TTIC 31210:
Advanced Natural Language Processing

Kevin Gimpel

Spring 2017

Lecture 11:
Inference in Bayesian NLP

- Project proposals due Wednesday, May 10

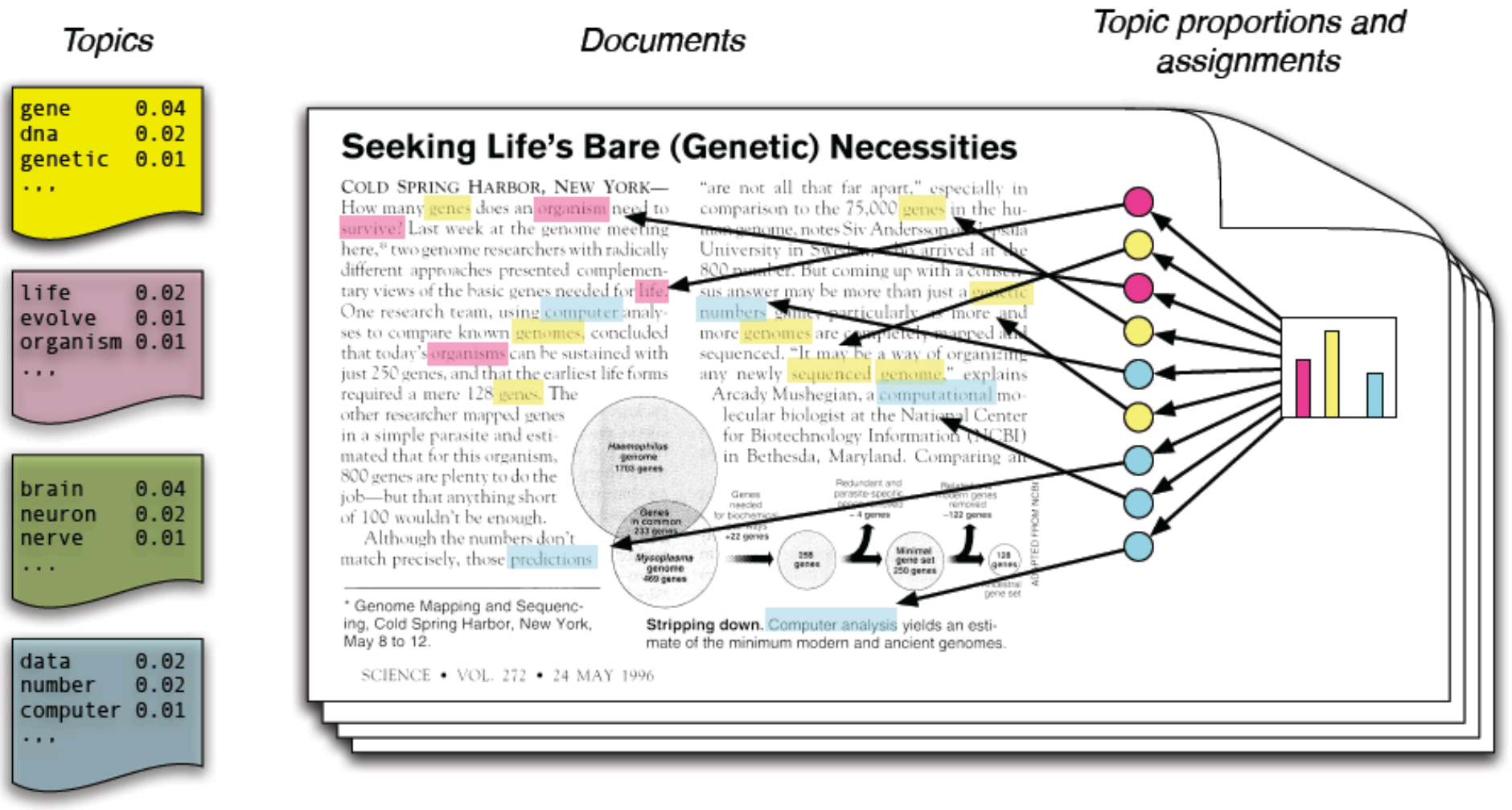
Bayesian Modeling

- this typically means:
  - defining parameters as random variables and giving them prior distributions
  - marginalizing out random variables whenever possible (like those parameter random variables)
  - aiming to compute posterior over all latent variables conditioned on observations (and maybe some hyperparameters)

Motivation

- a Bayesian approach is more impactful when there are latent variables
- NLP has a lot of models with latent variables
- unsupervised learning in NLP: “consider the unseen output as a latent variable”

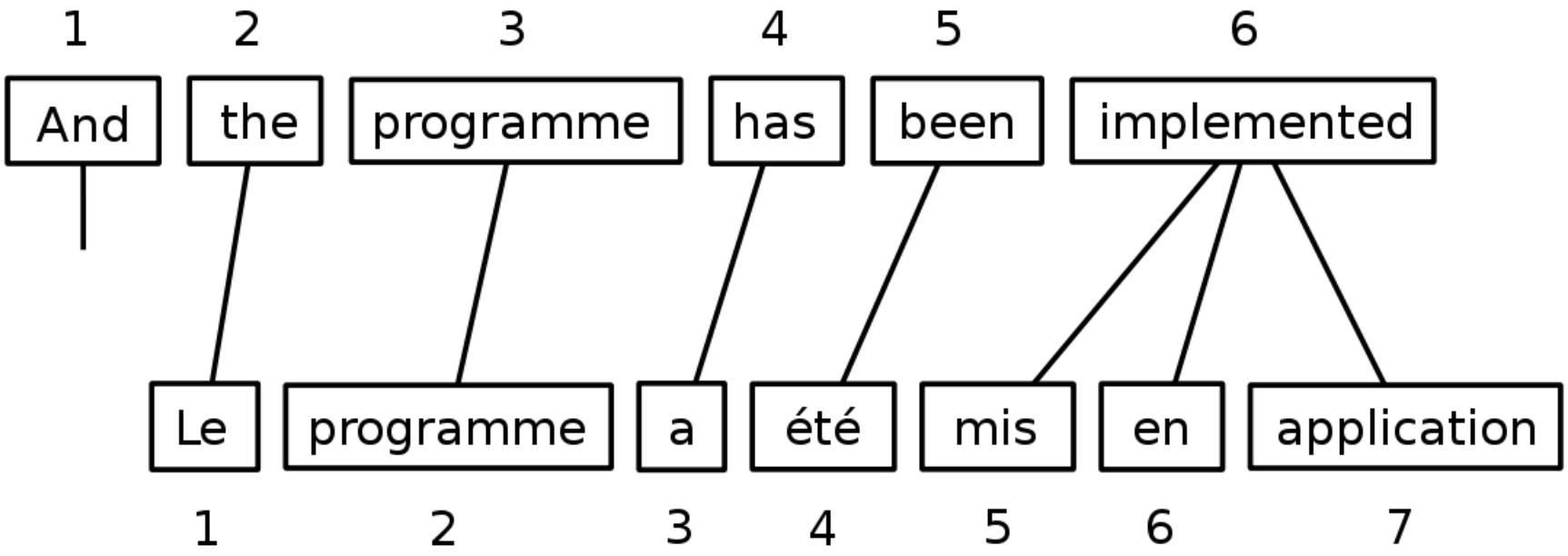
Topic Modeling



Blei et al. (2003)

Word Alignment

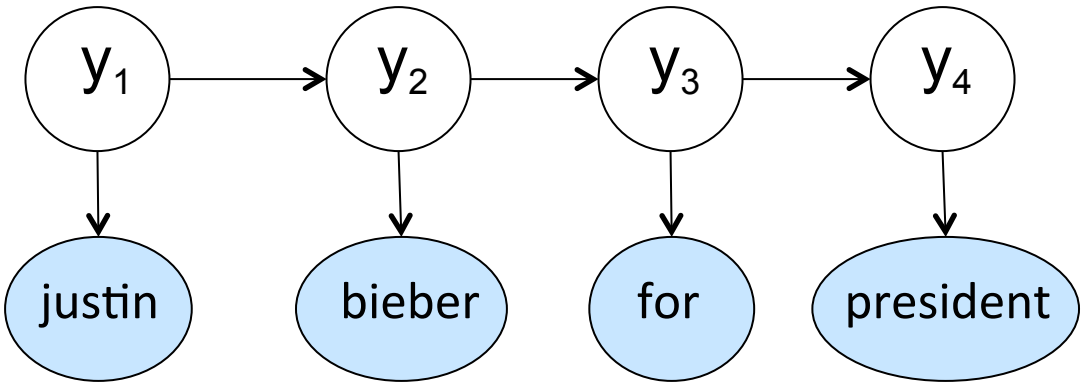
parallel sentences are observed, word alignments are latent variables:



Brown et al. (1990)

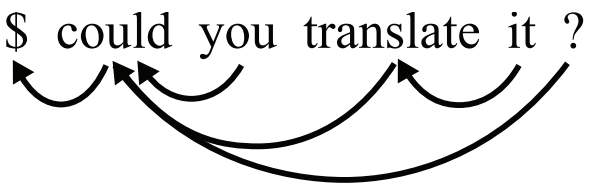
Unsupervised Part-of-Speech Tagging

sentences are observed, part-of-speech tags are treated as latent variables:



Unsupervised Dependency Parsing

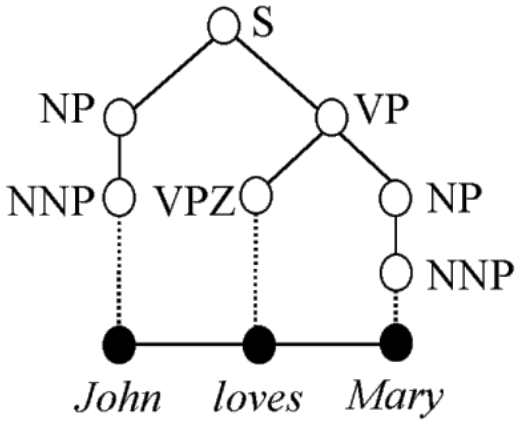
sentences are observed, dependency parse trees are treated as latent variables:



Klein & Manning
Smith & Eisner

Latent Syntactic Categories for Parsing

- split Penn Treebank syntactic categories into finer subcategories



NNP

NNP-0	Jr.	Goldman	INC.
NNP-1	Bush	Noriega	Peters
NNP-2	J.	E.	L.
NNP-3	York	Francisco	Street
NNP-4	Inc	Exchange	Co
NNP-5	Inc.	Corp.	Co.

RB

RB-0	recently	previously	still
RB-1	here	back	now
RB-2	very	highly	relatively
RB-3	so	too	as
RB-4	also	now	still
RB-5	however	Now	However

Petrov (2009)

Morphological Segmentation

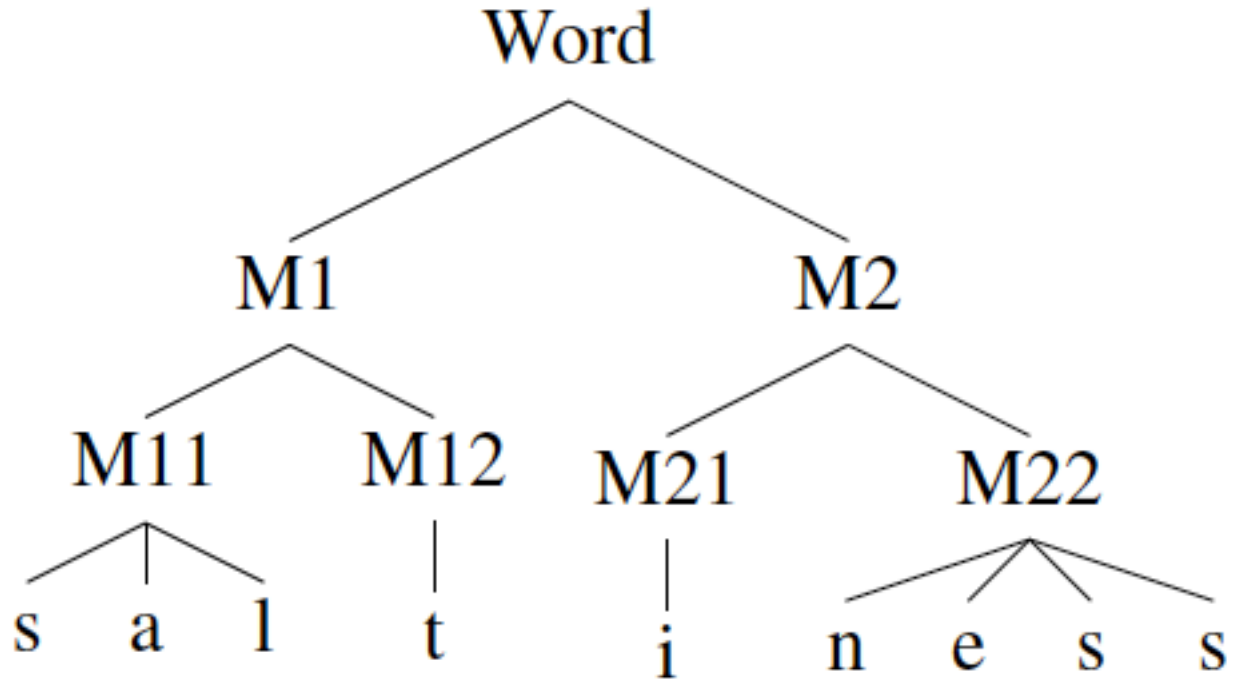
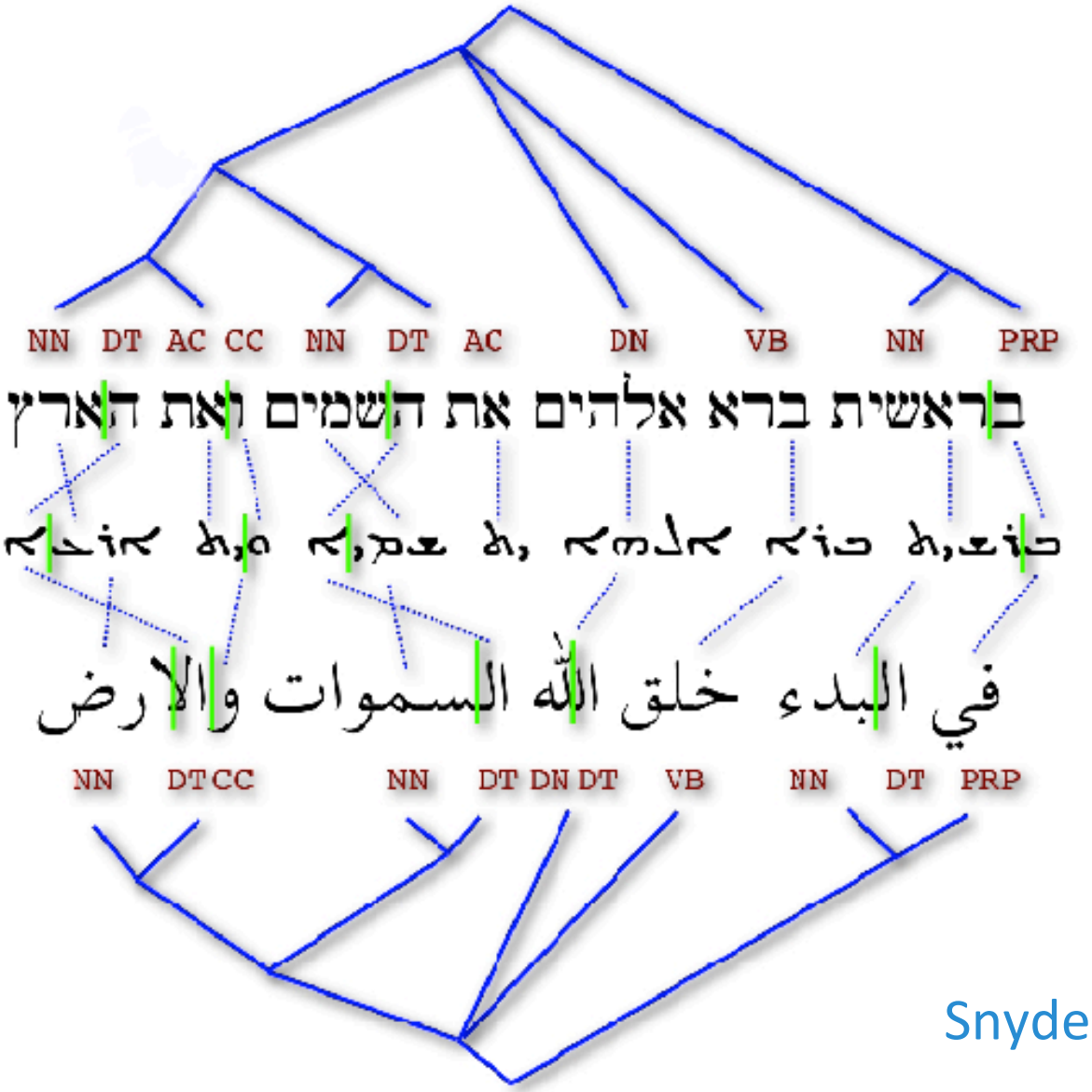


Figure 1: The parse tree generated by the metagrammar of depth 2 for the word *saltiness*.

Sirts & Goldwater (2013)

Morphological Segmentations, POS, and Syntactic Trees



Snyder & Barzilay

Generative Stories

- we hypothesize latent variables through which data are generated
- define “generative story” that describes how latent variables are generated, then how data is generated using latent variables
- we parameterize the distributions & add parameter generation to generative story

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 $p(X \mid z, \theta)$

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

Inference

- in general, inference roughly means “calculate statistical quantities of interest”
- examples:
  - compute the mode of some random variables when conditioning on some and marginalizing out others
  - compute marginals of some random variables (variable posteriors when marginalizing out everything else)
  - compute posterior distribution over some subset of random variables

Learning?

- in Bayesian NLP, there's often no "learning"
- there is only "inference"
- just define model and do inference to calculate what we want to calculate
  - no parameters are being estimated from data*
  - we are not optimizing any loss function*
  - there is no gradient descent*

* typically

Generative Story Template

1: Draw a set of parameters θ from $p(\theta \mid \alpha)$

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

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

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

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

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

$$= p(\theta \mid \alpha) \left(\prod_{i=1}^{n} p(z^{(i)} \mid \theta)\, p(x^{(i)} \mid z^{(i)}, \theta) \right)$$

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: $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$

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

$$= p(\theta \mid \alpha) \left(\prod_{i=1}^{n} p(z^{(i)} \mid \theta)\, p(x^{(i)} \mid z^{(i)}, \theta) \right)$$

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

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

Markov Chain Monte Carlo (MCMC)

- MCMC algorithms are widely used in Bayesian modeling but also useful more generally
- can be used to generate samples from distributions that are hard to sample from
- samples can be used to estimate quantities of interest
- these estimates are unbiased

Gibbs Sampling

- Gibbs sampling is the simplest and most widely-used MCMC algorithm (at least in NLP)

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 $U_i \leftarrow u$

Gibbs Sampling Template

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 $U_i \leftarrow u$

At convergence, each time we update any value of any random variable in $U_1, \ldots, U_p$, we have another sample from the posterior

these samples can be used to estimate any quantity of interest while offering some nice theoretical properties

Disadvantages of Gibbs Sampling?

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 $U_i \leftarrow u$

nearby samples are not necessarily uncorrelated, so it can take many samples for good estimates, especially of rare events

guarantees are at convergence, "burn-in" time can be hard to estimate & depends on initialization

LDA Inference Cheat Sheet

1: For each topic, draw a multinomial word distribution $\beta_k \sim$ Dirichlet(ψ)

2: For each document

a: Draw a multinomial topic distribution $\theta \sim$ Dirichlet(α)

b: For each position in document

i: Draw a topic $z \sim$ Multinomial(θ)

ii: Draw a word $w \sim$ Multinomial(β_z)

$a \sim$ Multinomial(θ)

$p(A = a_i \mid \theta) = \theta_i$

$\theta \sim$ Dirichlet(α)

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

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 $U_i \leftarrow u$

$K =$ # topics

$N =$ # documents

$M =$ # words in each document

$V =$ # words in vocabulary