distribution!

posterior parameters: $\alpha' = \alpha + y$

Gibbs Sampling Template

$U_1, ..., U_p =$ latent variables

$U_{-i} =$ all latent variables other than U_i

$\boldsymbol{X} =$ all observed data and hyperparameters

Gibbs sampling:

initialize all U_i to values u_i

repeat until convergence:

sample u from $p(U_i \mid u_{-i}, \boldsymbol{X})$

set $U_i \leftarrow u$

LDA Generative Story

1: For each topic $k = 1...K$, draw multinomial word distribution $\beta_k \sim \text{Dirichlet}(\psi)$

2: For each document i:

  a: Draw a multinomial topic distribution $\theta^{(i)} \sim \text{Dirichlet}(\alpha)$

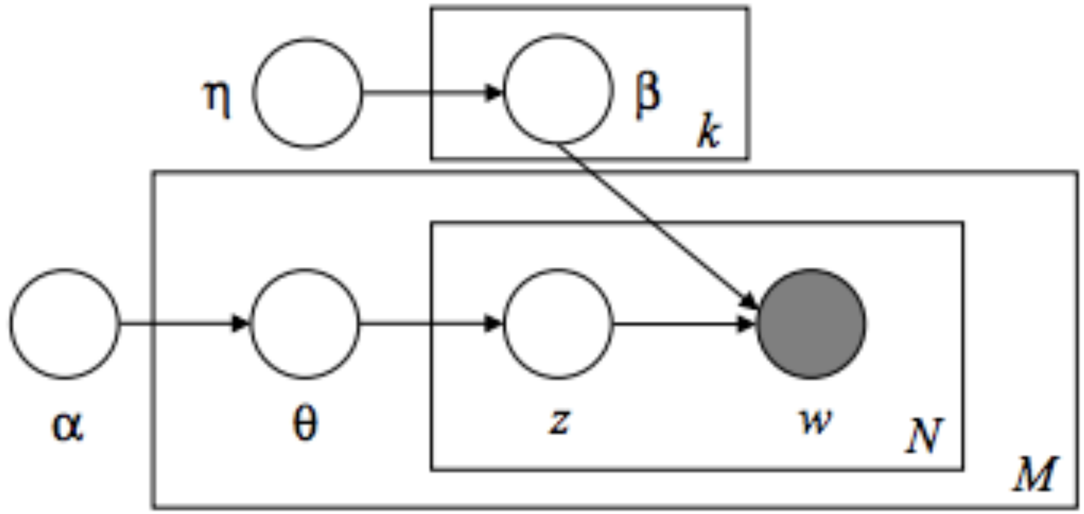
  b: For each position j in document i:

    i: Draw a topic $z^{(i,j)} \sim \text{Multinomial}(\theta^{(i)})$

    ii: Draw a word $w^{(i,j)} \sim \text{Multinomial}(\beta_{z^{(i,j)}})$

K = # topics
N = # documents
M = # words in each document
V = # words in vocabulary

Graphical Model for LDA



Gibbs Sampling for LDA

$$Z^{(i,j)} \mid \text{everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot,w^{(i,j)}})$$

$$\theta^{(i)} \in \mathbb{R}^K$$

$$\beta \in \mathbb{R}^{K \times V}$$

Gibbs Sampling for LDA

$$Z^{(i,j)} \mid \text{everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$$

$$\theta^{(i)} \mid \text{everything else} \sim \text{Dirichlet}(\alpha + m^{(i)})$$

$$\beta_k \mid \text{everything else} \sim \text{Dirichlet}(\psi + n_k)$$

$\theta^{(i)} \in \mathbb{R}^K$

$\beta \in \mathbb{R}^{K \times V}$

$m_k^{(i)} =$ # words in doc i from topic k

$n_{k,v} =$ # of times word v appears with topic k in any document

LDA

Generative Story:

$\beta_k \sim \text{Dirichlet}(\psi)$

$\theta^{(i)} \sim \text{Dirichlet}(\alpha)$

$Z^{(i,j)} \sim \text{Multinomial}(\theta^{(i)})$

Posteriors:

$\beta_k \mid \text{ everything else} \sim \text{Dirichlet}(\psi + n_k)$

$\theta^{(i)} \mid \text{ everything else} \sim \text{Dirichlet}(\alpha + m^{(i)})$

$Z^{(i,j)} \mid \text{ everything else} \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$

- we now have a way to generate samples from the posterior for the LDA model
- how should we do the following?
  - get topic assignments for each word in the document collection?
  - get topic distribution for a document?
  - get estimates of topic-word distributions for each topic?

Key Quantities

$$p(x, z, \theta \mid \alpha) = p(\theta \mid \alpha)\, p(z \mid \theta)\, p(x \mid z, \theta)$$

Our data is a set of samples: $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$

posterior: $p(z^{(1)}, \ldots, z^{(n)}, \theta \mid x^{(1)}, \ldots, x^{(n)}, \alpha)$

collapsed posterior: $p(z^{(1)}, \ldots, z^{(n)} \mid x^{(1)}, \ldots, x^{(n)}, \alpha)$

Collapsed Gibbs Sampling for LDA

Posterior: $Z^{(i,j)} \mid Z^{-(i,j)}, \theta, \beta, \boldsymbol{w}, \alpha, \psi \sim \mathrm{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$

Collapsed: $Z^{(i,j)} \mid Z^{-(i,j)}, \boldsymbol{w}, \alpha, \psi \sim ?$

- the collapsed posterior is tricky to work with because all latent variables become coupled
- i.e., we now have fewer independence assumptions to help us simplify things
- [on board]

Collapsed Gibbs Sampling for LDA

Posterior: $Z^{(i,j)} \mid Z^{-(i,j)}, \theta, \beta, \boldsymbol{w}, \alpha, \psi \sim \text{Multinomial}(\theta^{(i)} \odot \beta_{\cdot, w^{(i,j)}})$

Collapsed: $p(Z^{(i,j)} \mid Z^{-(i,j)}, \boldsymbol{w}, \alpha, \psi) = \dfrac{\psi + [n_{-(i,j),k}]_{w^{(i,j)}}}{V\psi + \sum_v [n_{-(i,j),k}]_v} \times \dfrac{\alpha + [m_{-(i,j)}]_k}{K\alpha + \sum_{k'} [m_{-(i,j)}]_{k'}}$

Roadmap

- intro (1 lecture)
- deep learning for NLP (5 lectures)
- structured prediction (4.5 lectures)
- generative models, latent variables, unsupervised learning, variational autoencoders (1.5 lectures)
- Bayesian methods in NLP (2 lectures)
- **Bayesian nonparametrics in NLP (1.5 lectures)**
- research tips & other topics (0.5 lectures)

“Nonparametric”?

- nonparametric does not mean “no parameters”
- it means that “the number of parameters grows as the data grows”
- for our purposes, think of it as “some component of the model permits an unbounded set of something”

Parametric or Nonparametric?

Model	Parametric or Nonparametric?

*parametric if vocab fixed

Parametric or Nonparametric?

Model	Parametric or Nonparametric?
Gaussian Mixture Model (GMM)	parametric
Hidden Markov Model (with GMM emissions)	parametric
Hidden Markov Model (for part-of-speech tagging, with multinomial emissions)	nonparametric*
n-gram language models	nonparametric*
LDA	nonparametric*
LSTM language model	nonparametric*
character-level LSTM language model	parametric (assuming fixed set of characters)

*parametric if vocab fixed

- “nonparametric modeling” in terms of vocab has a lot of simple engineering solutions:
  - use UNK for unknown words, do smoothing of high-order *n*-grams, etc.

- in this case, unbounded part of model is mostly determined by observed data, heuristics are useful

- modeling gets more interesting when unbounded part of model relates to **latent variables**

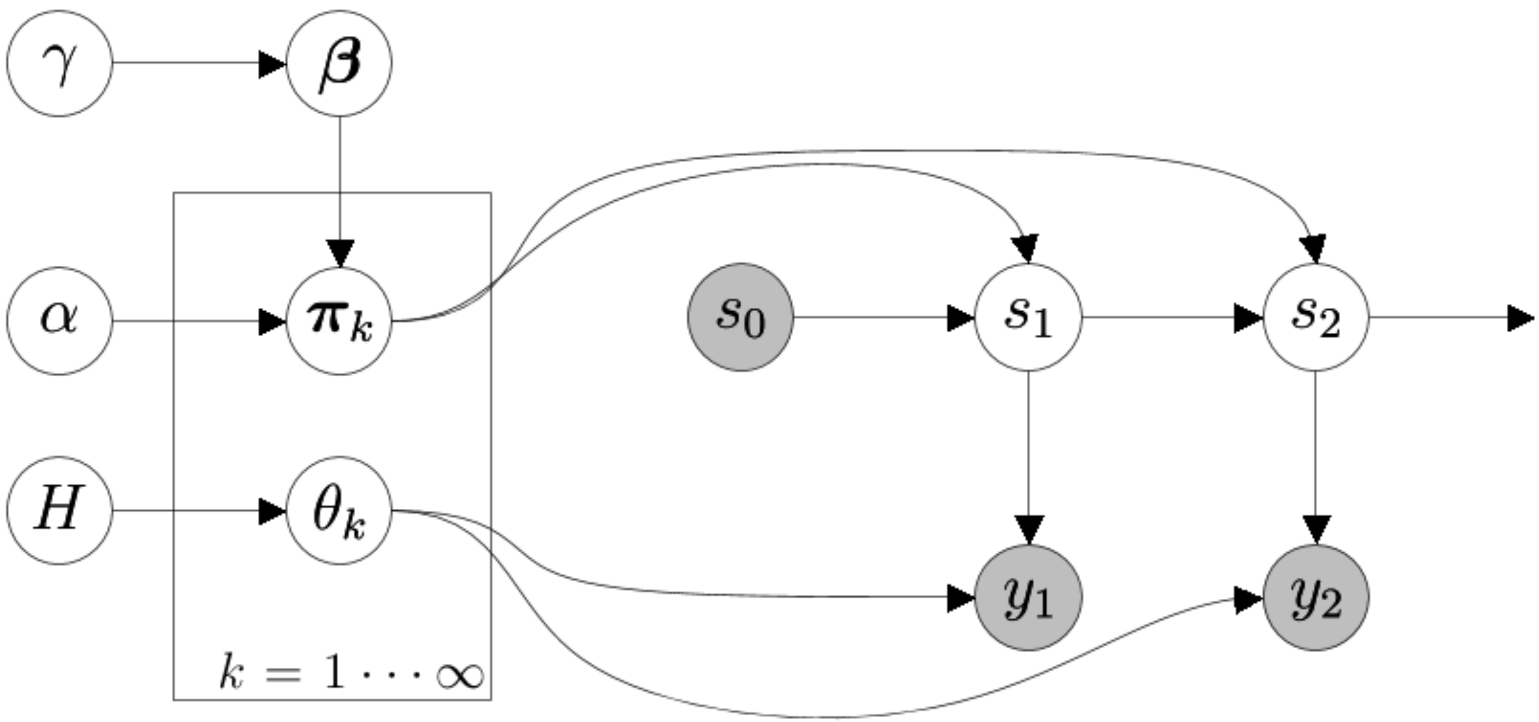
- when might you want to permit an unbounded set of latent items in a model?

Infinite Mixture Model

- number of mixture components is unbounded (grows depending on the data)
- e.g., LDA with an unbounded set of topics

“Infinite” HMM

- HMMs permit infinite sequences already
- what’s new here?
- infinite number of **hidden states**:



Beal et al. (2002), Teh et al. (2005)

"Infinite" PCFG

- PCFGs can already handle infinite-length derivations
- "infinite" here means an infinite number of **nonterminals**:

HDP-PCFG

$\boldsymbol{\beta} \sim \mathrm{GEM}(\alpha)$ [draw top-level symbol weights]

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

- $\phi_z^T \sim \mathrm{Dirichlet}(\alpha^T)$ [draw rule type parameters]
- $\phi_z^E \sim \mathrm{Dirichlet}(\alpha^E)$ [draw emission parameters]
- $\phi_z^B \sim \mathrm{DP}(\alpha^B, \boldsymbol{\beta}\boldsymbol{\beta}^T)$ [draw binary production parameters]

For each node i in the parse tree:

- $t_i \sim \mathrm{Multinomial}(\phi_{z_i}^T)$ [choose rule type]
- If $t_i = \textsc{Emission}$:
  - $x_i \sim \mathrm{Multinomial}(\phi_{z_i}^E)$ [emit terminal symbol]
- If $t_i = \textsc{Binary-Production}$:
  - $(z_{L(i)}, z_{R(i)}) \sim \mathrm{Multinomial}(\phi_{z_i}^B)$ [generate children symbols]

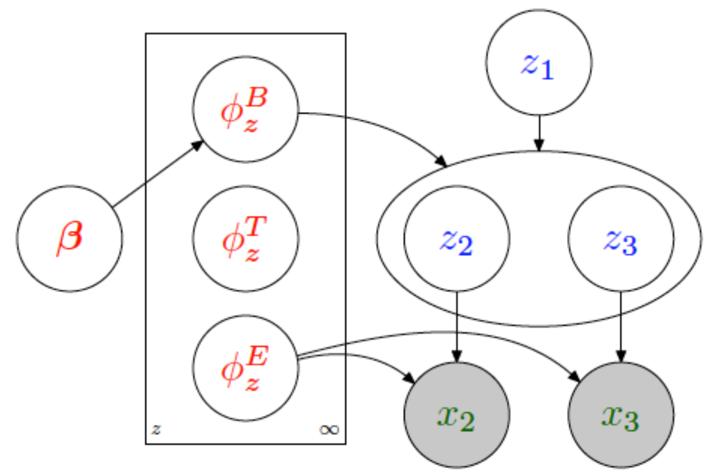


Figure 2: The definition and graphical model of the HDP-PCFG. Since parse trees have unknown structure, there is no convenient way of representing them in the visual language of traditional graphical models. Instead, we show a simple fixed example tree. Node 1 has two children, 2 and 3, each of which has one observed terminal child. We use $L(i)$ and $R(i)$ to denote the left and right children of node i.

Liang et al. (2007), Finkel et al. (2007)

- when might you want to permit an unbounded set of latent items in a model?
  - # topics in LDA
  - # Gaussians in a Gaussian Mixture Model
  - # hidden states in an HMM
  - # nonterminals in a PCFG
  - morph lexicon for morphological segmentation
  - lexicon for Chinese word segmentation
  - # coreference chains in coreference resolution
  - # senses for a word when learning sense-specific word embeddings
  - # dimensions in an embedding (?!)

- we need priors over distributions that permit an unbounded set of items

LDA Generative Story

1: For each topic $k = 1...K$, draw multinomial word distribution $\beta_k \sim \text{Dirichlet}(\psi)$

2: For each document i:

  a: Draw a multinomial topic distribution $\theta^{(i)} \sim \text{Dirichlet}(\alpha)$

  b: For each position j in document i:

    i: Draw a topic $z^{(i,j)} \sim \text{Multinomial}(\theta^{(i)})$

    ii: Draw a word $w^{(i,j)} \sim \text{Multinomial}(\beta_{z^{(i,j)}})$

K = # topics
N = # documents
M = # words in each document
V = # words in vocabulary

dimensionality of alpha must be *K* (the number of topics)

Dirichlet Process (DP)

- "distribution over distributions"
- unlike Dirichlet distribution, DP does not require pre-specifying number of components
- we'll now describe how a DP generates a distribution over an unbounded set of items

Running Example

- let’s say we’re trying to segment words into morphological units without any supervision:
  – walking → walk + ing
  – restarted → re + start + ed

- what is the unbounded set of latent items here?
  – lexicon of possible morphological units

Dirichlet Process (DP)

- contains a "base distribution" G_0
- simple example base distribution for our morph lexicon:

$$G_0(m) = p_{\text{len}}(|m|) \prod_{i=1}^{|m|} p_{\text{char}}(m_i)$$

- e.g., probability of "ing":

$$G_0(\text{"ing"}) = p_{\text{len}}(3)\ p_{\text{char}}(\text{i})\ p_{\text{char}}(\text{n})\ p_{\text{char}}(\text{g})$$

Dirichlet Processes

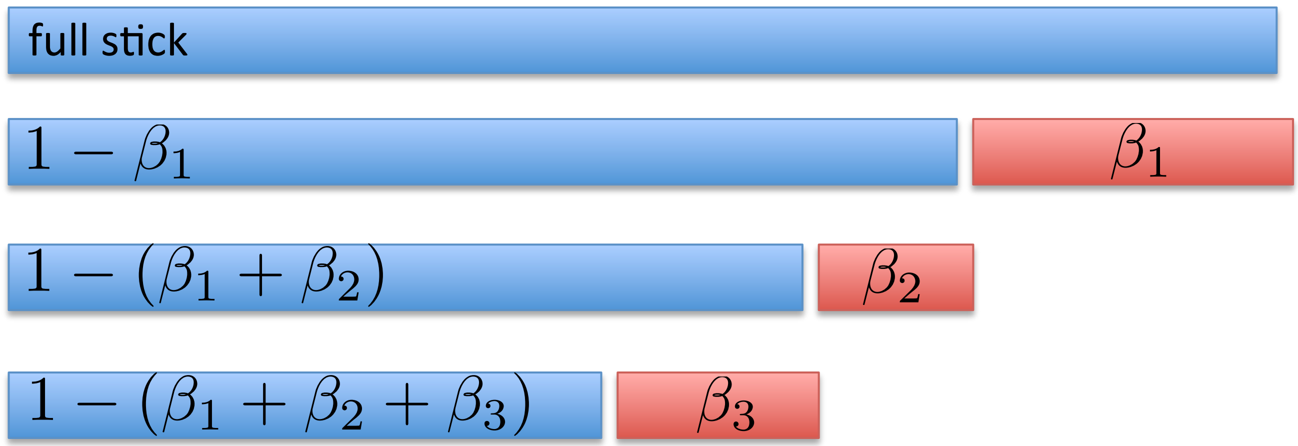
- our unbounded distribution over items will choose its items by sampling from the base distribution
- base distribution typically has an infinite set of items with nonzero probability, as in our example:

$$G_0(m) = p_{\text{len}}(|m|) \prod_{i=1}^{|m|} p_{\text{char}}(m_i)$$

Items and Probabilities

- base distribution provides the items ("atoms"), as many as we want
- where do their probabilities come from?
- we need an infinite set of probabilities that sum to 1
- DPs have another parameter: concentration (strength) parameter *s*

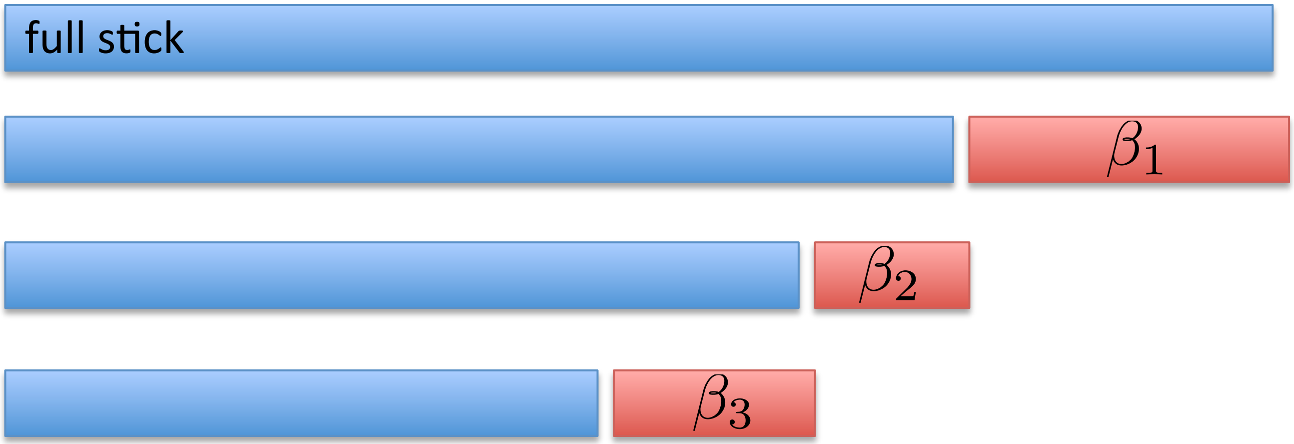
Stick-Breaking Process



…

- the betas form an infinite sequence that sums to 1
- they provide probabilities for an infinite set of items!

Stick-Breaking Process



$$\nu_k \sim \mathrm{Beta}(1, s)$$

$$\beta_k = \nu_k \prod_{j=1}^{k-1} (1 - \nu_j)$$

- the betas form an infinite sequence that sums to 1
- they provide probabilities for an infinite set of items!

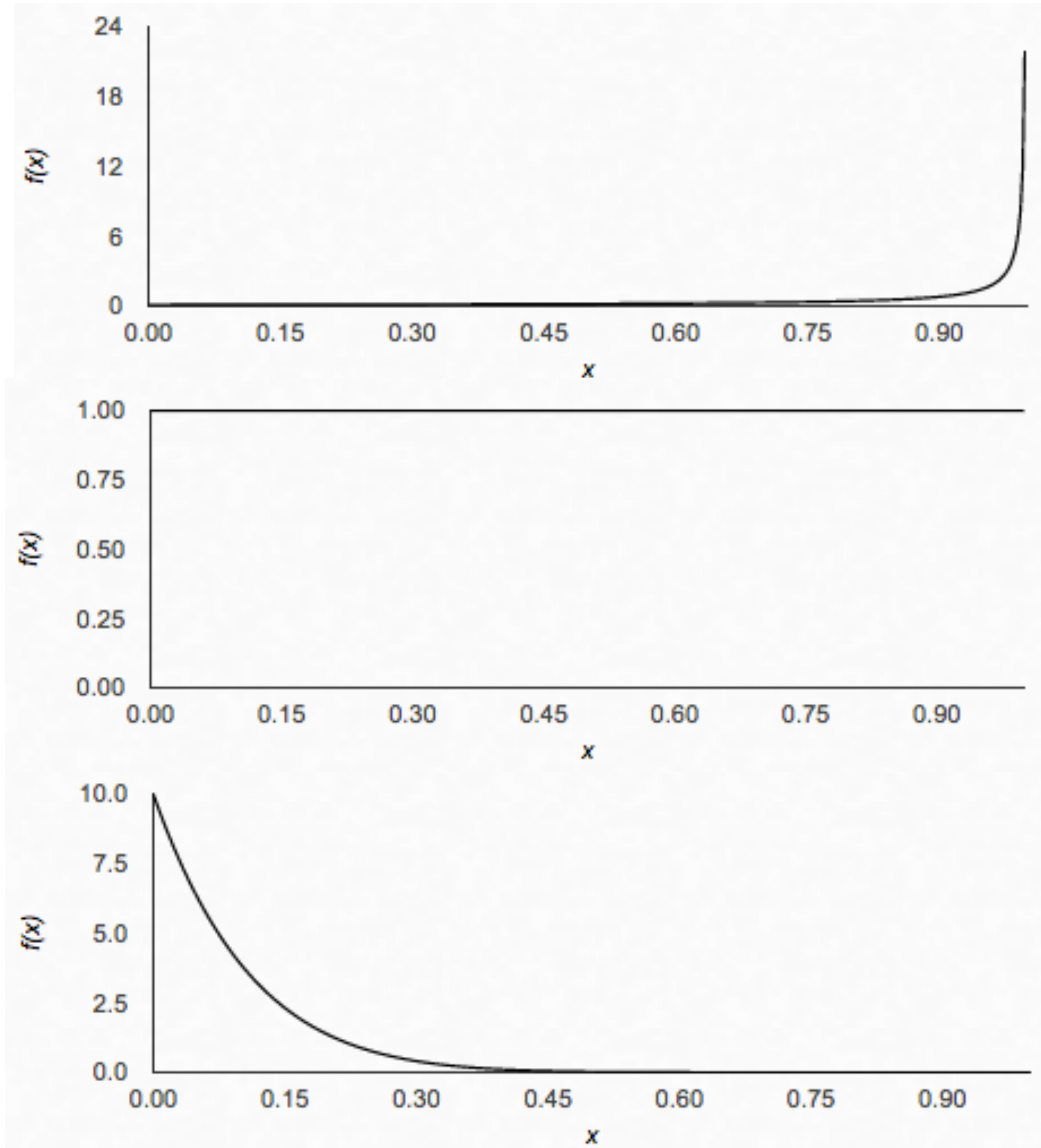
$\nu_k \sim \text{Beta}(1, s)$

Beta Distribution

$x \sim \text{Beta}(1, 0.1)$

$x \sim \text{Beta}(1, 1)$

$x \sim \text{Beta}(1, 10)$



Stick-Breaking with High Concentration (s = 10)

full stick

$$\nu_k \sim \mathrm{Beta}(1, s)$$

$$\beta_k = \nu_k \prod_{j=1}^{k-1} (1 - \nu_j)$$

Stick-Breaking with High Concentration (*s* = 10)

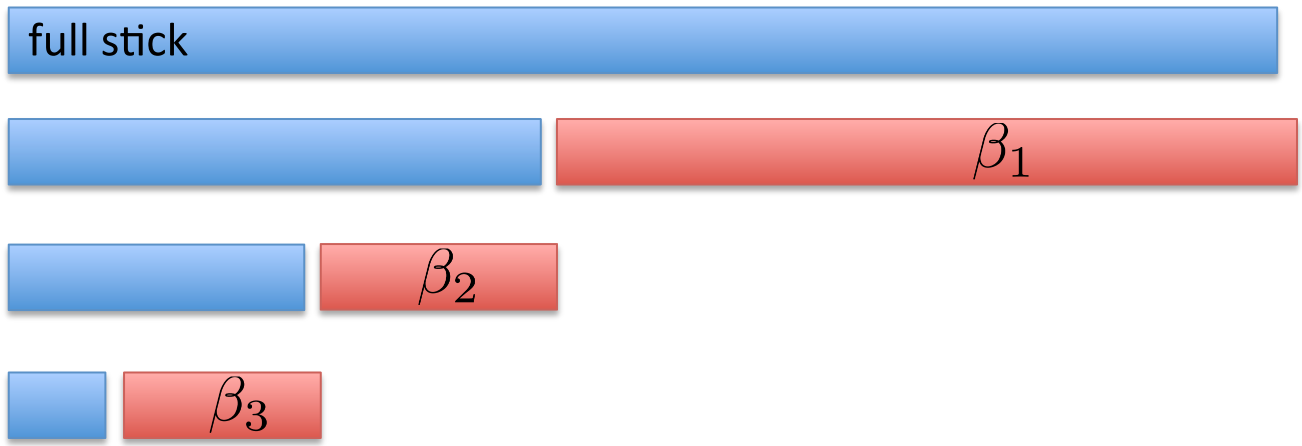
- high concentration = more of probability mass preserved for other pieces in the stick

Stick-Breaking with Low Concentration (s = 0.1)

$$\nu_k \sim \mathrm{Beta}(1, s)$$

$$\beta_k = \nu_k \prod_{j=1}^{k-1} (1 - \nu_j)$$

Stick-Breaking with Low Concentration (s = 0.1)



- low concentration = stronger power law effects in resulting probabilities

A Draw G from a DP

draw infinite probabilities from stick-breaking process with parameter s

1: $\beta \sim \mathrm{GEM}(s)$

draw atoms from base distribution

2: $\theta_1, \theta_2, \ldots \sim G_0$

atoms can be repeated!

3: the distribution G is defined as:

$$G(\theta) = \sum_{k=1}^{\infty} \beta_k \mathbb{I}[\theta = \theta_k]$$

A Draw G from a DP

1: $\beta \sim \text{GEM}(s)$ ← draw infinite probabilities from stick-breaking process with parameter s

2: $\theta_1, \theta_2, \ldots \sim G_0$ ← draw atoms from base distribution

atoms can be repeated!

3: the distribution G is defined as:

$$G(\theta) = \sum_{k=1}^{\infty} \beta_k \mathbb{I}[\theta = \theta_k]$$

$$G(\text{“ing”}) = \sum_{k=1}^{\infty} \beta_k \mathbb{I}[\text{“ing”} = \theta_k]$$

- the stick-breaking construction of the DP is useful for specifying models and defining inference algorithms

Dirichlet Process Mixture Model

- generative story for dataset $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$:

1: $\beta \sim \text{GEM}(s)$

2: $\theta_1, \theta_2, \ldots \sim G_0$ ← what should the base distribution be?

3: for $i = 1 \ldots n$, $z^{(i)} \sim \beta$

4: for $i = 1 \ldots n$, $x^{(i)} \sim p(x^{(i)} \mid \theta_{z^{(i)}})$

- each *x* is generated from a single mixture component
- the number of mixture components is unbounded