



Topic Models

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

Topic Modelling



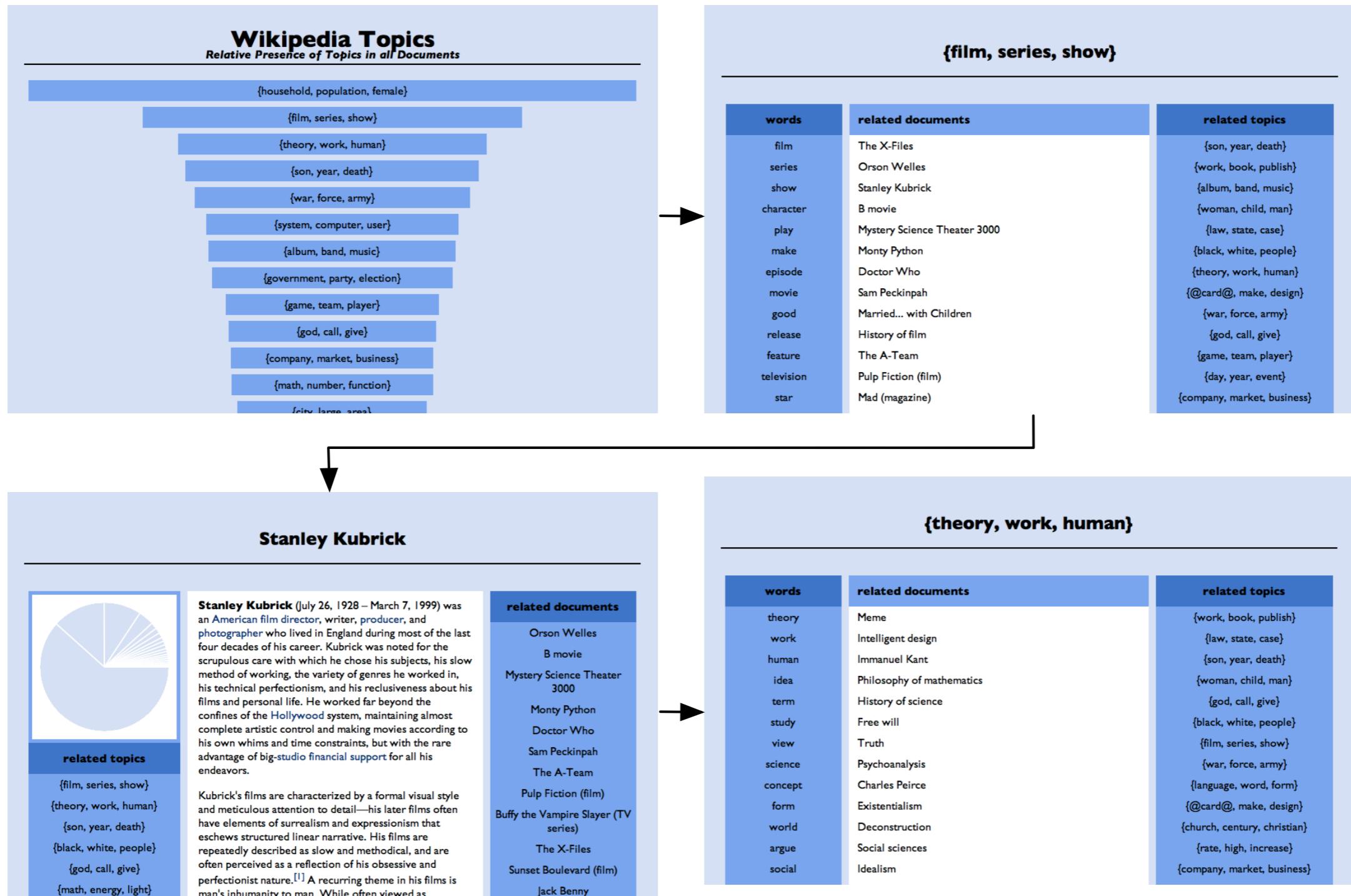
Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.

- 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.

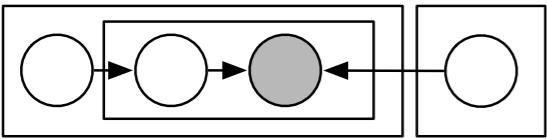
Discover topics from a corpus

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

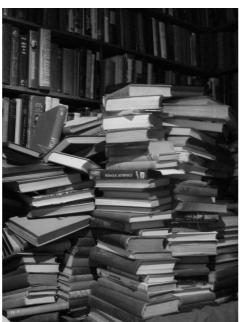
Organize and browse large corpora



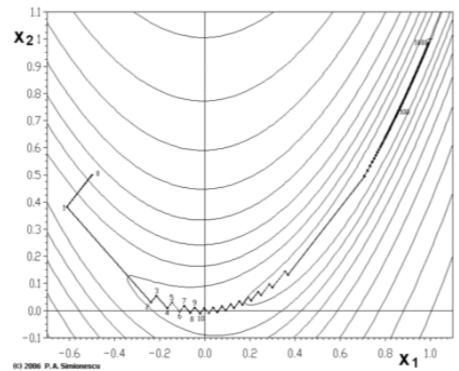
Assumptions



Data



Inference algorithm



Discovered structure

Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts

Gerard Salton, James Allan, Chris Vast amounts of text material are now available in processing. Here, approaches are outlined for manipulating subject areas in accordance with user needs. In particular, text themes, traversing texts selectively, and reflect text content.

Many kinds of texts are currently available in machine-readable form and are amenable to automatic processing. Because the available databases are large and cover many topics, it is often possible to find what needs to be provided to users interested in the topic. It has been suggested that links be established between the various databases; for example, particular context paragraphs to other paragraphs containing related information. The use of hypertext, for example, makes it possible for the reader to switch from one part of a document to another, perhaps to related text elements (1). Unfortunately, until now, viable methods for automatically creating large hypertext structures and for using such hypertext structures effectively have not been available. Here we give methods for creating text relations, particularly for texts relevant to medical and scientific databases. In particular, we outline procedures for determining text themes, thematic relations, and thematic consistency statements that reflect text contexts.

Text Analysis and Retrieval: The Smart System

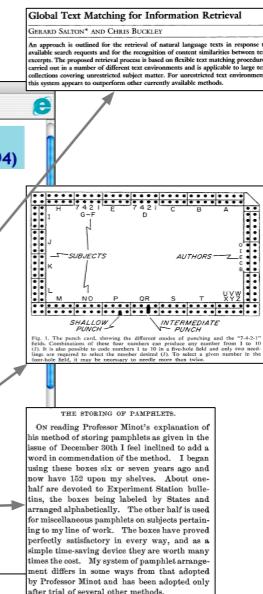
The first step in the retrieval process is to model all retrieval model, all informa-
tion as well as informa-
tion by sets of terms, or terms
associated with the
information. In principle
one can build a collection
of a thesaurus, but best
constructing such a thesaurus
is difficult. One way is to
derive the terms under
consideration from the text
content.

Because the text
contains repeated
information, some
sign high weights
and lower weights
for other terms.

Another approach
is to let the user
choose which kind
is the well-known
frequency (f) in
frequency, which
is a low frequency
($f < 1$), and which
they occur in

TOPIC	PROB.
data computer system information network	0.30
information library text index libraries	0.19
two three four different single	0.16

<p>Text Analysis and Retrieval: The Smart System</p> <p>The smart system is a sophisticated text retrieval tool, developed over the past 30 years, that is based on the vector space model.</p> <p>The authors are in the Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, USA.</p>	<p>to determine the most appropriate terms assigned to a text content.</p> <p>Besides the term for content, we propose to introduce a term-weighting scheme that associates higher and lower weights to A. A powerful term-weighting scheme is based on term frequency (f_t), which is the frequency of term t in P with a low frequency (L). Such terms usually occur in P.</p> <p>When a term t is represented by weighted vectors $D_t = (d_{t1}, d_{t2}, \dots)$, it is possible to measure the similarity between pairs of vectors of term similarity. This is,</p>
SCIENCE • VOL 257	DOCUMENT
<p>"Global Text Matching for Information Retrieval" (1991) "Automatic Text Analysis" (1970) "Gauging Similarity with n-Grams: Language-Independent Categorization of Text" (1995) "Developments in Automatic Text Retrieval" (1991) "Simple and Rapid Method for the Coding of Punched Cards" (1962) "Data Processing by Optical Coincidence" (1961) "Pattern-Alignment Memory" (1976) "The Storing of Pamphlets" (1899) "A Punched-Card Technique for Computing Means, Standard Deviations, and the Product-Moment Correlation Coefficient and for Listing Scattergrams" (1946)</p>	





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?*

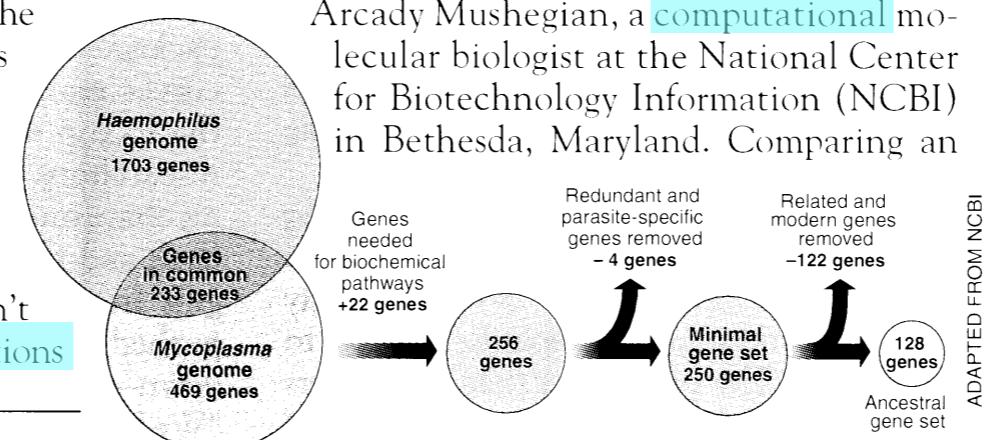
Latent Dirichlet Allocation (LDA)

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.

Generative model for LDA

Topics

```
gene      0.04
dna       0.02
genetic   0.01
...

```

```
life      0.02
evolve   0.01
organism 0.01
...

```

```
brain     0.04
neuron   0.02
nerve    0.01
...

```

```
data     0.02
number  0.02
computer 0.01
...

```

Documents

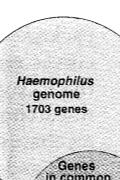
Seeking Life's Bare (Genetic) Necessities

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

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

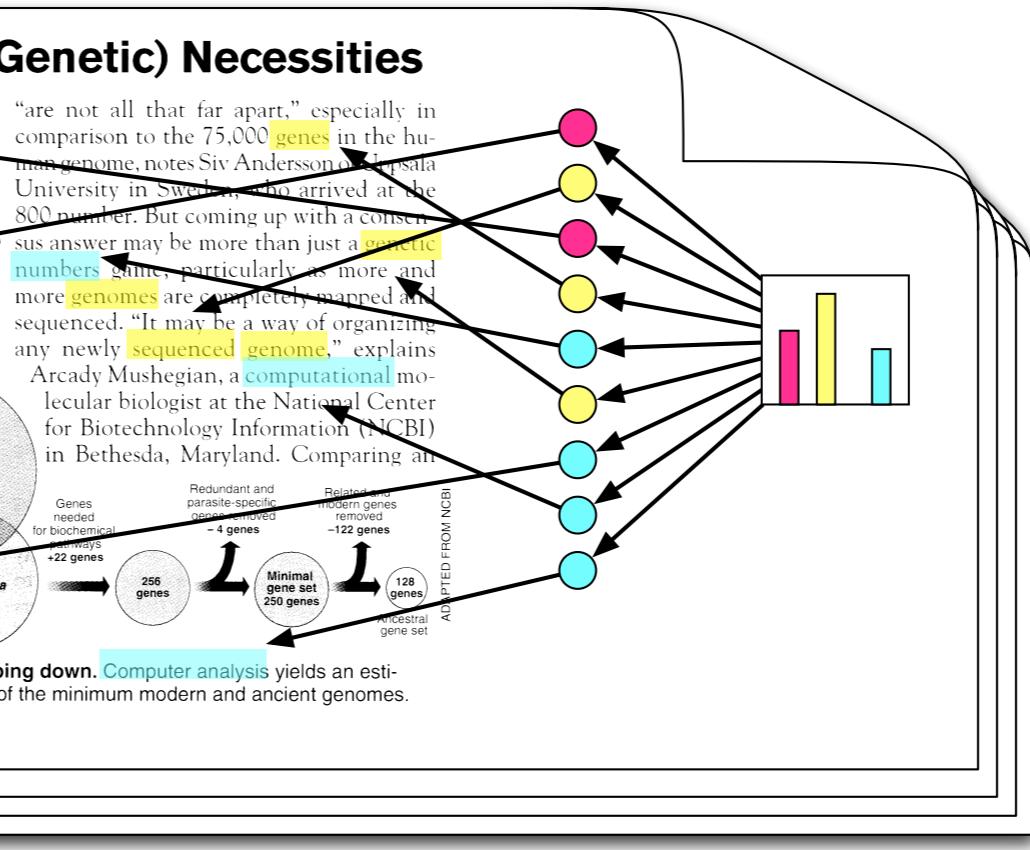
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes 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 all



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

SCIENCE • VOL. 272 • 24 MAY 1996

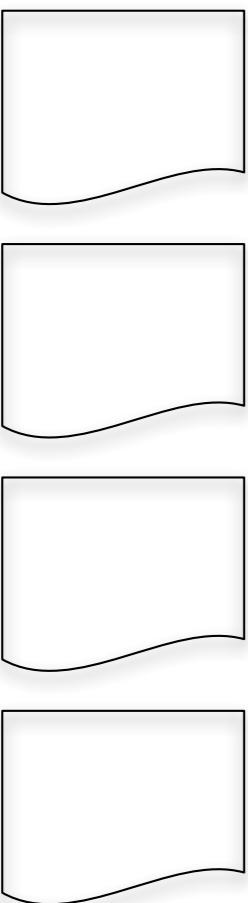
Topic proportions and assignments



- 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

Seeking Life's Bare (Genetic) Necessities

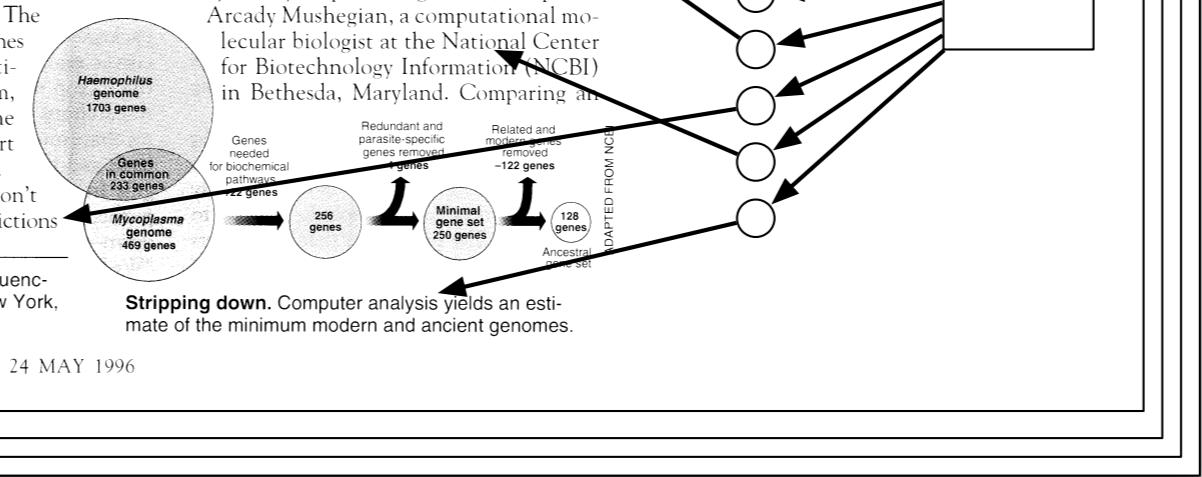
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known 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

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

SCIENCE • VOL. 272 • 24 MAY 1996

"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



Topic proportions and assignments

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

The posterior

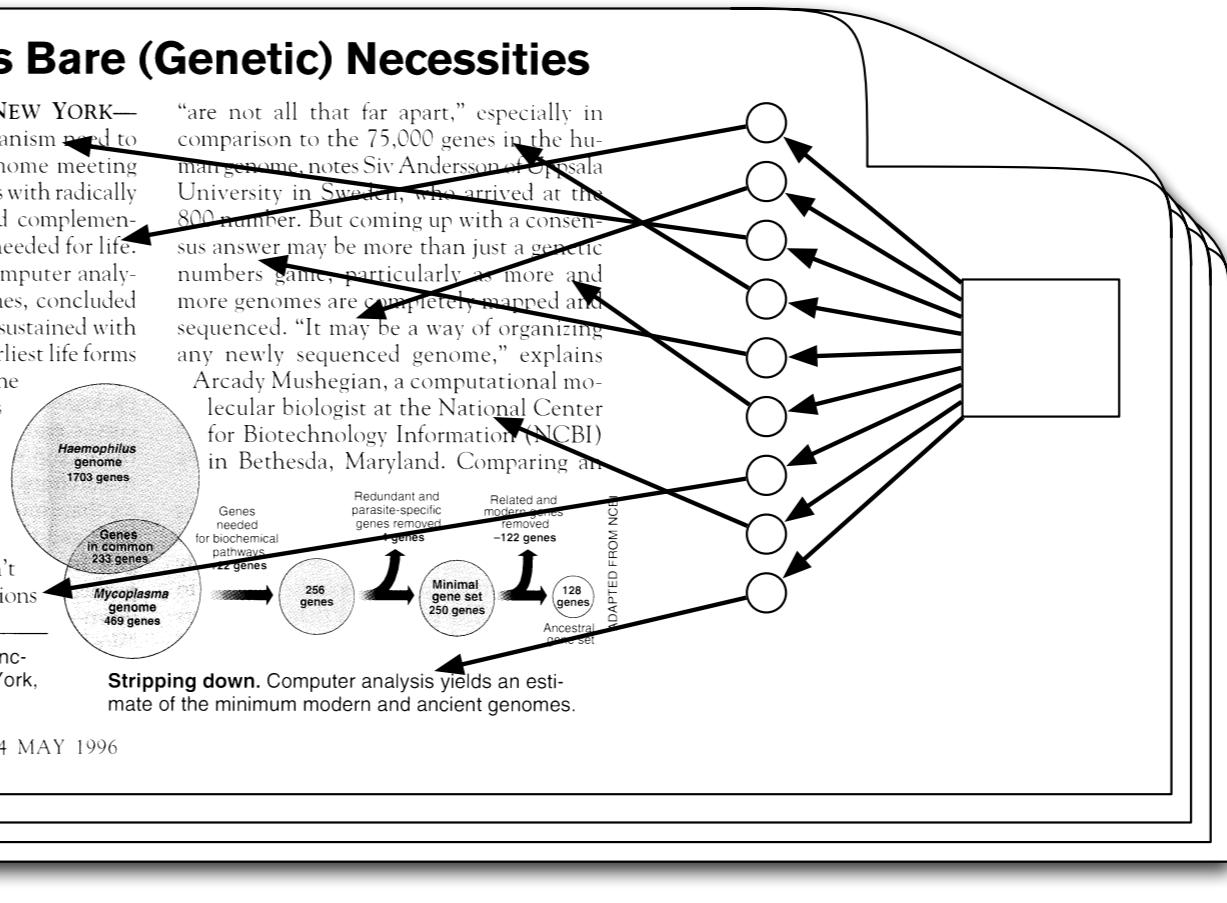
Topics



Documents

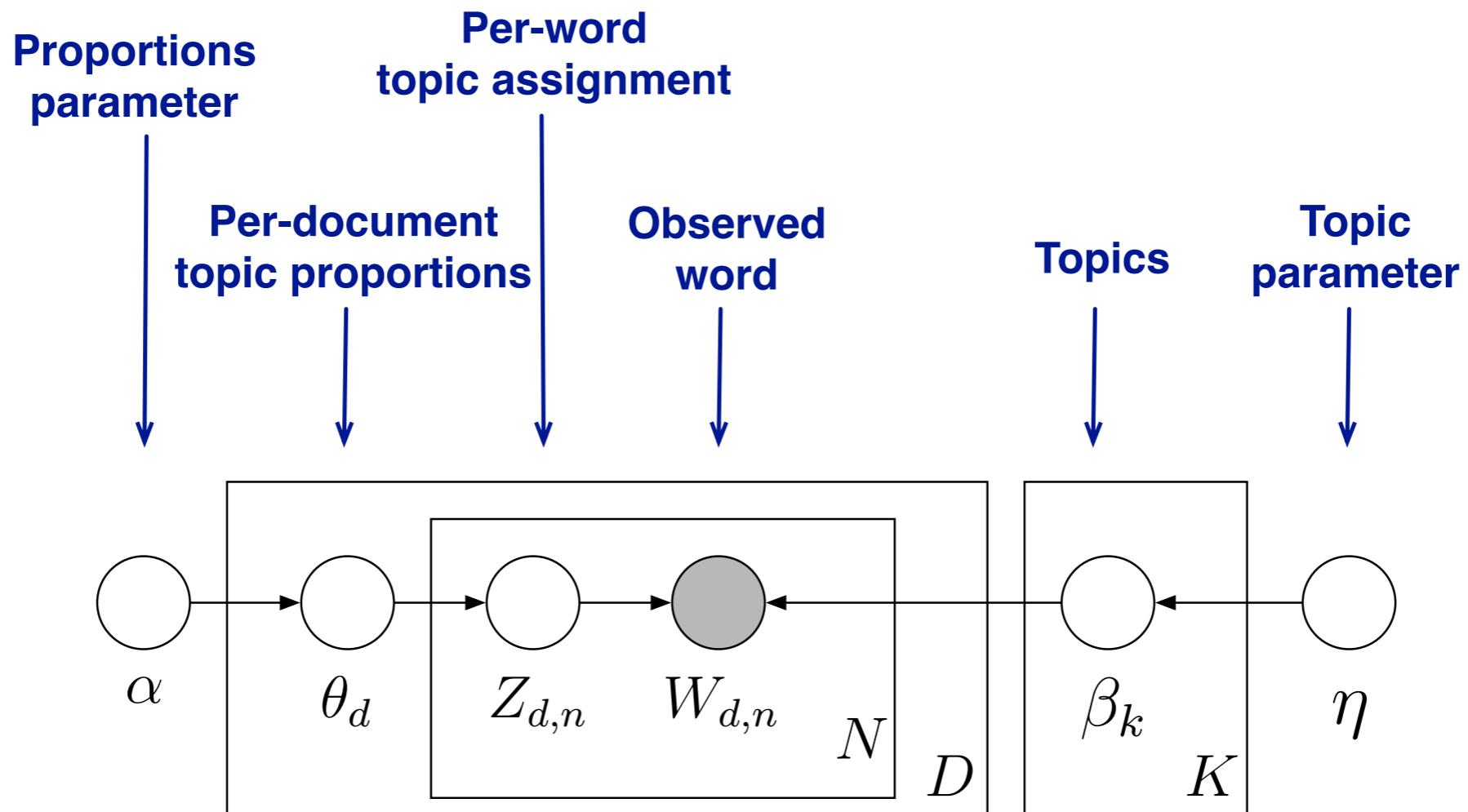


Topic proportions and assignments



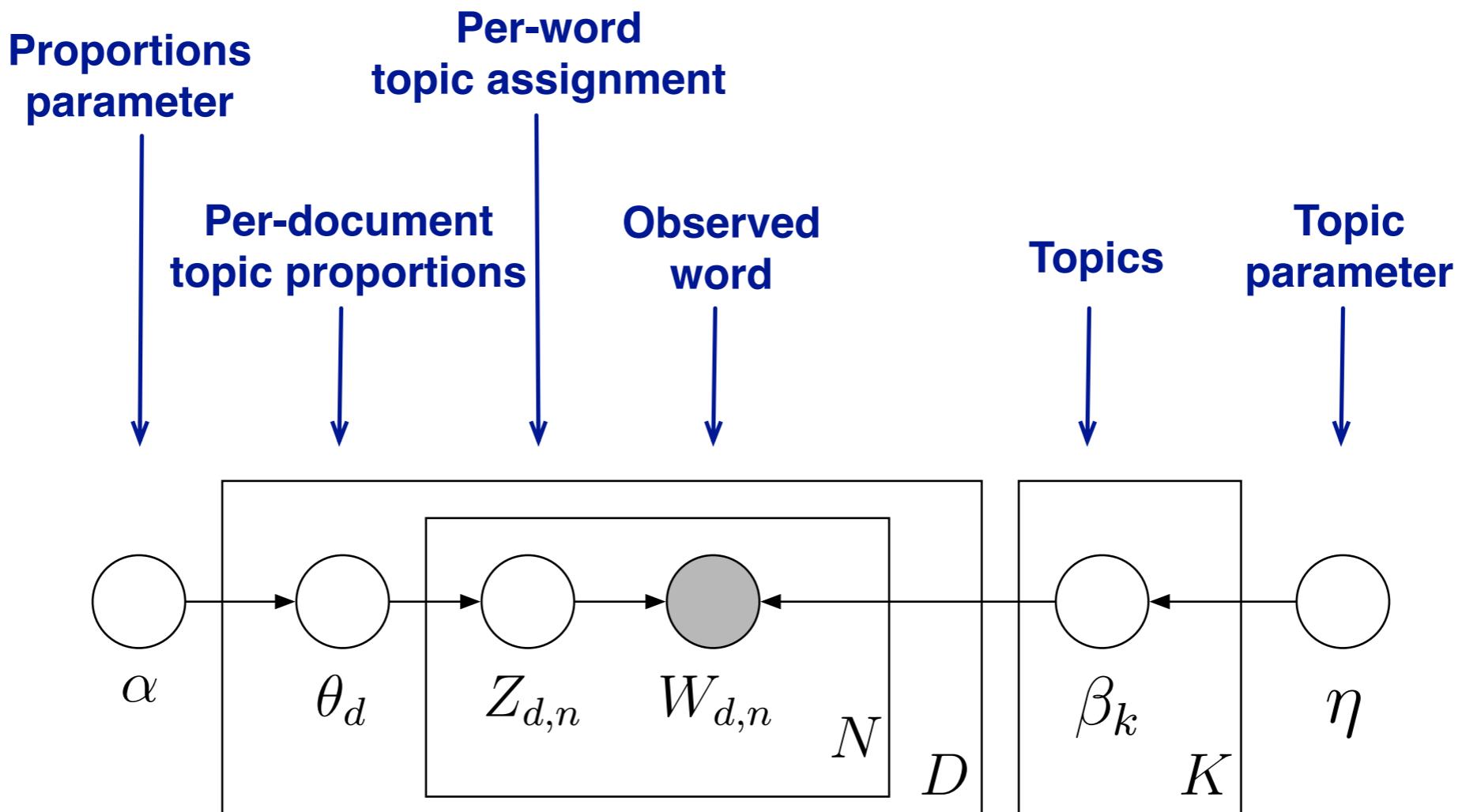
- Our goal is to **infer** the hidden variables
- I.e., compute their distribution conditioned on the documents
 $p(\text{topics, proportions, assignments} \mid \text{documents})$

LDA as a graphical model



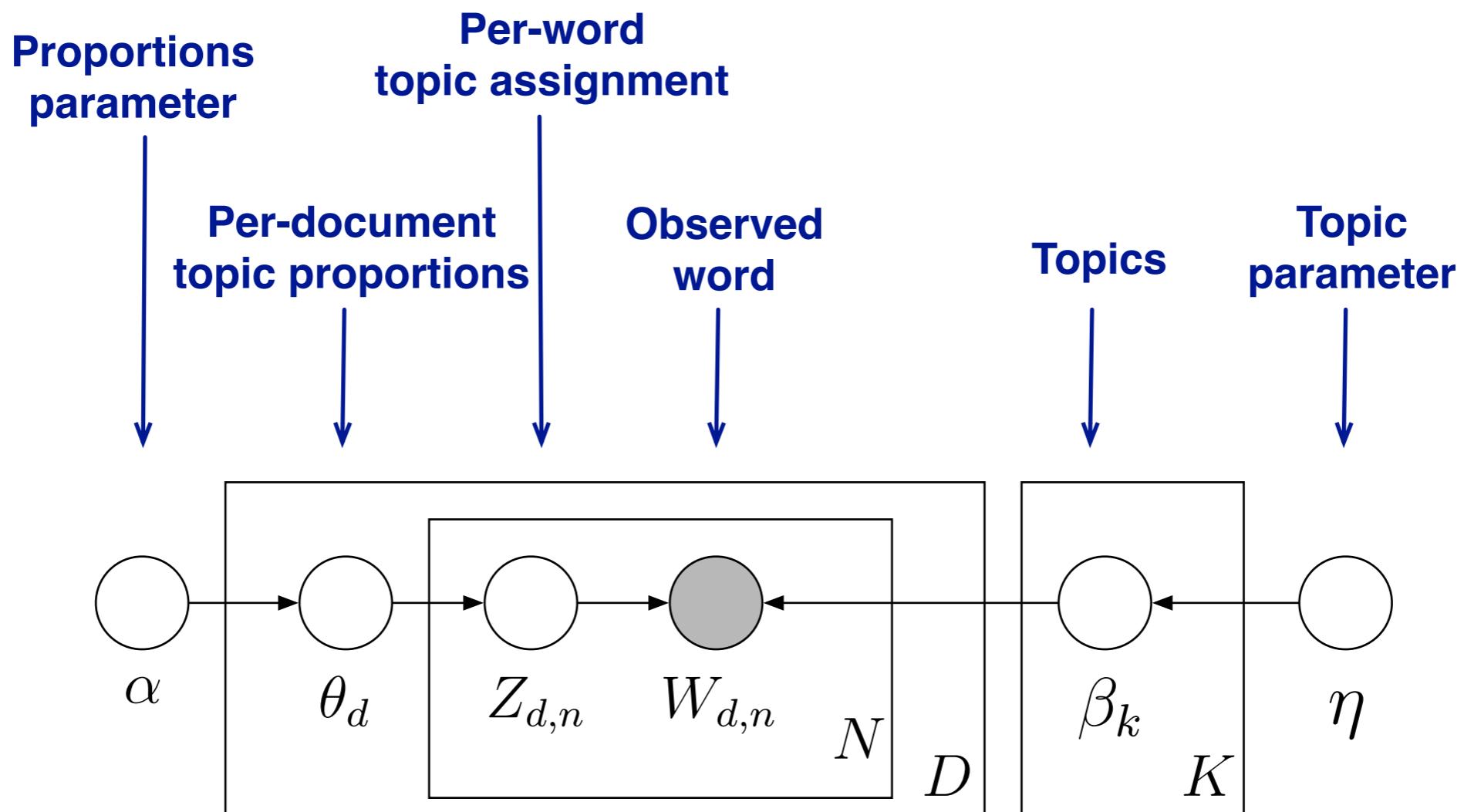
- Encodes our assumptions about the data
- Connects to algorithms for computing with data
- See *Pattern Recognition and Machine Learning* (Bishop, 2006).

LDA as a graphical model



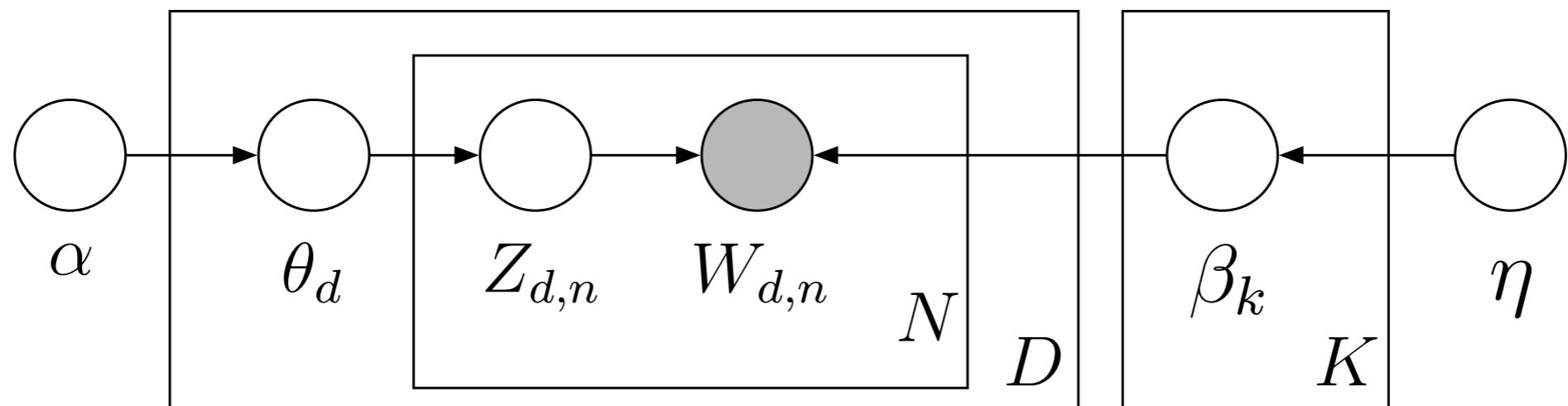
- Nodes are random variables; edges indicate dependence.
- Shaded nodes are observed.
- Plates indicate replicated variables.

LDA as a graphical model



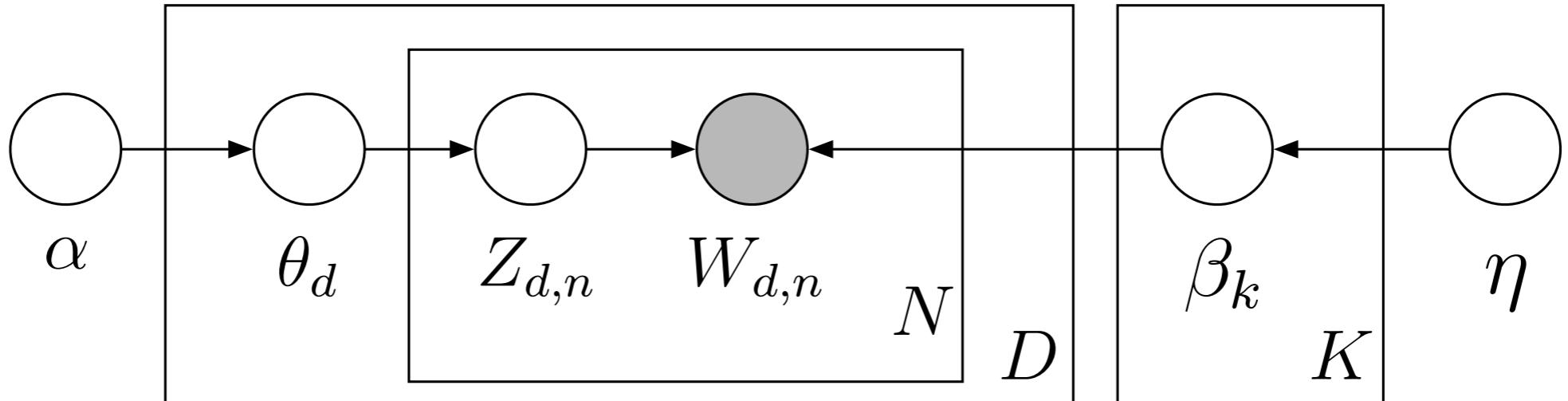
$$\prod_{i=1}^K p(\beta_i | \eta) \prod_{d=1}^D p(\theta_d | \alpha) \left(\prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

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

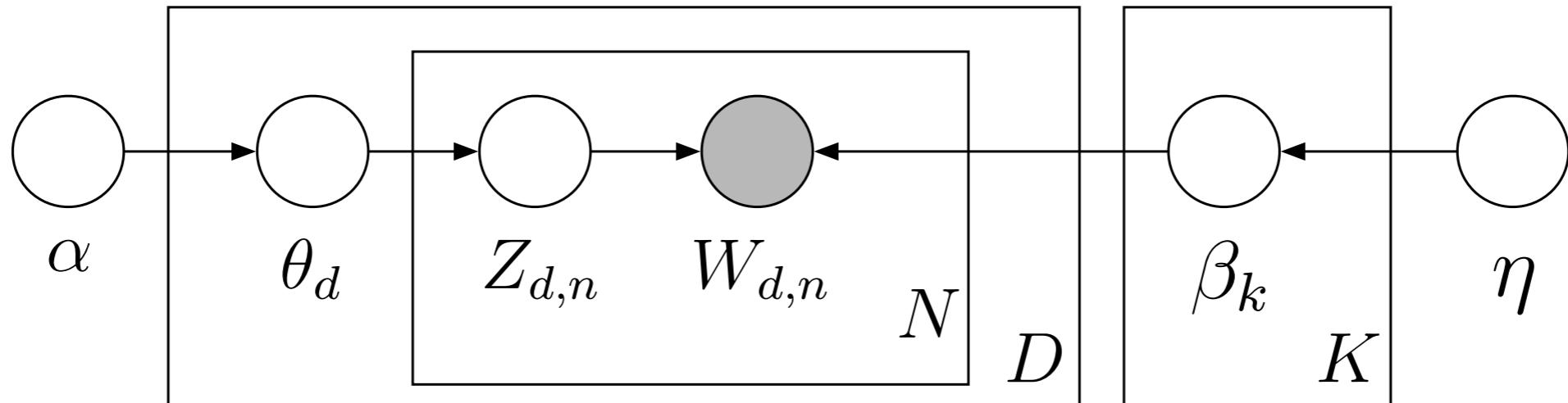


Approximate posterior inference algorithms

- 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



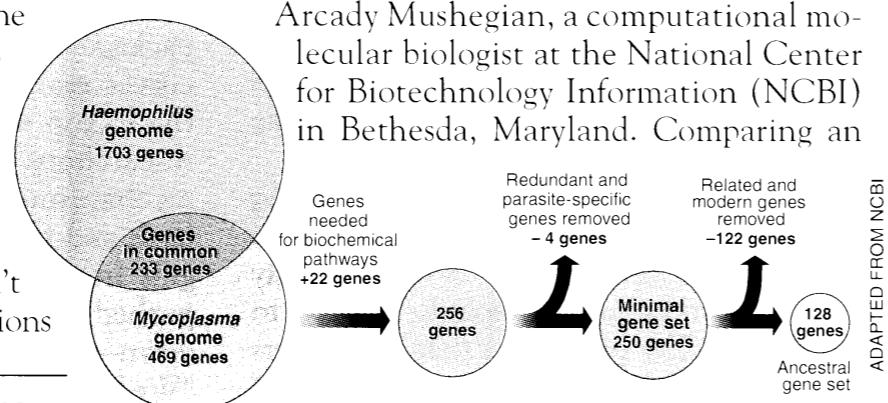
- **Data:** The OCR'ed collection of *Science* from 1990–2000
 - 17K documents
 - 11M words
 - 20K unique terms (stop words and rare words removed)
- **Model:** 100-topic LDA model using variational inference.

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

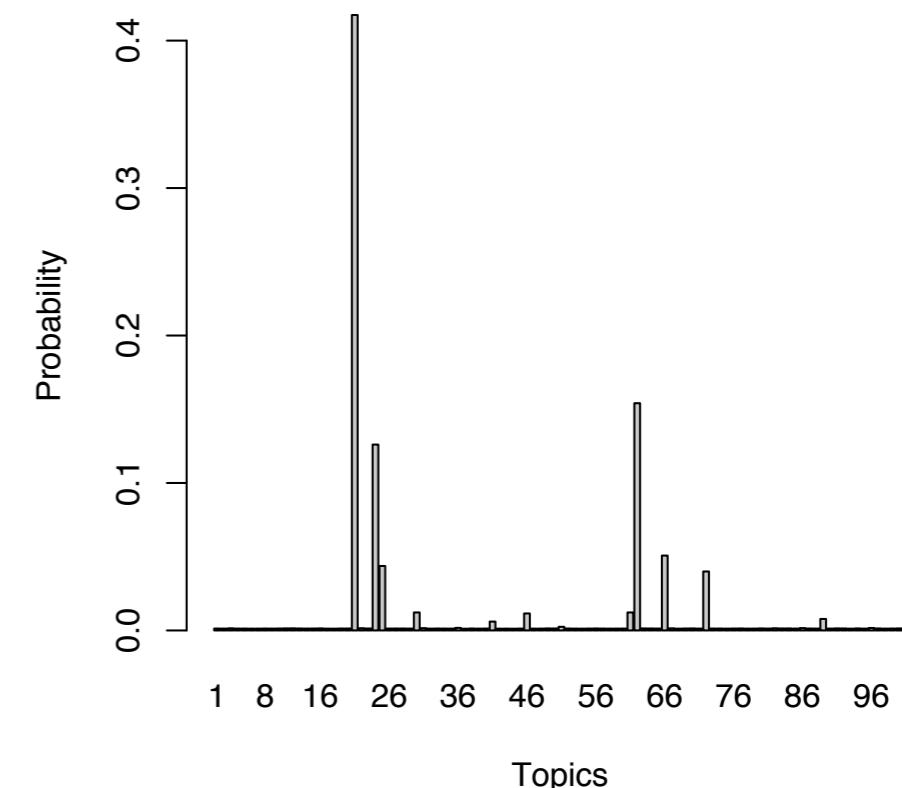
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes 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

Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts

Gerard Salton, James Allan, Chris Buckley,

Vast amounts of text material are now available in machine-readable form and are amenable to automatic processing. Here, approaches are outlined for manipulating and accessing subject areas in accordance with user needs. In particular, methods for mining text themes, traversing texts selectively, and extracting statements that reflect text content.

Many kinds of texts are currently available in machine-readable form and are amenable to automatic processing. Because the available databases are large and cover many different subject areas, automatic aids must be provided to users interested in accessing the data. It has been suggested that links be placed between related pieces of text, connecting, for example, particular text paragraphs to other paragraphs covering related subject matter. Such a linked text structure, often called hypertext, makes it possible for the reader to start with particular text passages and use the linked structure to find related text elements (1). Unfortunately, until now, viable methods for automatically building large hypertext structures and for using such structures in a sophisticated way have not been available. Here we give methods for constructing text relation maps and for using text relations to access and use text databases. In particular, we outline procedures for determining text themes, traversing texts selectively, and extracting summary statements that reflect text content.

Text Analysis and Retrieval: The Smart System

The Smart system is a sophisticated text retrieval tool, developed over the past 30 years, that is based on the vector space model.

The authors are in the Department of Computer Science, Cornell University, Ithaca, NY 14853-7501, USA.

model of retrieval model, all information as well as information represented by sets, or vectors, is typically a word, associated with the document. In principle chosen from a complete thesaurus, but because of the need for unrestricted topics to derive the terms under consideration, terms assigned to a text content.

Because the term for content representation introduce a term-weighting scheme that assigns high weights to high frequency and lower weights to low frequency terms. A powerful term-weighting scheme is the well-known term frequency (tf) times inverse document frequency (df), which is given by $f_i = \frac{f_i}{\sum_j f_j}$. Such terms distinguish which they occur frequently.

When all texts are represented by weighted vectors $D_i = (d_{i1}, d_{i2}, \dots, d_{in})$, the weight assigned to each dimension is a similarity measure between pairs of vectors. Thus, given two vectors D_i and D_j , their similarity is given by

SCIENCE • VOL



"Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts" (1994)

TOPIC	PROB
data computer system information network	0.30
information library text index libraries	0.19
two three four different single	0.16

DOCUMENT	SCORE
"Global Text Matching for Information Retrieval" (1991)	0.2570
"Automatic Text Analysis" (1970)	0.3110
"Gauging Similarity with n-Grams: Language-Independent Categorization of Text" (1995)	0.3210
"Developments in Automatic Text Retrieval" (1991)	0.3480
"Simple and Rapid Method for the Coding of Punched Cards" (1962)	0.3610
"Data Processing by Optical Coincidence" (1961)	0.4290
"Pattern-Analyzing Memory" (1976)	0.4320
"The Storing of Pamphlets" (1899)	0.4440
"A Punched-Card Technique for Computing Means, Standard Deviations, and the Product-Moment Correlation Coefficient and for Listing Scattergrams" (1946)	0.4550

Global Text Matching for Information Retrieval

GERARD SALTON* AND CHRIS BUCKLEY

An approach is outlined for the retrieval of natural language texts in response to available search requests and for the recognition of content similarities between text excerpts. The proposed retrieval process is based on flexible text matching procedures carried out in a number of different text environments and is applicable to large text collections covering unrestricted subject matter. For unrestricted text environments this system appears to outperform other currently available methods.

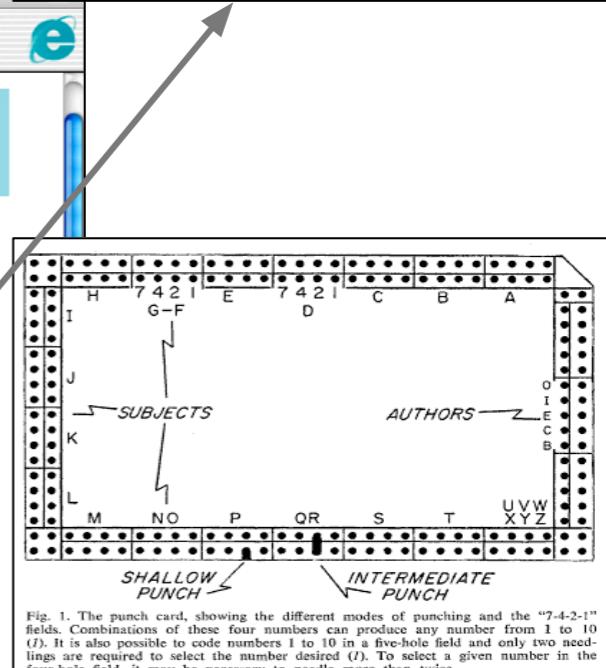


Fig. 1. The punch card, showing the different modes of punching and the "7-4-2-1" fields. Combinations of these four numbers can produce any number from 1 to 10 (J). It is also possible to code numbers 1 to 10 in a five-hole field and only two needlings are required to select the number desired (I). To select a given number in the four-hole field, it may be necessary to needle more than twice.

THE STORING OF PAMPHLETS.

ON reading Professor Minot's explanation of his method of storing pamphlets as given in the issue of December 30th I feel inclined to add a word in commendation of the method. I began using these boxes six or seven years ago and now have 152 upon my shelves. About one-half are devoted to Experiment Station bulletins, the boxes being labeled by States and arranged alphabetically. The other half is used for miscellaneous pamphlets on subjects pertaining to my line of work. The boxes have proved perfectly satisfactory in every way, and as a simple time-saving device they are worth many times the cost. My system of pamphlet arrangement differs in some ways from that adopted by Professor Minot and has been adopted only after trial of several other methods.



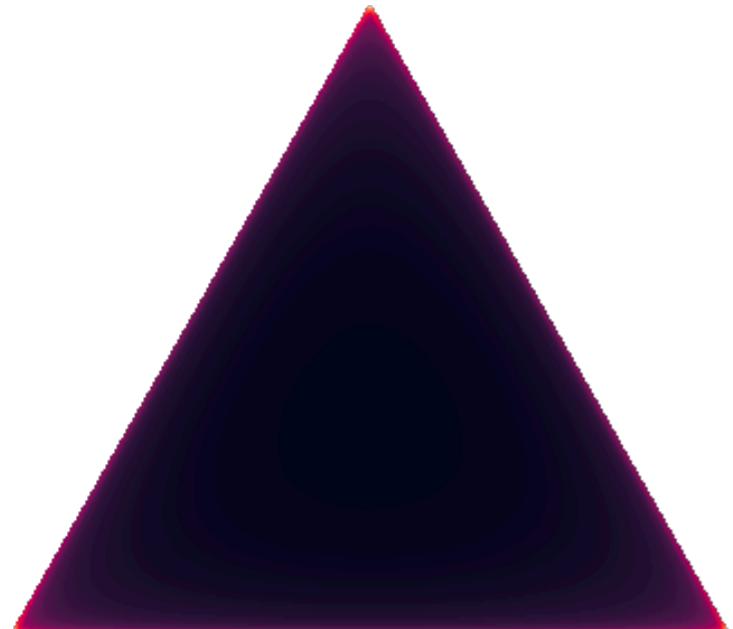
The Dirichlet distribution

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

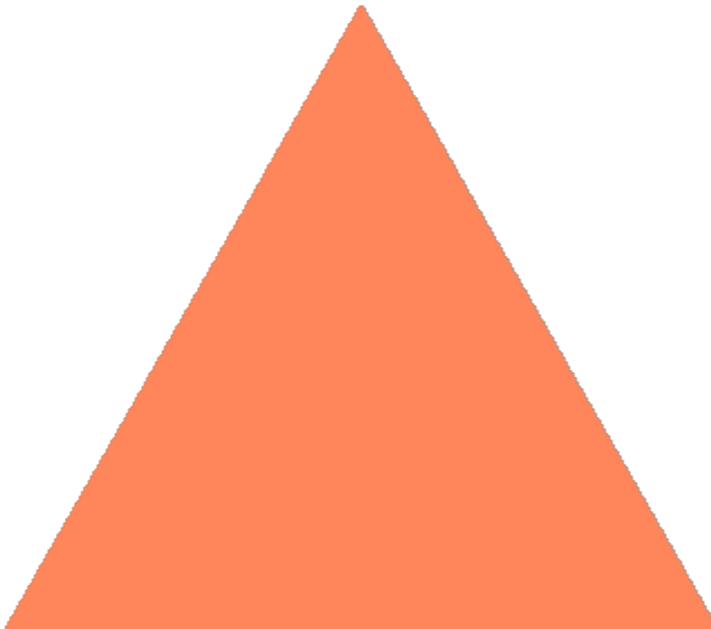
$$p(\theta | \vec{\alpha}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_i \theta_i^{\alpha_i - 1}.$$

- 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.

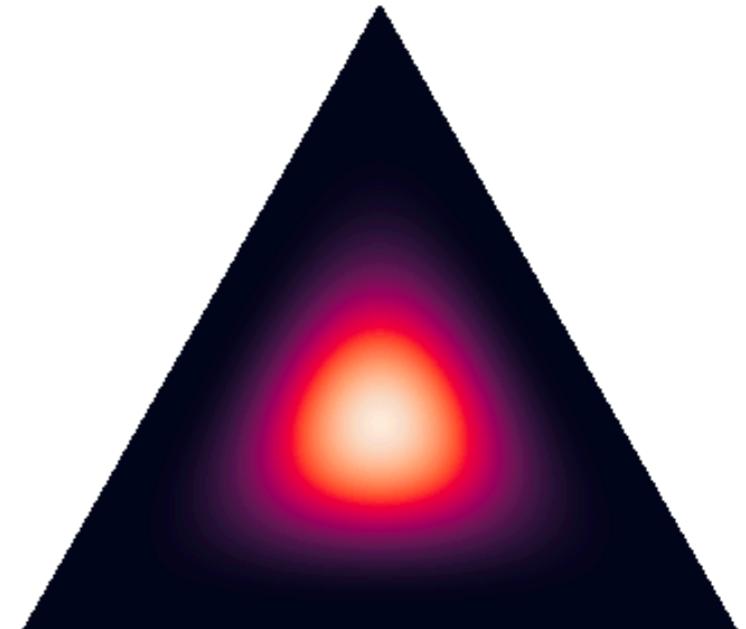
$(0.85, 0.85, 0.85)$



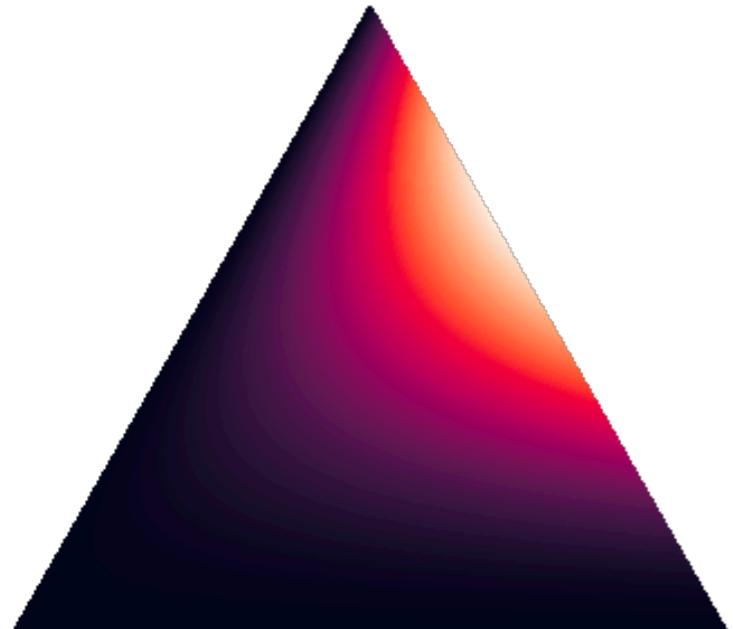
$(1, 1, 1)$



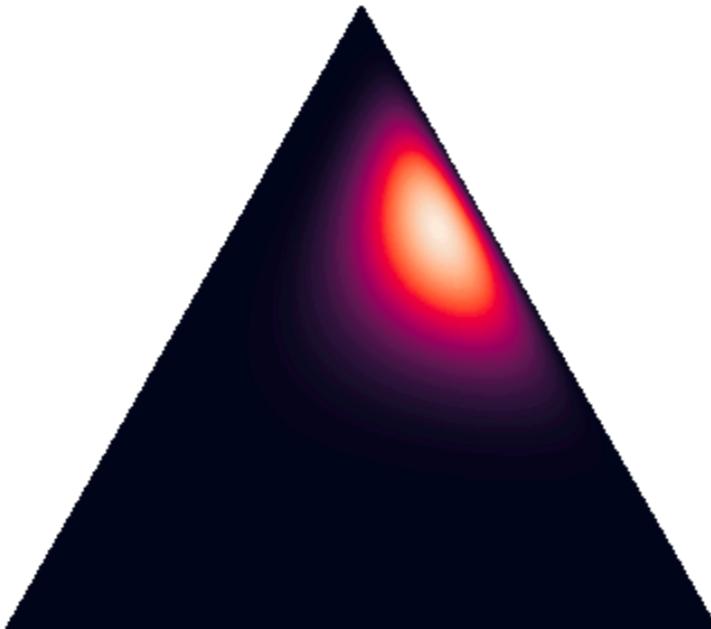
$(5, 5, 5)$



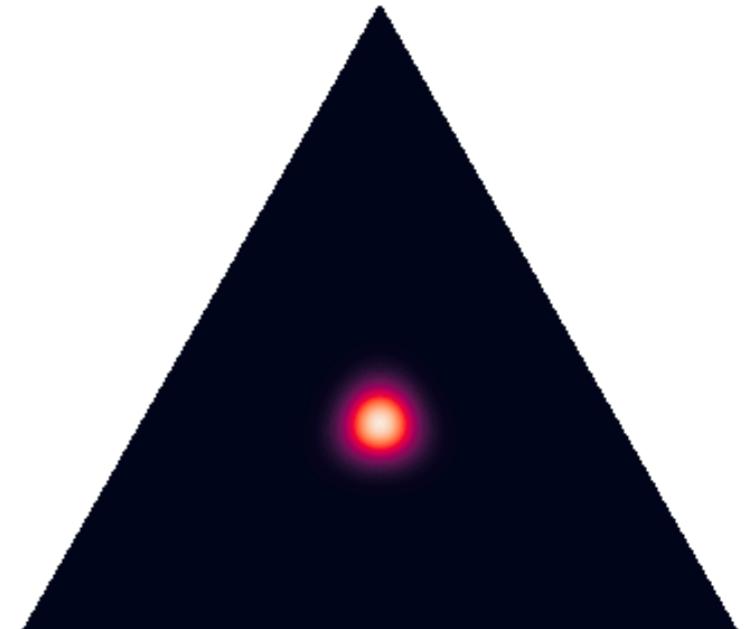
$(1, 2, 3)$



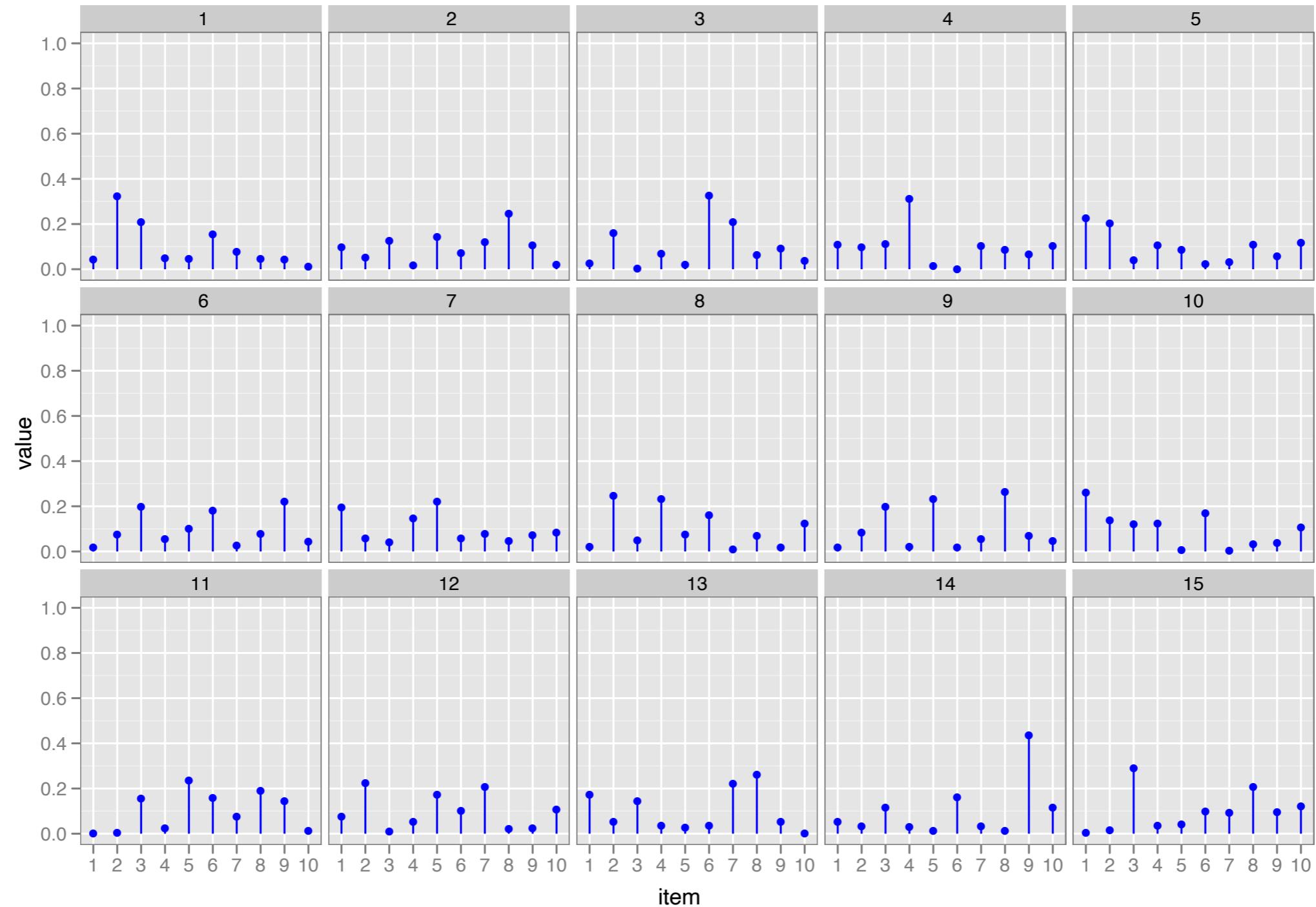
$(2, 5, 10)$



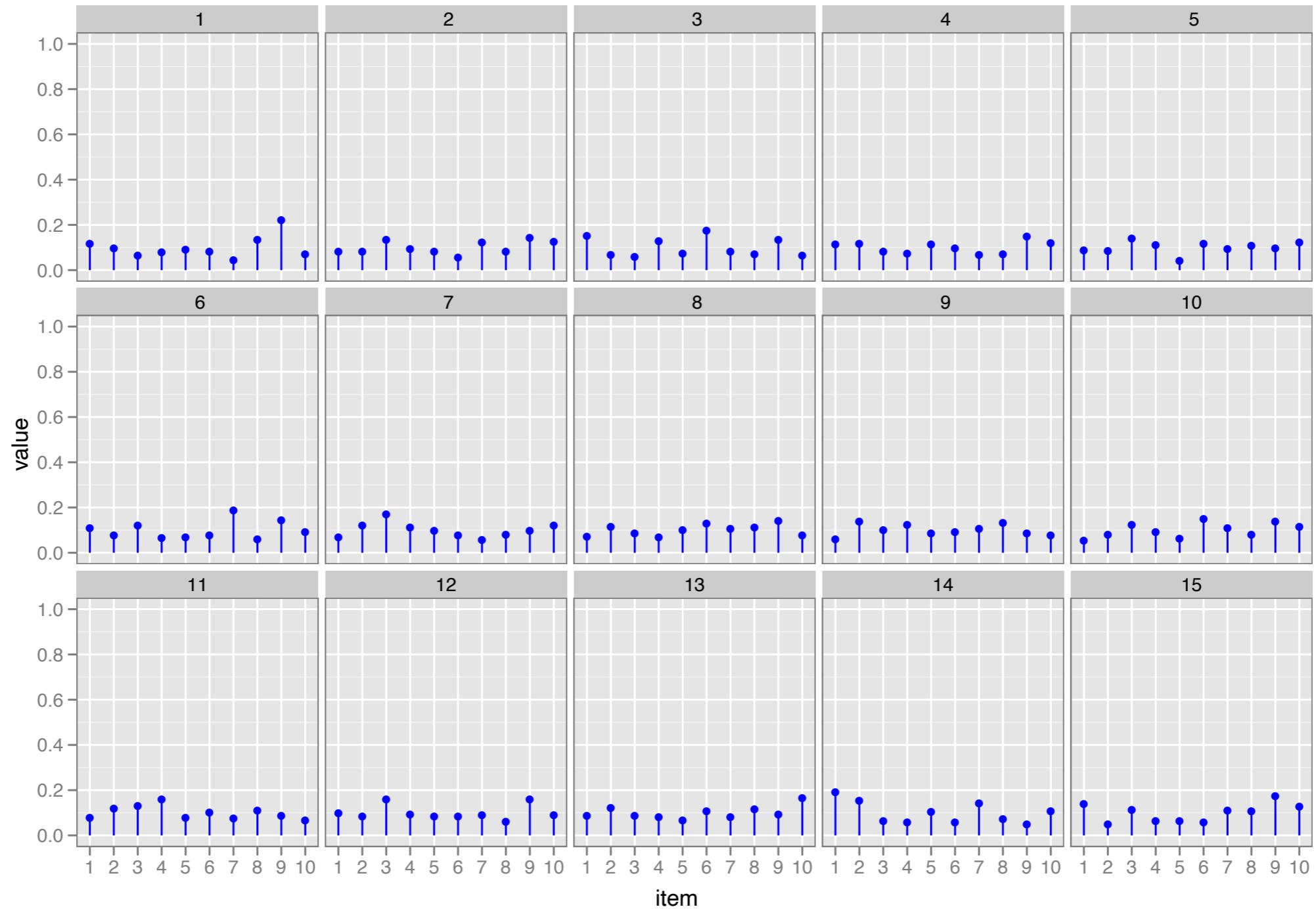
$(50, 50, 50)$



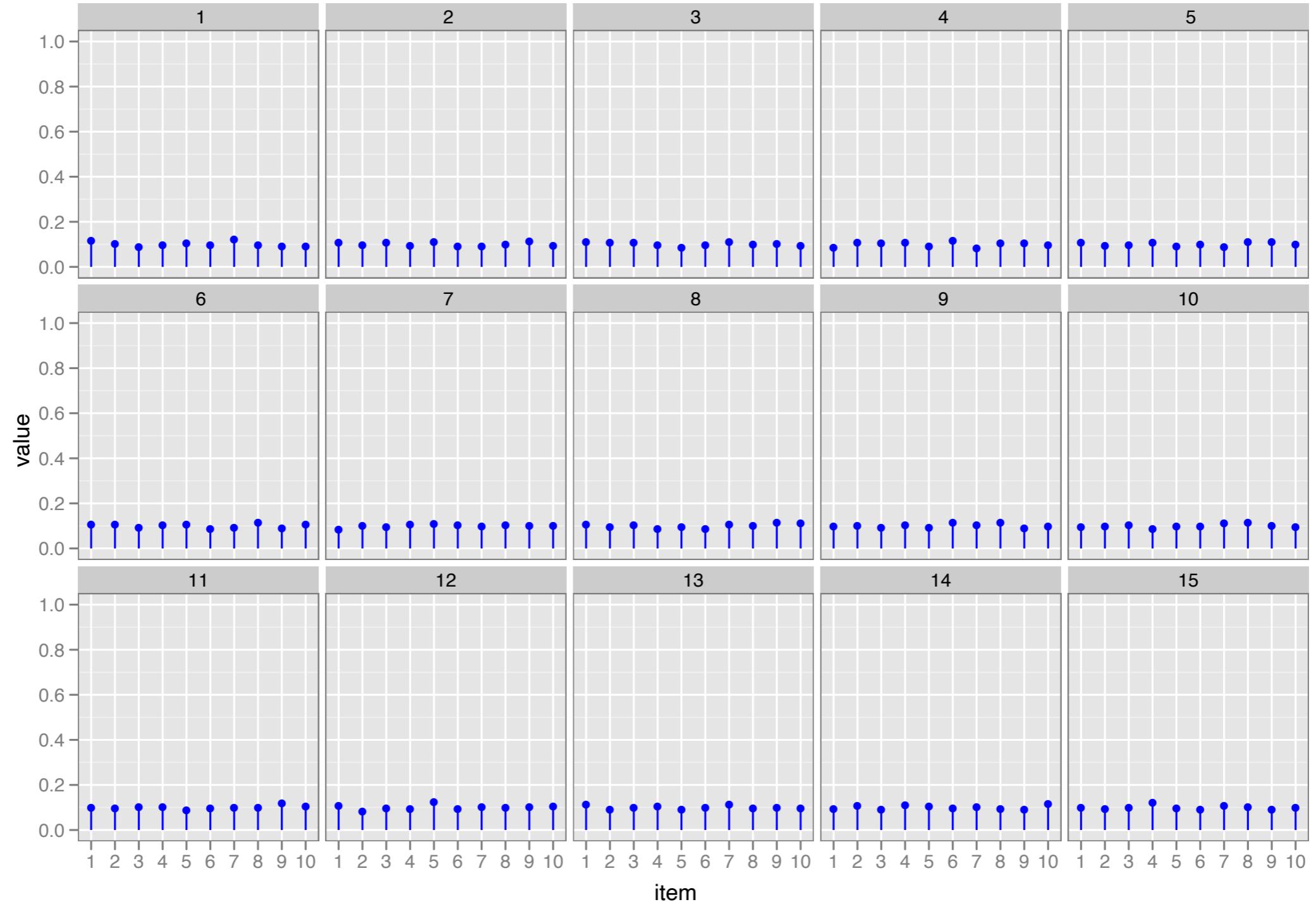
$\alpha = 1$



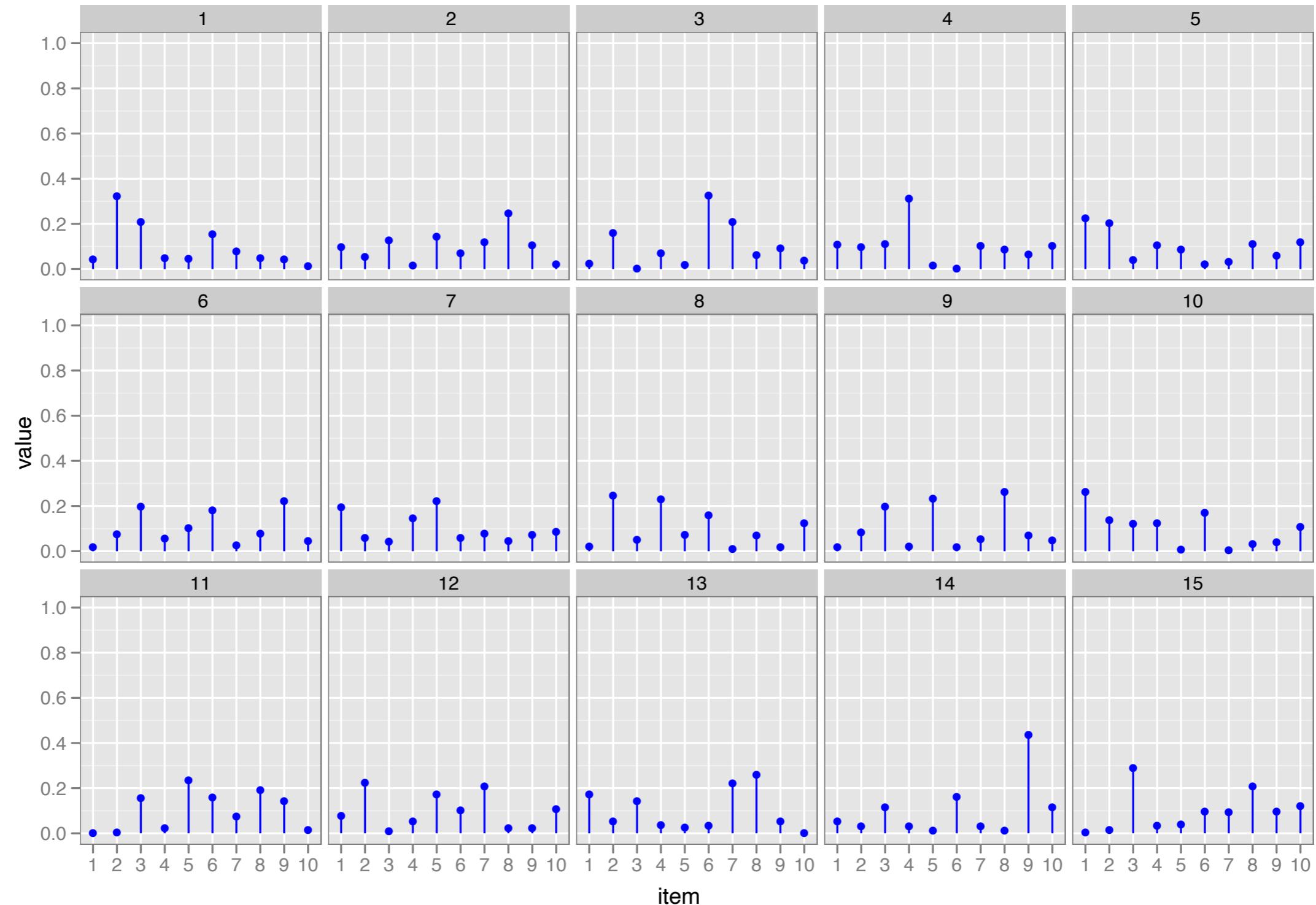
$\alpha = 10$



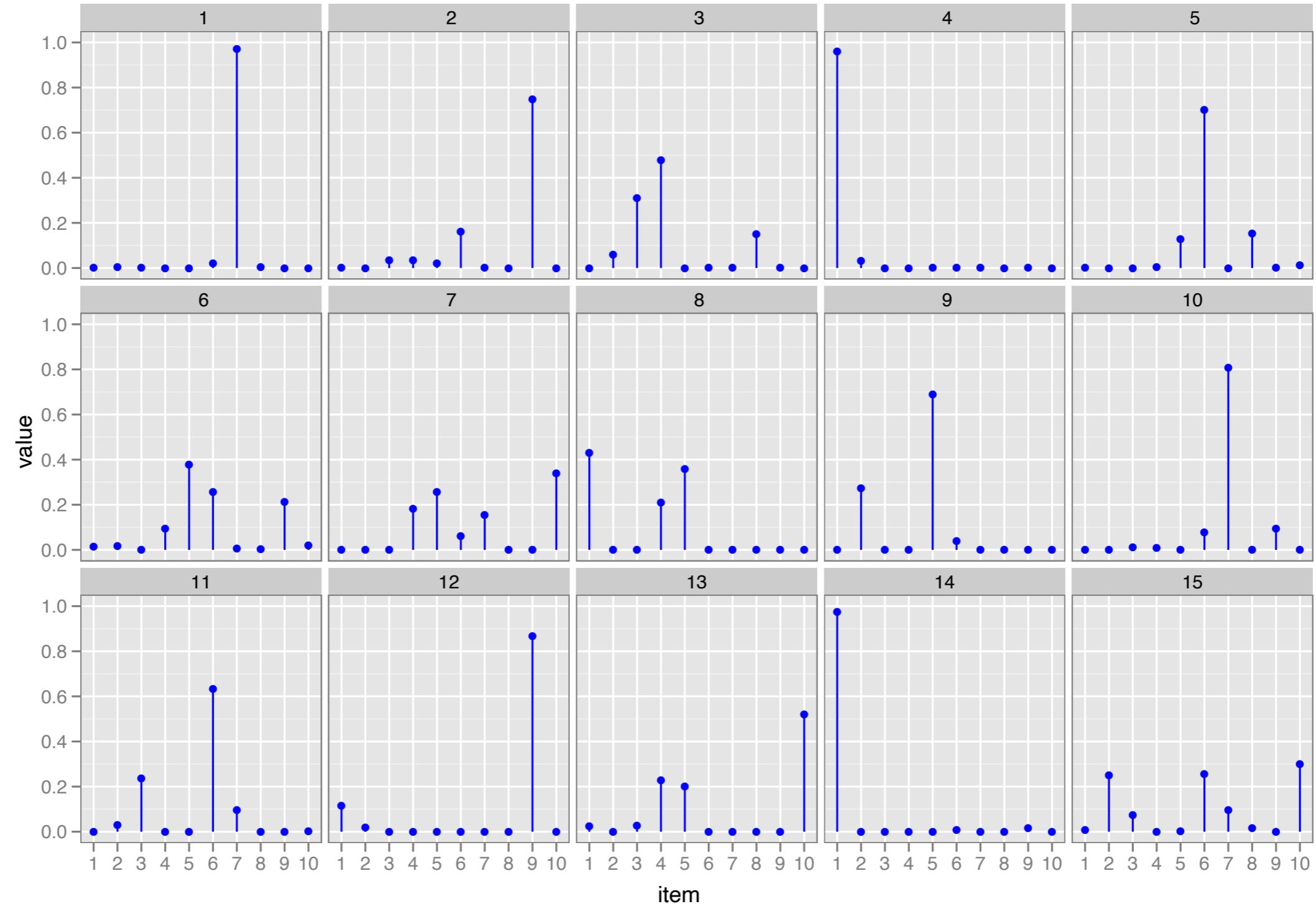
$$\alpha = 100$$



$\alpha = 1$



$$\alpha = 0.1$$



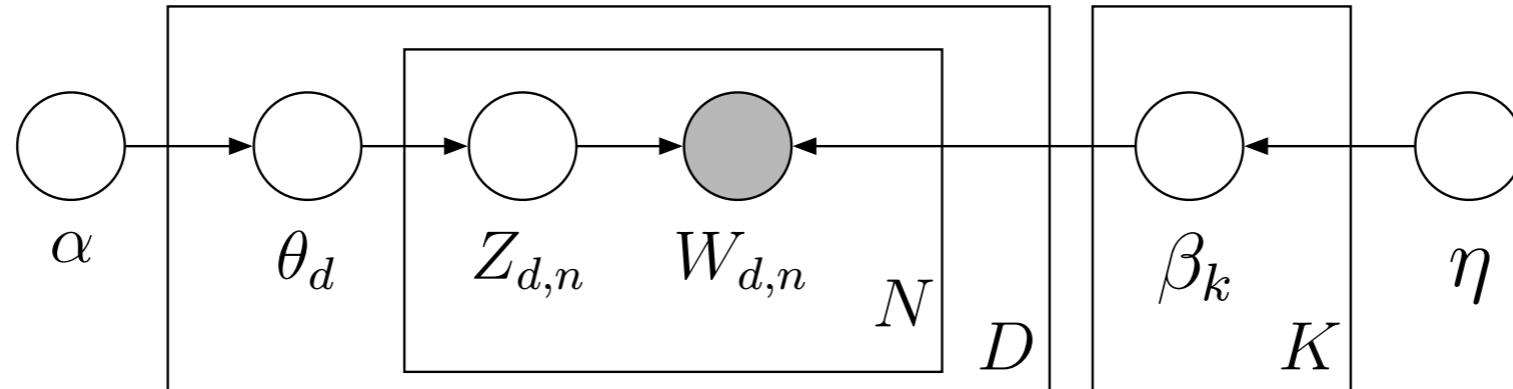


Why LDA works?

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

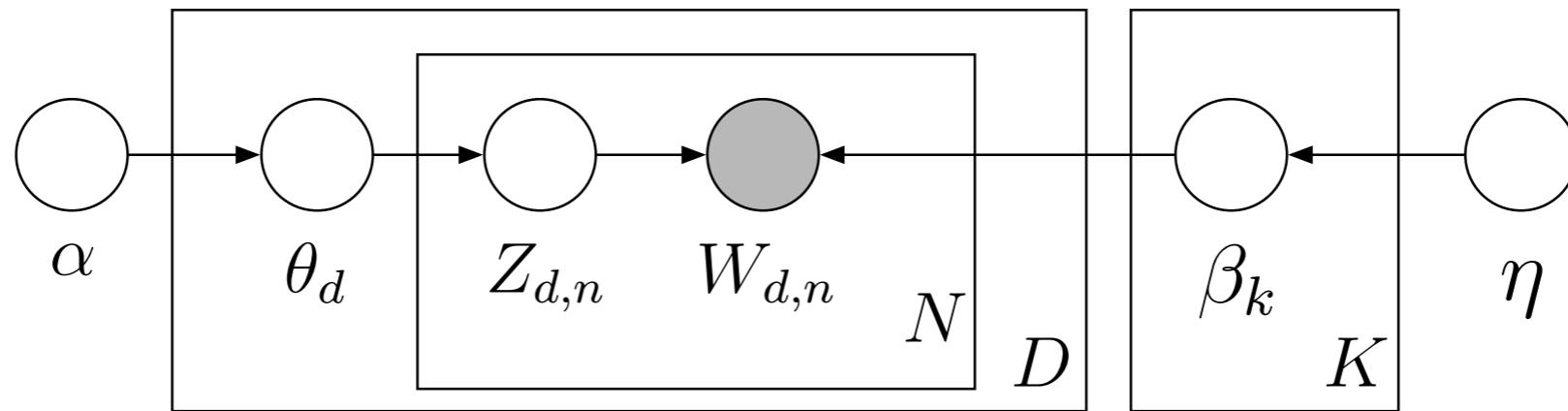


- The joint distribution of the latent variables and documents is

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

- The posterior of the latent variables given the documents is

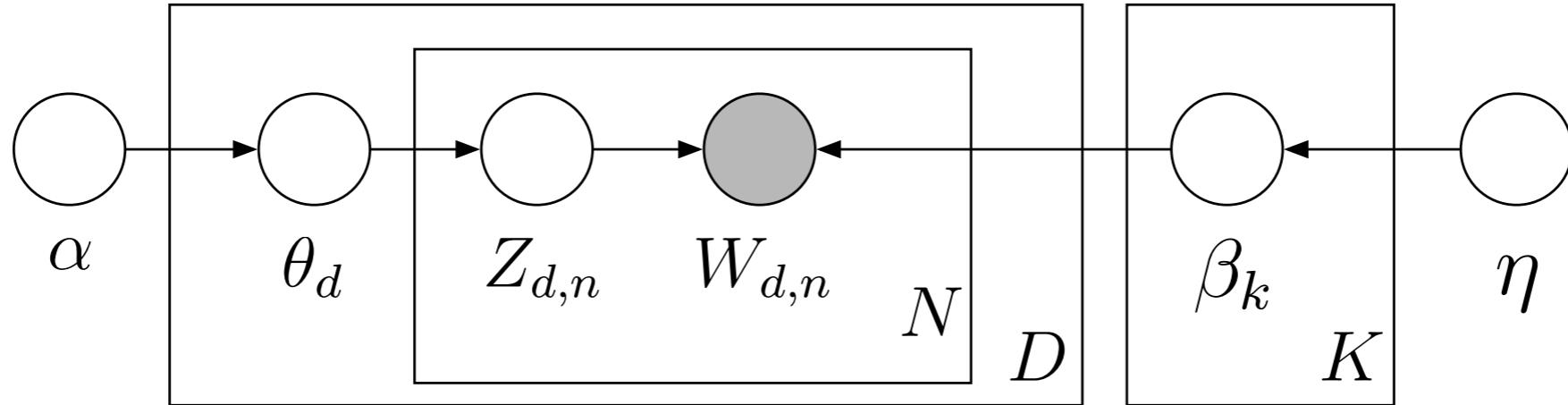
$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D,1:N} | w_{1:D,1:N}).$$



- 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.



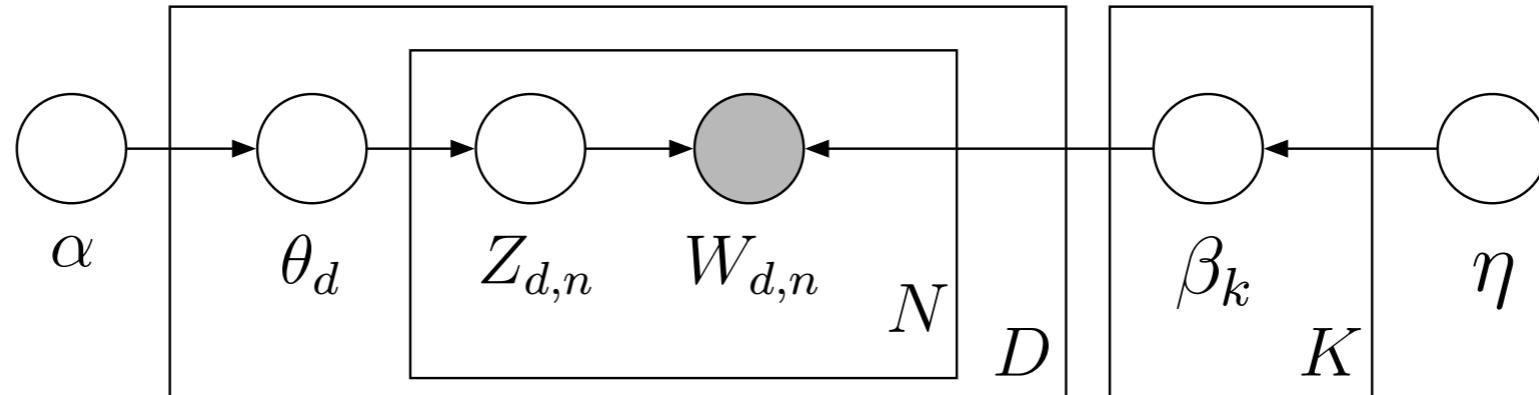
- There is a large literature on approximating the posterior.
- We will focus on
 - Gibbs sampling
 - Mean-field variational methods (batch and online)



MCMC

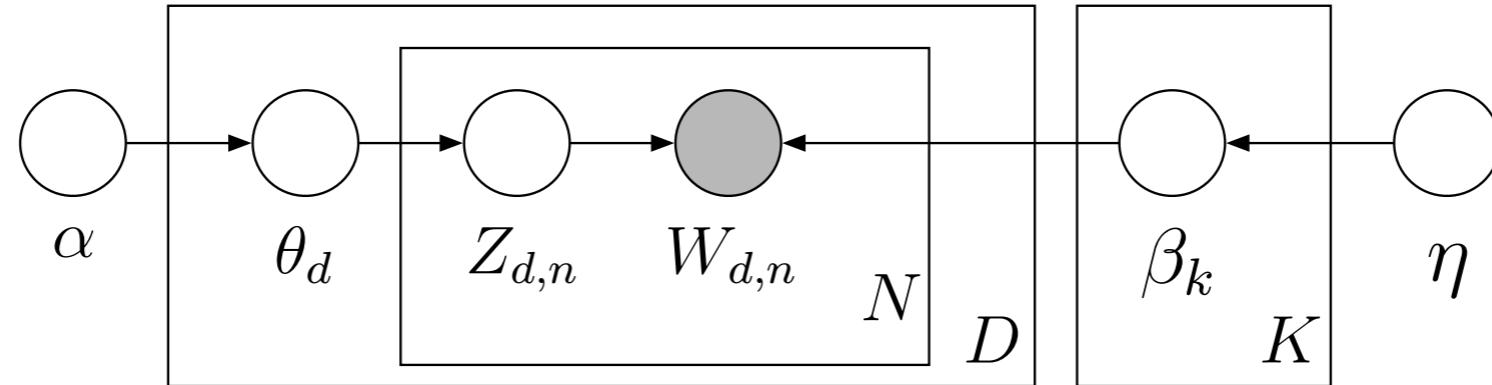
- Construct a **Markov chain** on the hidden variables, whose limiting distribution is the posterior.
- Collect **independent samples** from that distribution; approximate the posterior with them
- In **Gibbs sampling** the chain is defined by the conditional distribution of each hidden variable given 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 assignments $Z_{d,n}$
- Global variables are shared by the corpus
 - Topics β_k

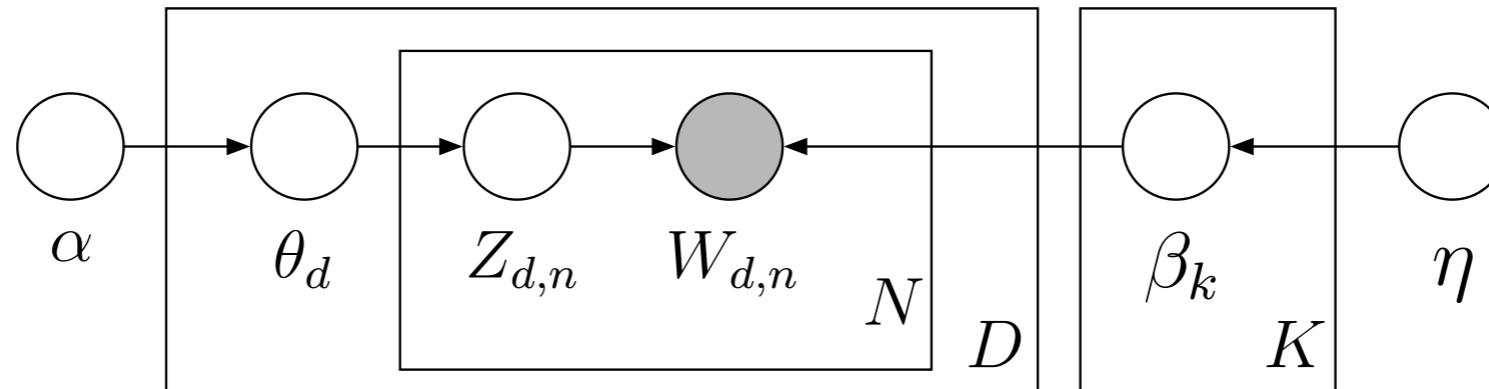
Local and global variables



- 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

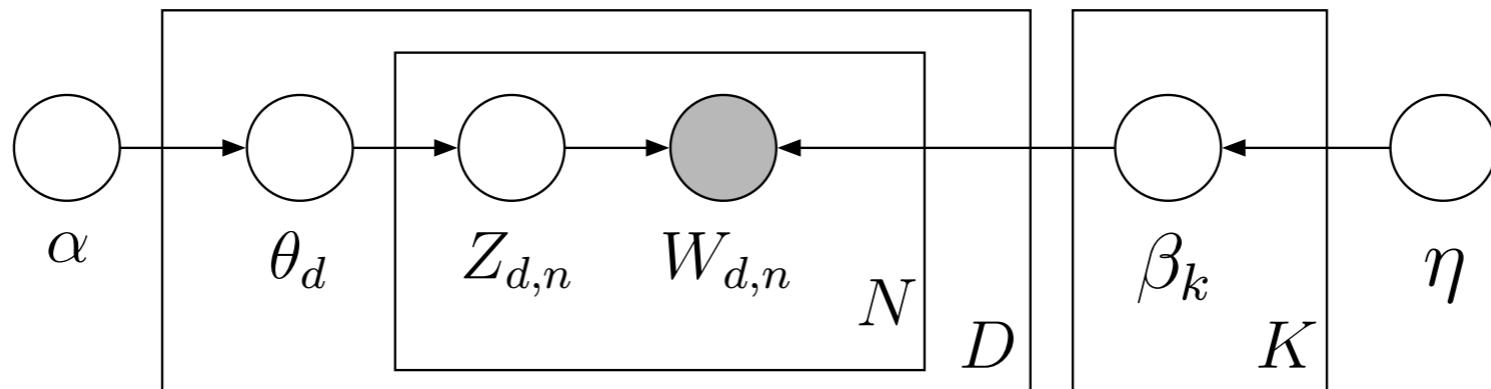


- We observe words $\mathbf{w} = w_{1:N}$. The Markov chain is defined on $\{\theta, z_{1:N}\}$, the topic proportions and topic assignments.
- Some notation—

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

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

Local Gibbs sampling for LDA



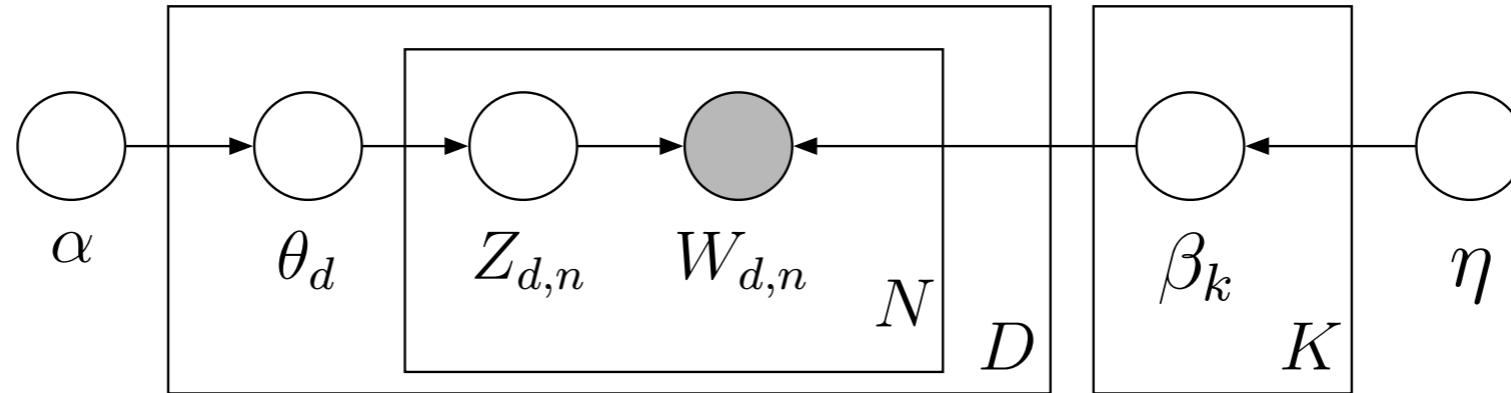
A simple Gibbs sampler is

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

where

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

Collapsed local Gibbs sampling



- The topic proportions θ can be integrated out,

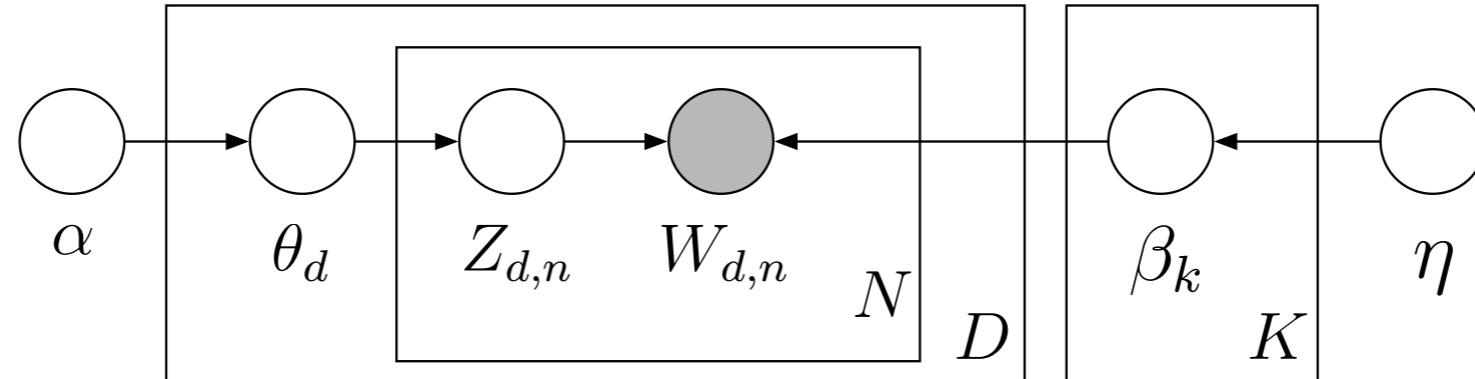
$$p(z_n | z_{-n}, \mathbf{w}) = p(w_n | \beta_{1:K}) \cdot \int_{\theta} p(z_n | \theta) p(\theta | z_{-n}) d\theta$$

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

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

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

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, θ_d) for each document.
- Then, sample each topic from

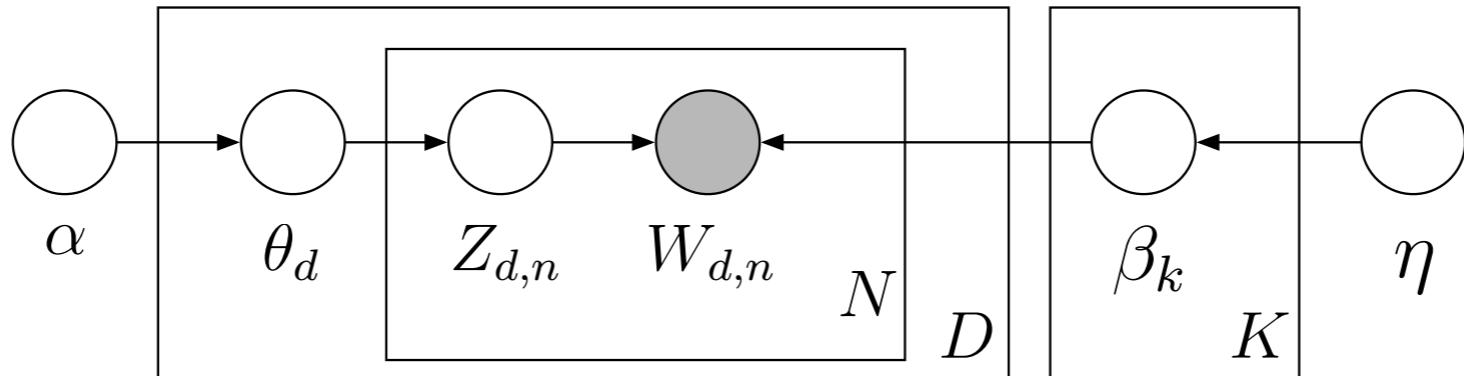
$$\beta_k \mid \mathbf{z}_{1:D}, \mathbf{W} \sim \text{Dir}(\lambda_k),$$

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 assignments $\mathbf{z}_{1:D}$,

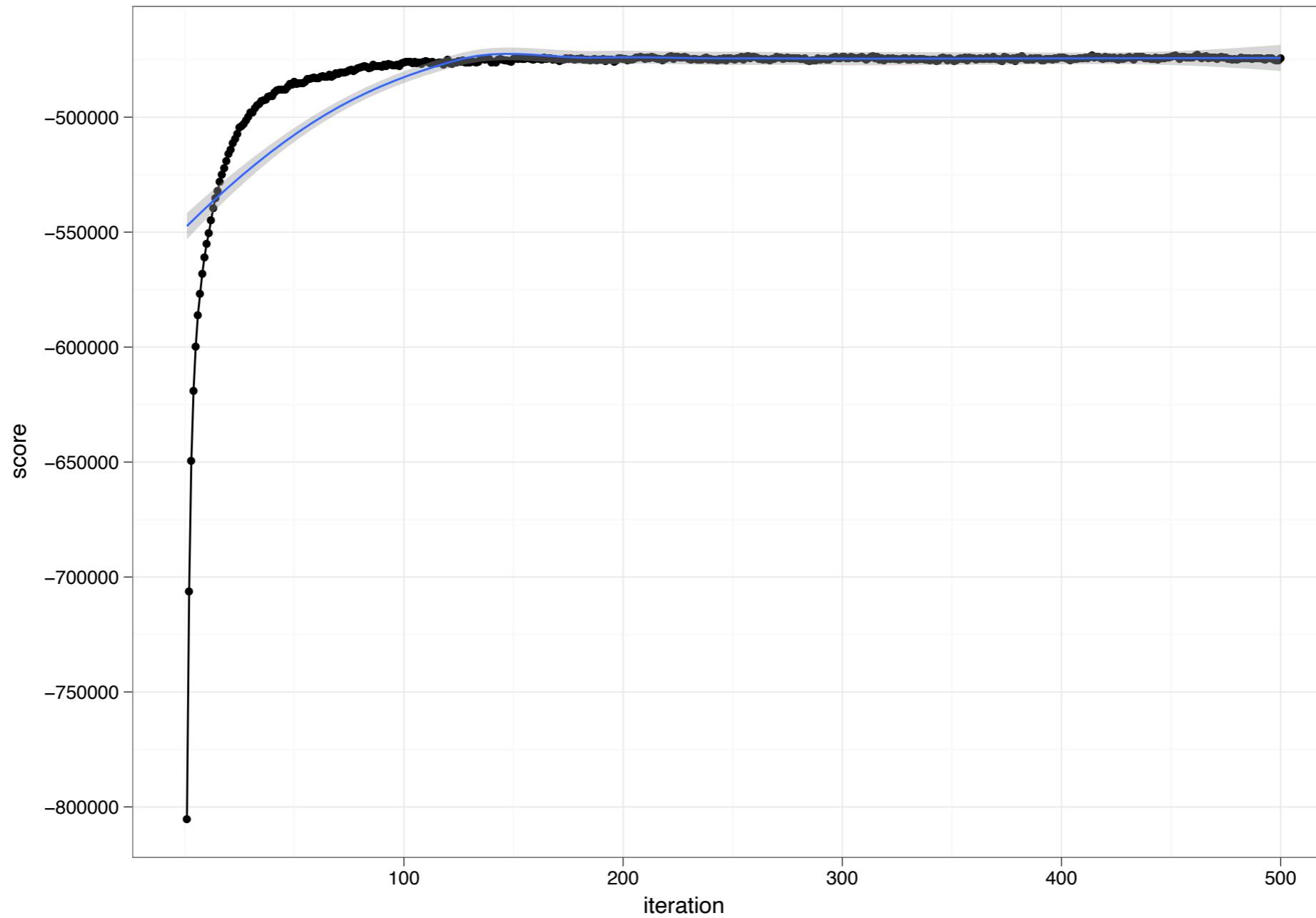
$$p(z_{n,d} = k | \mathbf{z}_{-(n,d)}, \mathbf{W}) \propto \left(\frac{m_k(\mathbf{z}_{-(n,d)}, \mathbf{W}) + \eta}{\sum_v m_k^v(\mathbf{z}_{-(n,d)}) + V\eta} \right) (n_k(z_{-i}) + \alpha)$$

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

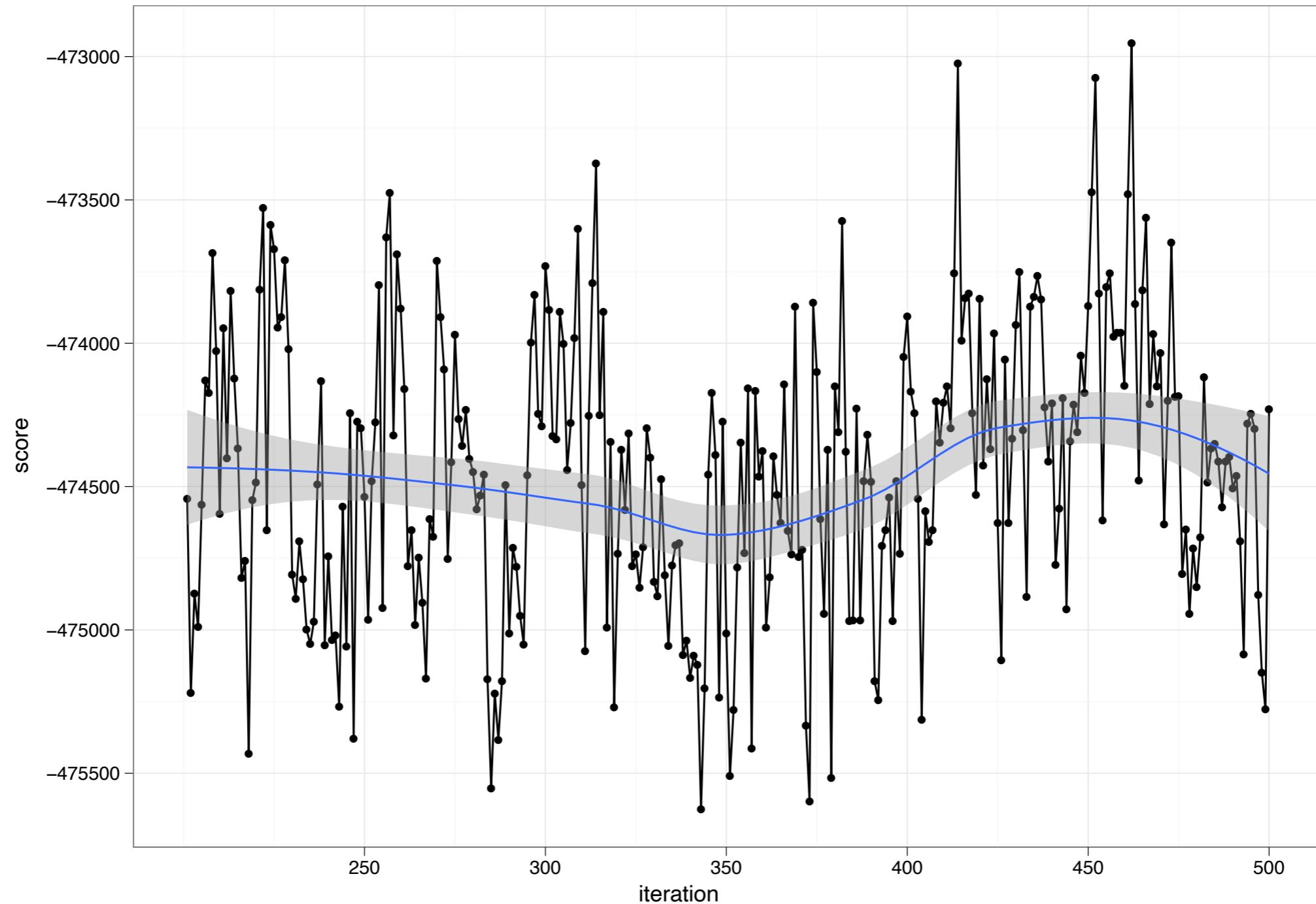
Gibbs sampling for LDA in practice

- In practice:
 - ① Obtain a corpus of documents \mathbf{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

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

- 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)

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

(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 $p(z | x)$.

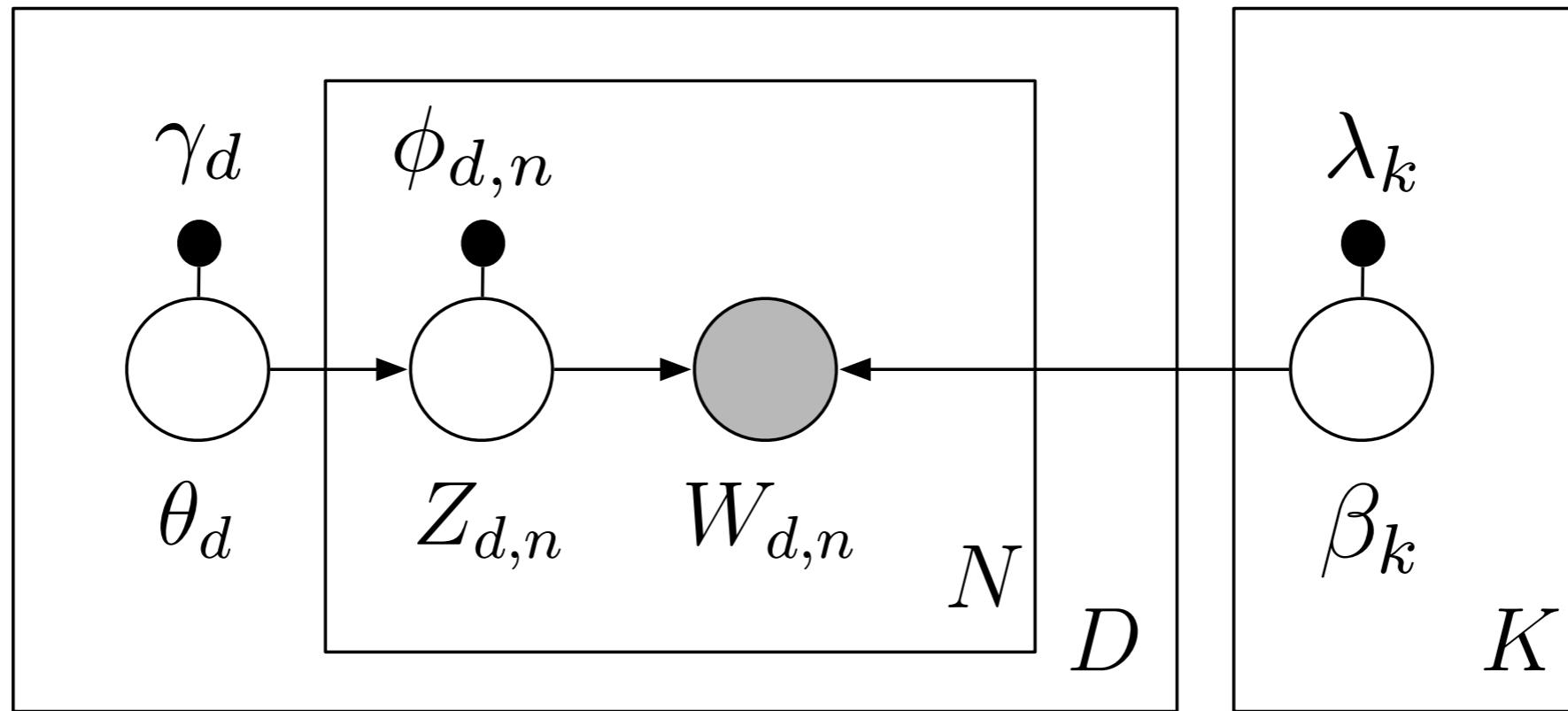
Mean-field variational inference

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

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

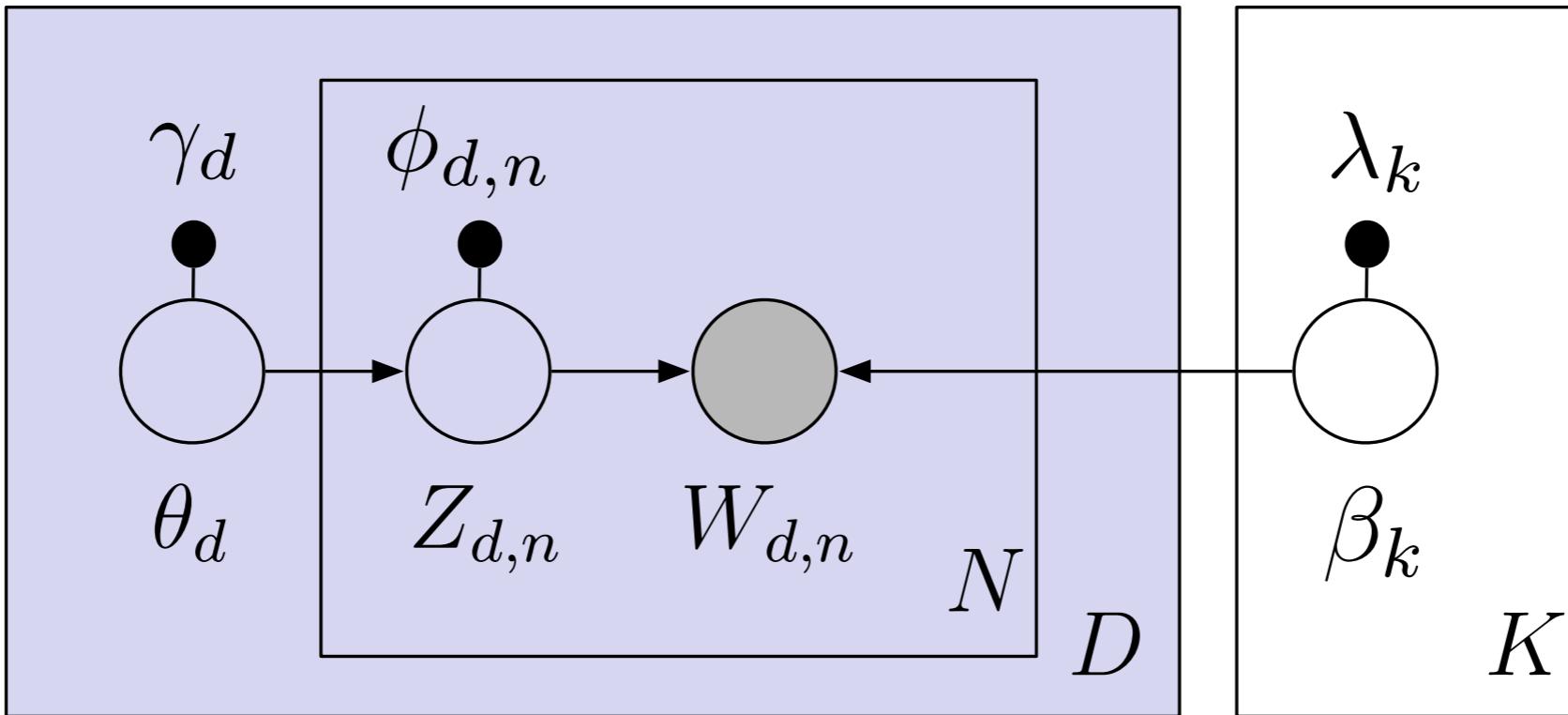
- Each latent variable is independently governed by its own variational parameter ν_m .
- In the true posterior they can exhibit dependence.
(Often, this is what makes exact inference difficult.)

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.

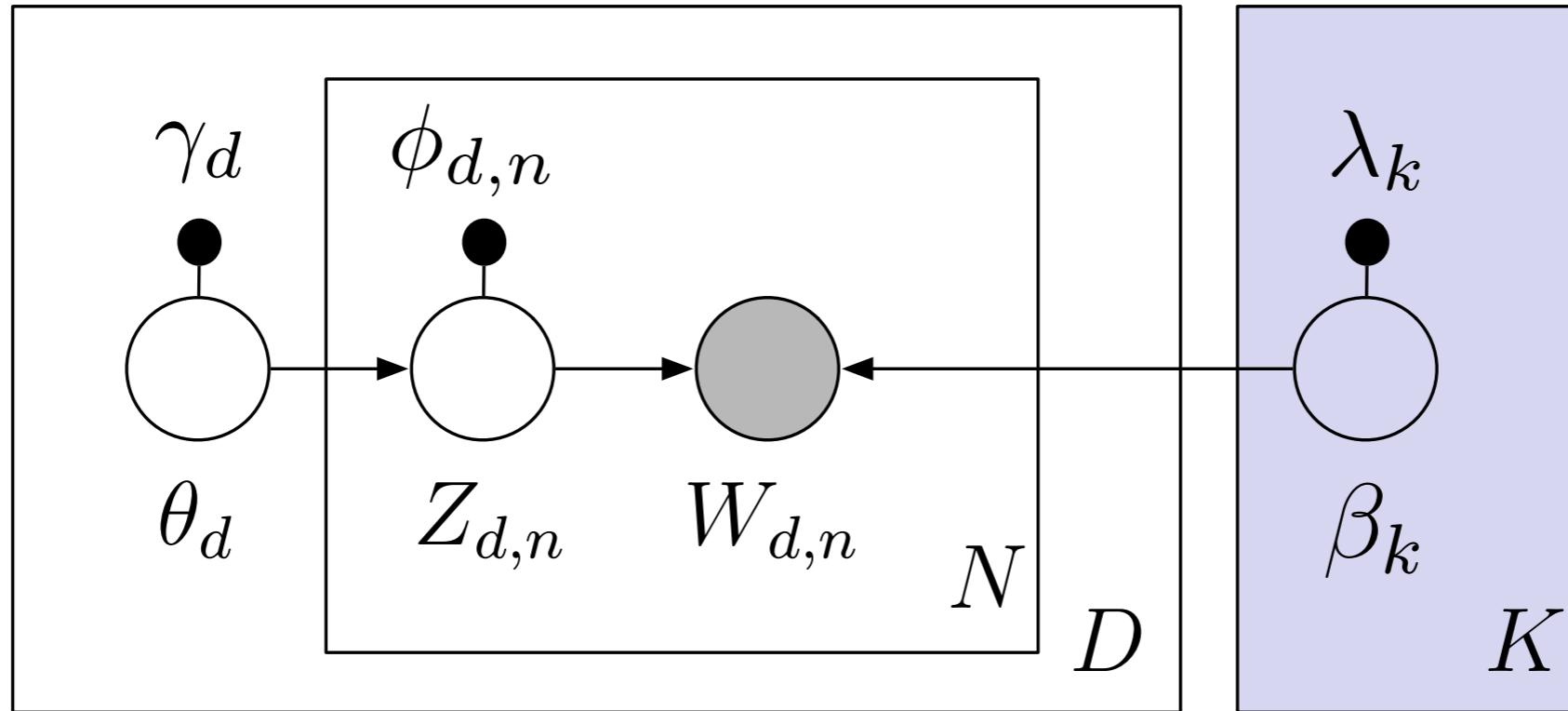
Variational inference for LDA



- 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  $\lambda_{1:K}$  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
```

“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
```

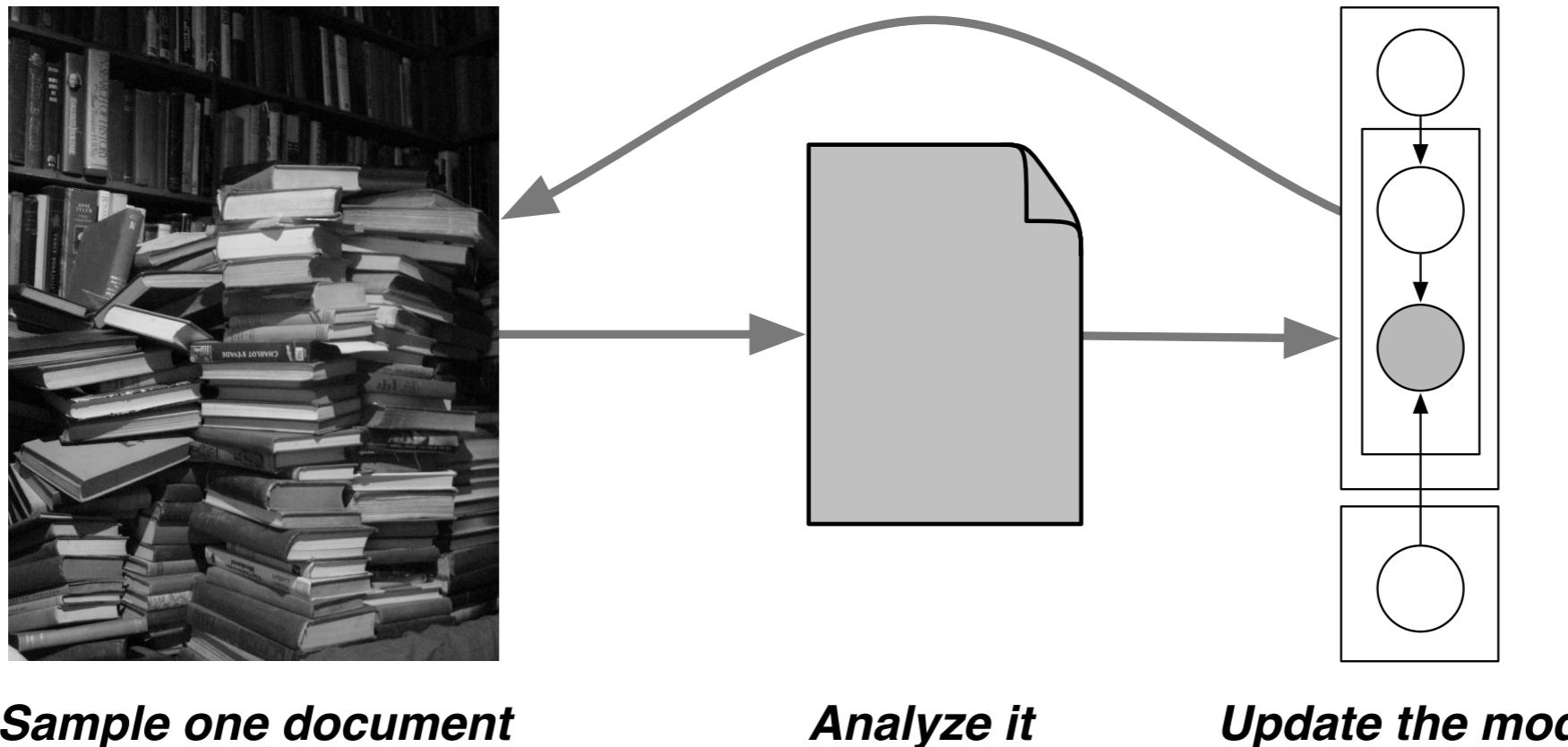
Do variational inference for each document.

“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  $\frac{1}{K} \sum_k |\text{change in } \gamma_{d,k}| < \epsilon$   
9:   end for  
10:  Set  $\lambda_k = \eta + \sum_d \sum_n w_{d,n} \phi_{d,n}$   
11: end while
```

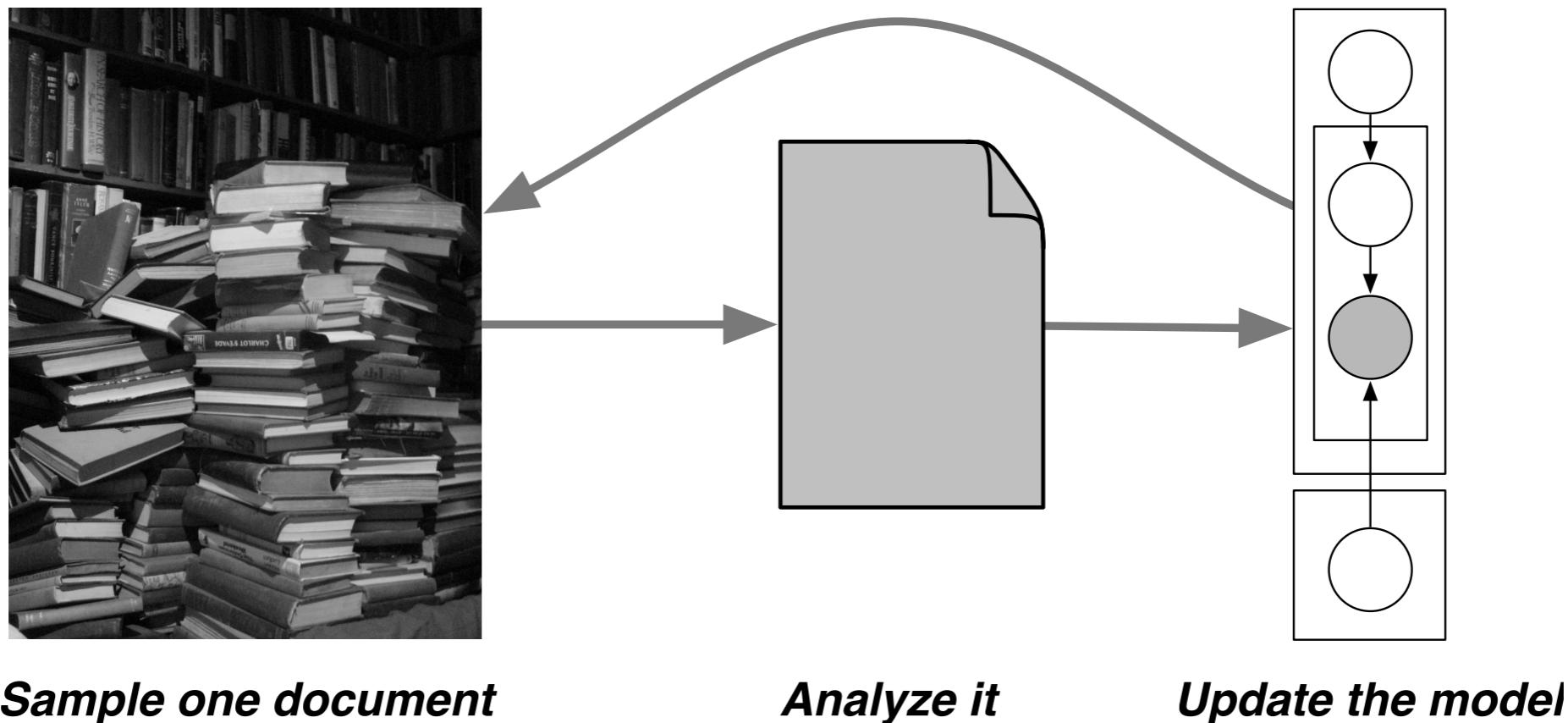
Update the posterior estimates of the topics based on the “E step.”

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



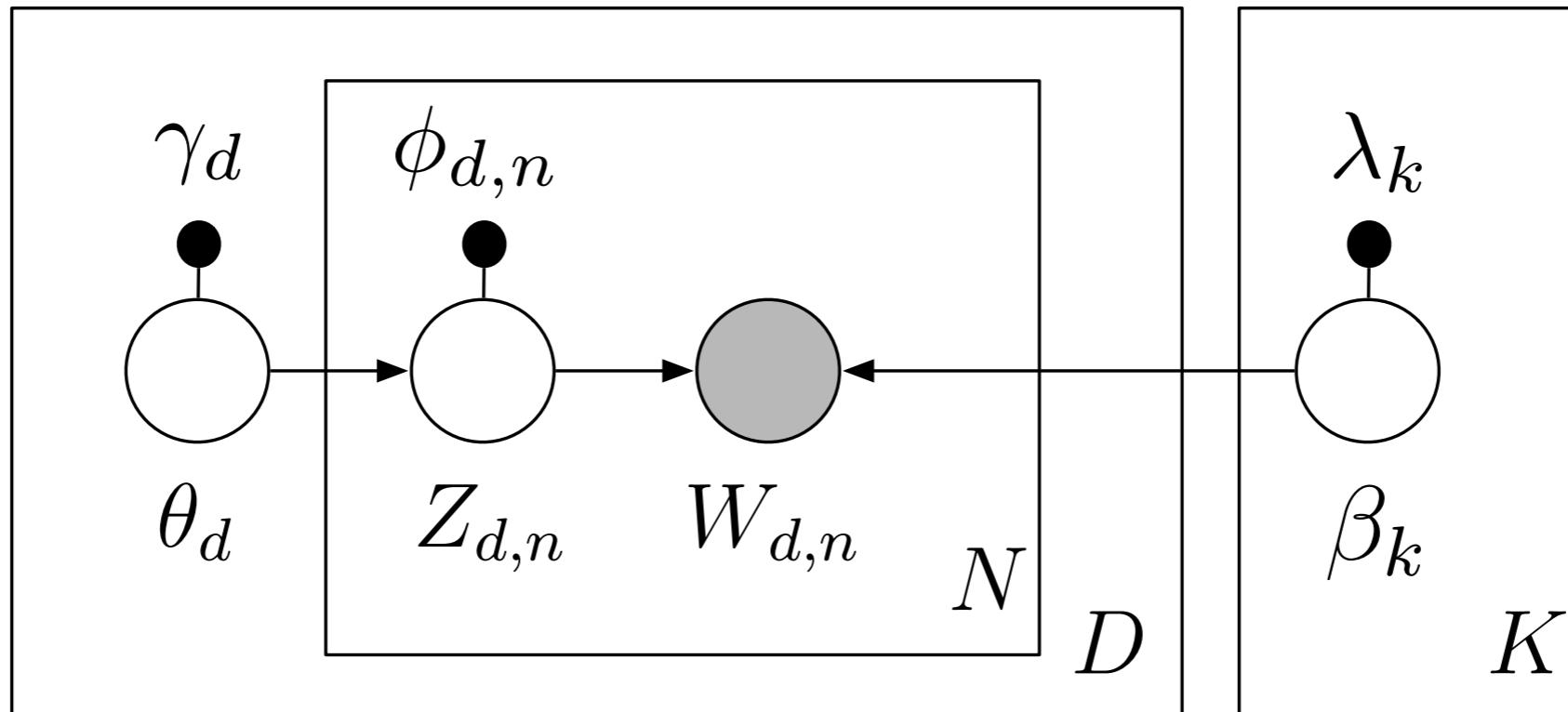
- Our goal is to use this (and related) models for analyzing massive collections of millions of documents.
- But, in the first step of batch inference we estimate the posterior for *every document* based on randomly initialized topics.

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



- 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.

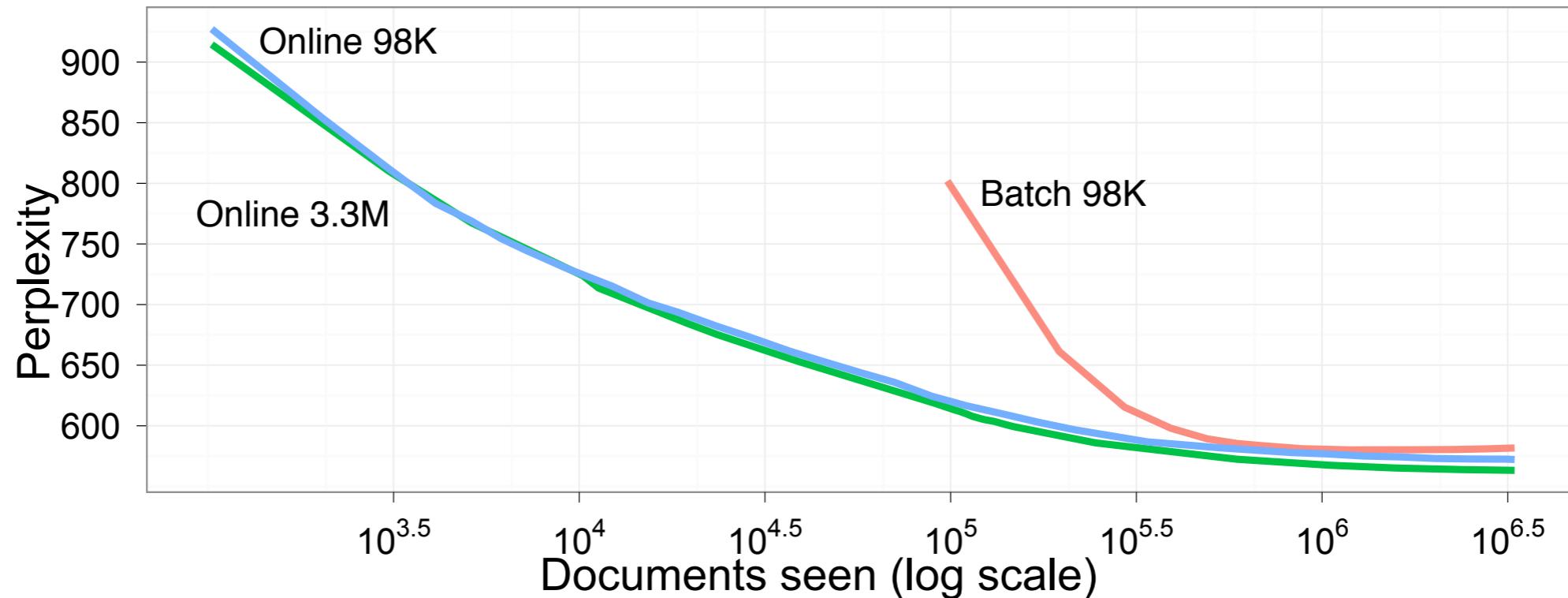
Online variational inference for LDA (sketch)

- 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

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

Analyzing 3.3M articles from Wikipedia



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

Why does this work?

A STOCHASTIC APPROXIMATION METHOD¹

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, \dots 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/~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/