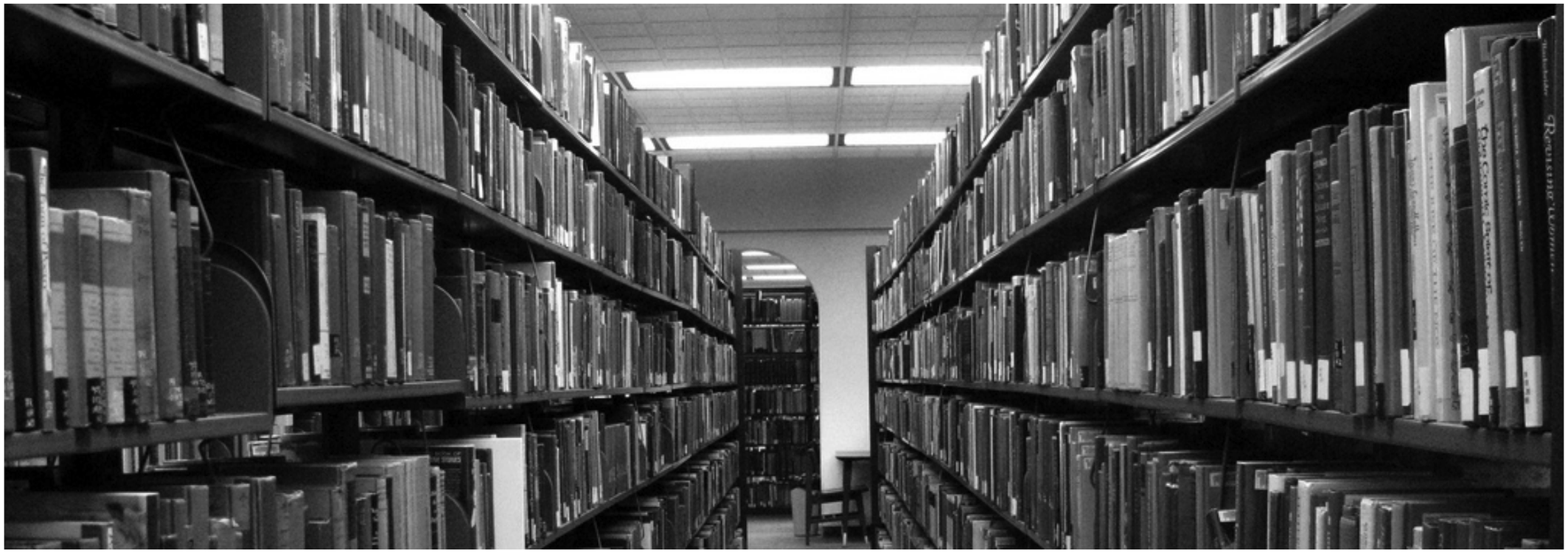
Topic Models

Some slides are adapted from Prof. David Blei's tutorial.

Topic Modelling



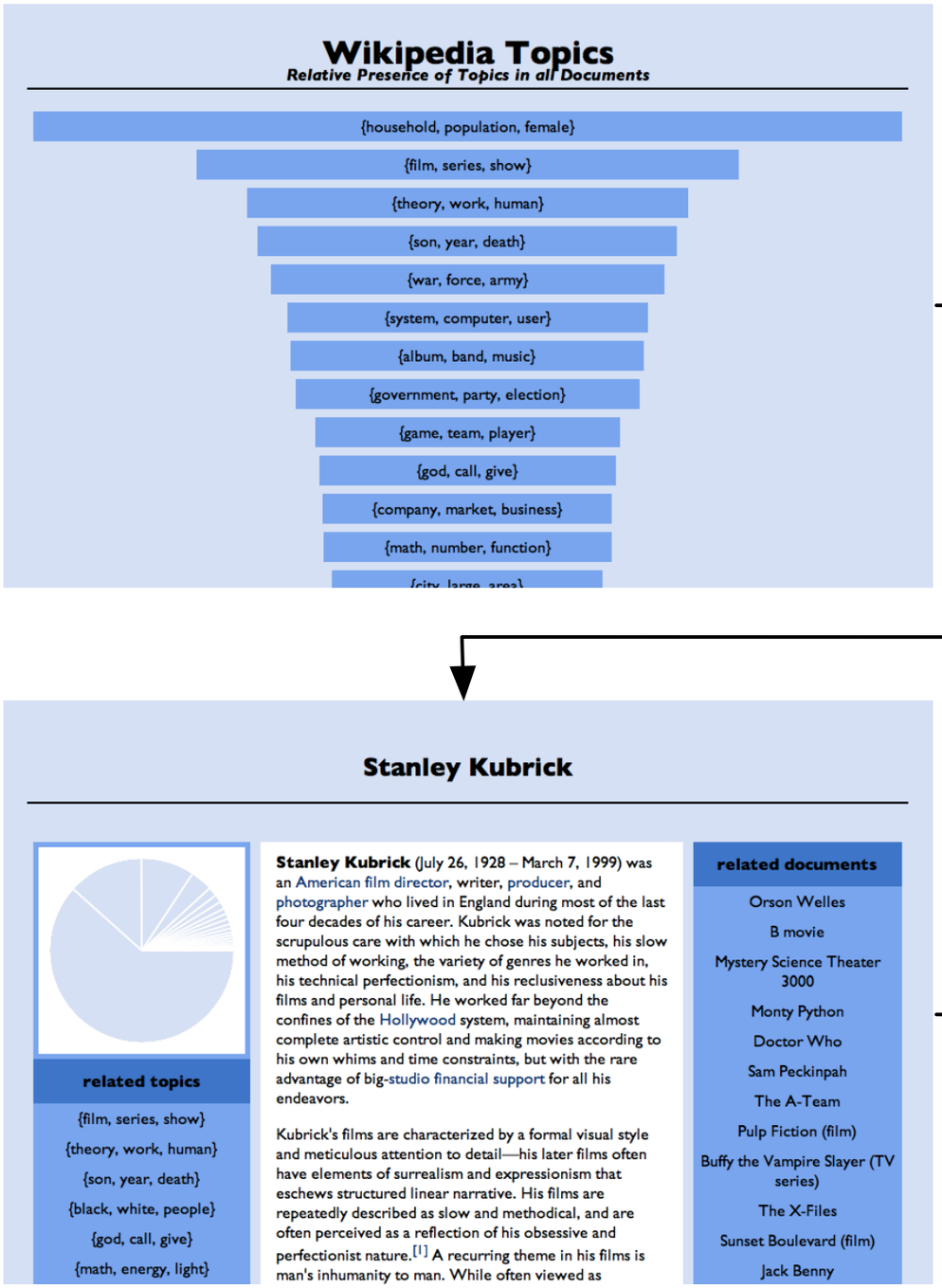
Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.

1. Discover the hidden themes that pervade the collection.
2. Annotate the documents according to those themes.
3. Use annotations to organize, summarize, and search the texts.

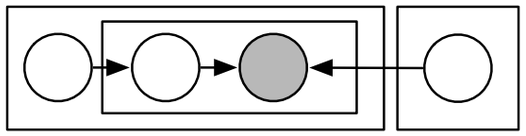
Discover topics from a corpus

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

Organize and browse large corpora

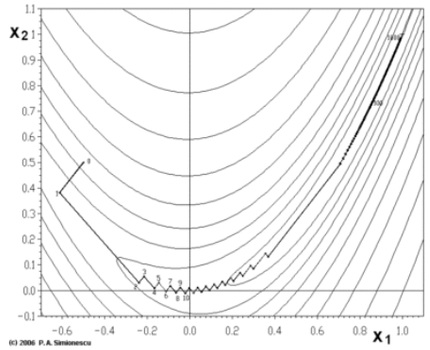


Assumptions

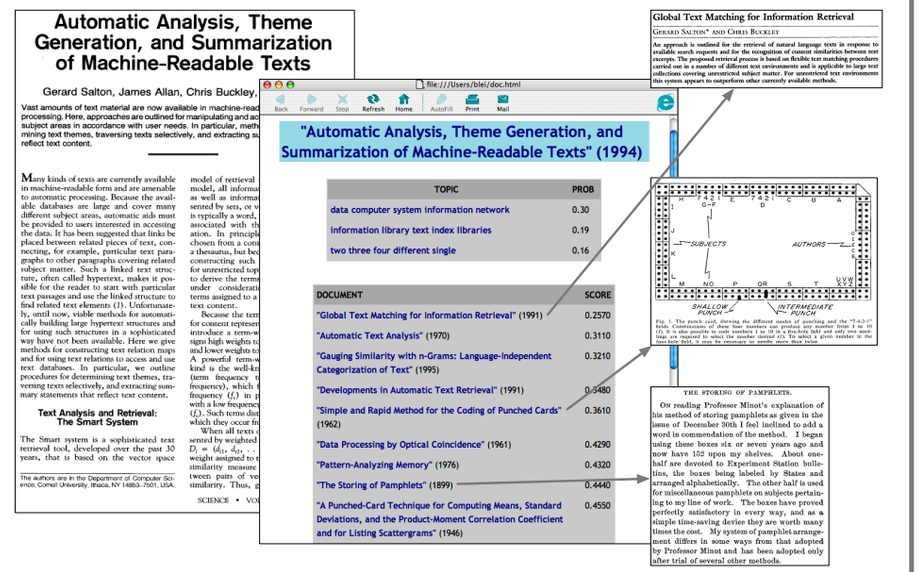


Data

Inference algorithm



Discovered structure



Probabilistic Modeling

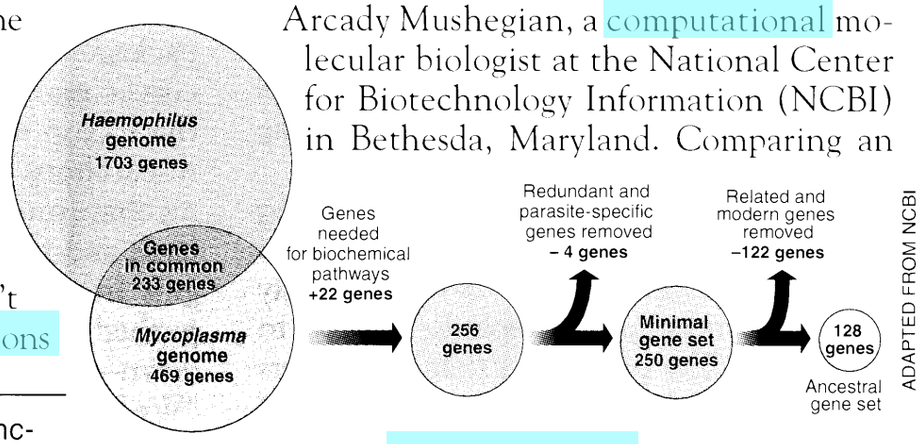
1. Data are assumed to be observed from a generative probabilistic process that includes hidden variables.
   - *In text, the hidden variables are the thematic structure.*
2. Infer the hidden structure using posterior inference
   - *What are the topics that describe this collection?*
3. Situate new data into the estimated model.
   - *How does a new document fit into the topic structure?*

Latent Dirichlet Allocation (LDA)

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

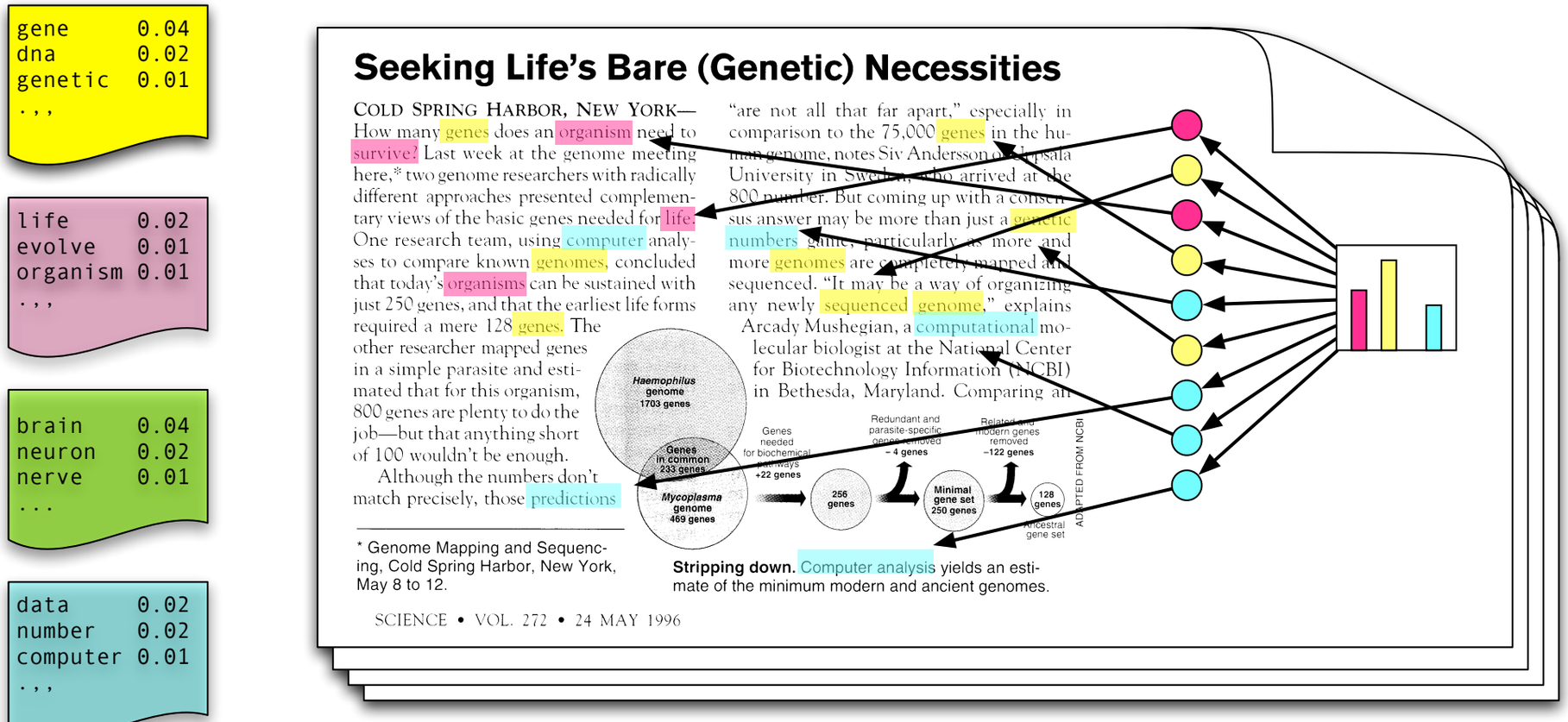
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Simple intuition: Documents exhibit multiple topics.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet allocation." *Journal of Machine Learning Research* 3.Jan (2003): 993-1022.

Generative model for LDA

Topics | *Documents* | *Topic proportions and assignments*



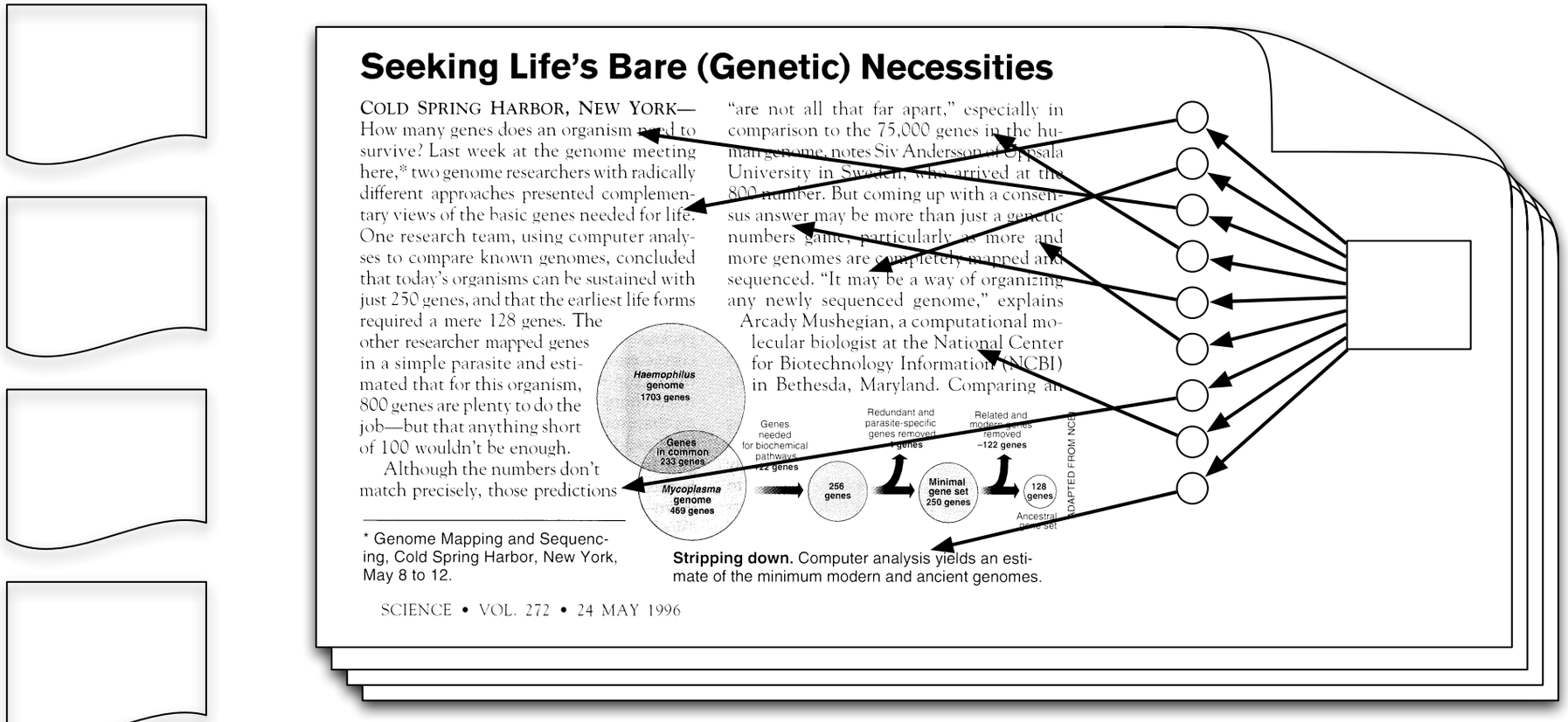
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

The posterior

Topics

Documents

Topic proportions and assignments



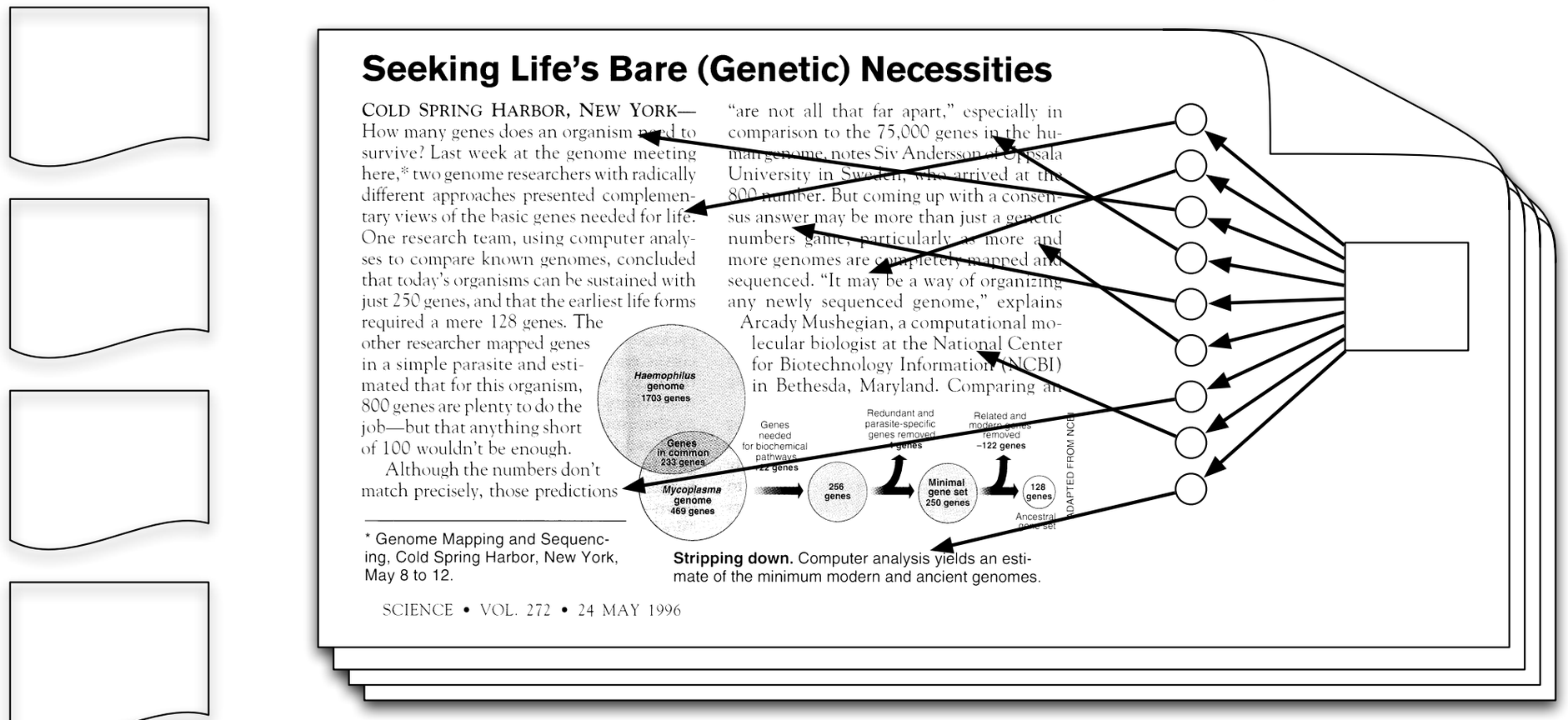
- In reality, we only observe the documents
- The other structure are **hidden variables**

The posterior

Topics

Documents

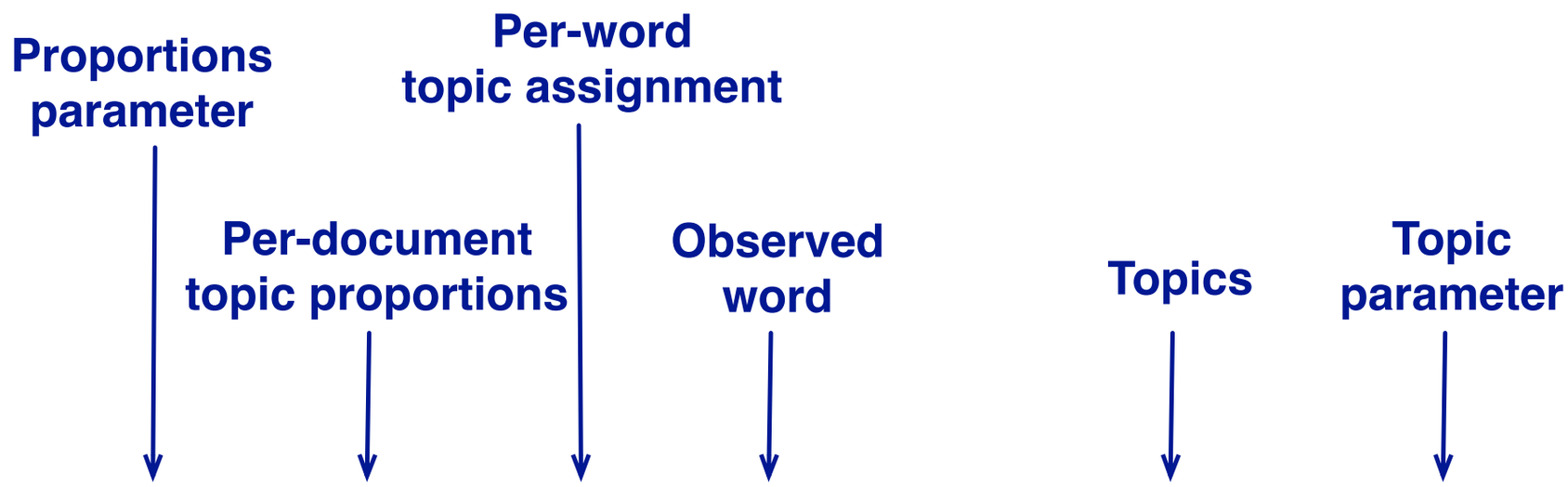
Topic proportions and assignments

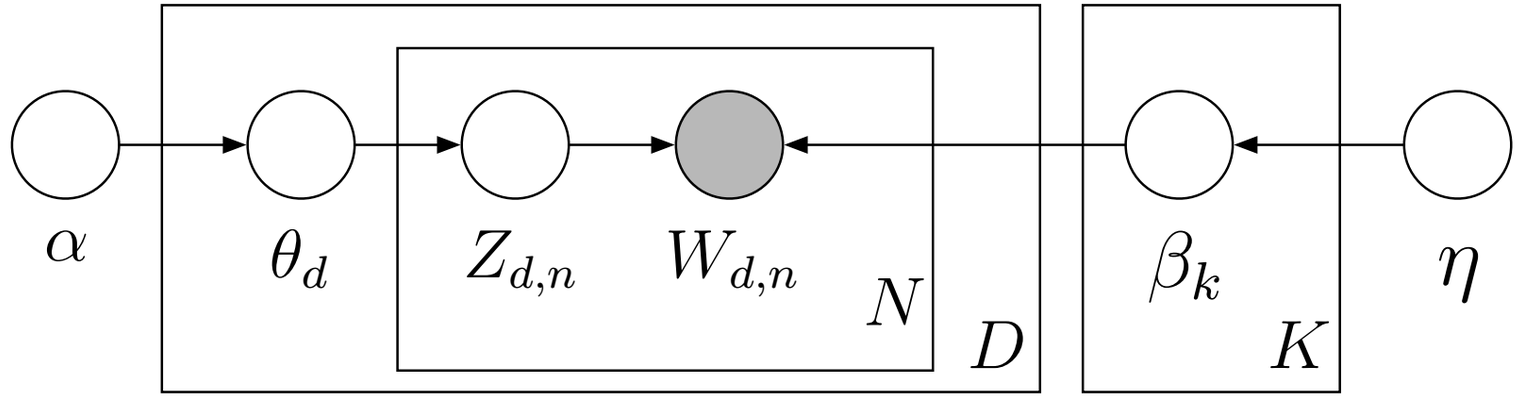


- Our goal is to **infer** the hidden variables
- I.e., compute their distribution conditioned on the documents

$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

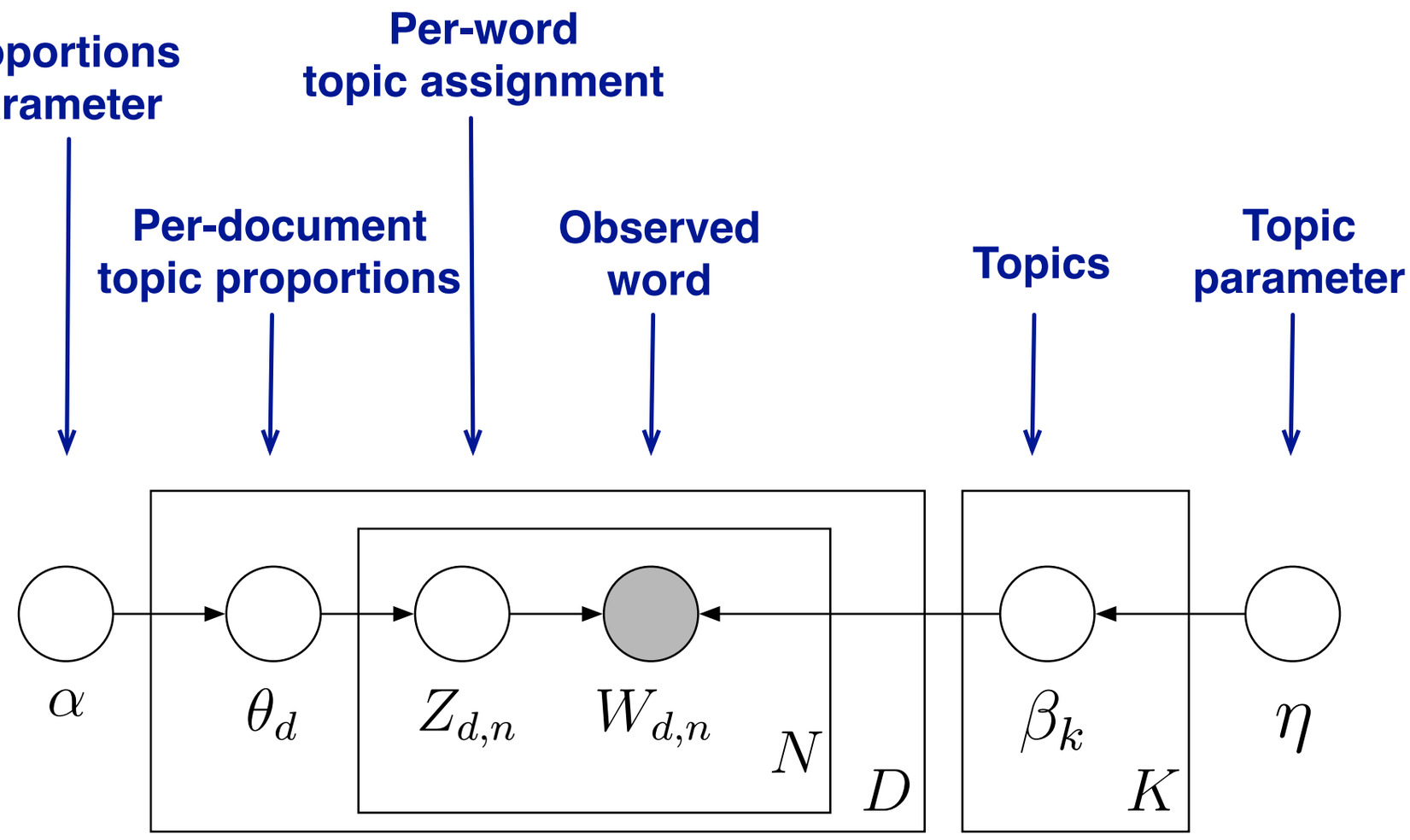
LDA as a graphical model





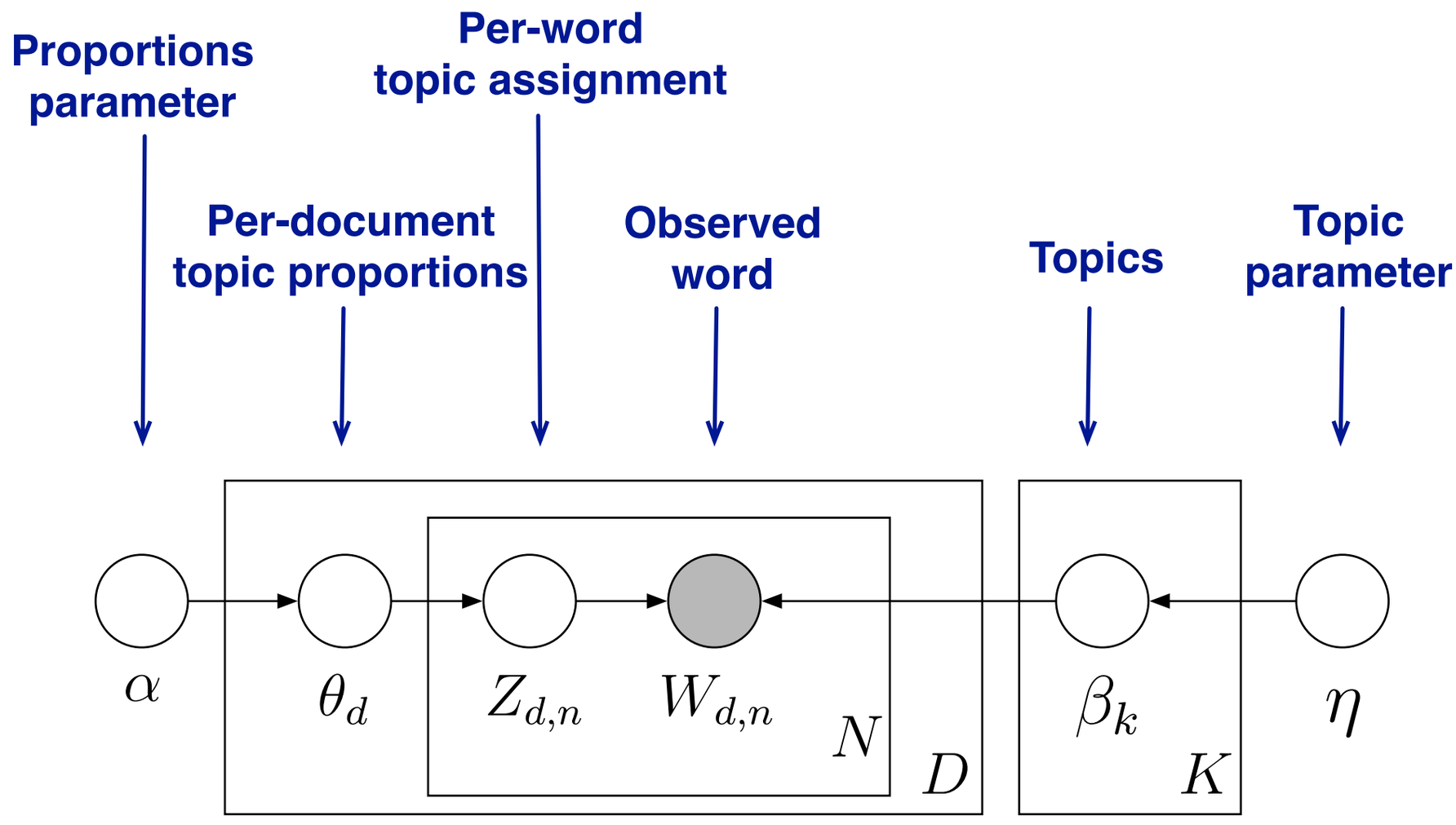
- Encodes our assumptions about the data
- Connects to algorithms for computing with data
- See *Pattern Recognition and Machine Learning* (Bishop, 2006).

LDA as a graphical model



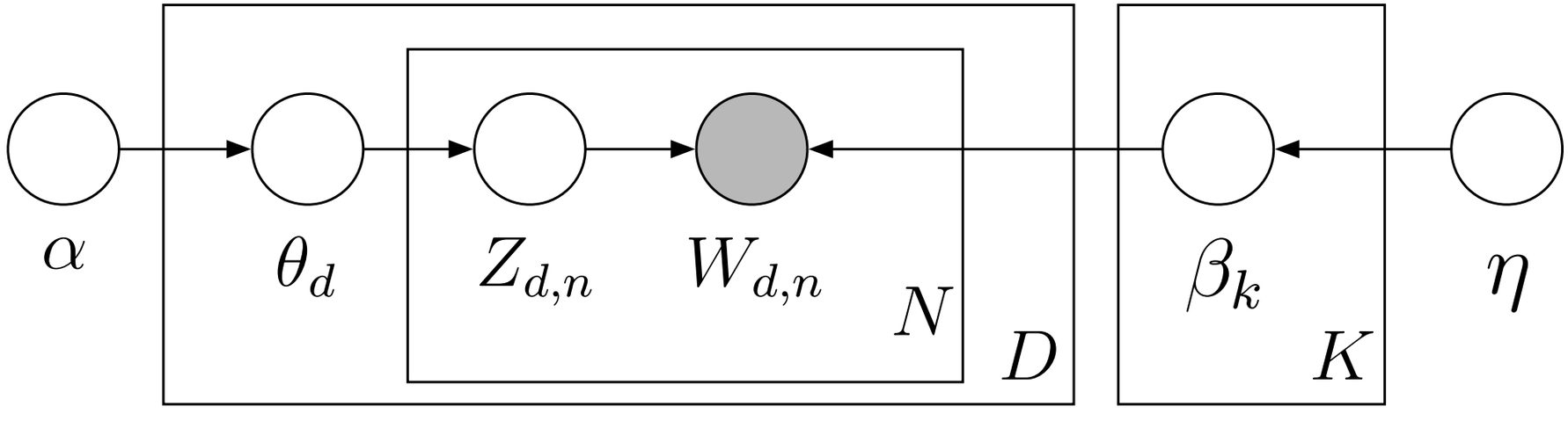
- Nodes are random variables; edges indicate dependence.
- Shaded nodes are observed.
- Plates indicate replicated variables.

LDA as a graphical model



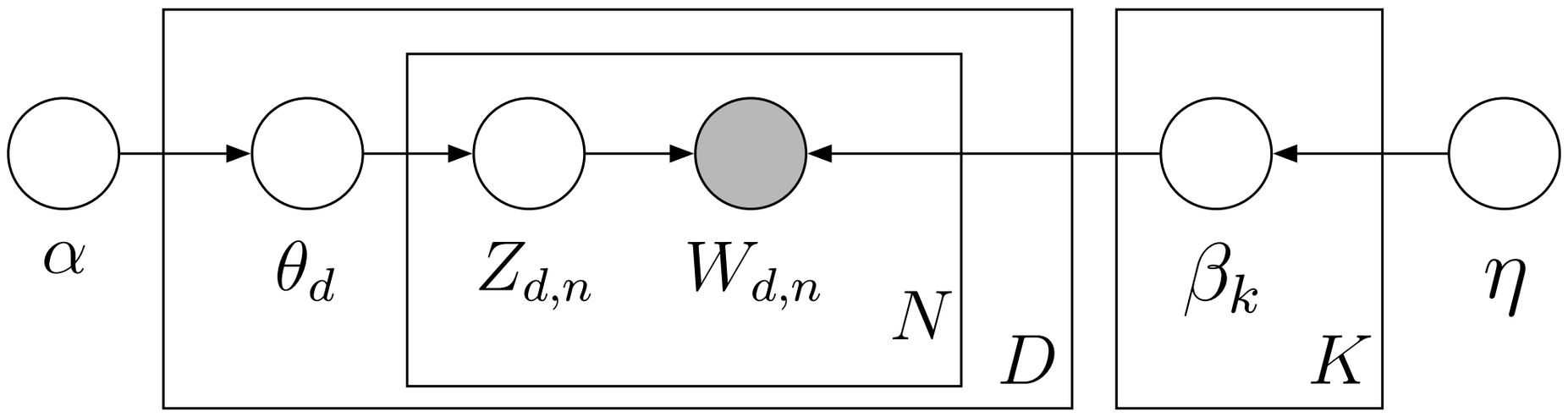
$$\prod_{i=1}^{K} p(\beta_i \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right)$$

LDA



- This joint defines a posterior.
- From a collection of documents, infer
  - Per-word topic assignment $z_{d,n}$
  - Per-document topic proportions θ_d
  - Per-corpus topic distributions β_k
- Then use posterior expectations to perform the task at hand, e.g., information retrieval, document similarity, exploration, ...

LDA

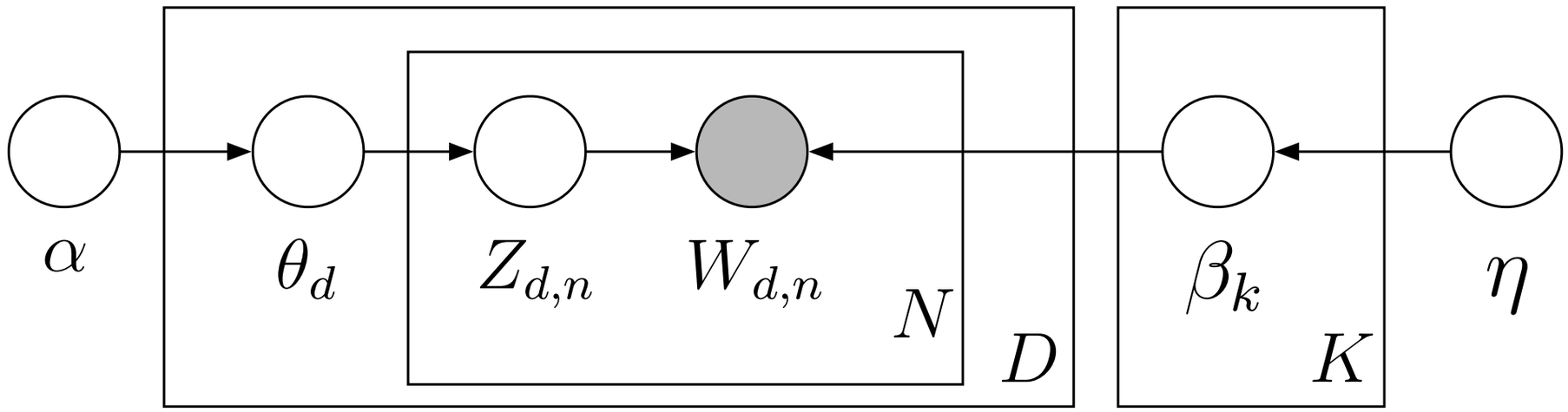


Approximate posterior inference algorithms

- Mean field variational methods (Blei et al., 2001, 2003)
- Expectation propagation (Minka and Lafferty, 2002)
- Collapsed Gibbs sampling (Griffiths and Steyvers, 2002)
- Collapsed variational inference (Teh et al., 2006)
- Online variational inference (Hoffman et al., 2010)

Also see Mukherjee and Blei (2009) and Asuncion et al. (2009).

Example inference

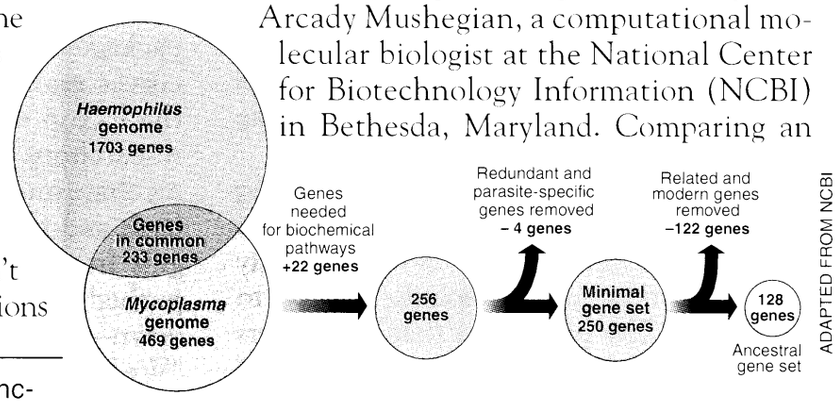


- **Data**: The OCR'ed collection of *Science* from 1990–2000
  - 17K documents
  - 11M words
  - 20K unique terms (stop words and rare words removed)
- **Model**: 100-topic LDA model using variational inference.

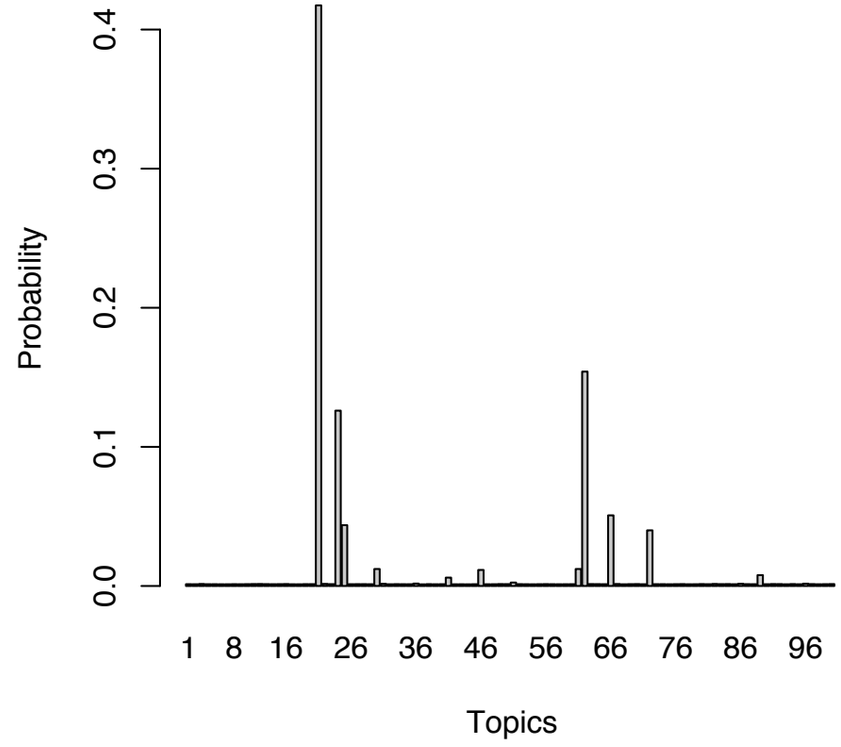
Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



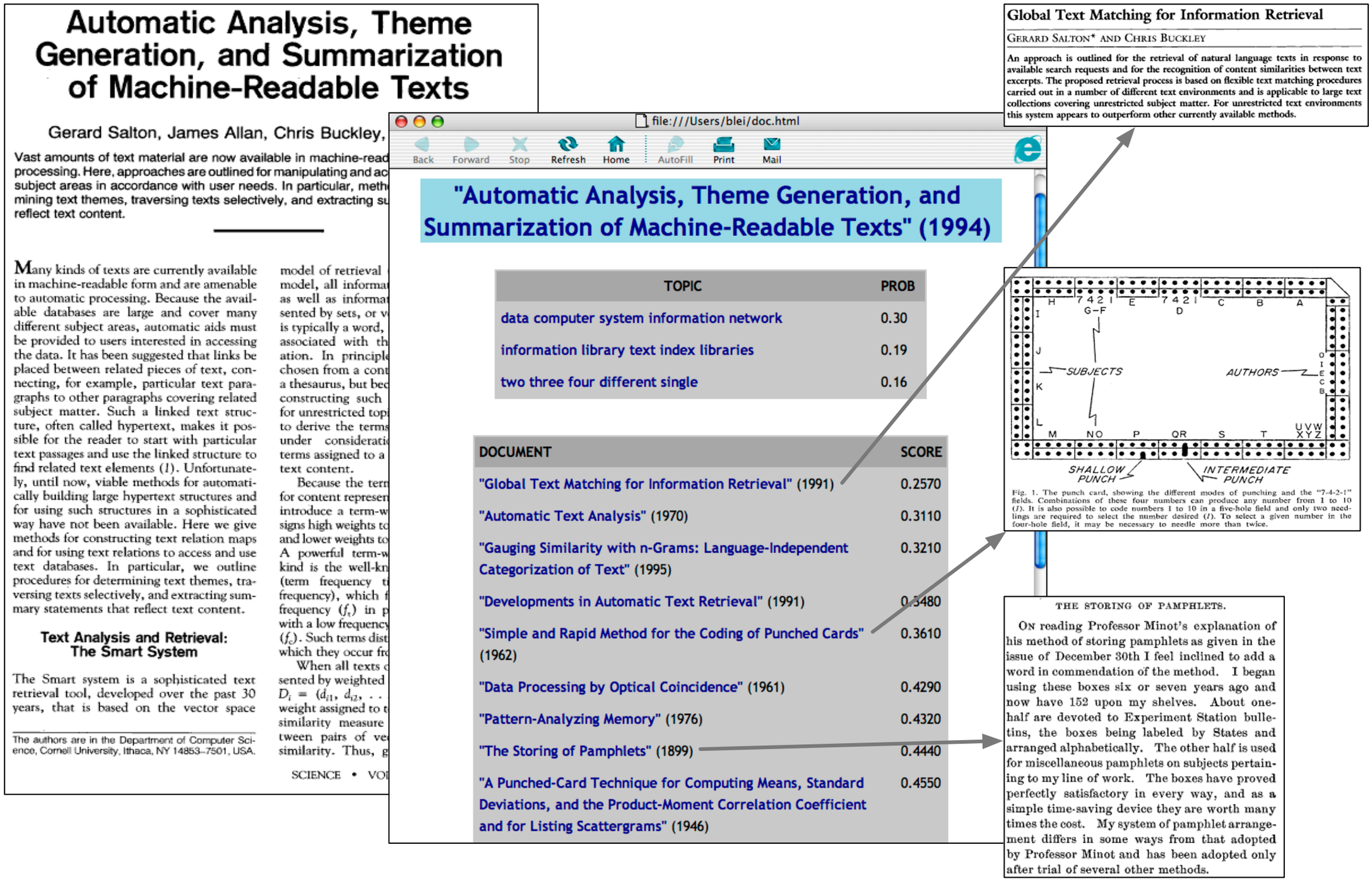
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.



* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

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

Used to explore and browse document collections



The Dirichlet distribution

- The Dirichlet distribution is an exponential family distribution over the simplex, i.e., positive vectors that sum to one

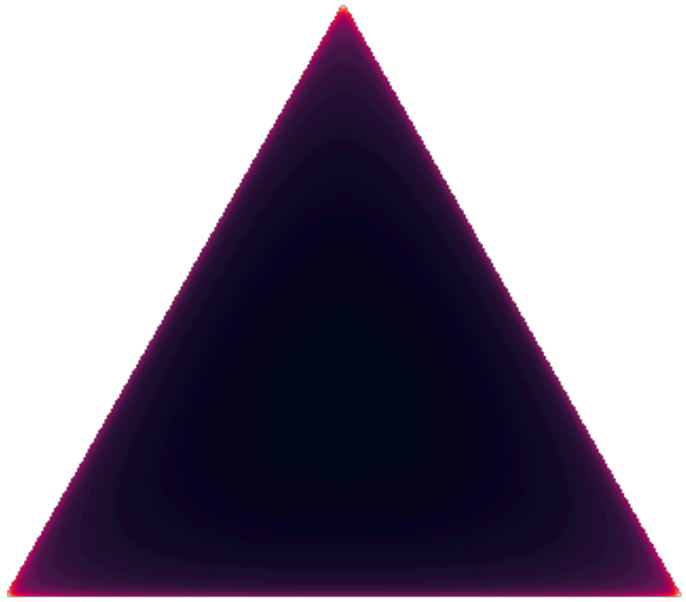
$$p(\theta \mid \vec{\alpha}) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}.$$

- It is **conjugate** to the multinomial. Given a multinomial observation, the posterior distribution of θ is a Dirichlet.

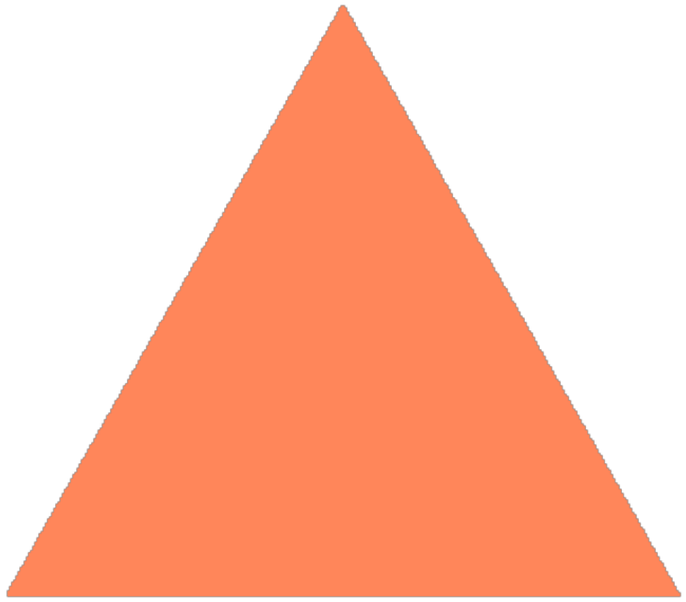
- The parameter α controls the mean shape and sparsity of θ.

- The topic proportions are a K dimensional Dirichlet.
  The topics are a V dimensional Dirichlet.

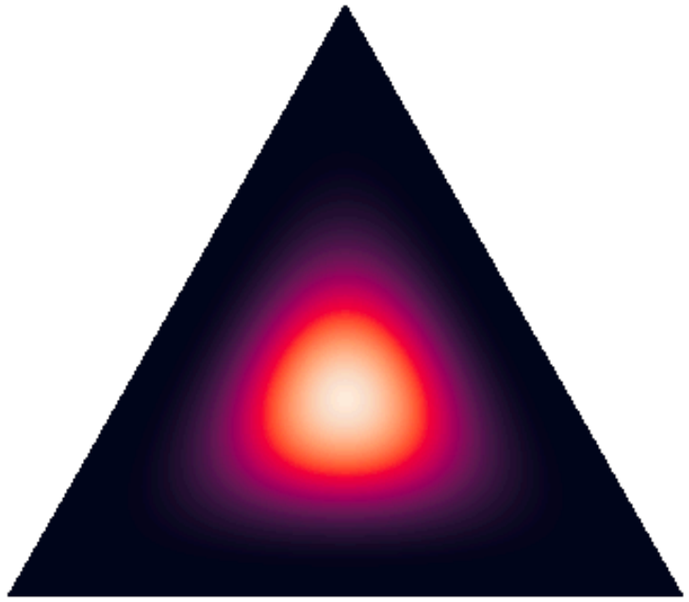
(0.85, 0.85, 0.85)



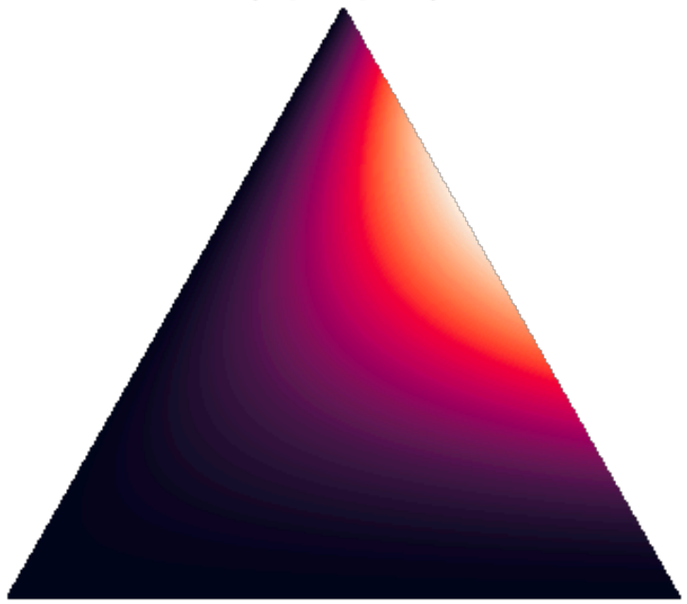
(1, 1, 1)



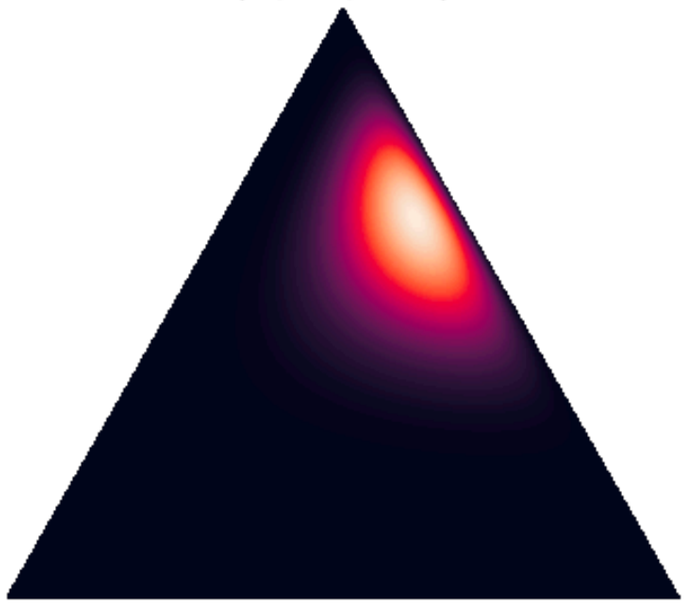
(5, 5, 5)



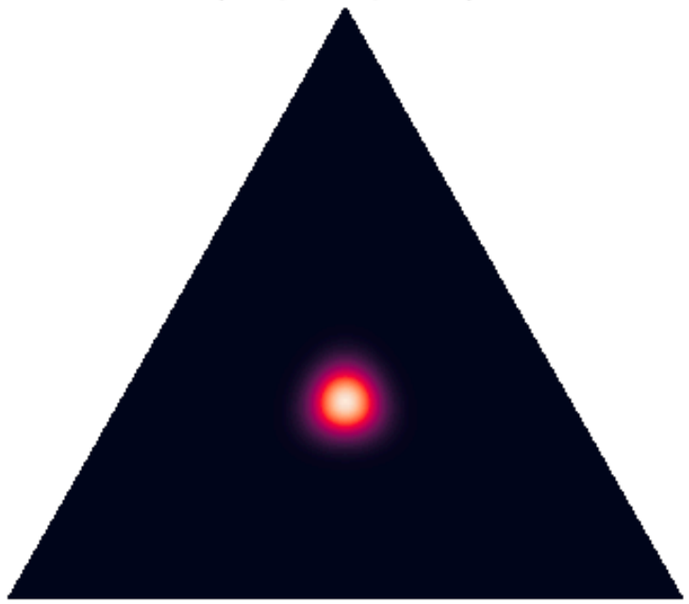
(1, 2, 3)



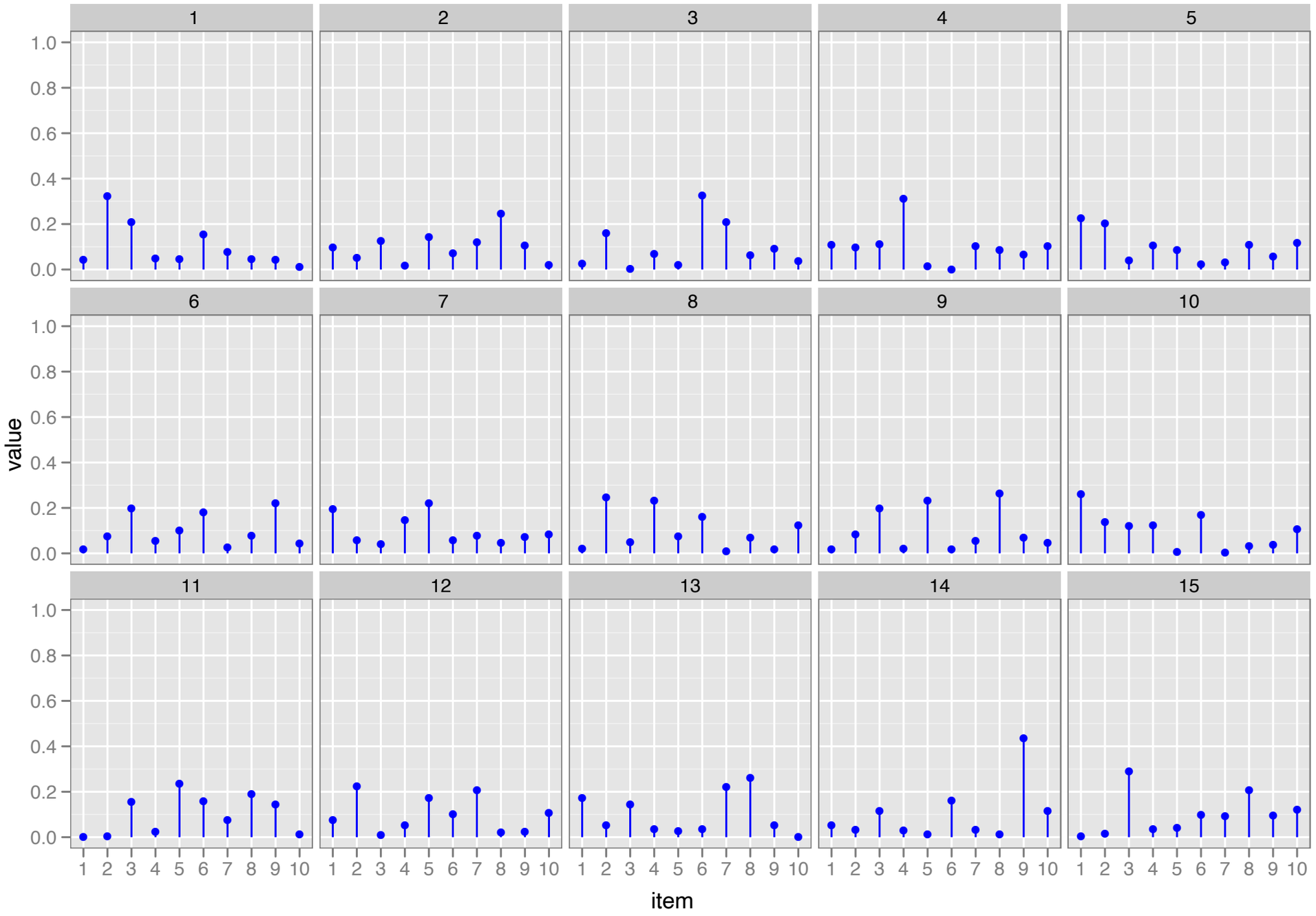
(2, 5, 10)



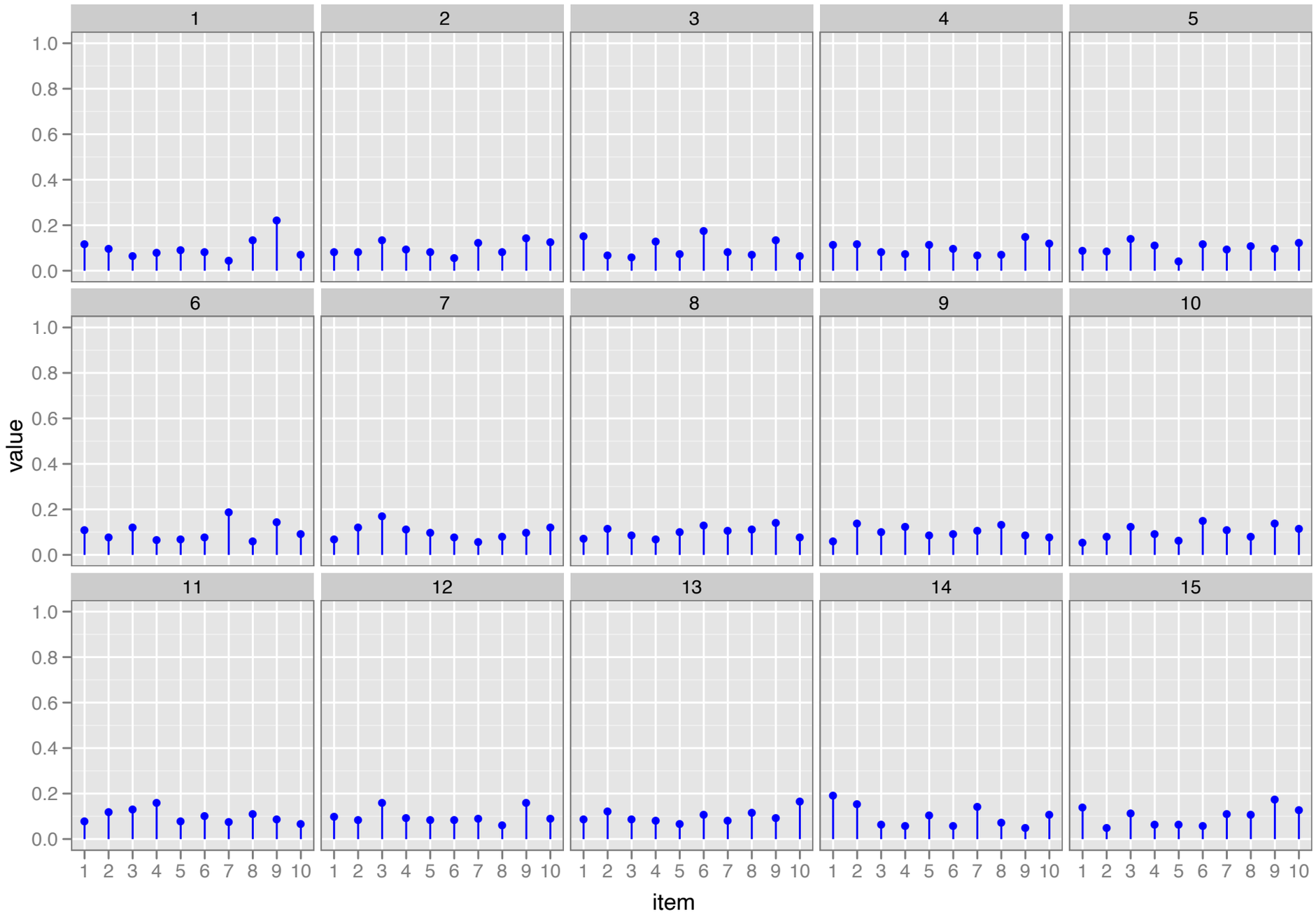
(50, 50, 50)



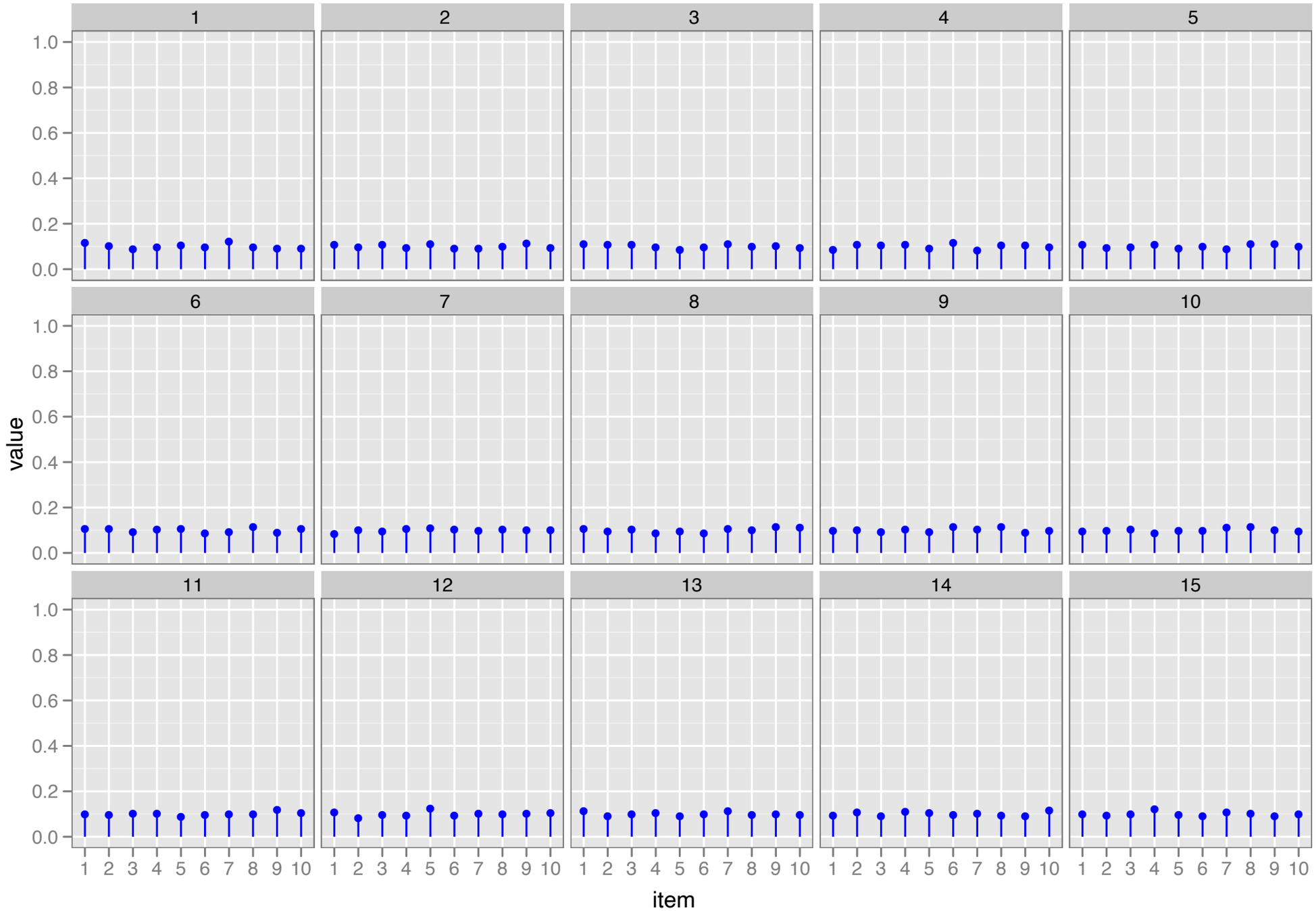
$\alpha = 1$


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
value
item
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10

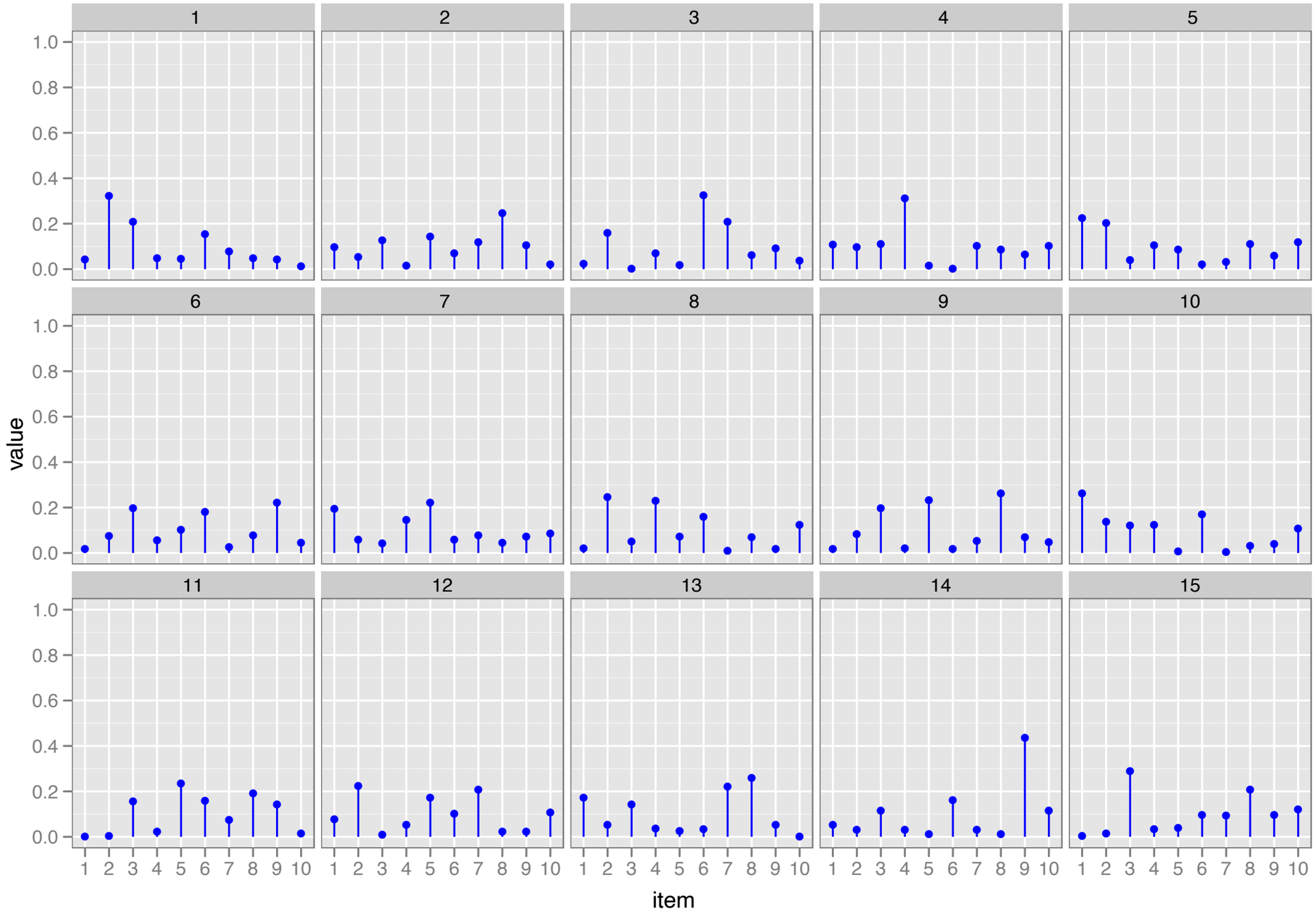
$\alpha = 10$


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
value
item
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10

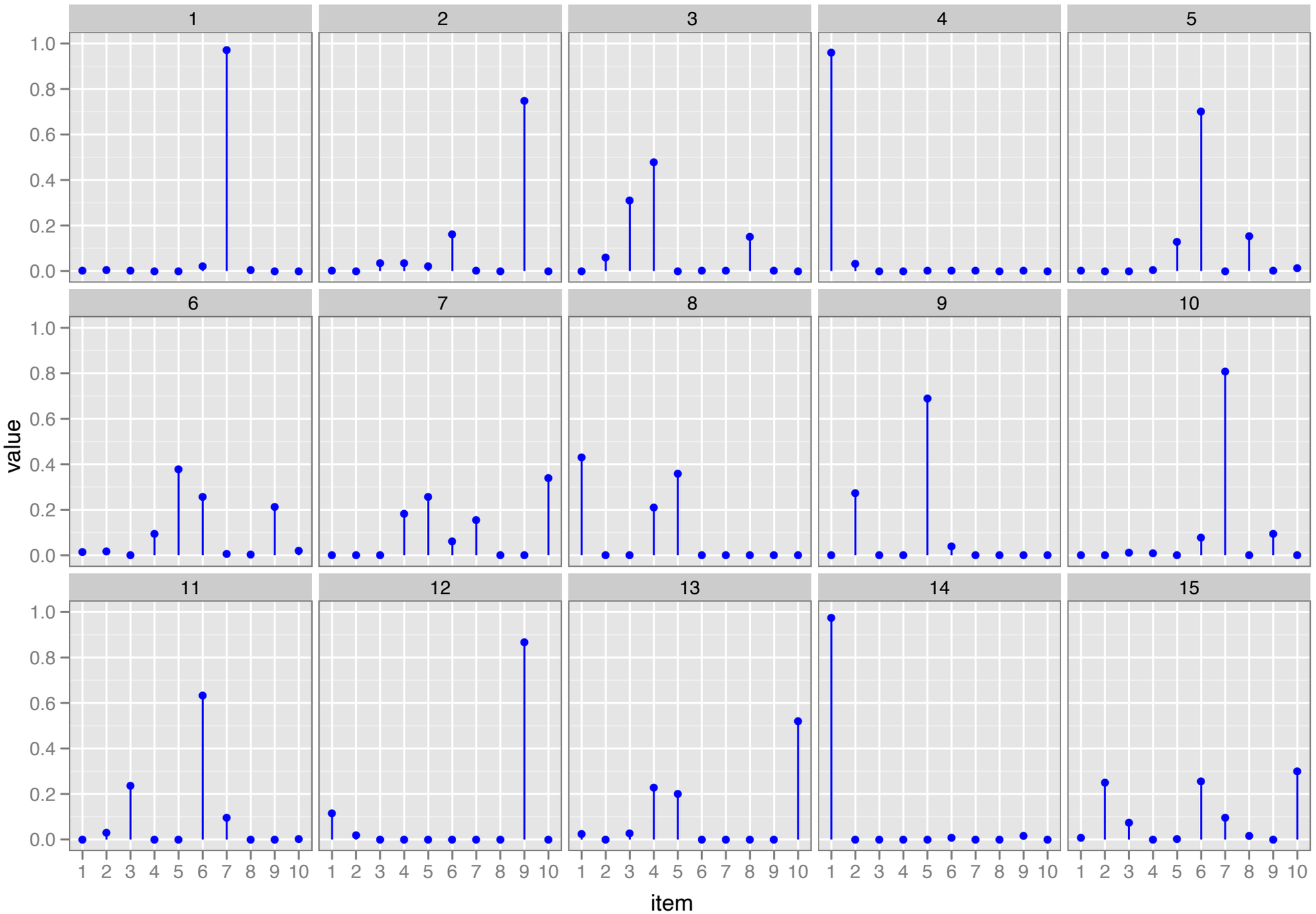
$\alpha = 100$



$\alpha = 1$


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
value
item
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10

$\alpha = 0.1$

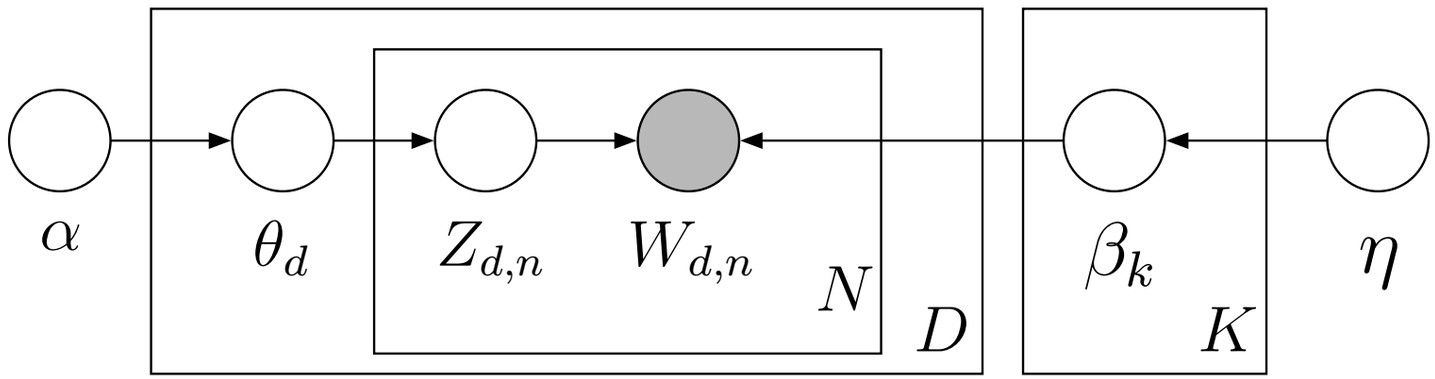

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
value
item
0.0
0.2
0.4
0.6
0.8
1.0
1 2 3 4 5 6 7 8 9 10

Why LDA works?

Why does the LDA posterior put “topical” words together?

- Word probabilities are maximized by dividing the words among the topics. (More terms means more mass to be spread around.)
- In a mixture, this is enough to find clusters of co-occurring words.
- In LDA, the Dirichlet on the topic proportions can encourage sparsity, i.e., a document is penalized for using many topics.
- Loosely, this can be thought of as softening the strict definition of “co-occurrence” in a mixture model.
- This flexibility leads to sets of terms that more tightly co-occur.

Posterior inference for LDA

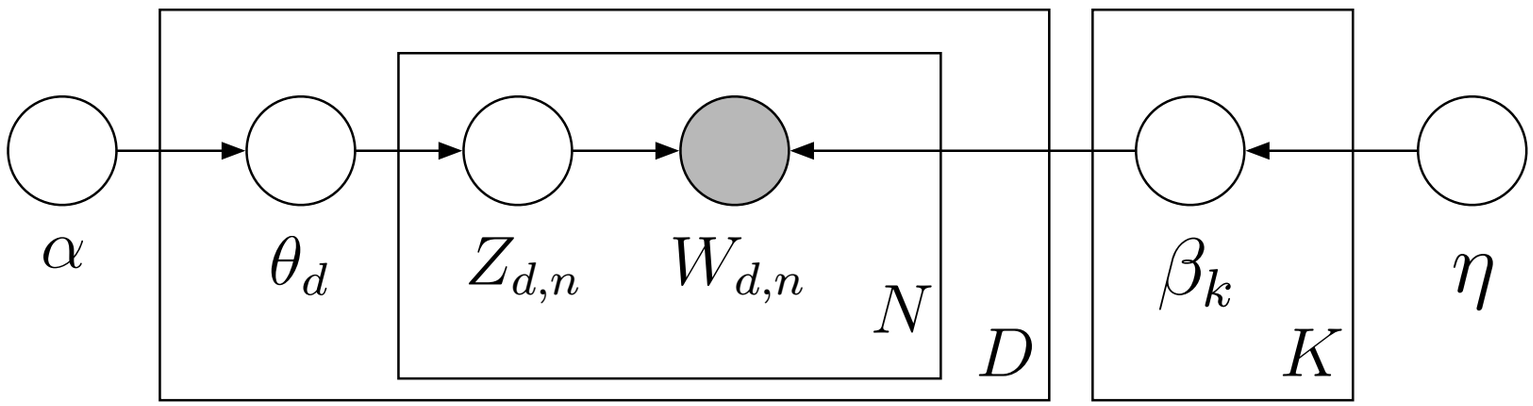


- The joint distribution of the latent variables and documents is

$$\prod_{i=1}^{K} p(\beta_i \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right).$$

- The posterior of the latent variables given the documents is

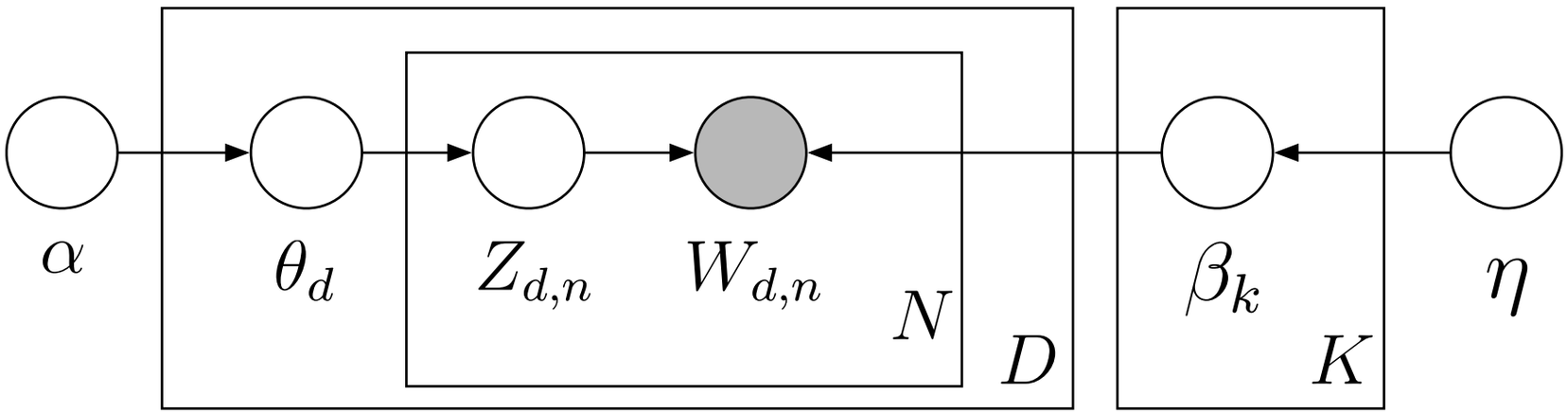
$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D,1:N} \mid w_{1:D,1:N}).$$



- This is equal to

$$\frac{p(\beta_{1:K}, \theta_{1:D}, \mathbf{z}_{1:D}, \mathbf{w}_{1:D})}{\int_{\beta_{1:K}} \int_{\theta_{1:D}} \sum_{\mathbf{z}_{1:D}} p(\beta_{1:K}, \theta_{1:D}, \mathbf{z}_{1:D}, \mathbf{w}_{1:D})}.$$

- We can't compute the denominator, the marginal $p(\mathbf{w}_{1:D})$.
- This is the crux of the inference problem.



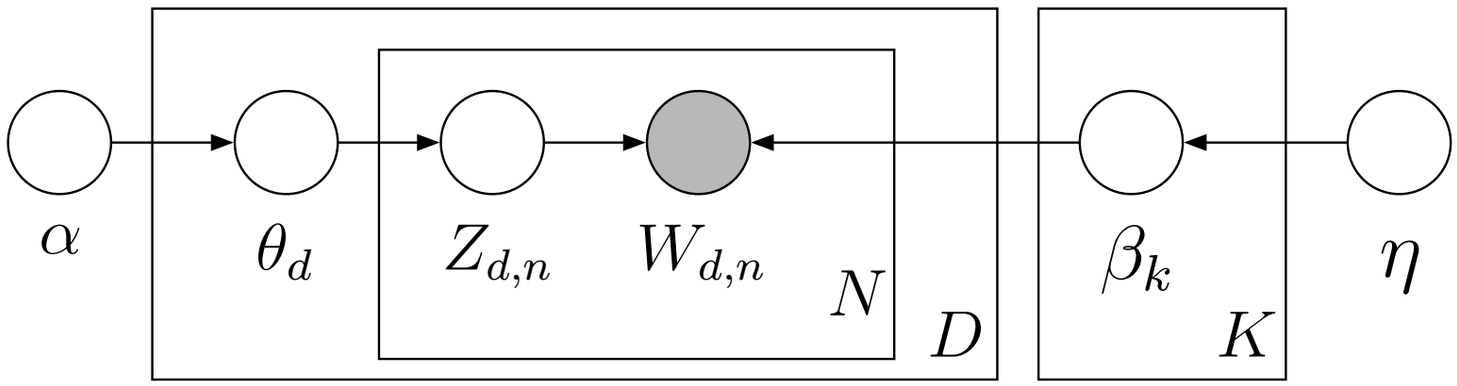
- There is a large literature on approximating the posterior.

- We will focus on
  - Gibbs sampling
  - Mean-field variational methods (batch and online)

MCMC

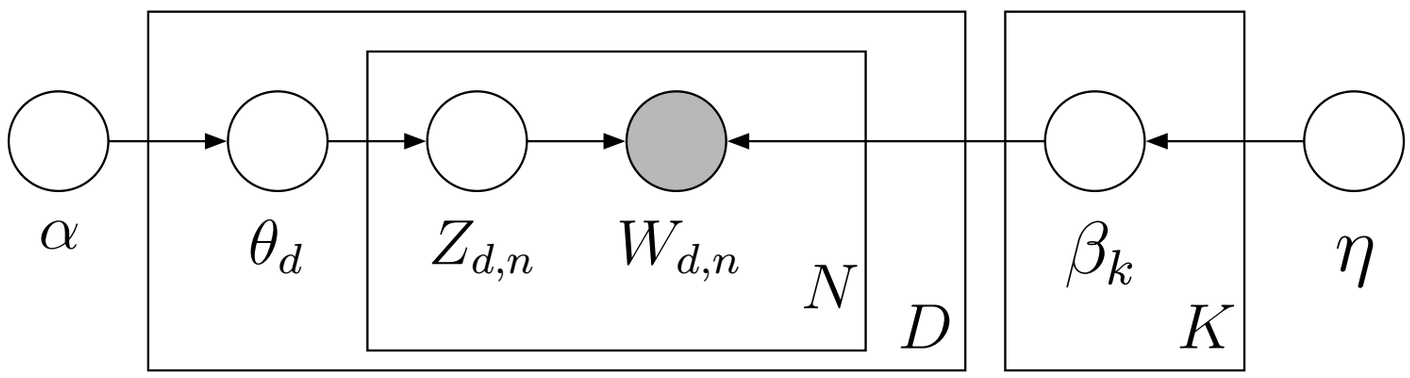
- Construct a **Markov chain** on the hidden variables, whose limiting distribution is the posterior.
- Collect **independent samples** from that distribution; approximate the posterior with them
- In **Gibbs sampling** the chain is defined by the conditional distribution of each hidden variable given observations and the current setting of the other hidden variables.

Local and global variables



- Local variables are local to each document
  - Topic proportions θ_d
  - Topic assignemnts $z_{d,n}$
- Global variables are shared by the corpus
  - Topics β_k

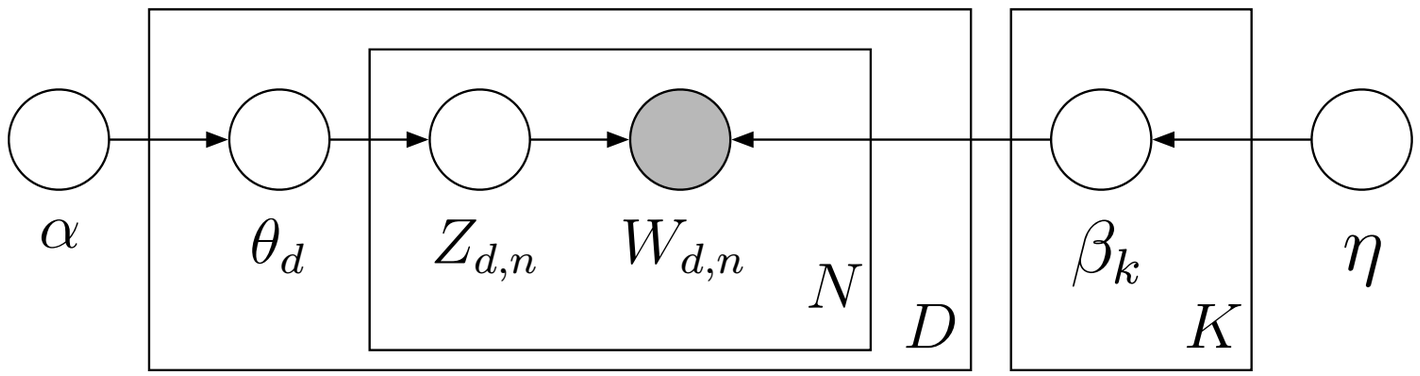
Local and global variables



- Assume the topics are fixed.
- Even “local inference” is intractable,

$$p(\theta, z_{1:N} \mid w_{1:N}, \beta_{1:K}) = \frac{p(\theta)\prod_{n=1}^{N} p(z_n \mid \theta)p(w_n \mid \beta_{z_n})}{\int_\theta p(\theta)\prod_{n=1}^{N}\sum_{z_n} p(z_n \mid \theta)p(w_n \mid \beta_{z_n})}.$$

Local Gibbs sampling for LDA

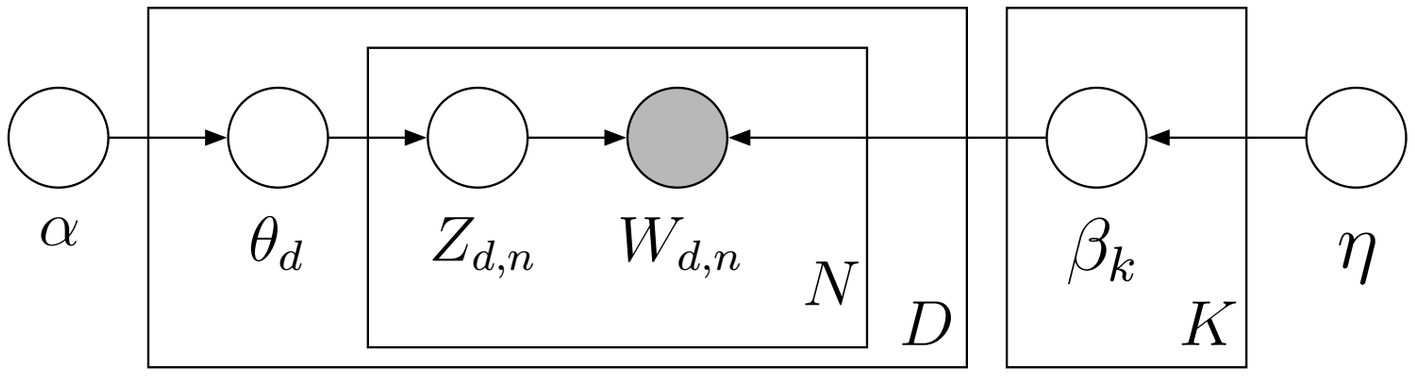


- We observe words $\mathbf{w} = w_{1:N}$. The Markov chain is defined on $\{\theta, z_{1:N}\}$, the topic proportions and topic assignments.
- Some notation—

$$\begin{aligned} n(z_{1:N}) &= \sum_{n=1}^{N} z_n \\ m_k(\mathbf{z}_{1:D}, \mathbf{W}) &= \sum_{d=1}^{D} \sum_{n=1}^{N} z_{d,n}^k w_{d,n}. \end{aligned}$$

- $n(z_{1:N})$ are topic counts;
  $m_k(z_{1:N}, \mathbf{W})$ are within-topic word counts.

Local Gibbs sampling for LDA



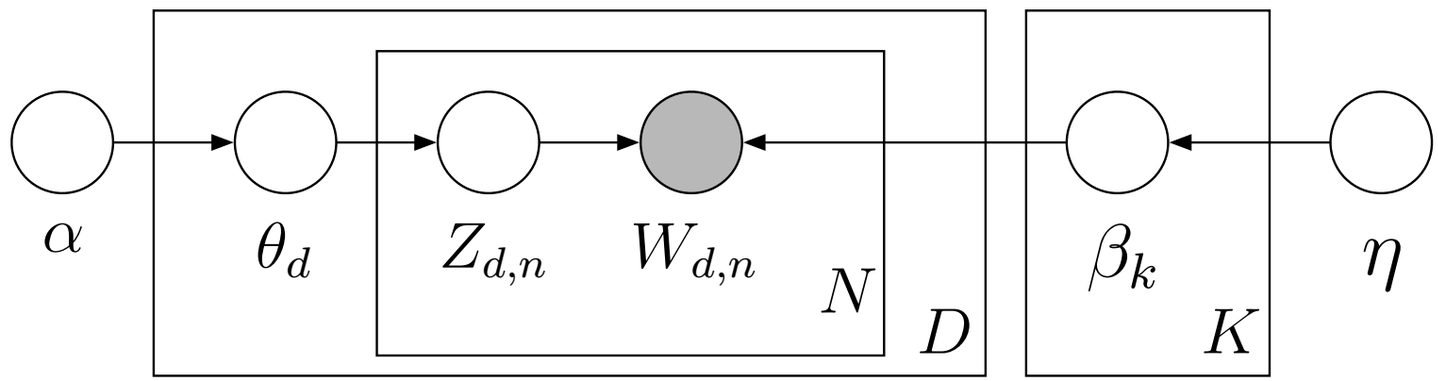
A simple Gibbs sampler is

$$\begin{aligned} \theta \,|\, \mathbf{w}, z_{1:N} &\sim \mathrm{Dir}(\gamma) \\ z_n \,|\, \theta, \mathbf{w} &\sim \mathrm{Mult}(\phi_n) \end{aligned}$$

where

$$\begin{aligned} \gamma &= \alpha + n(z_{1:N}) \\ \phi_n &\propto \theta \cdot p(w_n \,|\, \beta_{1:K}). \end{aligned}$$

Collapsed local Gibbs sampling



- The topic proportions θ can be integrated out,

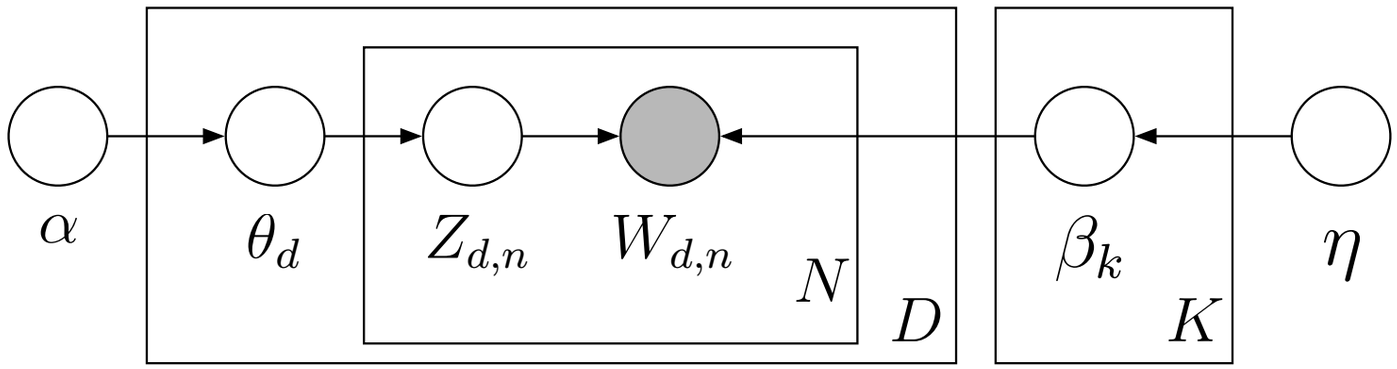
$$p(z_n \mid z_{-n}, \mathbf{w}) = p(w_n \mid \beta_{1:K}) \cdot \int_\theta p(z_n \mid \theta) p(\theta \mid z_{-n}) d\theta$$

- A collapsed Gibbs sampler constructs a chain on $z_{1:N}$,

$$z_n \mid z_{-n}, \mathbf{w} \sim \mathrm{Mult}(\phi_n),$$

where $\phi_n \propto p(w_n \mid \beta_{1:K})(n(z_{-n}) + \alpha)$.

Sampling the topics



- We observe the corpus $\mathbf{W} = \mathbf{w}_{1:D}$.
- We define the chain on $\{\mathbf{z}_{1:D}, \theta_{1:D}, \beta_{1:K}\}$.
- First, sample latent variables $(\mathbf{z}_d, \theta_d)$ for each document.
- Then, sample each topic from

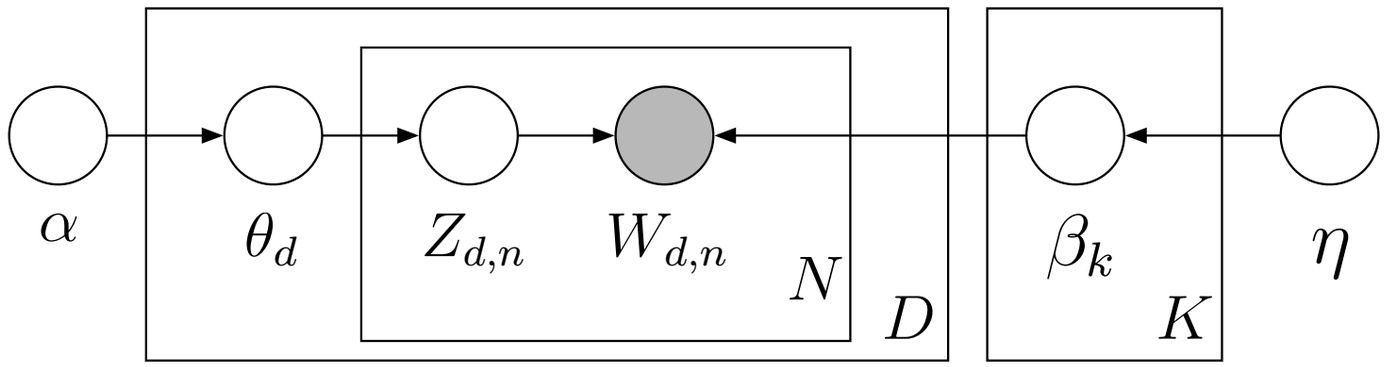
$$\beta_k \,|\, \mathbf{z}_{1:D}, \mathbf{W} \sim \mathrm{Dir}(\lambda_k),$$

where

$$\lambda_k := \eta + m_k(\mathbf{z}_{1:D}, \mathbf{W}).$$

Recall $m_k(\mathbf{z}_{1:D}, \mathbf{W})$ are words counts for topic k.

Collapsed Gibbs sampling with topics



- We can integrate out the topics $\beta_{1:K}$ too.
- The sampler is defined on the topic assigments $\mathbf{z}_{1:D}$,

$$p(z_{n,d} = k \,|\, \mathbf{z}_{-(n,d)}, \mathbf{W}) \propto \left(\frac{m_k(\mathbf{z}_{-(n,d)}, \mathbf{W}) + \eta}{\sum_v m_k^v(\mathbf{z}_{-(n,d)}) + V\eta} \right) (n_k(z_{-i}) + \alpha)$$

- This is an excellent Gibbs sampler for LDA. It was developed by Giffiths and Steyvers (2002) and is widely used.

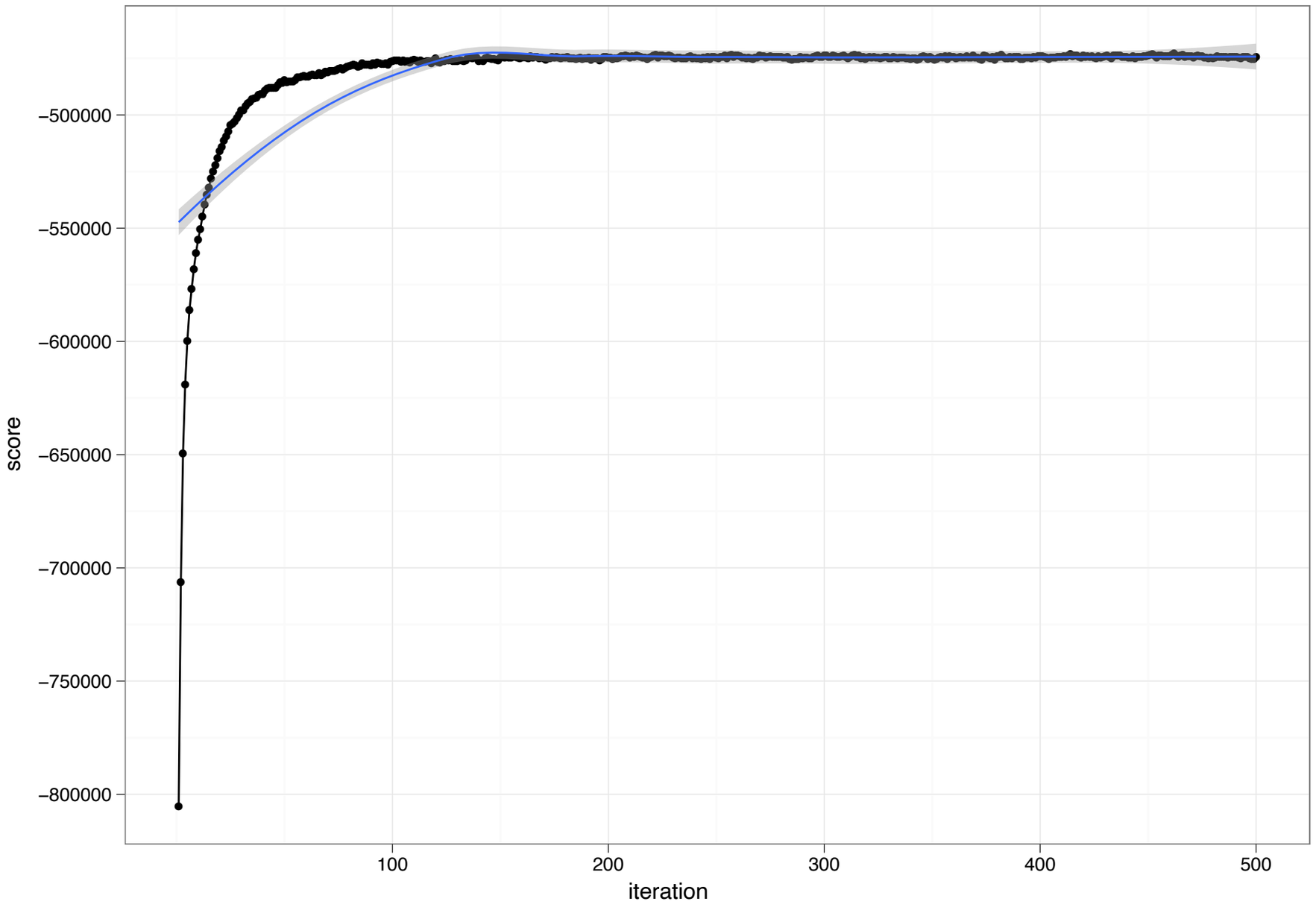
Gibbs sampling for LDA in practice

- In practice:
  1. Obtain a corpus of documents **W**
  2. Run the Gibbs sampler for some number of iterations.
  3. Store states at some lag, or store the MAP state.

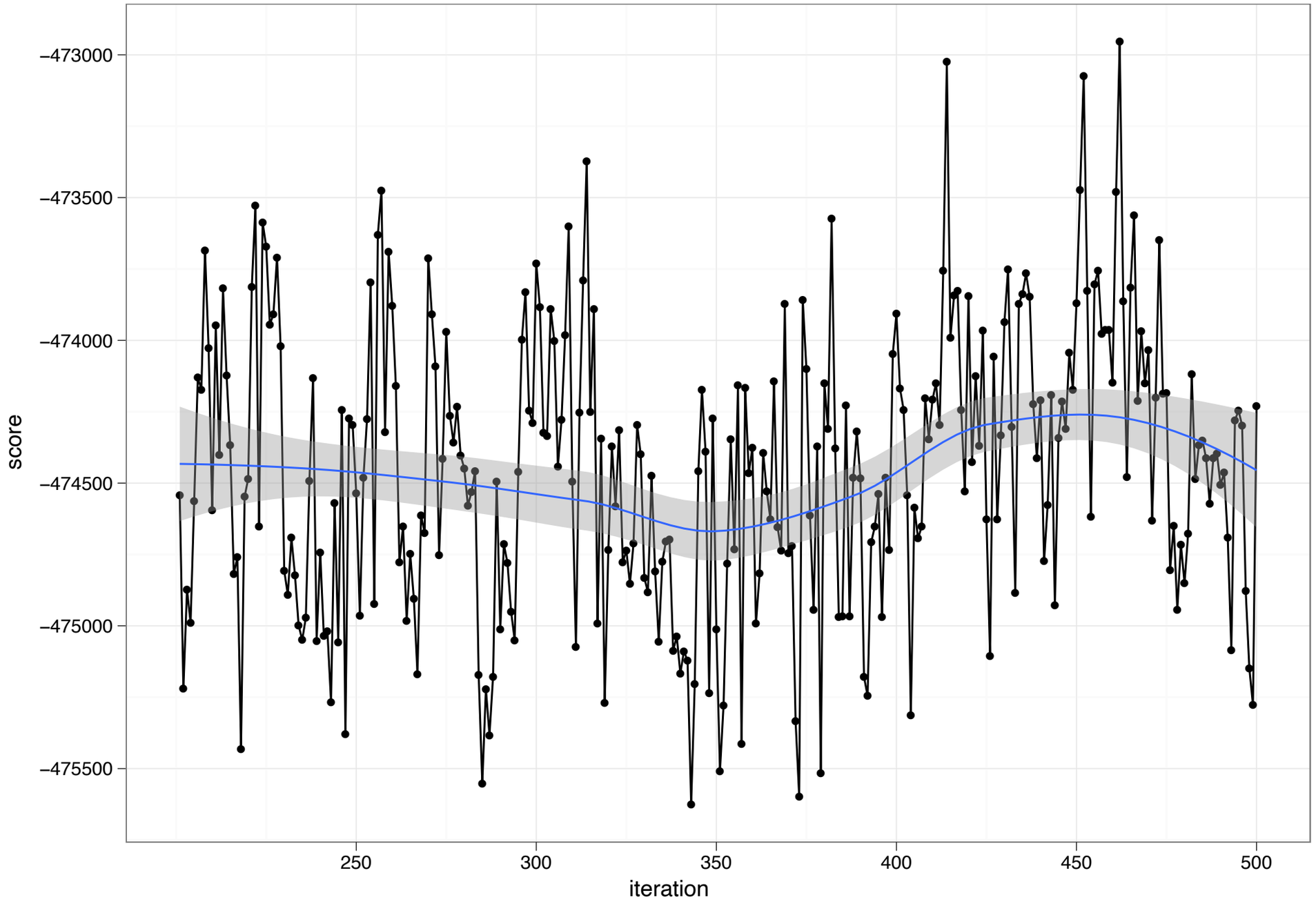
- Look at counts like $m_k(\mathbf{z}_{1:D}, \mathbf{W})$ to investigate the topics; look at $n(\mathbf{z}_d)$ to investigate how each document exhibits them.

- **A good habit: Assess the convergence of the chain.**
  - Monitor the log probability of the state & observations. (Its exponential is proportional to the posterior.)
  - Do something fancier, e.g., Raftery and Lewis (1992).

Assessing convergence example



Assessing convergence example



Gibbs sampling for LDA

- Simple algorithm for sampling from a complex distribution.
- Works well in practice. Is the best first algorithm to try.
- However
  - Can be slow for very large data sets
  - It is difficult to handle nonconjugacy; it is hard to generalize to the dynamic topic model and correlated topic model.

Variational inference

- Variational inference replaces sampling with **optimization**.
- The main idea—
    - Place a distribution over the hidden variables with free parameters, called **variational parameters**.
        - Optimize the variational parameters to make the distribution close (in KL divergence) to the true posterior
- In some settings, variational inference is faster than MCMC.
- It is easier to handle nonconjugate pairs of distributions with variational inference. (This is important in the CTM, DTM, etc.)

Variational inference (in general)

- Let $x = x_{1:N}$ be observed variables;
  let $z = z_{1:M}$ be the latent variables.

- Our goal is to compute the posterior distribution

$$p(z \mid x) = \frac{p(z, x)}{\int p(z, x) dz}$$

- For many interesting distributions, the marginal likelihood of the observations is difficult to efficiently compute.

Variational inference

- Introduce a distribution over the latent variables $q_\nu(z)$, parameterized by *variational parameters* ν.
- Use Jensen's inequality to bound the log probability of the observations, (Jordan et al., 1999)

$$\begin{aligned}
\log p(x) &= \log \int p(z, x) dz \\
&= \log \int p(z, x) \frac{q_\nu(z)}{q_\nu(z)} dz \\
&\geq \mathrm{E}_{q_\nu}[\log p(Z, x)] - \mathrm{E}_{q_\nu}[\log q_\nu(Z)]
\end{aligned}$$

  (J. McAuliffe calls this the **evidence lower bound**, or ELBO.)

- Optimize the variational parameters to tighten this bound.
- This is the same as finding the member of the family q_ν that is closest in KL divergence to $p(z \mid x)$.

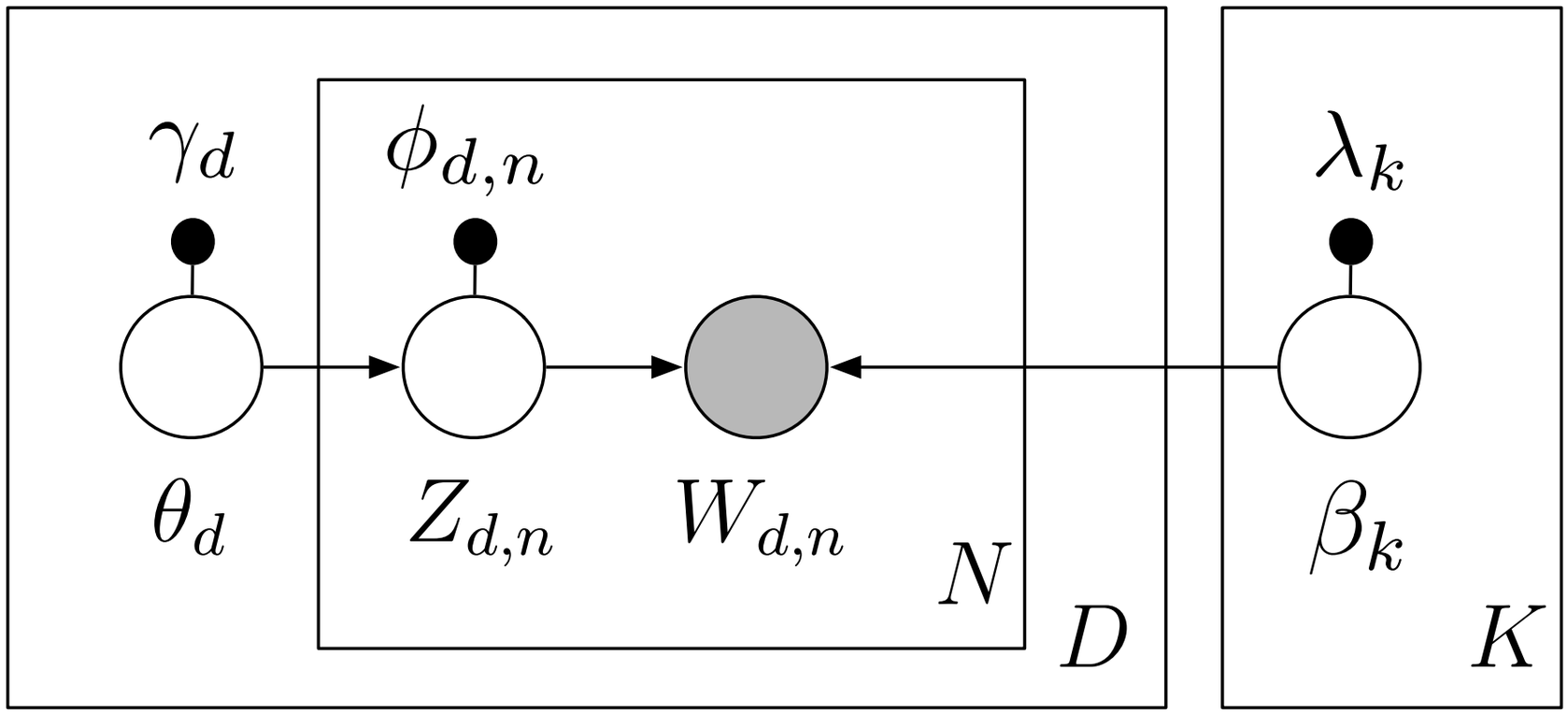
Mean-field variational inference

- Complexity is determined by the factorization of q_ν
- In *mean field variational inference* q_ν is fully factored

$$q_\nu(z) = \prod_{m=1}^{M} q_{\nu_m}(z_m).$$

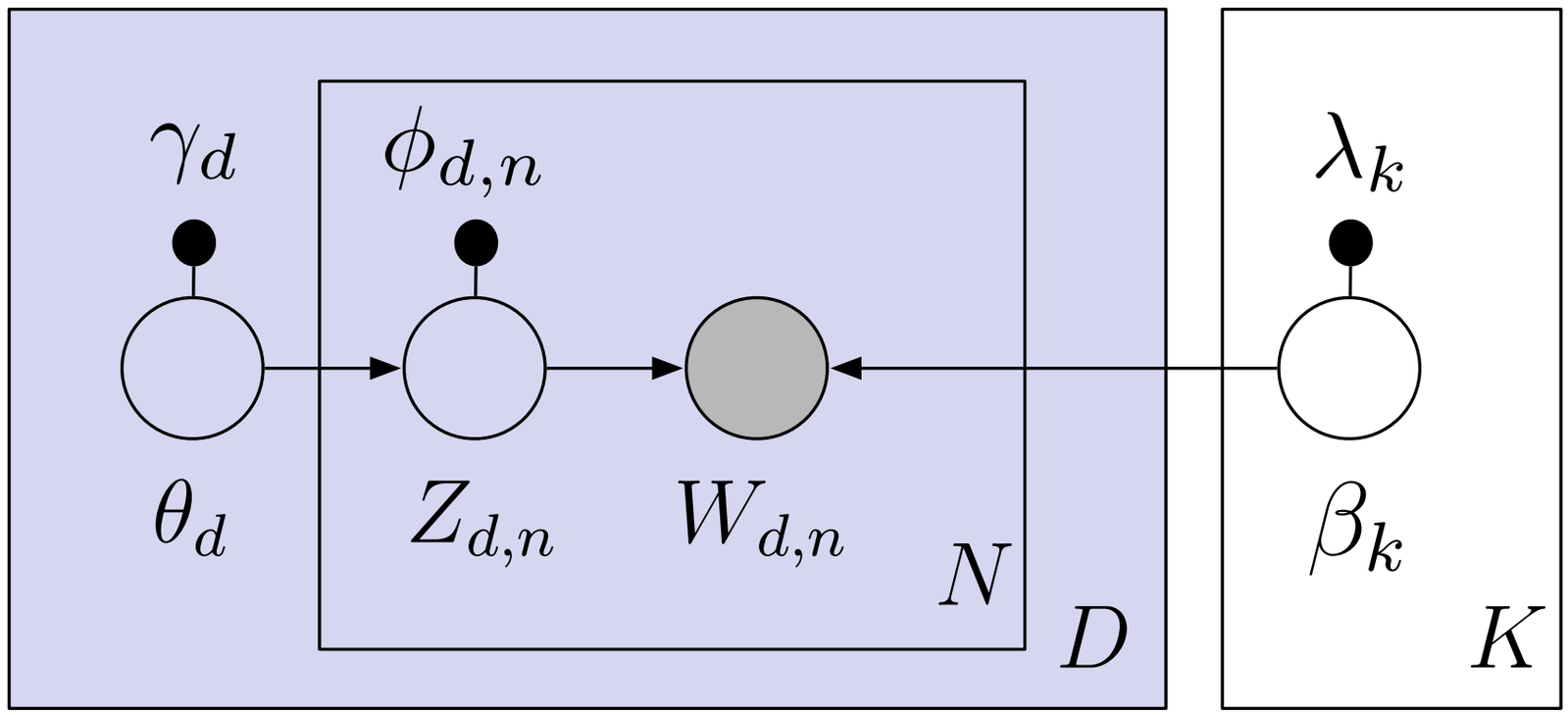
- Each latent variable is independently governed by its own variational parameter ν_m.
- In the true posterior they can exhibit dependence. (Often, this is what makes exact inference difficult.)

Variational inference for LDA



- The *mean field distribution* places a variational parameter on each hidden variable.
- Optimize these with coordinate ascent, iteratively optimizing each parameter while holding the others fixed.

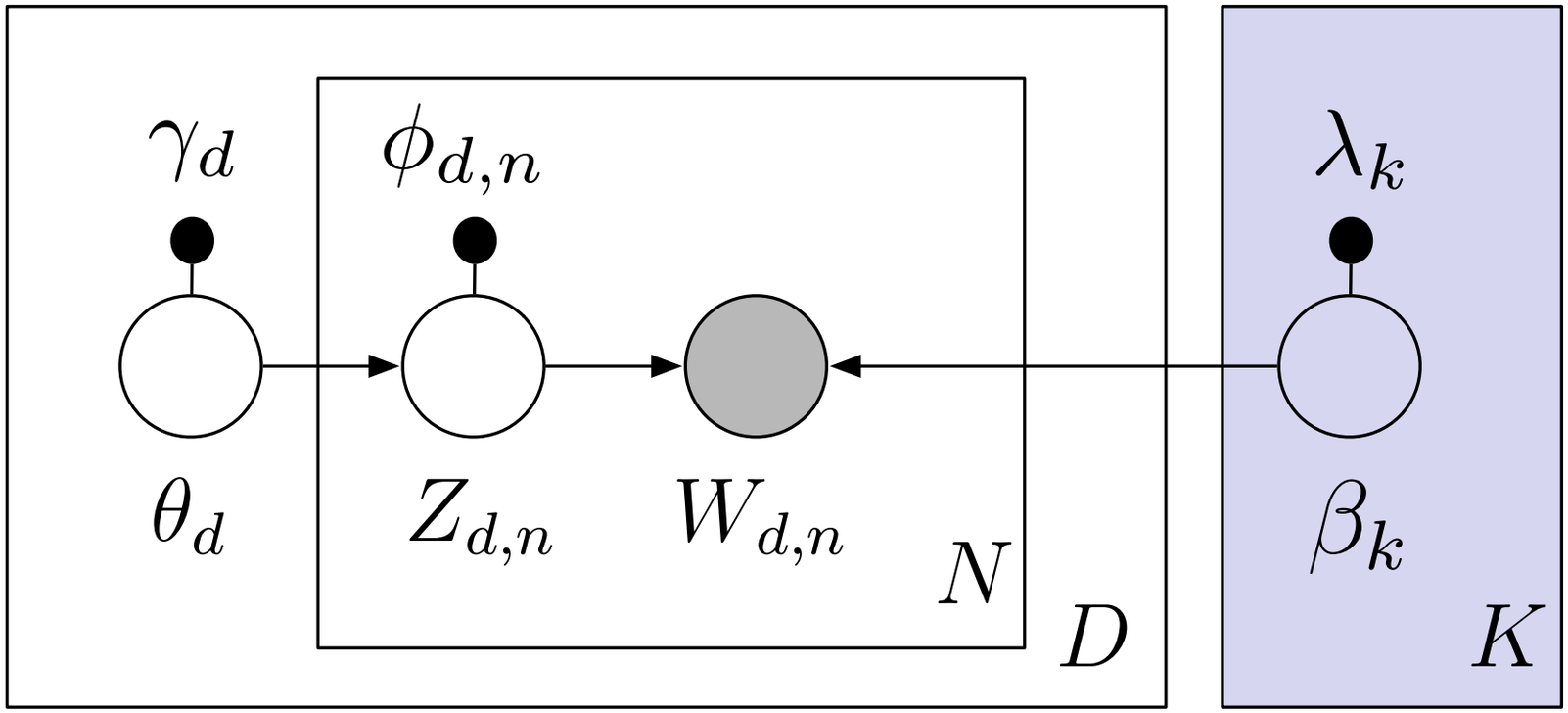
Variational inference for LDA



- In the "local step" we iteratively update the parameters for each document, holding the topic parameters fixed.

$$\begin{aligned}\gamma^{(t+1)} &= \alpha + \sum_{n=1}^{N} \phi_n^{(t)} \\ \phi_n^{(t+1)} &\propto \exp\{\mathbb{E}_q[\log \theta] + \mathbb{E}_q[\log \beta_{\cdot,w_n}]\}.\end{aligned}$$

Variational inference for LDA



- In the “global step” we aggregate the parameters computed from the local step and update the parameters for the topics,

$$\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}.$$

Variational inference for LDA (sketch)

1: Initialize topics randomly.
2: **repeat**
3: **for** each document **do**
4: **repeat**
5: Update the topic assignment variational parameters.
6: Update the topic proportions variational parameters.
7: **until** document objective converges
8: **end for**
9: Update the topics from aggregated per-document parameters.
10: **until** corpus objective converges.

Variational inference for LDA

1: Initialize topics $\boldsymbol{\lambda}_{1:K}$ randomly.
2: **while** relative improvement in $\mathcal{L}(\mathbf{w}, \phi, \boldsymbol{\gamma}, \boldsymbol{\lambda}) > \epsilon$ **do**
3: **for** $d = 1$ to D **do**
4: Initialize $\gamma_{d,k} = 1$.
5: **repeat**
6: Set $\phi_{d,n} \propto \exp\{\mathbb{E}_q[\log \theta_d] + \mathbb{E}_q[\log \beta_{\cdot,w_n}]\}$
7: Set $\gamma_d = \alpha + \sum_n \phi_{d,n}$
8: **until** $\frac{1}{K} \sum_k |\text{change in } \gamma_{d,k}| < \epsilon$
9: **end for**
10: Set $\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}$
11: **end while**

"E step"

1: Initialize topics $\lambda_{1:K}$ randomly.
2: **while** relative improvement in $\mathcal{L}(\mathbf{w}, \phi, \gamma, \lambda) > \epsilon$ **do**
3: **for** $d = 1$ to D **do**
4: Initialize $\gamma_{d,k} = 1$.
5: **repeat**
6: Set $\phi_{d,n} \propto \exp\{\mathbb{E}_q[\log \theta_d] + \mathbb{E}_q[\log \beta_{\cdot,w_n}]\}$
7: Set $\gamma_d = \alpha + \sum_n \phi_{d,n}$
8: **until** $\frac{1}{K} \sum_k |\text{change in } \gamma_{d,k}| < \epsilon$
9: **end for**
10: Set $\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}$
11: **end while**

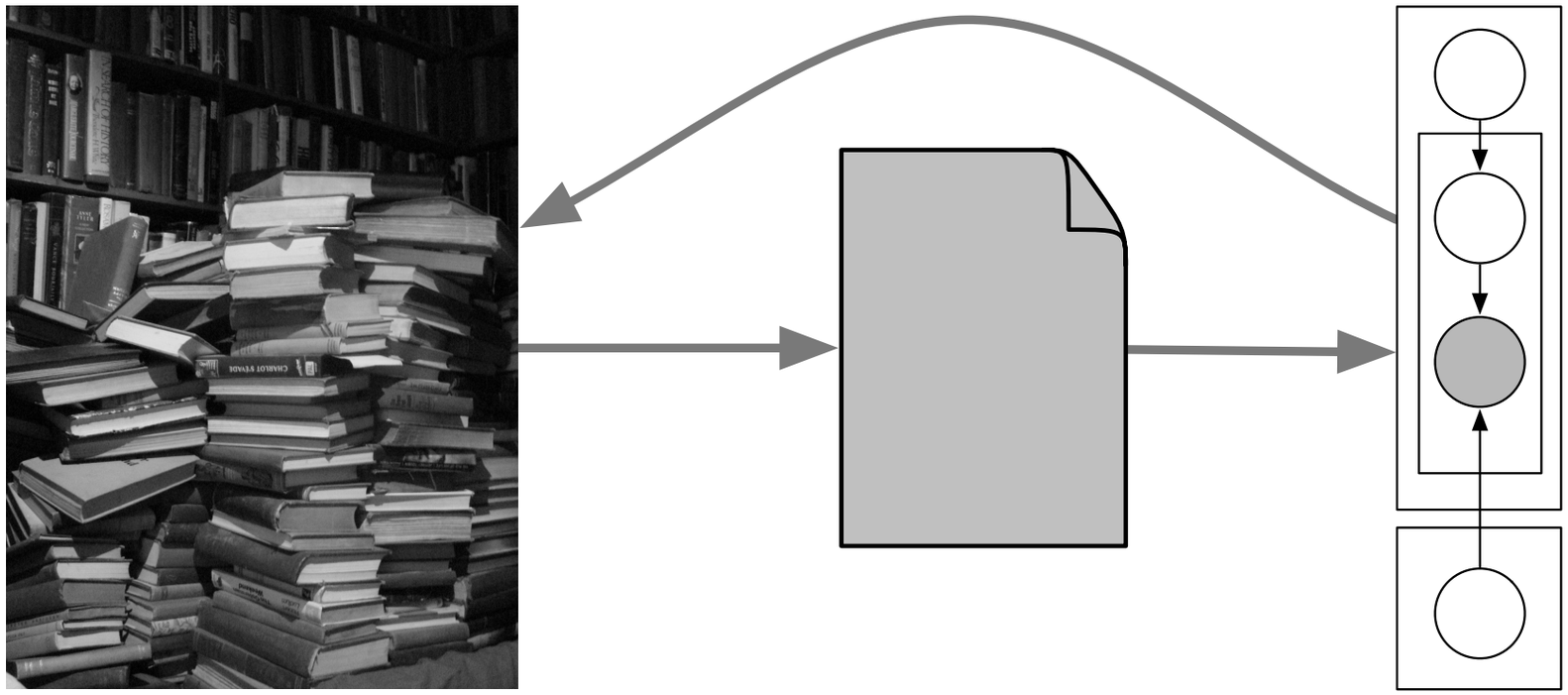
Do variational inference for each document.

"M step"

1: Initialize topics $\lambda_{1:K}$ randomly.
2: **while** relative improvement in $\mathcal{L}(\mathbf{w}, \phi, \gamma, \lambda) > \epsilon$ **do**
3: **for** $d = 1$ to D **do**
4: Initialize $\gamma_{d,k} = 1$.
5: **repeat**
6: Set $\phi_{d,n} \propto \exp\{\mathbb{E}_q[\log \theta_d] + \mathbb{E}_q[\log \beta_{\cdot,w_n}]\}$
7: Set $\gamma_d = \alpha + \sum_n \phi_{d,n}$
8: **until** $\frac{1}{K} \sum_k |$ change in $\gamma_{d,k}| < \epsilon$
9: **end for**
10: Set $\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}$
11: **end while**

Update the posterior estimates of the topics based on the "E step."

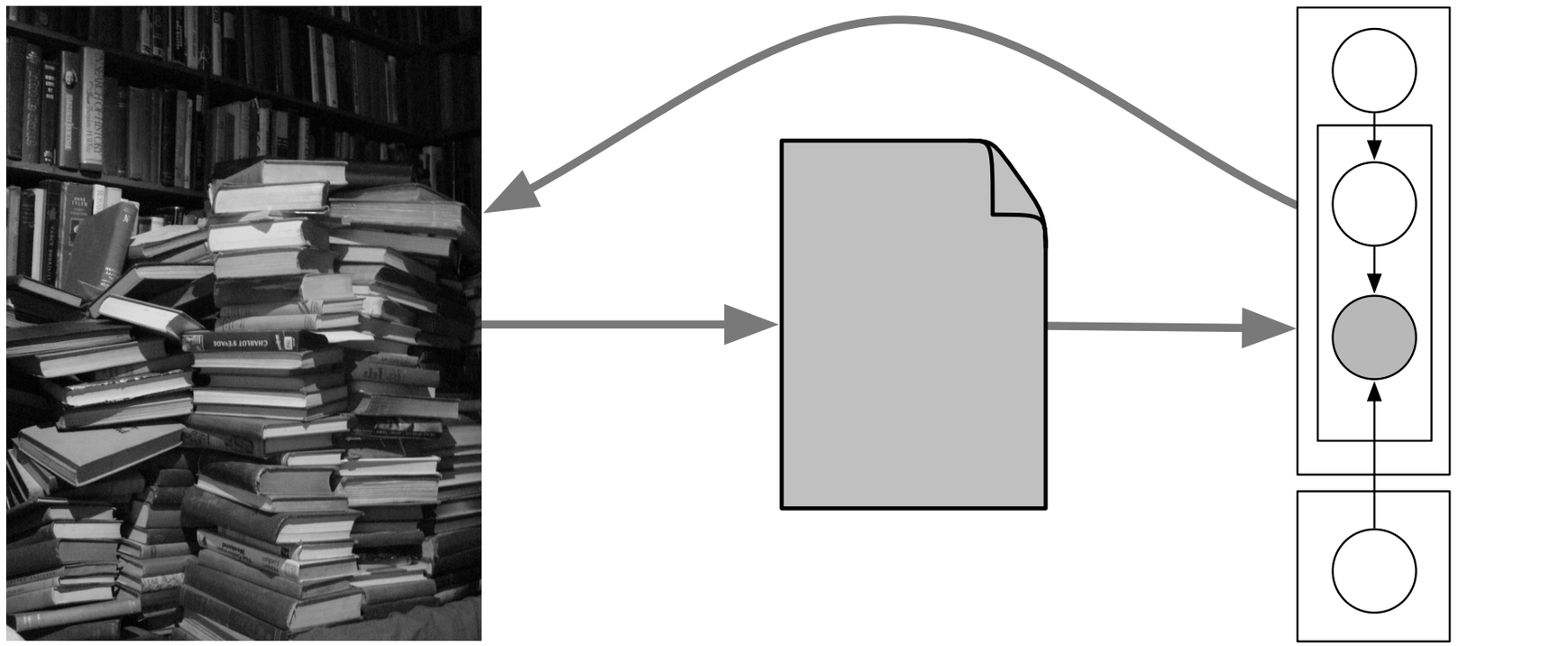
Online inference for LDA (with M. Hoffman and F. Bach)



Sample one document ***Analyze it*** ***Update the model***

- Our goal is to use this (and related) models for analyzing massive collections of millions of documents.
- But, in the first step of batch inference we estimate the posterior for *every document* based on randomly initialized topics.

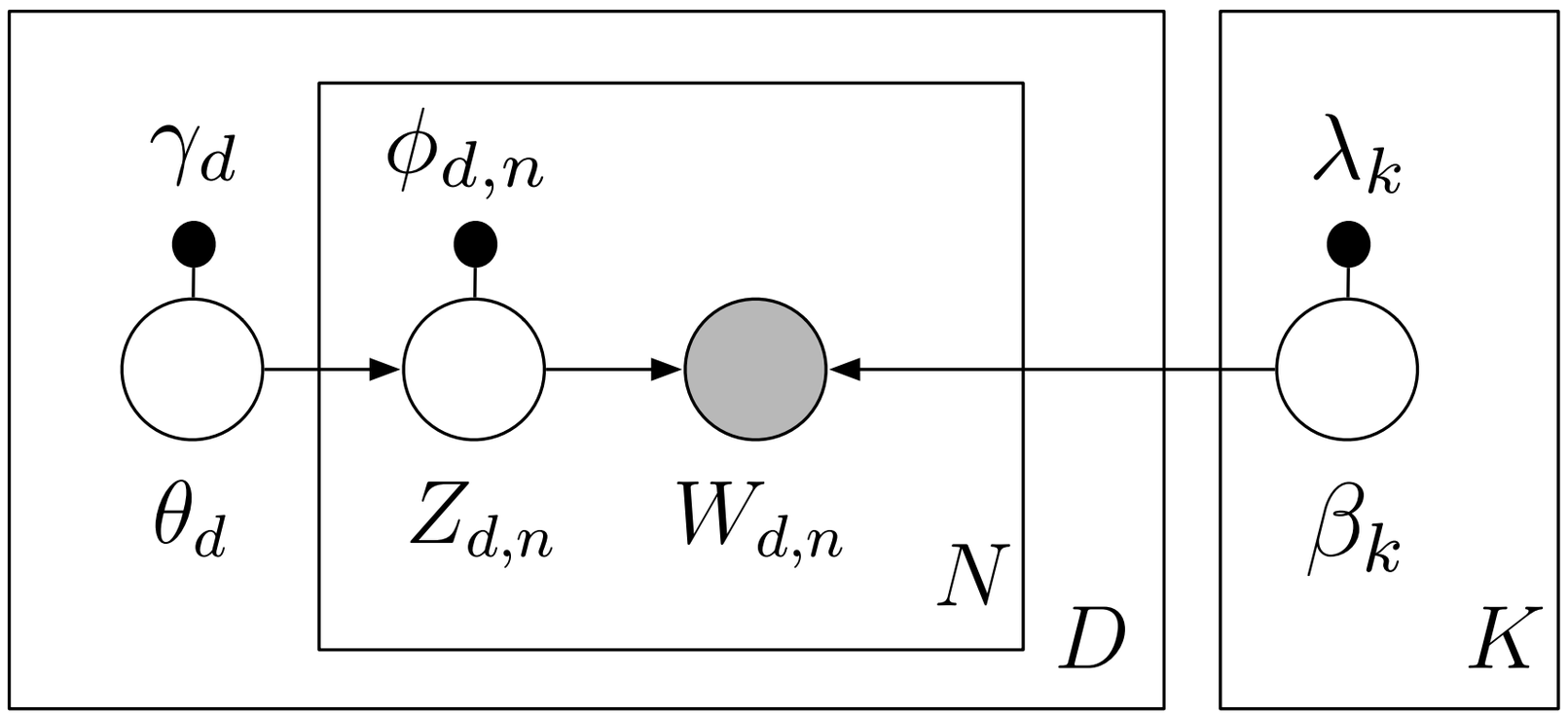
Online inference for LDA (with M. Hoffman and F. Bach)



Sample one document ***Analyze it*** ***Update the model***

- Online variational inference is much more efficient.
- It allows us to easily analyze millions of documents.
- It lets us develop topic models on streaming collections.

Online inference for LDA



1. Randomly pick a document.
2. Perform local variational inference with the current topics.
3. Form “fake” topics, treating the sampled document as though it were the only document in the collection.
4. Update the topics to be a weighted average of the fake topics and current topics.

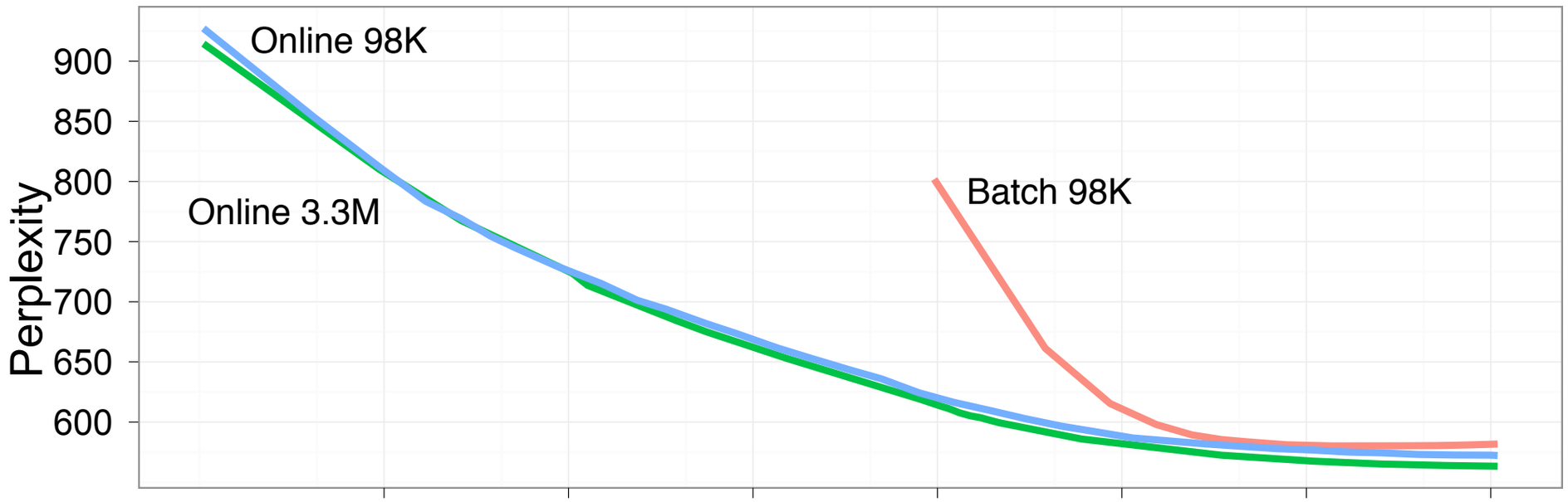
Online variational inference for LDA (sketch)

1: Define an appropriate sequence of weights.
2: Initialize topics randomly.
3: **for** ever **do**
4: Choose a random document d.
5: **repeat**
6: Update the topic assignment variational parameters.
7: Update the topic proportions variational parameters.
8: **until** document objective converges
9: Compute topics as though d is the only document.
10: Set the topics to a weighted average of the current topics and the topics from step 9.
11: **end for**

On-line variational inference for LDA

Define $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$
Initialize $\boldsymbol{\lambda}$ randomly.
for $t = 0$ to ∞ **do**
    Choose a random document w_t
    Initialize $\gamma_{tk} = 1$. (The constant 1 is arbitrary.)
    repeat
        Set $\phi_{t,n} \propto \exp\{\mathbb{E}_q[\log \theta_t] + \mathbb{E}_q[\log \beta_{\cdot,w_n}]\}$
        Set $\gamma_t = \alpha + \sum_n \phi_{t,n}$
    until $\frac{1}{K}\sum_k |\text{change in } \gamma_{t,k}| < \epsilon$
    Compute $\tilde{\lambda}_k = \eta + D\sum_n w_{t,n}\phi_{t,n}$
    Set $\lambda_k = (1-\rho_t)\lambda_k + \rho_t\tilde{\lambda}_k$.
end for

Analyzing 3.3M articles from Wikipedia



Documents analyzed	**2048**	**4096**	**8192**	**12288**	**16384**	**32768**	**49152**	**65536**
Top eight words	systems	systems	service	service	service	business	business	business
	road	health	systems	systems	companies	service	service	industry
	made	communication	health	companies	systems	companies	companies	service
	service	service	companies	business	business	industry	industry	companies
	announced	billion	market	company	company	company	services	services
	national	language	communication	billion	industry	management	company	company
	west	care	company	health	market	systems	management	management
	language	road	billion	industry	billion	services	public	public

Why does this work?

A STOCHASTIC APPROXIMATION METHOD[1]

By Herbert Robbins and Sutton Monro

University of North Carolina

1. Summary. Let $M(x)$ denote the expected value at level x of the response to a certain experiment. $M(x)$ is assumed to be a monotone function of x but is unknown to the experimenter, and it is desired to find the solution $x = \theta$ of the equation $M(x) = \alpha$, where α is a given constant. We give a method for making successive experiments at levels $x_1, x_2, \cdots$ in such a way that x_n will tend to θ in probability.

- Why waste time with the real gradient, when a cheaper noisy estimate of the gradient will do (Robbins and Monro, 1951)?
- Idea: Follow a noisy estimate of the gradient with a step-size.
- By decreasing the step-size according to a certain schedule, we guarantee convergence to a local optimum.
- See Hoffman et al. (2010) and Sato (2001).

Online inference is promising, in general

- Stochastic variational methods are a general way to approximate the posterior for massive/streaming data.
- No need to process the whole data set in advance; can easily link to web APIs and other data sources
- Powerful algorithm for topic modeling, and can be adapted hierarchical models for many types of data.
- Software and papers: www.cs.princeton.edu/$\sim$blei/

Implementations of LDA

There are many available implementations of topic modeling—

LDA-C*	A C implementation of LDA
HDP*	A C implementation of the HDP ("infinite LDA")
Online LDA*	A python package for LDA on massive data
LDA in R*	Package in R for many topic models
LingPipe	Java toolkit for NLP and computational linguistics
Mallet	Java toolkit for statistical NLP
TMVE*	A python package to build browsers from topic models

* available at www.cs.princeton.edu/∼blei/