Unsupervised Learning and Topic Modeling

Yoav Goldberg

Bar Ilan University

(with slides from David Blei, Zornitsa Kozerava)

Previously

Supervised Learning

- Get labeled training data
- Represent data as (features, label) pairs
- Train a classifier / model to predict labels based on features

Today

- What if we don't have training data?
- Can we still do something useful?

Unsupervised Learning

Things we can do without labeled data

Unsupervised Methods

Option 1: “Naturally occurring” labels / bootstrap

- Be creative and find data which can be used as labels.
  - e.g., **we want to identify paragraphs.** Maybe some website indicate this via their HTML tags?
- Automatically create your own training set
  - Write simple rule-based system to collect easy examples
    - high precision, low recall
  - Use the easy examples as training data
    - Hope it will generalize well.
    - Careful not to overlap your features with the rules too much!

Unsupervised Methods

Option 1: “Naturally occurring” labels / semi-supervised

- Be creative and find data which can be used as labels.
  - **Want to identify sentiment?** Look at tweets with happy and sad emojis.

Unsupervised Methods

Option 1: “Naturally occurring” labels / semi-supervised

- Be creative and find data which can be used as labels.
  - **Want to identify sentiment?** Look at tweets with happy and sad emojis.
  - what are the pros and cons here?

Unsupervised Methods

Option 1: “Naturally occurring” labels / semi-supervised

- Be creative and find data which can be used as labels.
  - **Want to identify sentiment?** Look at tweets with happy and sad emojis.
  - what are the pros and cons here?
- Can also use the *proxy* naturally occurring data for representation learning.
  - The Felbo et al 2017 paper. (next slides)

Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm

Bjarke Felbo[1], Alan Mislove[2], Anders Søgaard[3], Iyad Rahwan[1], Sune Lehmann[4]

Use Emoji Prediction
to learn good representations
for sentiment

Table 2: The number of tweets in the pretraining dataset associated with each emoji in millions.

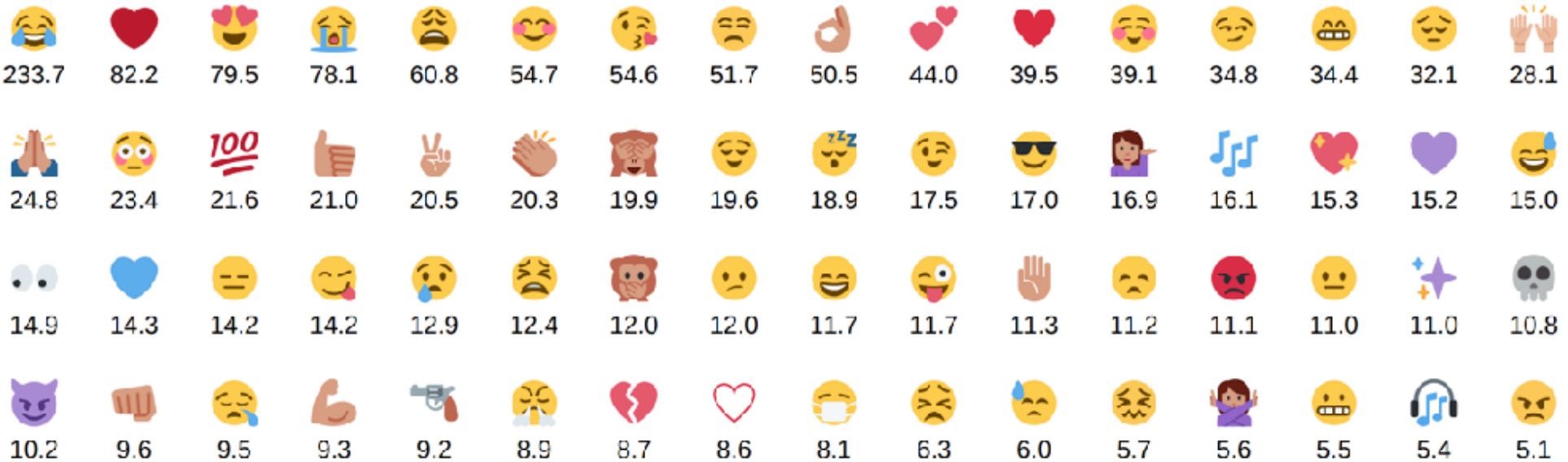


Table 1: Example sentences scored by our model. For each text the top five most likely emojis are shown with the model's probability estimates.

I love mom's cooking	49.1%	8.8%	3.1%	3.0%	2.9%
I love how you never reply back..	14.0%	8.3%	6.3%	5.4%	5.1%
I love cruising with my homies	34.0%	6.6%	5.7%	4.1%	3.8%
I love messing with yo mind!!	17.2%	11.8%	8.0%	6.4%	5.3%
I love you and now you're just gone..	39.1%	11.0%	7.3%	5.3%	4.5%
This is shit	7.0%	6.4%	6.0%	6.0%	5.8%
This is the shit	10.9%	9.7%	6.5%	5.7%	4.8%

why is this useful?

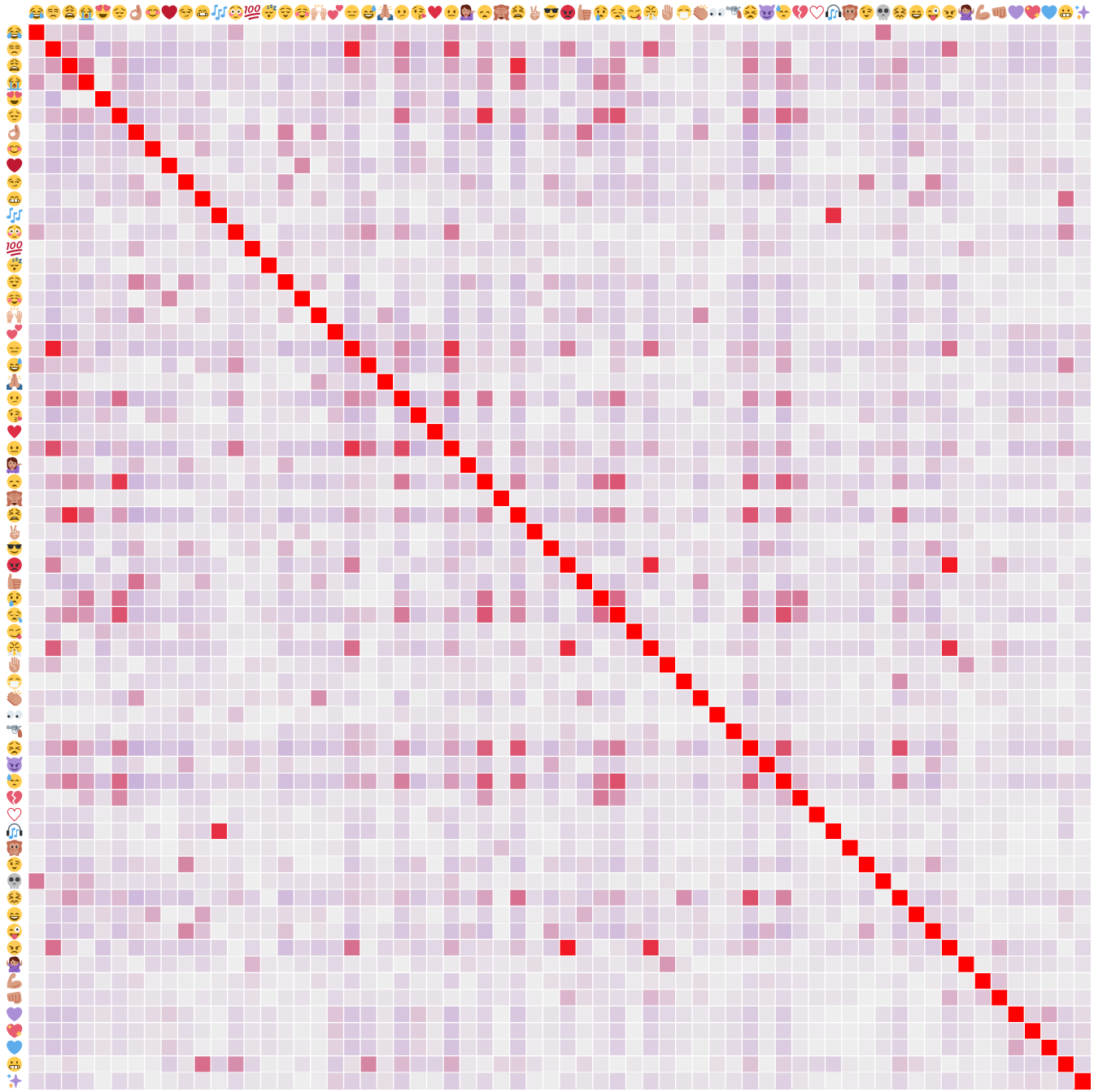


Figure 7: Correlation matrix of the model's predictions on the pretraining test set.

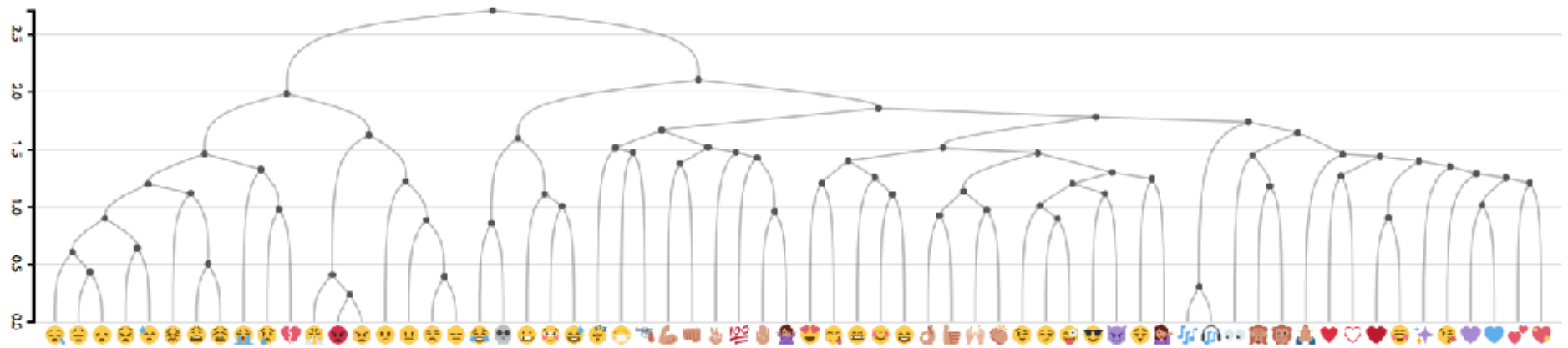


Figure 6: Hierarchical clustering of the DeepMoji model's predictions across categories on the test set. The dendrogram shows how the model learns to group emojis into overall categories and subcategories based on emotional content. The y-axis is the distance on the correlation matrix of the model's predictions measured using average linkage.

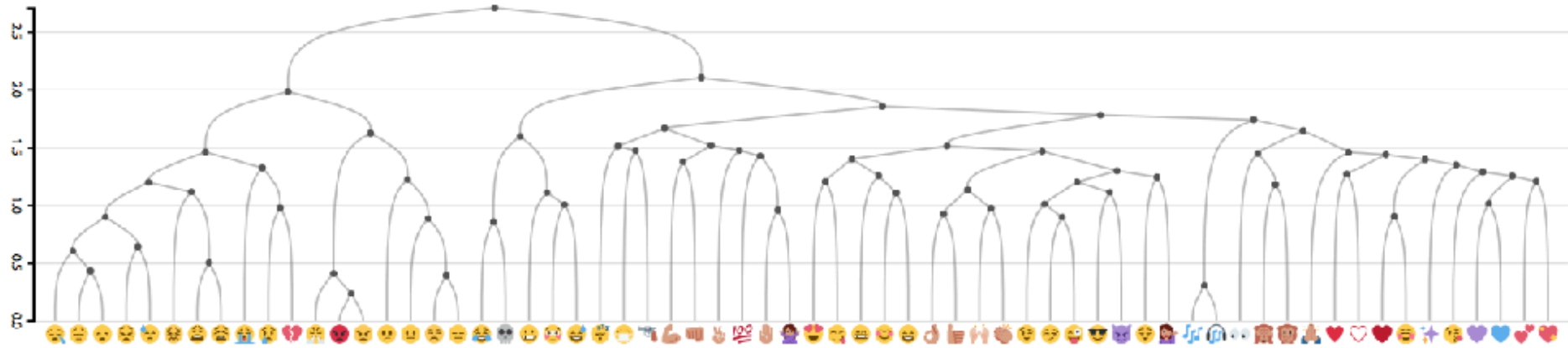


Figure 6: Hierarchical clustering of the DeepMoji model's predictions across categories on the test set. The dendrogram shows how the model learns to group emojis into overall categories and subcategories based on emotional content. The y-axis is the distance on the correlation matrix of the model's predictions measured using average linkage.

Predicting a large set can be indicative of more coarse-grained trends.

model

(main idea. extra details in paper.)

- Train RNN (LSTM) to predict emojis based on a tweet.
  - Result: encoder that takes a tweet and returns a **vector** which is useful for predicting emojis.
- Take (smaller) sentiment dataset.
  - Encode sentences to vectors using above encoder.
  - Train to predict sentiment from vectors.

Unsupervised Methods

Option 2: Write and algorithm and hope it works

- Example: assignment 3.
  - Represent words by their contexts
  - Define the co-occurrence metric (PMI, word2vec)
  - Define similarity measure (cosine)
  - Use this to get a useful result – lists of similar words
- Can be very effective
- But no “learning” involved.
- What to do when this doesn’t work?

Unsupervised Methods

Option 3: Obtain Cheap / Easy Annotations

- Make easy annotation tasks for humans
  - Pose annotation as natural questions that are easy to answer.
- But how to come up with the right questions?

Unsupervised Methods

Option 3: Obtain Cheap / Easy Annotations

- Measure human behavior
  - Eye-tracking when reading
  - Mouse-movement when reading
  - Keyboard clicks when writing
  - ...
- How can these be leveraged to obtain useful data for learning?

Unsupervised Methods

Option 5: Latent-variable generative modeling

- Define a "generative story" of how the data was generated
  - This story doesn't have to be very convincing or realistic
- The story can include "latent variables", stuff that you would like to see but you don't
  - For example: HMM POS-tagging, where we treat the tags as latent.
- Search for an assignment of latent variables such that the data has high probability under the model.
  - Usually, this search is hard.
  - Approximate!
    - EM
    - **MCMC (Gibbs sampling)**

Example: HMM

- We want to train a POS-tagger, but don't have labeled data.
- We do have a dictionary, associating some words with their possible POS tags, and also a lot of text.
- We will use the dictionary and the text to train a bigram HMM model.

Unsupervised Learning

Example: HMM

The Bigram-HMM generative story:

To generate a tagged sentence $(w, t) = (w_1, \ldots, w_n, t_1, \ldots, t_n)$:

- Start with tag t_0 = START.
- For i in $1, \ldots, n$:
  - Draw a random tag t_i from the transition distribution $P(t_i|t_{i-1})$
  - Draw a random word w_i from the tag distribution $P(w_i|t_i)$

Recall the supervised case

- We observe both the words and the tags.
- We estimate $q = P(t_i|t_{i-1})$ and $e = P(w_i|t_i)$ based on our observations.
- Done

Notation – Discrete Distributions

we say that $X \sim Discrete(\theta, k)$ iff:

- X can get one of k values
- θ is a vector with k entries
- $\theta_i \geq 0$
- $\sum_i \theta_i = 1$
- $P(X = i) = \theta_i$

Example

$p(t_j|t_{j-1})$ is a discrete distribution.

$$t_j \sim Discrete(\theta, |T|)$$

Where:

- $|T|$ is the size of the tagset
- We can get a uniform distribution if we set:
  - $\theta_i = 1/|T|$
- We can also estimate θ from data using MLE:
  - $\theta_{t_j} = \frac{count(t_{j-1}, t_j)}{count(t_{j-1})}$

Example HMM:

The UNsupervised case

- We don't get to see the tags. They are *latent*.
- But, for a given tag assignment, we can:
  - Estimate parameters
  - Calculate corpus probability
- Search for tag assignments such that if we estimate parameters from them, and then use the parameters to calculate the corpus probability, we will get high probability.
- This search looks hard!
- And it is.
- Two possible approximations:
  - EM algorithm
  - **Gibbs sampling**

Gibbs sampling

$w = w_1, \ldots, w_n$
$t = t_1, \ldots, t_n$

- We are interested in the tag assignment that will maximize $P(w, t)$
- For a fixed w, $\arg\max_t P(w, t) = \arg\max_t P(t|w)$

Gibbs sampling

$w = w_1, \ldots, w_n$
$t = t_1, \ldots, t_n$

- We are interested in the tag assignment that will maximize $P(w, t)$
- For a fixed w, $\arg\max_t P(w, t) = \arg\max_t P(t|w)$
- If we could sample from $P(t|w)$, we will, with high probability, get t such that $P(t|w)$ is high.

Gibbs sampling

$w = w_1, \ldots, w_n$
$t = t_1, \ldots, t_n$

- We are interested in the tag assignment that will maximize $P(w, t)$
- For a fixed w, $\arg\max_t P(w, t) = \arg\max_t P(t|w)$
- If we could sample from $P(t|w)$, we will, with high probability, get t such that $P(t|w)$ is high.
- Ok. . . but how do we sample from $P(t|w)$?

Gibbs sampling

$w = w_1, \ldots, w_n$
$t = t_1, \ldots, t_n$

- We are interested in the tag assignment that will maximize $P(w, t)$
- For a fixed w, $\arg\max_t P(w, t) = \arg\max_t P(t|w)$
- If we could sample from $P(t|w)$, we will, with high probability, get t such that $P(t|w)$ is high.
- Ok. . . but how do we sample from $P(t|w)$?
- Gibbs sampling is a "magical" way of doing that
  - To uncover the magic, see Graphical Models class

Gibbs sampling

Main idea

- In order to sample $P(t|w) = P(t_1, t_2, \ldots, t_n|w)$:
- Start with a random assignment of $t_1, \ldots, t_n$. Then:
  - sample t_1 based on $t_2, \ldots, t_n, w$
    - $P(t_1|t_2, t_3, \ldots, t_n, w)$
  - sample t_2 based on $t_1, t_3, \ldots, t_n, w$
  - ...
  - sample t_k based on $t_1, \ldots, t_{k-1}, t_{k+1}, \ldots, t_n, w$
  - ... and so on
- After many iterations, we will get samples from $P(t|w)$

Gibbs sampling

Calculating $P(t_k|t_1,\ldots,t_{k-1},t_{k+1},\ldots,t_n,w)$

- Notation: $t^{-\mathbf{k}} = t_1,\ldots,t_{k-1},t_{k+1},\ldots,t_n$.
- We can estimate q and e as previously, based on w and the assignments to $t^{-\mathbf{k}}$.
- Now we get:

$$P(t_k|t^{-\mathbf{k}}) \propto q(t_k|t_{k-1})e(w_k|t_k)q(t_{k+1}|t_k)$$

  - (why? and what does $\propto$ means?)
- Calculate this for every possible value of t_k.
- Normalize

Draws from distributions

$$X \sim Discrete(\theta, k)$$

```
p = Math.random()

sum = 0.0

for i in 0...k-1 {
  sum += theta[i];
  if(sum >= p) return i
}
```

The Gibbs sampling algorithm

Sampling from $P(t|w)$ for $t = t_1, \ldots, t_n$

Initialize t with random values
Calculate parameters (collect counts) based on t,w.
for many iterations **do**
  for $i \in 1, \ldots, n$ **do**
    "forget" value of t_i (decrease counts)
    Calculate $P(t_i|t^{-\mathbf{i}})$ based on modified counts
    Sample new value for t_i from $P(t_i|t^{-\mathbf{i}})$

Putting it all together

Training HMM from text and dictionary using Gibbs sampling

For each word, assign a random tag from the set allowed by the dictionary
Calculate q, e based on this tag assignment
for many iterations **do**
  for every sentence **do**
    for $i \in 1, \ldots, length$ **do**
      "forget" value of t_i (decrease counts)
      Calculate $P(t_i|t^{-\mathbf{i}})$ based on modified counts
      (Set prob of tags not in dictionary to 0. Normalize.)
      Sample new value for t_i from $P(t_i|t^{-\mathbf{i}})$

Putting it all together

Training HMM from text and dictionary using Gibbs sampling

For each word, assign a random tag from the set allowed by the dictionary
Calculate q, e based on this tag assignment
for many iterations **do**
  for every sentence **do**
    for $i \in 1, \ldots, length$ **do**
      "forget" value of t_i (decrease counts)
      Calculate $P(t_i|t^{-\mathbf{i}})$ based on modified counts
      (Set prob of tags not in dictionary to 0. Normalize.)
      Sample new value for t_i from $P(t_i|t^{-\mathbf{i}})$
Calculate final q and e based on the final state
(can also average several states)

HMM - discussion

Why do you expect this to work?

Why do we need the tag dictionary?

Topic Modeling / LDA

The problem with information

www.betaversion.org/~stefano/linotype/news/26/

As more information becomes available, it becomes more difficult to access what we are looking for.

We need new tools to help us organize, search, and understand these vast amounts of information.

Topic modeling

Candida Hofer

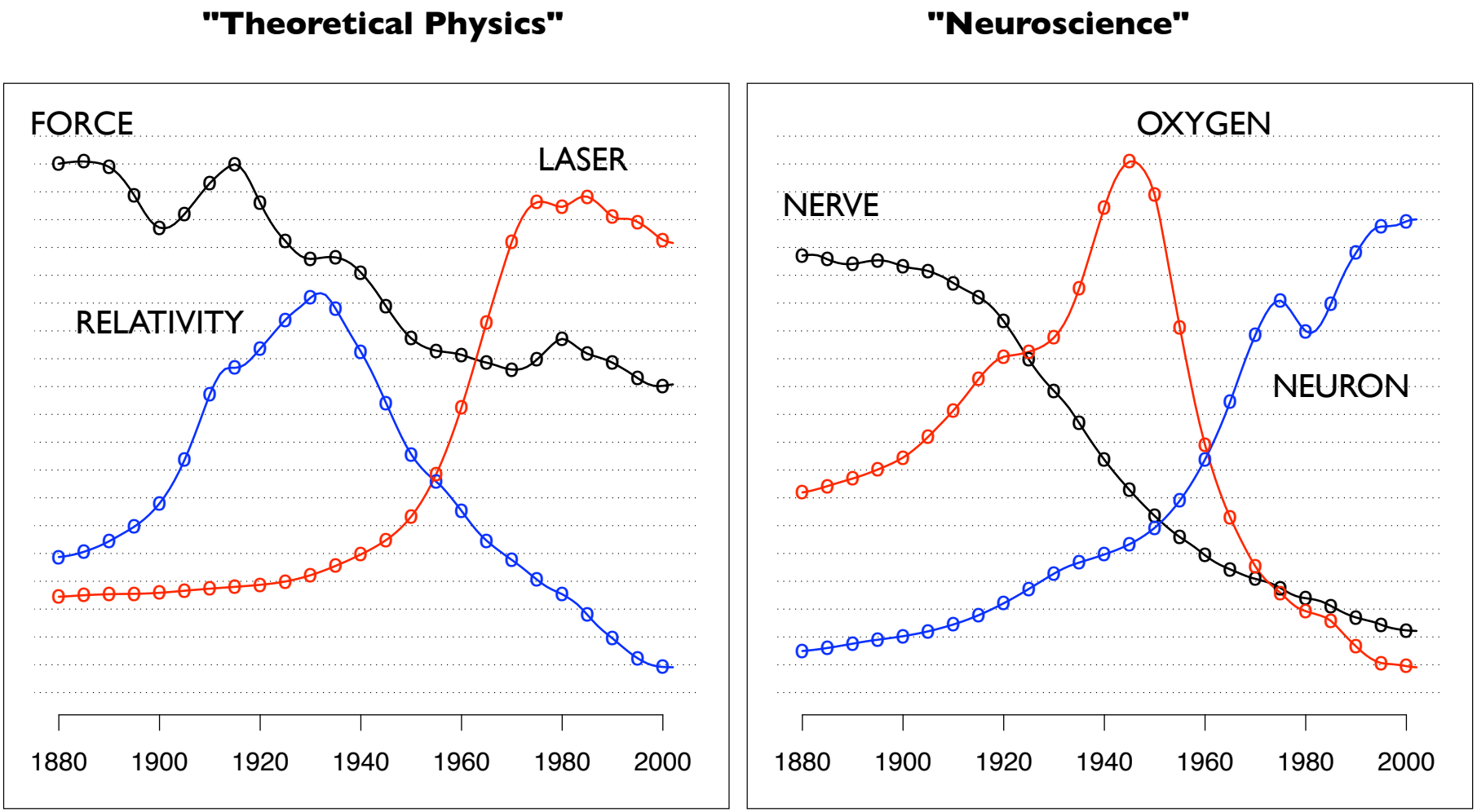
Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.

1. Uncover the hidden topical patterns that pervade the collection.
2. Annotate the documents according to those topics.
3. Use the annotations to organize, summarize, and search the texts.

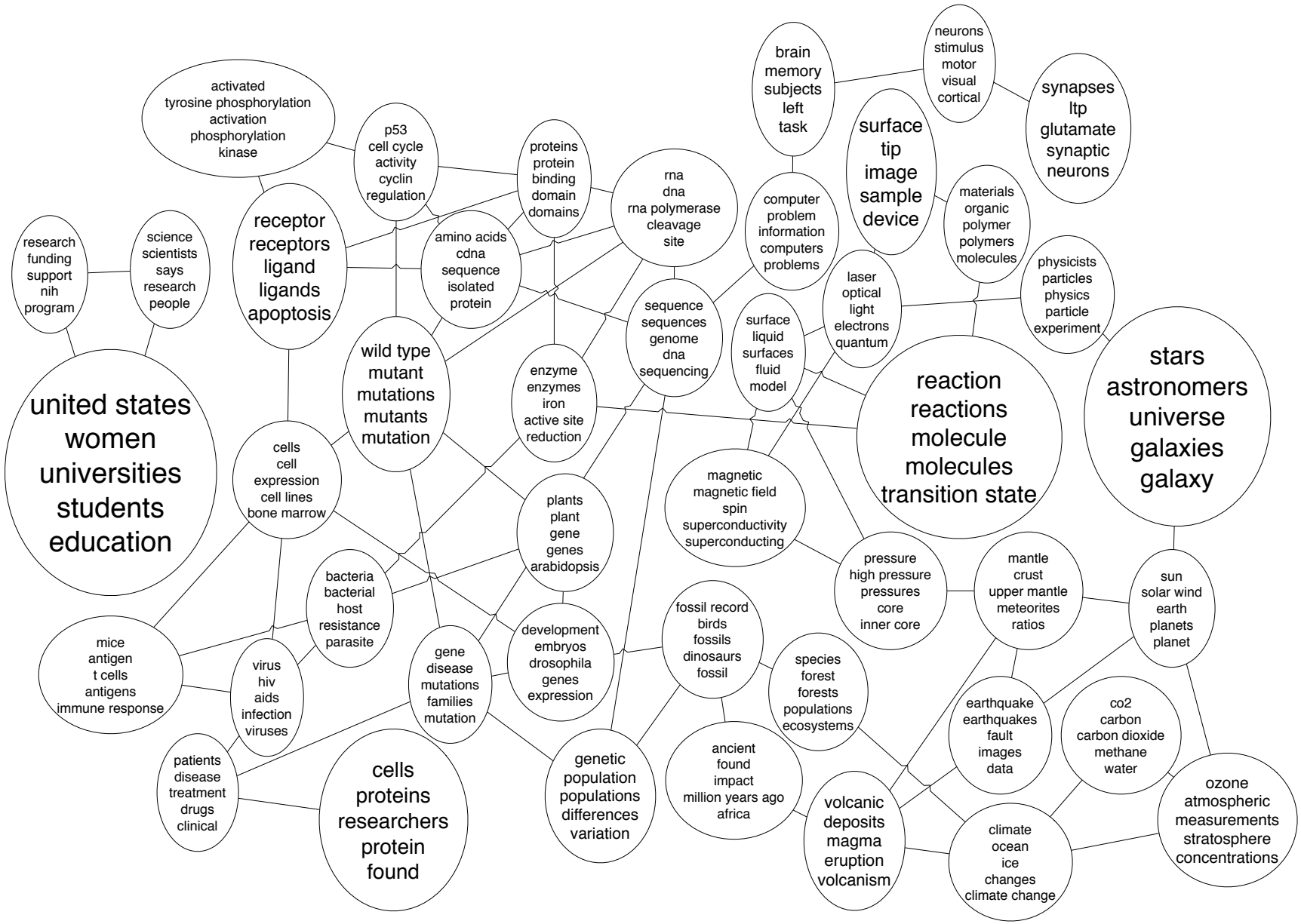
Discover topics from a corpus

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

Model the evolution of topics over time



Model connections between topics



Annotate images

SKY WATER TREE
MOUNTAIN PEOPLE

SCOTLAND WATER
FLOWER HILLS TREE

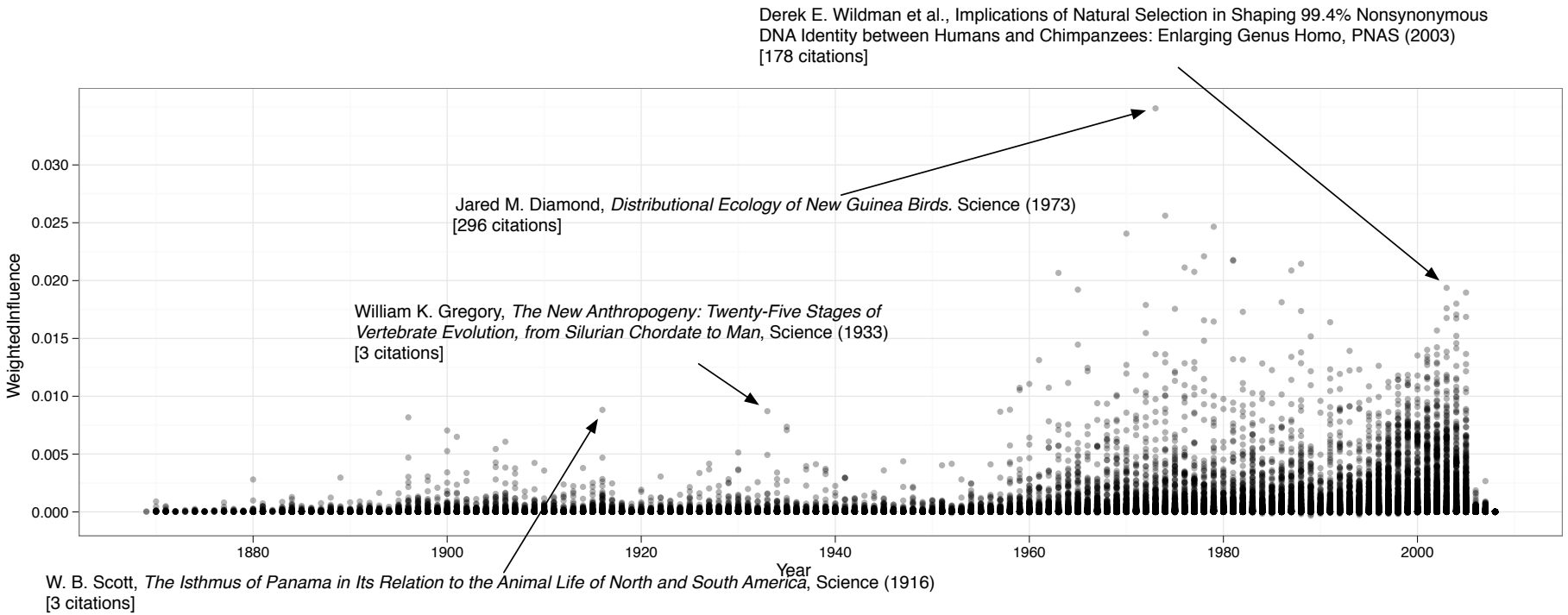
SKY WATER BUILDING
PEOPLE WATER

FISH WATER OCEAN
TREE CORAL

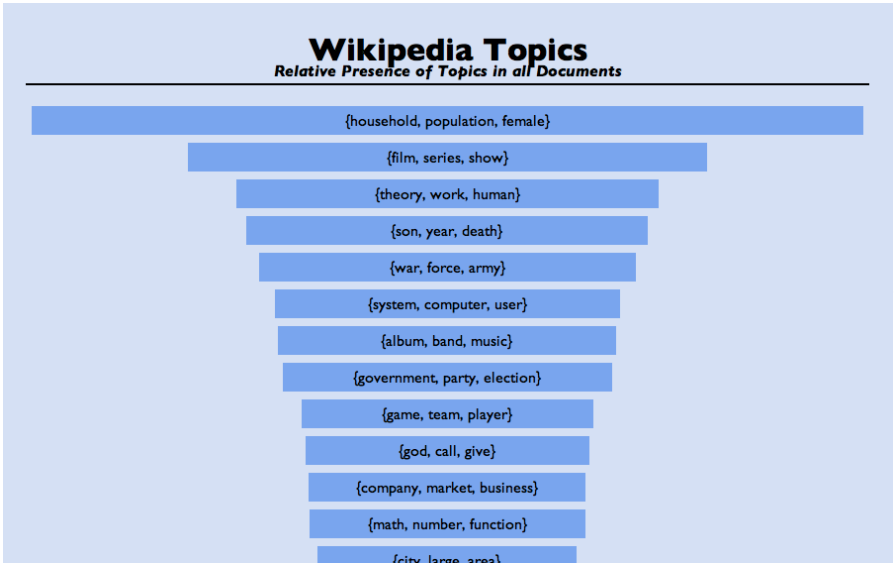
PEOPLE MARKET PATTERN
TEXTILE DISPLAY

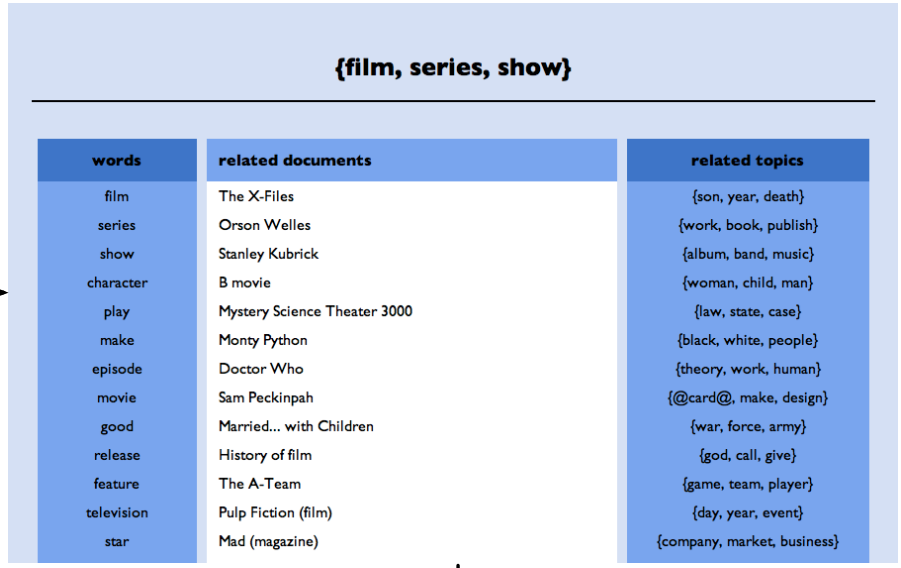
BIRDS NEST TREE
BRANCH LEAVES

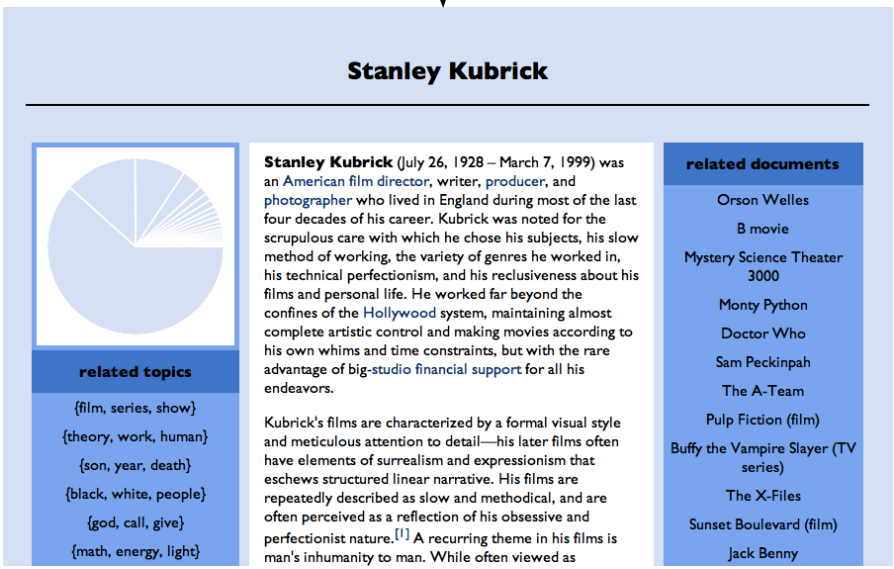
Discover influential articles

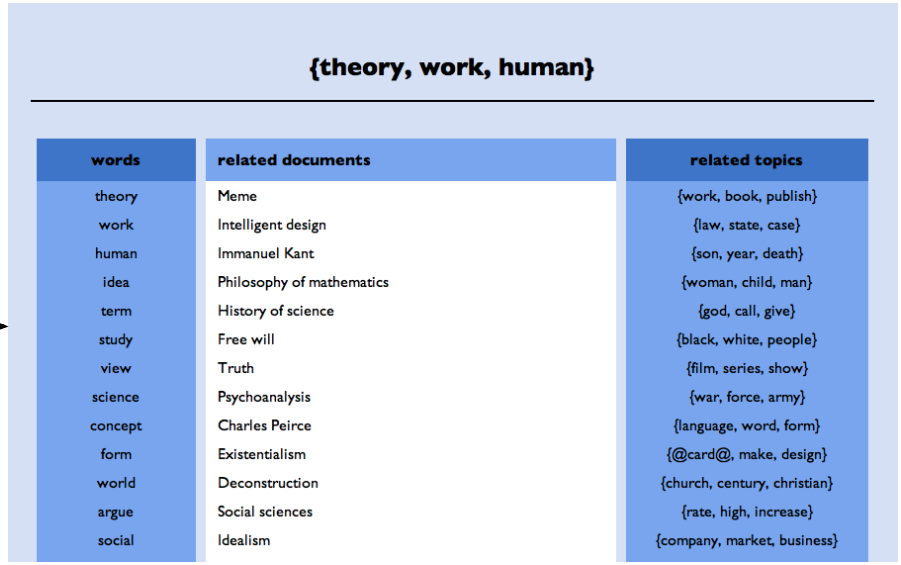


Organize and browse large corpora









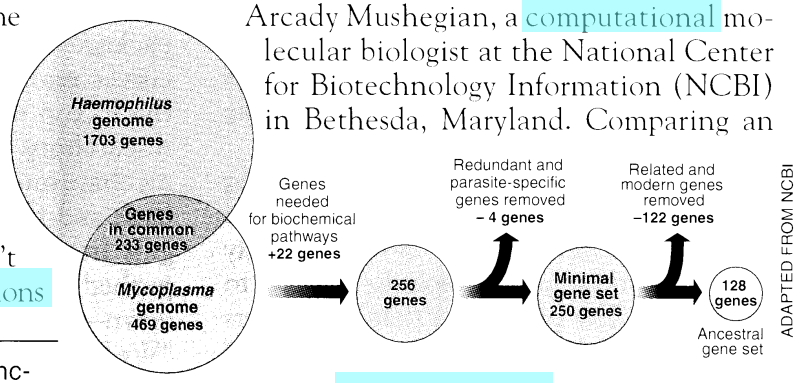
Latent Dirichlet Allocation

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



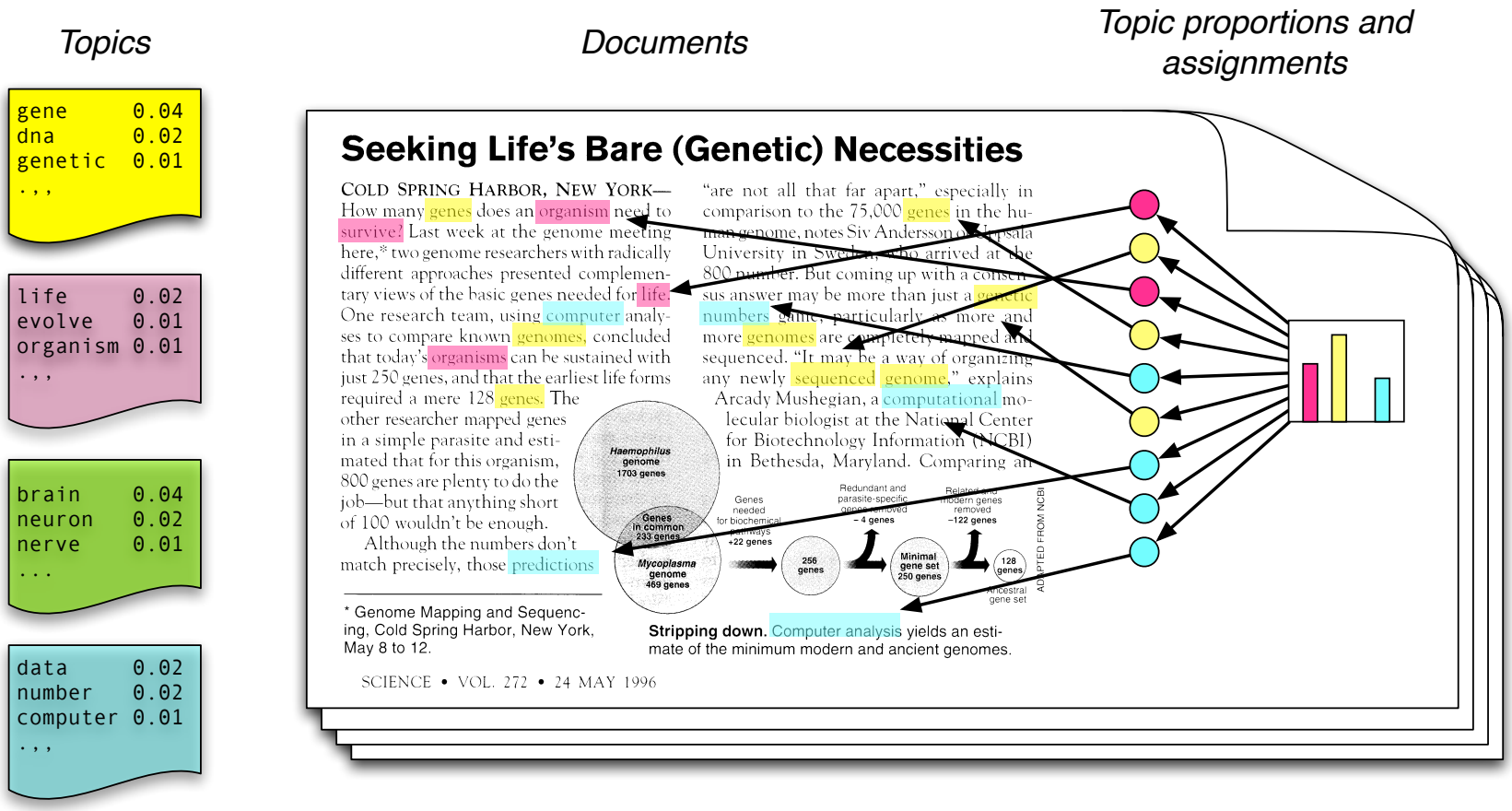
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

SCIENCE • VOL. 272 • 24 MAY 1996

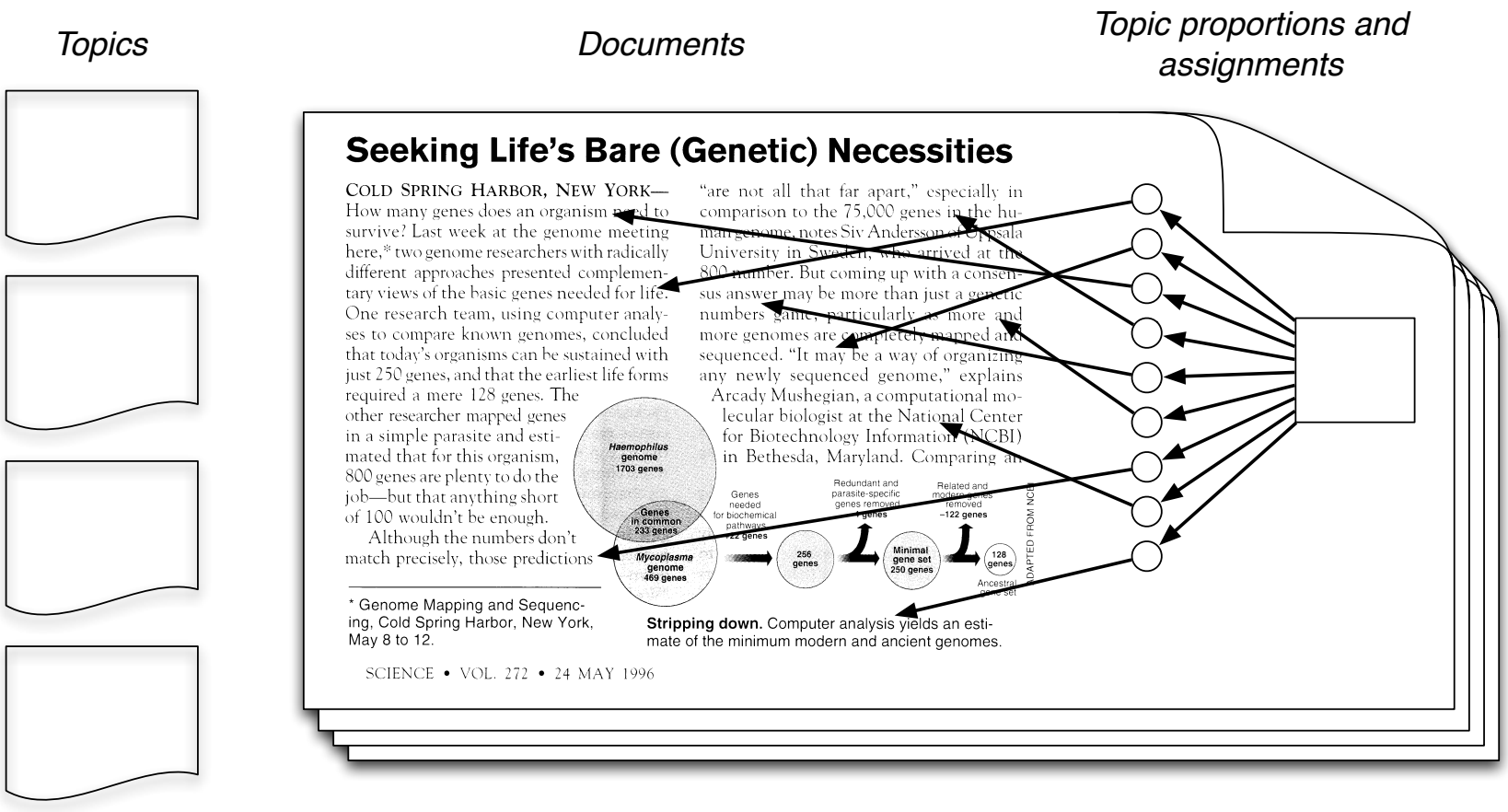
Simple intuition: Documents exhibit multiple topics.

Generative model for LDA



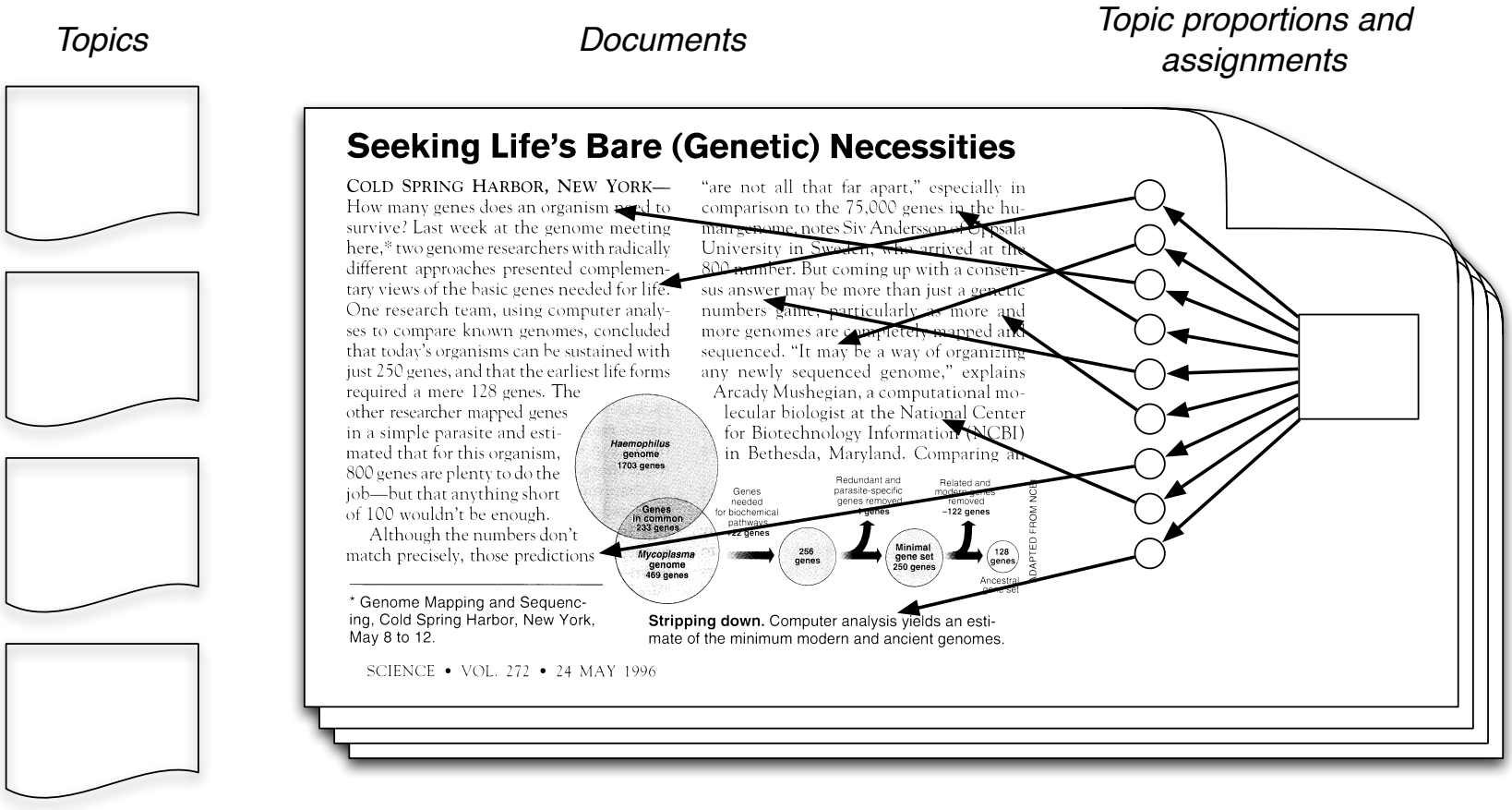
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

The posterior distribution



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

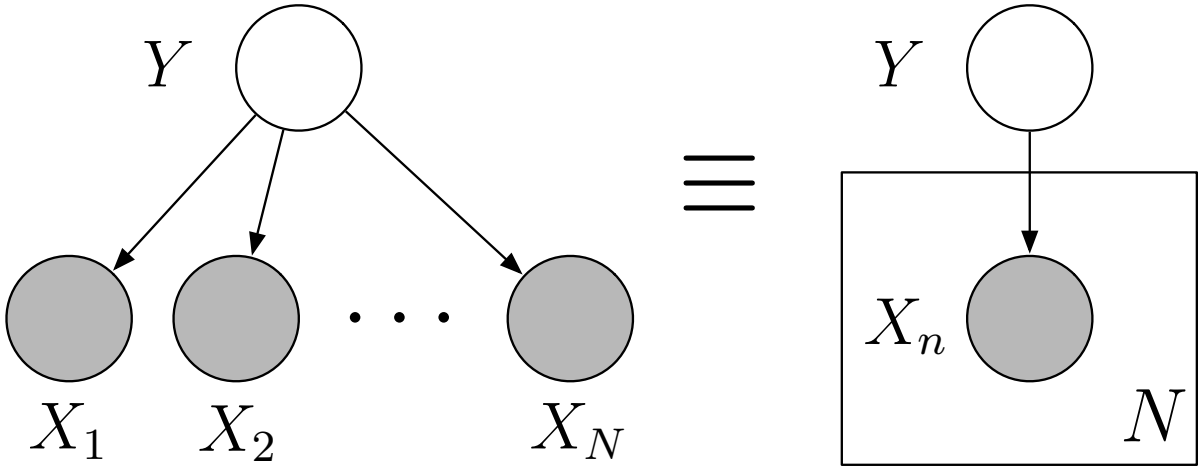
The posterior distribution



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

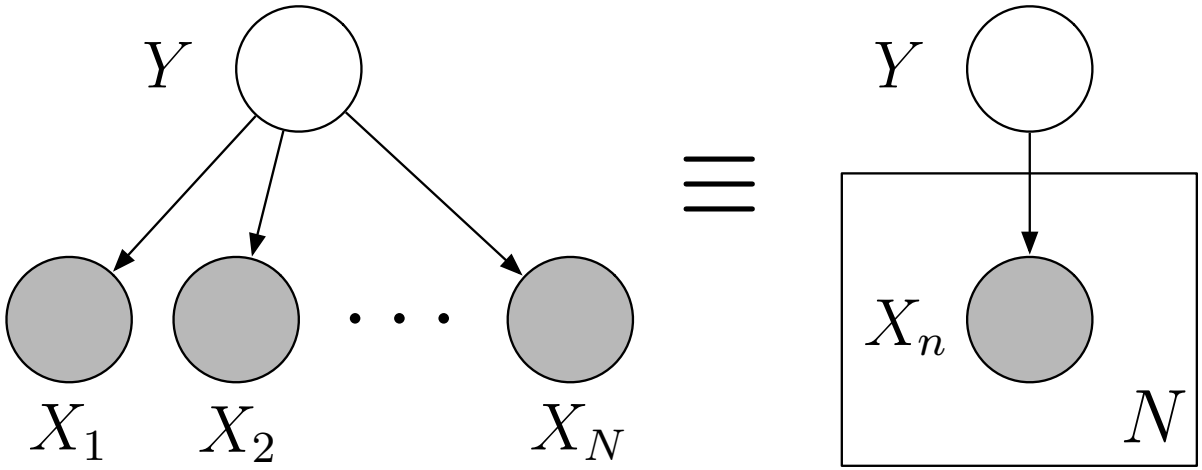
$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

Graphical models (Aside)



- Nodes are random variables
- Edges denote possible dependence
- Observed variables are shaded
- Plates denote replicated structure

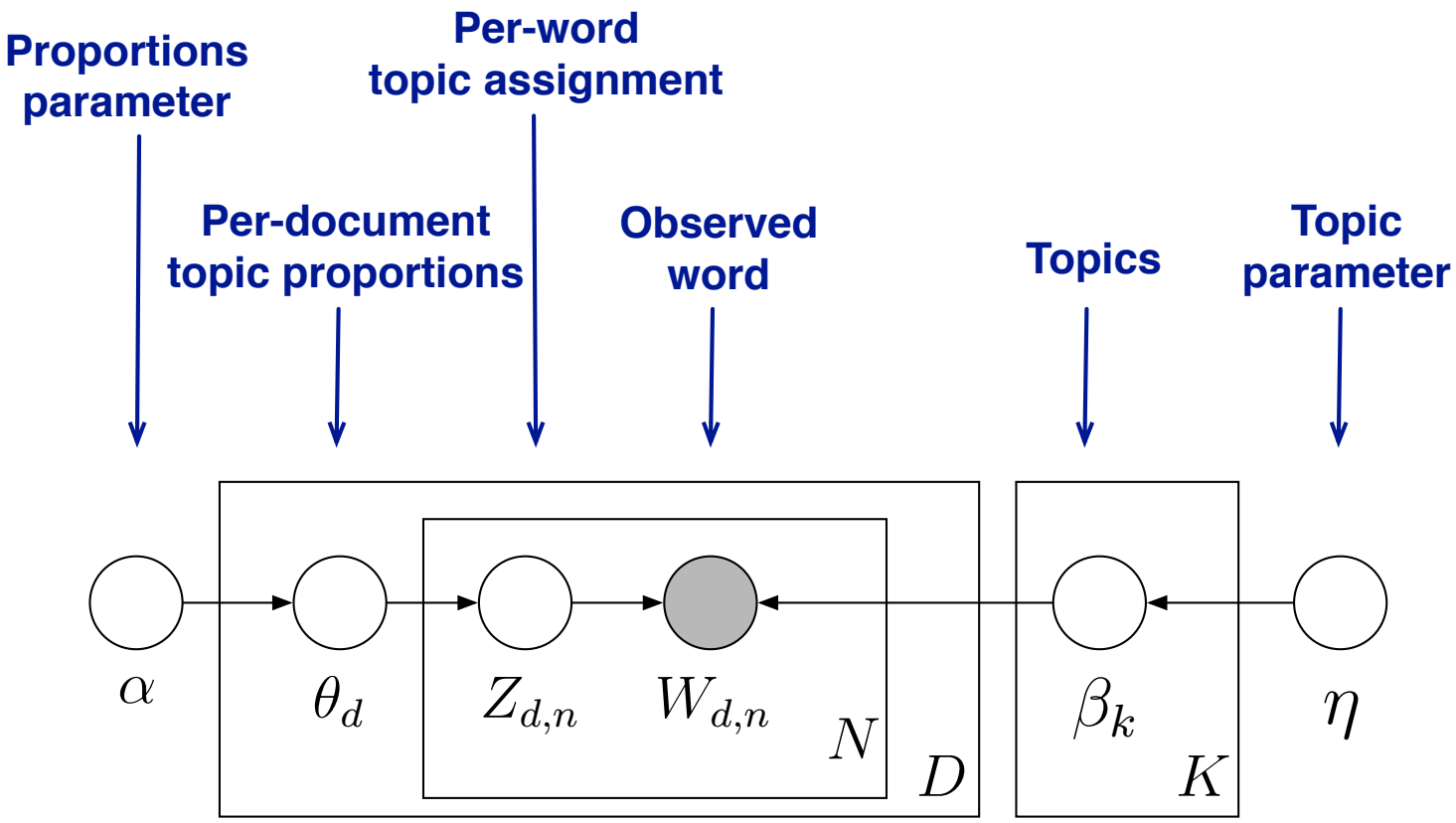
Graphical models (Aside)



- Structure of the graph defines the pattern of conditional dependence between the ensemble of random variables
- E.g., this graph corresponds to

$$p(y, x_1, \ldots, x_N) = p(y) \prod_{n=1}^{N} p(x_n \mid y)$$

LDA as a graphical model



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

LDA Generative Story

We have K topics, and a vocabulary V of $|V|$ words.
Each topic β^k is a distribution over words.

A document d is created by

- Sample length n_d from a Poisson distribution
  - (alternatively, assume n_d is given)
- Sample topic proportions θ^d from a Dirichlet distribution with parameter α.
- For each position $i \in 1, \ldots, n$:
  - Sample topic z_i from θ^d
  - Sample word w_i from the distribution β^{z_i}

LDA Generative Story

We have K topics, and a vocabulary V of $|V|$ words.
Each topic β^k is a distribution over words.

A document d is created by

- Sample length n_d from a Poisson distribution
  - (alternatively, assume n_d is given)
- Sample topic proportions θ^d from a Dirichlet distribution with parameter α.
- For each position $i \in 1, \ldots, n$:
  - Sample topic z_i from θ^d
  - Sample word w_i from the distribution β^{z_i}

Assumptions

- We do not care about the word-order ("bag of words")
- Each word is independent of the other words given its topic

LDA Generative Story

We have K topics, and a vocabulary V of $|V|$ words.
Each topic β^k is a distribution over words.

A document d is created by

- Sample length n_d from a Poisson distribution
  - (alternatively, assume n_d is given)
- Sample topic proportions θ^d from **a Dirichlet distribution with parameter α.**
- For each position $i \in 1, \ldots, n$:
  - Sample topic z_i from θ^d
  - Sample word w_i from the distribution β^{z_i}

Assumptions

- We do not care about the word-order ("bag of words")
- Each word is independent of the other words given its topic

The Dirichlet Distribution

- The Dirichlet distribution is a "distribution over distributions"
- When you sample $\theta \sim \textsc{Dirichlet}(\alpha, K)$:
  - θ is a K-dim vector
  - $\theta_i \geq 0$
  - $\sum_i \theta_i = 1$

The Dirichlet Distribution

- The Dirichlet distribution is a "distribution over distributions"
- When you sample $\theta \sim \text{DIRICHLET}(\alpha, K)$:
  - θ is a K-dim vector
  - $\theta_i \geq 0$
  - $\sum_i \theta_i = 1$

The probability of seeing a particular vector θ is:

$$P_{\text{DIRICHLET}(\alpha,K)}(\theta) = \frac{\prod_{i=1}^{K} \theta_i^{\alpha_i - 1}}{B(\alpha)}$$

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$$

- Γ is the gamma function, generalization of factorial.

- Generally, α is a k-dim vector, but we will assume "symmetric" dirichlet, in which α is a single scalar (and $\alpha_i = \alpha$ for all $i \in \{1, \ldots, K\}$)

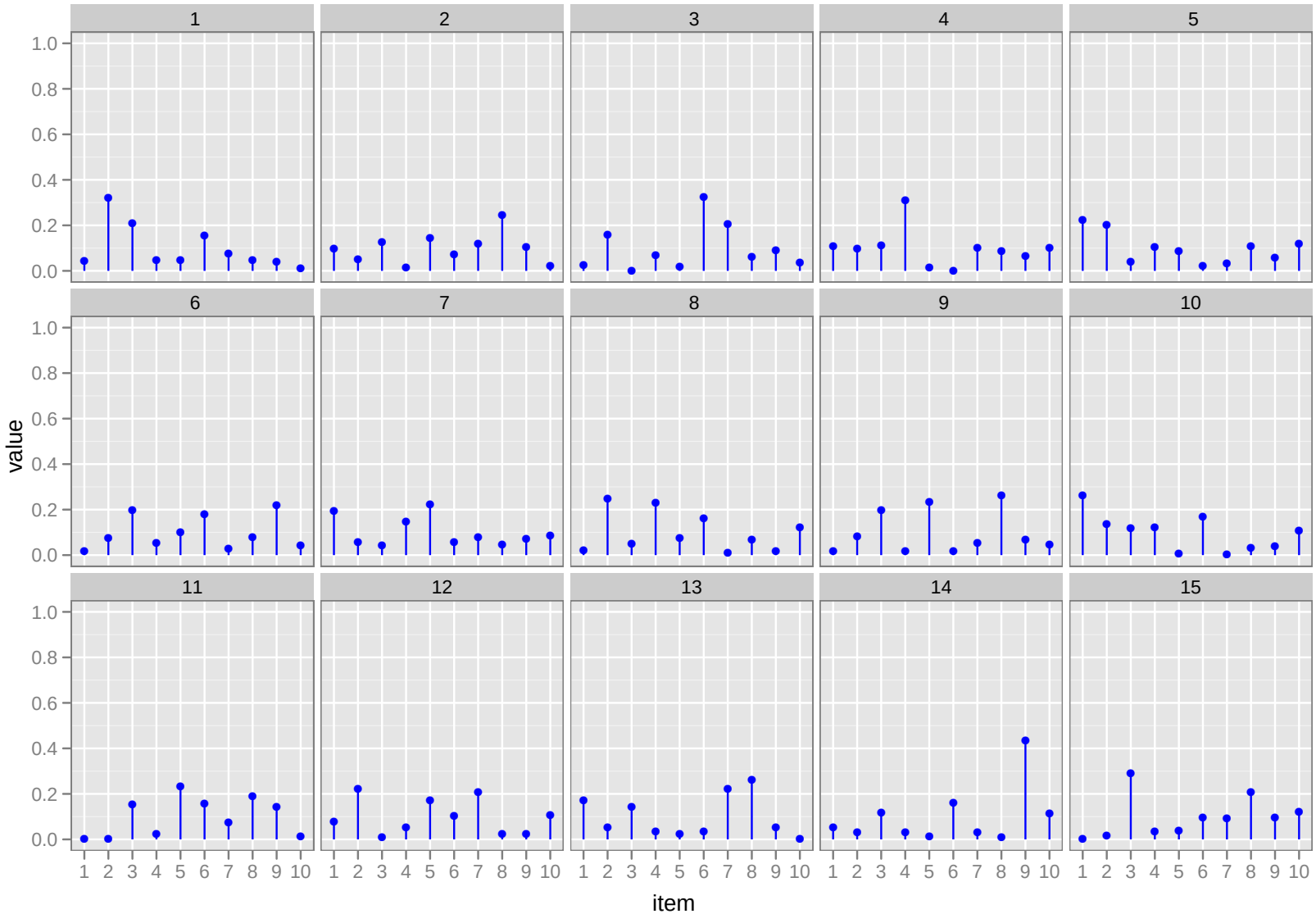
The Dirichlet Distribution

The Dirichlet distribution is a “distribution over distributions”

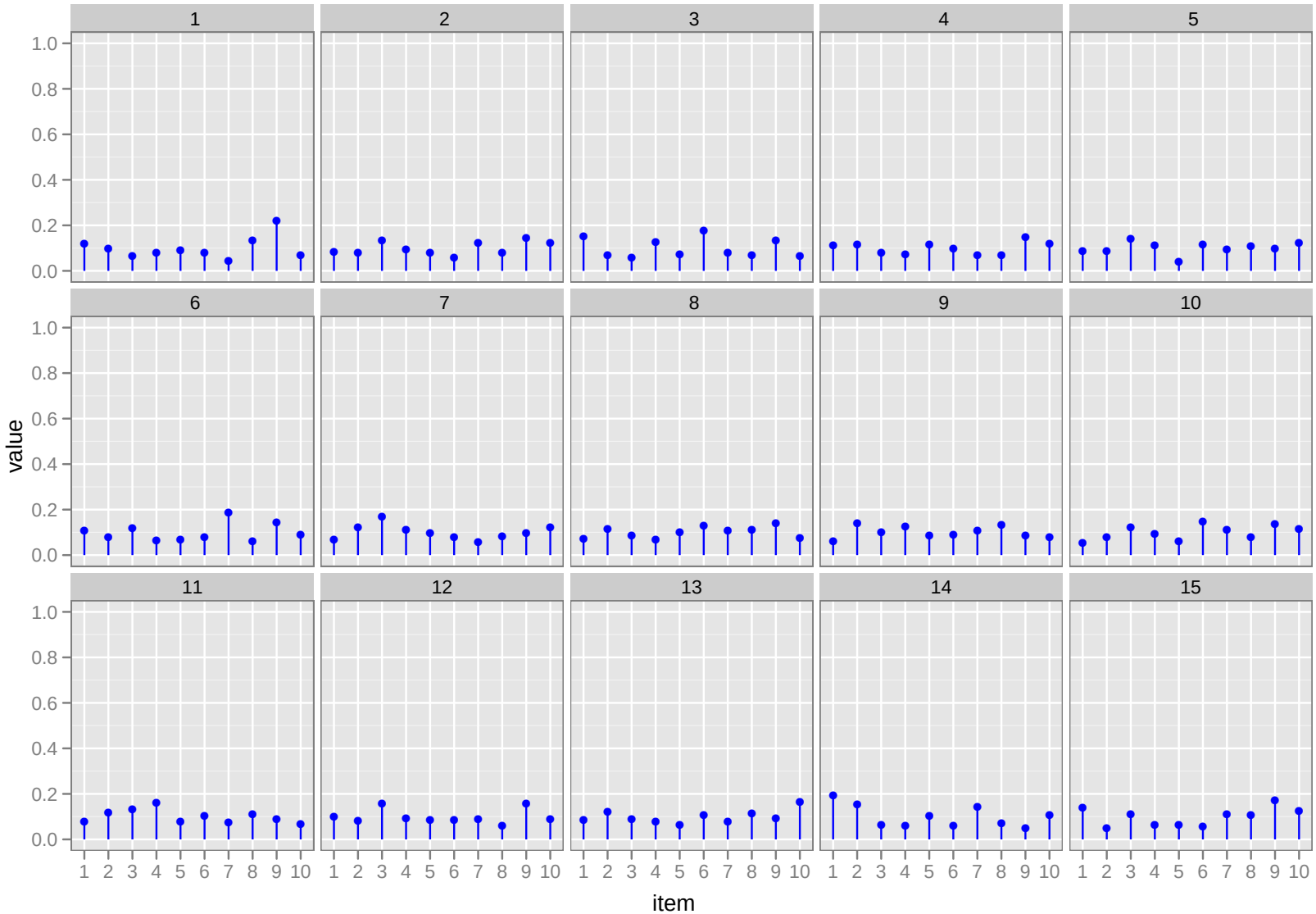
$$P_p(\theta|\alpha) = \frac{\prod_{i=1}^{K} \theta_i^{\alpha_i - 1}}{B(\alpha)}$$

- α controls the shape, mean and sparsity of θ

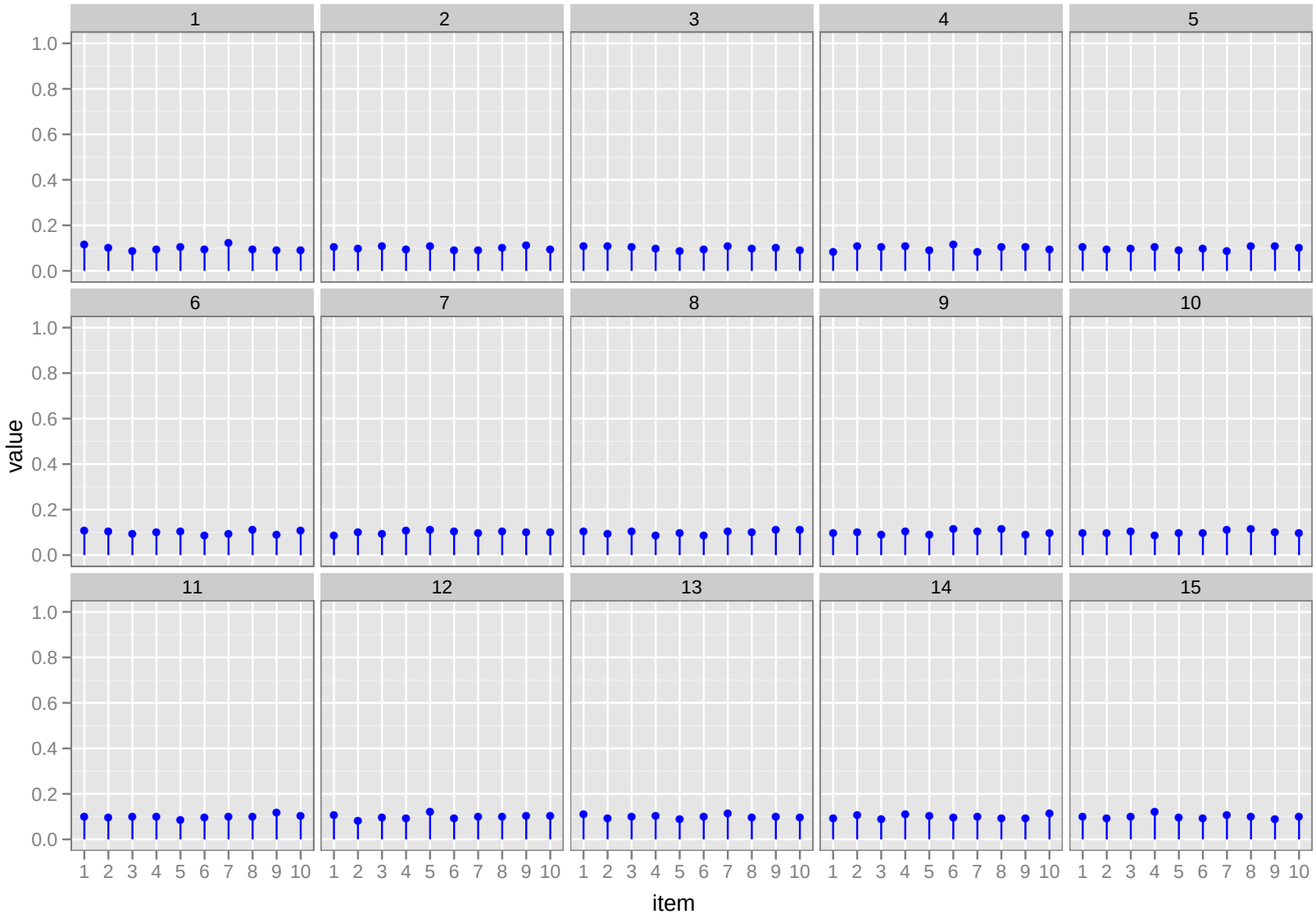
$\alpha = 1$


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

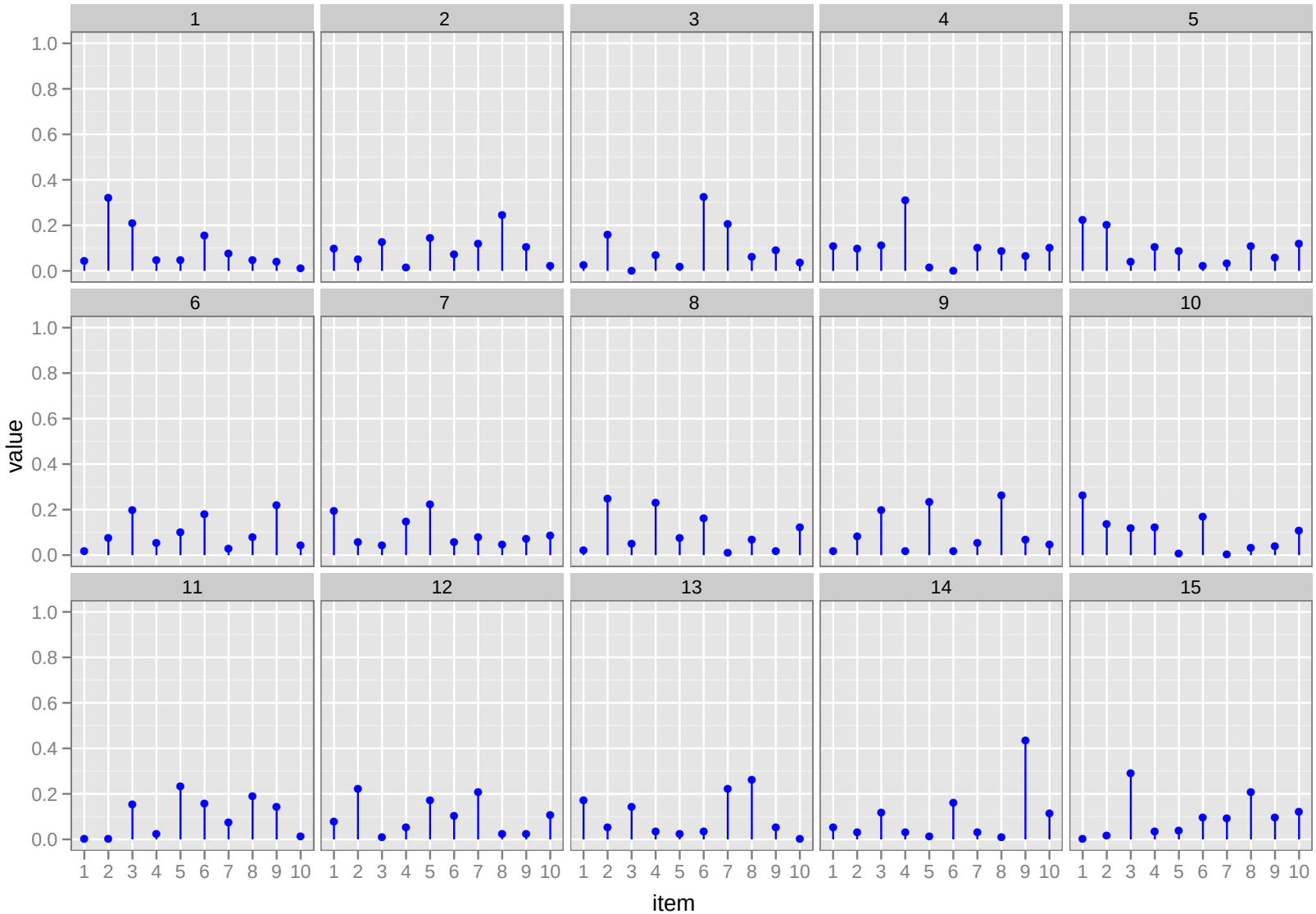
$\alpha = 10$


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

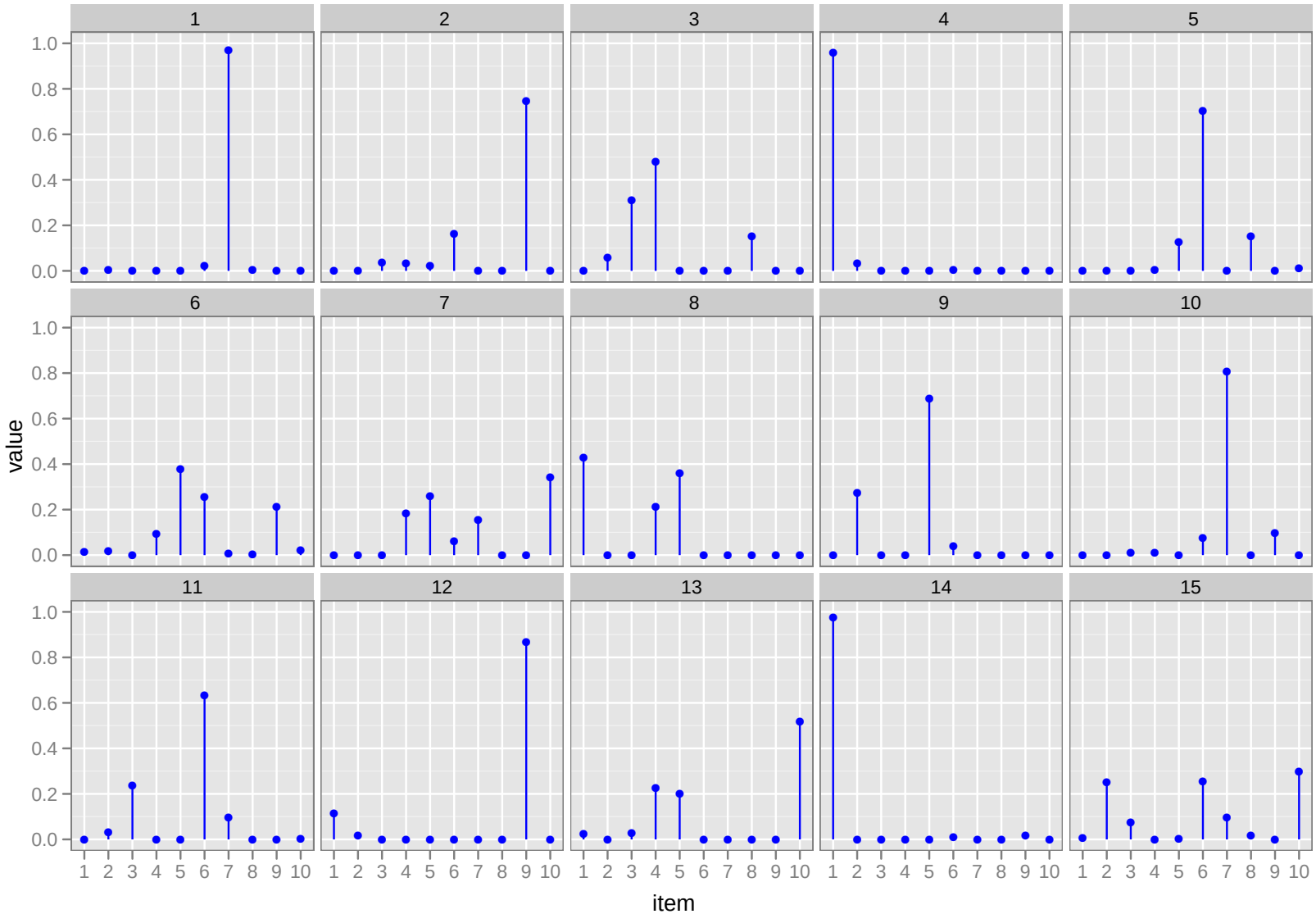
$\alpha = 100$


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

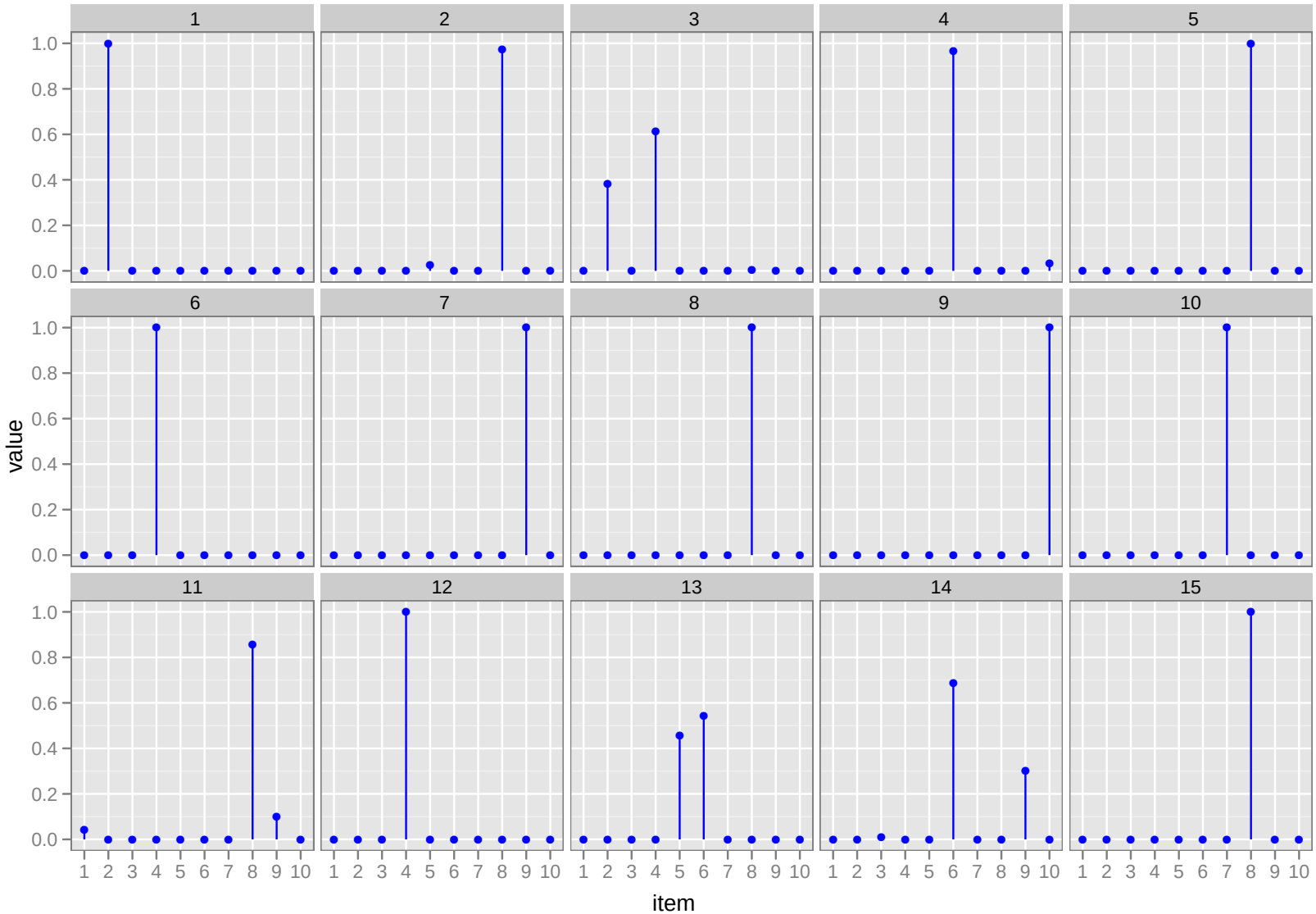
$\alpha = 1$


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

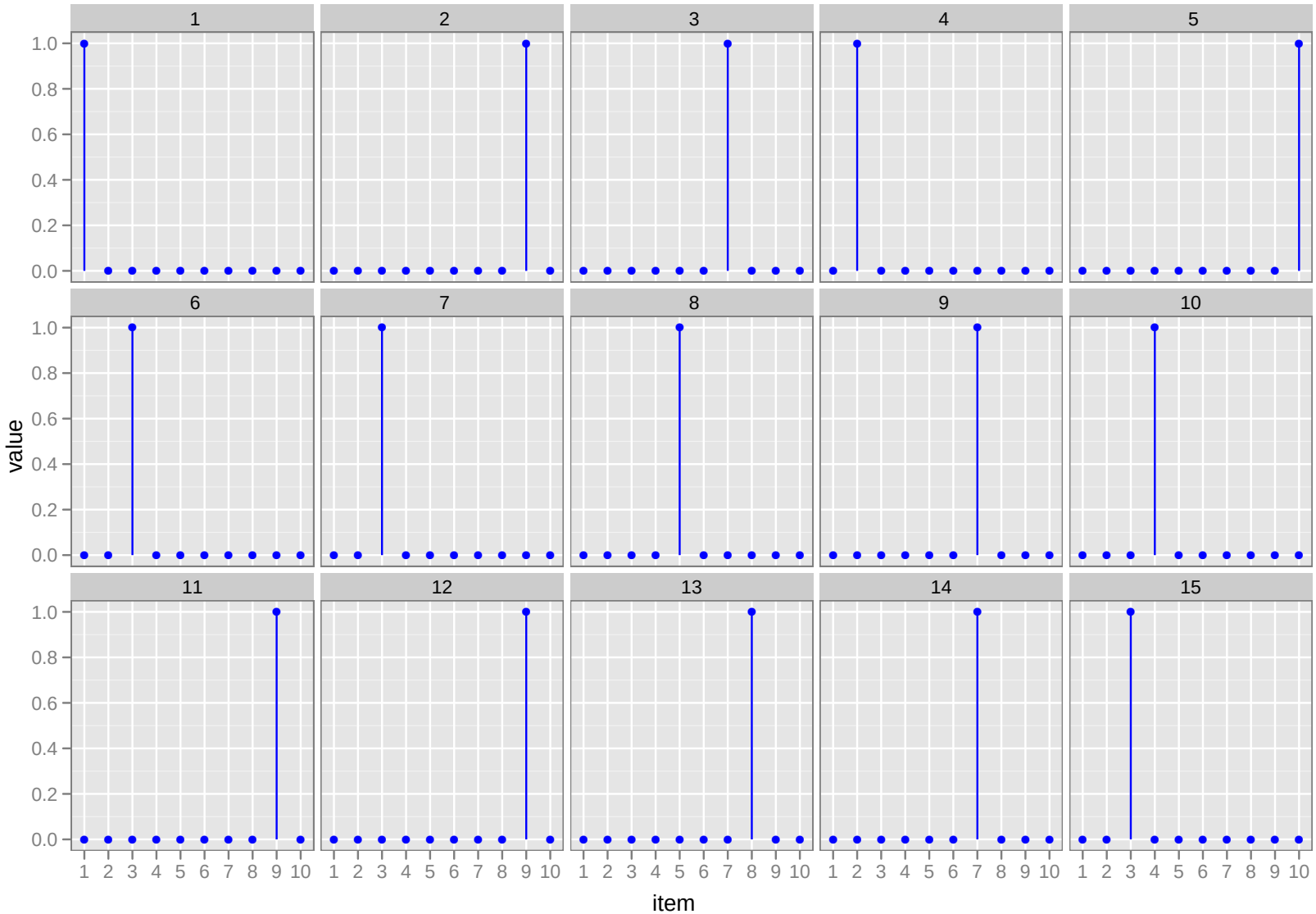
$\alpha = 0.1$


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

$\alpha = 0.01$


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

$\alpha = 0.001$


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

The Dirichlet Distribution

For draws θ from a symmetric dirichlet distribution:

- $\alpha = 1$ All θ are equally likely
- $\alpha > 1$ Uniform θ are more likely
- $\alpha < 1$ Spikey θ are more likely

LDA Generative Story

We have K topics, and a vocabulary V of $|V|$ words.
Each topic β^k is a distribution over words. $\beta^k \sim Dirichlet(\eta, |V|)$

A document d is created by

- Sample length n_d from a Poisson distribution
  - (alternatively, assume n_d is given)
- Sample topic proportions θ^d from a Dirichlet distribution with parameter α.
- For each position $i \in 1, \ldots, n$:
  - Sample topic z_i from θ^d
  - Sample word w_i from the distribution β^{z_i}

LDA Generative Story

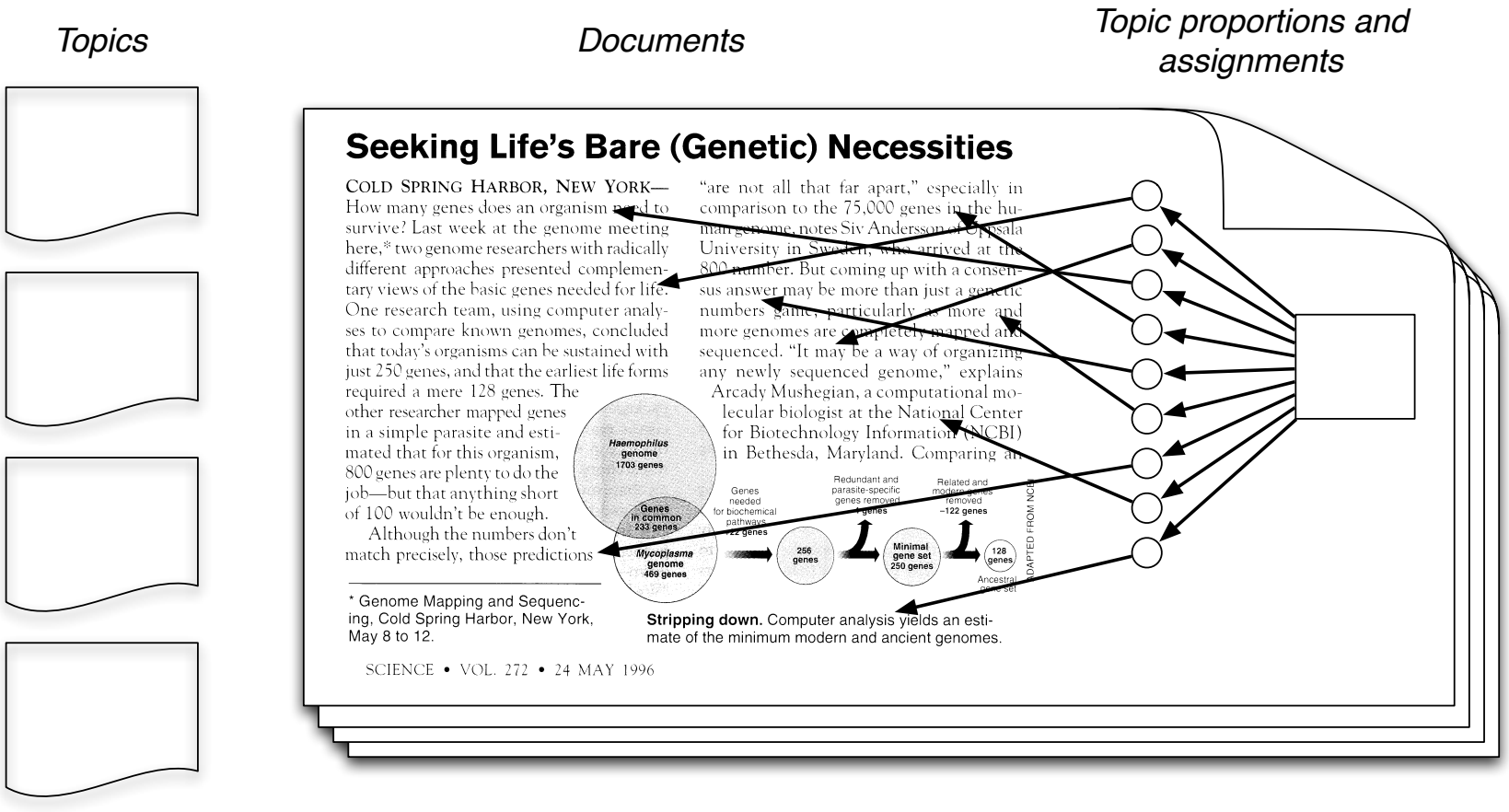
We have K topics, and a vocabulary V of $|V|$ words.
Each topic β^k is a distribution over words. $\beta^k \sim Dirichlet(\eta, |V|)$

A document d is created by

- Sample length n_d from a Poisson distribution
  - (alternatively, assume n_d is given)
- Sample topic proportions θ^d from a Dirichlet distribution with parameter α.
- For each position $i \in 1, \ldots, n$:
  - Sample topic z_i from θ^d
  - Sample word w_i from the distribution β^{z_i}

α controls how many topics we expect to see in our documents

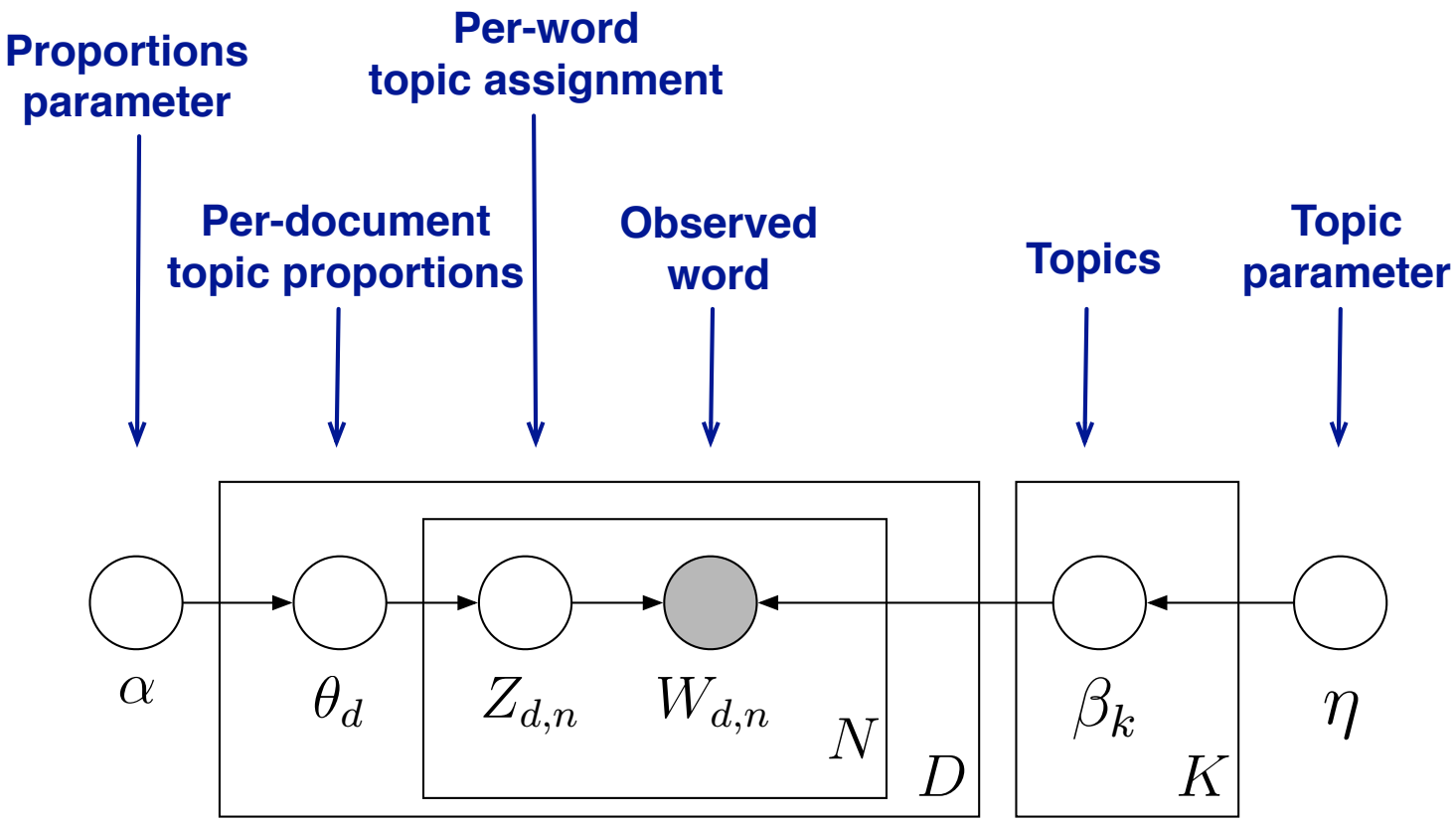
Latent Dirichlet allocation (LDA)



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

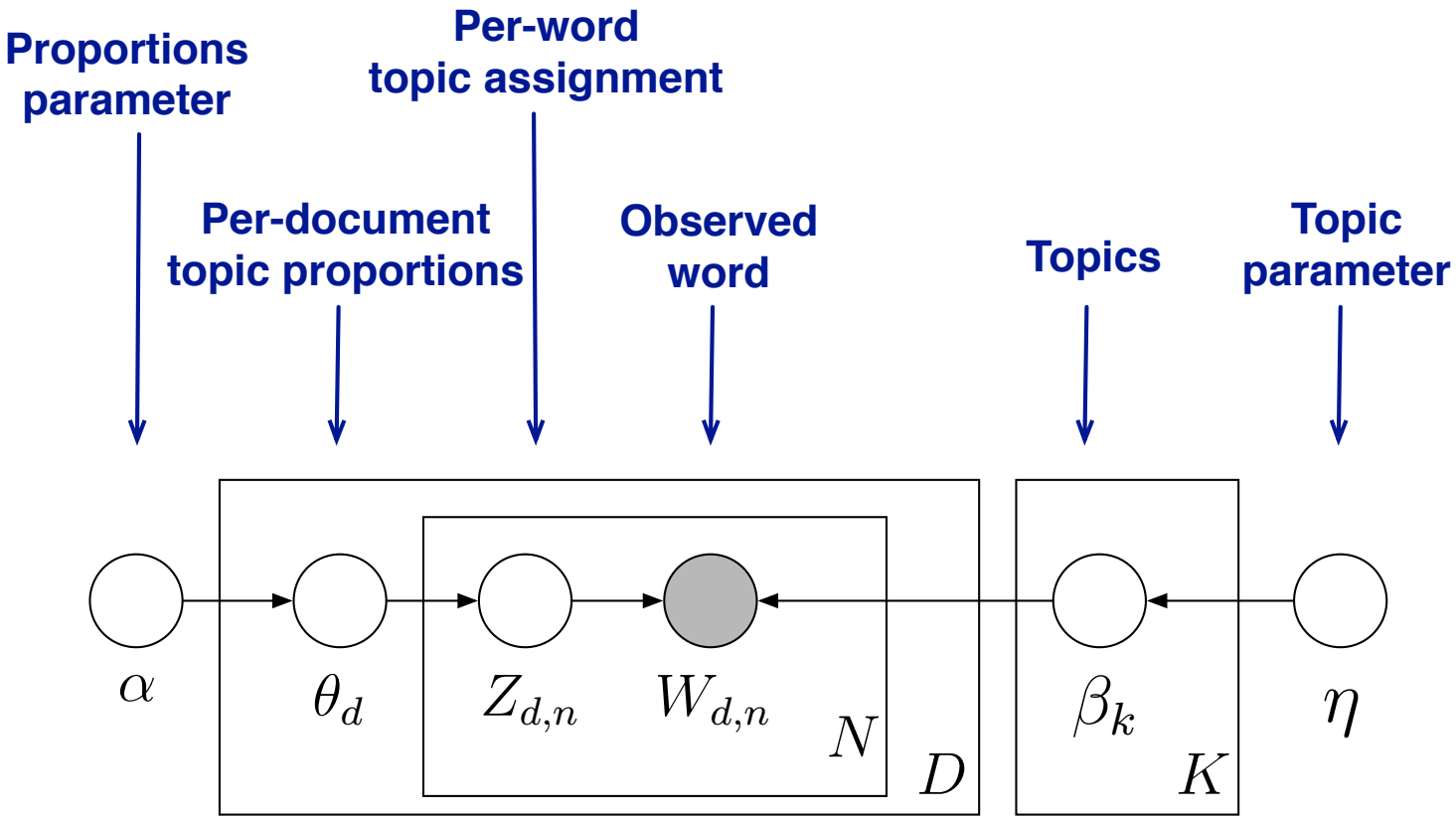
$$p(\text{topics, proportions, assignments} \mid \text{documents})$$

LDA as a graphical model



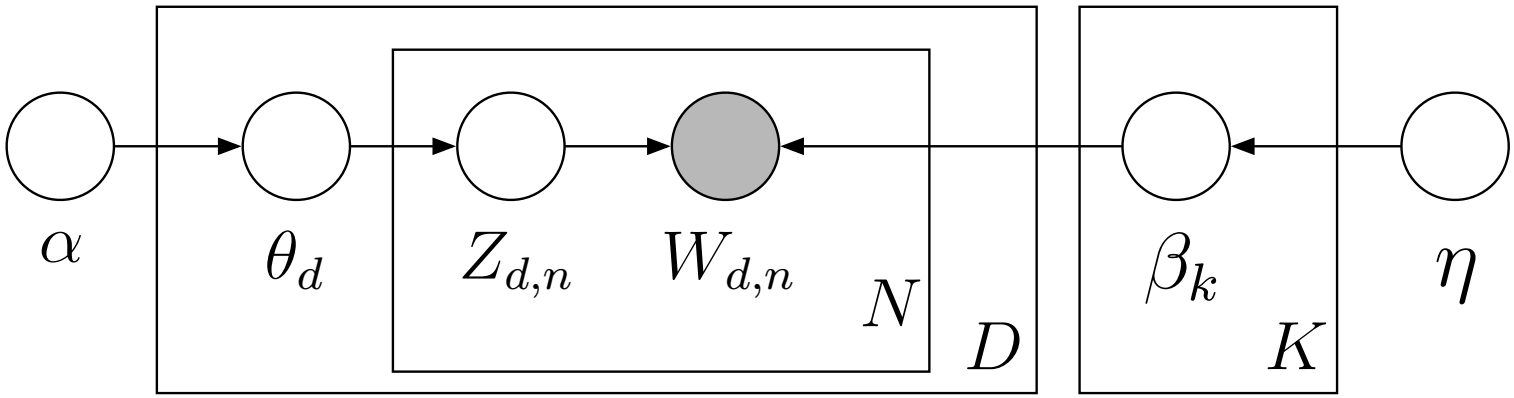
- Nodes are random variables; edges indicate dependence.
- Shaded nodes are observed; unshaded nodes are hidden.
- Plates indicate replicated variables.

LDA as a graphical model



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

LDA as a graphical model



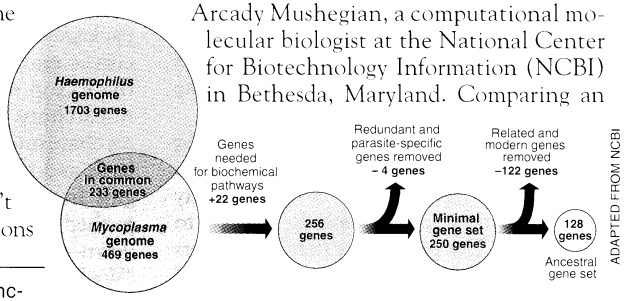
- This joint defines a posterior, $p(\theta, z, \beta \mid w)$.
- From a collection of documents, infer
  - Per-word topic assignment $z_{d,n}$
  - Per-document topic proportions θ_d
  - Per-corpus topic distributions β_k
- Then use posterior expectations to perform the task at hand: information retrieval, document similarity, exploration, and others.

Example 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.

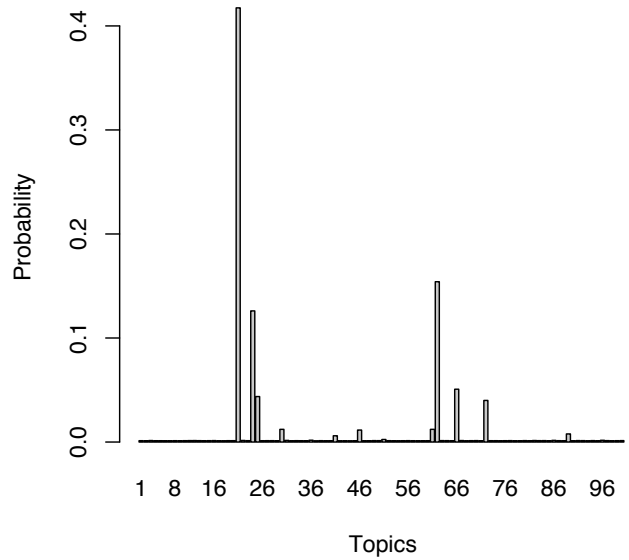
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just 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



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

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

SCIENCE • VOL. 272 • 24 MAY 1996



Example inference

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

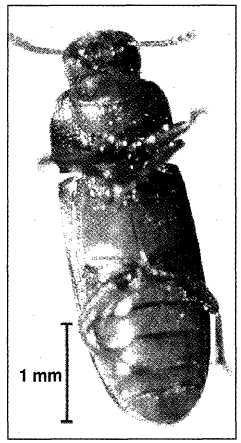
Example inference (II)

Chaotic Beetles

Charles Godfray and Michael Hassell

Ecologists have known since the pioneering work of May in the mid-1970s (*1*) that the population dynamics of animals and plants can be exceedingly complex. This complexity arises from two sources: The tangled web of interactions that constitute any natural community provide a myriad of different pathways for species to interact, both directly and indirectly. And even in isolated populations the nonlinear feedback processes present in all natural populations can result in complex dynamic behavior. Natural populations can show persistent oscillatory dynamics and chaos, the latter characterized by extreme sensitivity to initial conditions. If such chaotic dynamics were common in nature, then this would have important ramifications for the management and conservation of natural resources. On page 389 of this issue, Costantino *et al.* (*2*) provide the most convincing evidence to date of complex dynamics and chaos in a biological population—of the flour beetle, *Tribolium castaneum* (see figure).

It has proven extremely difficult to demonstrate complex dynamics in populations in the field. By its very nature, a chaotically fluctuating population will superficially resemble a stable or cyclic population buffeted by the normal random perturbations experienced by all species. Given a long enough time series, diagnostic tools from nonlinear mathematics can be used to identify the telltale signatures of chaos. In phase space, chaotic trajectories come to lie on "strange attractors," curious geometric objects with fractal structure and hence noninteger dimension. As they move over the surface of the attractor, sets of adjacent trajectories are pulled apart, then stretched and folded, so that it becomes impossible to predict exact population densities into the future. The strength of the mixing that gives rise to the extreme sensitivity to initial conditions can be measured mathematically estimating the Liapunov exponent, which is positive for chaotic dynamics and nonpositive otherwise. There have been many attempts to estimate attractor dimension and Liapunov exponents from time series data, and some candidate chaotic population have been identified (some insects, rodents, and most convincingly, human childhood diseases), but the statistical difficulties preclude any broad generalization (*3*).



Cannibalism and chaos. The flour beetle, *Tribolium castaneum*, exhibits chaotic population dynamics when the amount of cannibalism is altered in a mathematical model.

An alternative approach is to parameterize population models with data from natural populations and then compare their predictions with the dynamics in the field. This technique has been gaining popularity in recent years, helped by statistical advances in parameter estimation. Good ex-

The authors are in the Department of Biology, Imperial College at Silwood Park, Ascot, Berks, SL5 7PZ UK. E-mail: m.hassell@ic.ac.uk

Example inference (II)

problem	model	selection	species
problems	rate	male	forest
mathematical	constant	males	ecology
number	distribution	females	fish
new	time	sex	ecological
mathematics	number	species	conservation
university	size	female	diversity
two	values	evolution	population
first	value	populations	natural
numbers	average	population	ecosystems
work	rates	sexual	populations
time	data	behavior	endangered
mathematicians	density	evolutionary	tropical
chaos	measured	genetic	forests
chaotic	models	reproductive	ecosystem

Why does LDA work?

Why does LDA “work”?

- LDA trades off two goals.
  1. For each document, allocate its words to as few topics as possible.
  2. For each topic, assign high probability to as few terms as possible.
- These goals are at odds.
  - Putting a document in a single topic makes #2 hard:
    All of its words must have probability under that topic.
  - Putting very few words in each topic makes #1 hard:
    To cover a document’s words, it must assign many topics to it.
- Trading off these goals finds groups of tightly co-occurring words.

What do we get out of LDA?

- Topic assignments z
- Topic proportions (how strong is topic k in document j?)
- Topics distributions (how strong is word i in topic k?)
  - Also: which topics are related to word i?

What do we get out of LDA?

- Topic assignments z
- Topic proportions (how strong is topic k in document j?)
- Topics distributions (how strong is word i in topic k?)
  - Also: which topics are related to word i?

So?

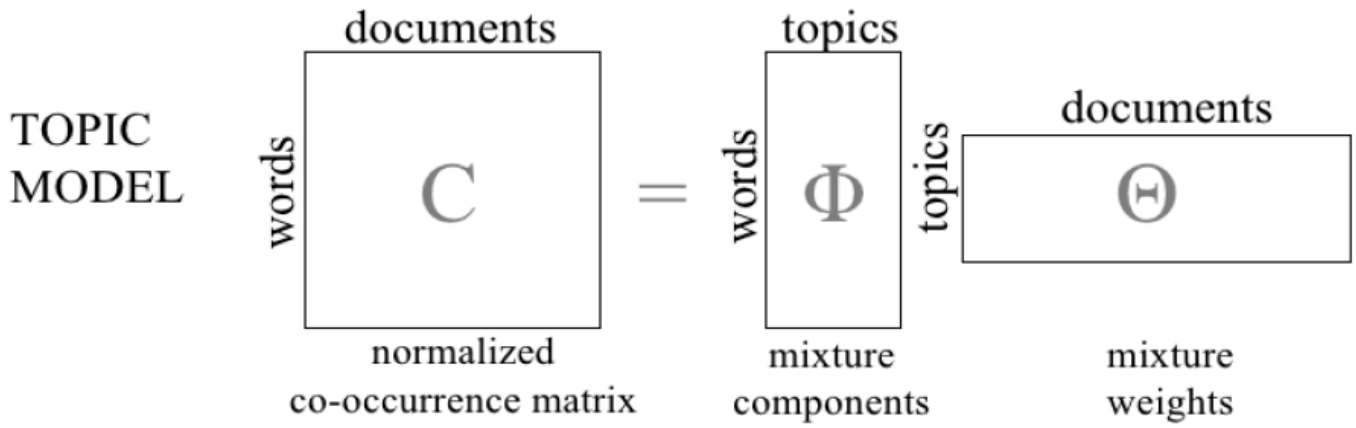
- Which topics are in our corpus?
- Find similar docs (by comparing "topic vectors" of docs)
- Find related words (by comparing "topic vectors" of words)
- Query expansion: find documents related to words X,Y,Z, even if all or some of these words did not appear in the document
- ...

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Topic Model

- P(tlk) for all t and k, is a term by topic matrix (gives which terms make up a topic)
- P(kldoc) for all k and doc, is a topic by document matrix (gives which topics are in a document)



EXAMPLE

Analysis of TASA Corpus

- Given a text collection written by first grade to college students

- Data has following characteristics
  - 26,000+ word types (stop words removed)
  - 37,000+ documents
  - 6,000,000+ word tokens

- Find topics in the data

Topics in the Educational Corpus (TASA)

- 37K docs, 26K words
- 1700 topics, e.g.:

Polysemy

PRINTING	**PLAY**	TEAM	JUDGE	HYPOTHESIS	STUDY
PAPER	PLAYS	GAME	TRIAL	EXPERIMENT	**TEST**
PRINT	STAGE	BASKETBALL	**COURT**	SCIENTIFIC	STUDYING
PRINTED	AUDIENCE	PLAYERS	CASE	OBSERVATIONS	HOMEWORK
TYPE	THEATER	PLAYER	JURY	SCIENTISTS	NEED
PROCESS	ACTORS	**PLAY**	ACCUSED	EXPERIMENTS	CLASS
INK	DRAMA	PLAYING	GUILTY	SCIENTIST	MATH
PRESS	SHAKESPEARE	SOCCER	DEFENDANT	EXPERIMENTAL	TRY
IMAGE	ACTOR	PLAYED	JUSTICE	**TEST**	TEACHER
PRINTER	THEATRE	BALL	**EVIDENCE**	METHOD	WRITE
PRINTS	PLAYWRIGHT	TEAMS	WITNESSES	HYPOTHESES	PLAN
PRINTERS	PERFORMANCE	BASKET	CRIME	TESTED	ARITHMETIC
COPY	DRAMATIC	FOOTBALL	LAWYER	**EVIDENCE**	ASSIGNMENT
COPIES	COSTUMES	SCORE	WITNESS	BASED	PLACE
FORM	COMEDY	**COURT**	ATTORNEY	OBSERVATION	STUDIED
OFFSET	TRAGEDY	GAMES	HEARING	SCIENCE	CAREFULLY
GRAPHIC	**CHARACTERS**	TRY	INNOCENT	FACTS	DECIDE
SURFACE	SCENES	COACH	DEFENSE	DATA	IMPORTANT
PRODUCED	OPERA	GYM	CHARGE	RESULTS	NOTEBOOK
CHARACTERS	PERFORMED	SHOT	CRIMINAL	EXPLANATION	REVIEW

Three Documents with the word “play”

(numbers & colors → topic assignments)

A Play^{082} is written^{082} to be performed^{082} on a stage^{082} before a live^{093} audience^{082} or before motion^{270} picture^{004} or television^{004} cameras^{004} (for later^{054} viewing^{004} by large^{202} audiences^{082}). A Play^{082} is written^{082} because playwrights^{082} have something

He was listening^{077} to music^{077} coming^{009} from a passing^{043} riverboat. The music^{077} had already captured^{006} his heart^{157} as well as his ear^{119}. It was jazz^{077}. Bix beiderbecke had already had music^{077} lessons^{077}. He wanted^{268} to play^{077} the cornet. And he wanted^{268} to play^{077} jazz^{077}

Jim^{296} plays^{166} the game^{166}. Jim^{296} likes^{081} the game^{166} for one. The game^{166} book^{254} helps^{081} jim^{296}. Don^{180} comes^{040} into the house^{038}. Don^{180} and jim^{296} read^{254} the game^{166} book^{254}. The boys^{020} see a game^{166} for two. The two boys^{020} play^{166} the game^{166}.

LDA Inference

How do we fit an LDA model to the data?

Fitting an LDA model to our data

Use an existing tool!

- Mallet (java)
- gensym (python)
- Many other tools available
  - (see David Blei's website)

Fitting an LDA model to our data

But how are the tools implemented?

And what if we want a slightly different story?

Fitting an LDA model to our data

But how are the tools implemented?

And what if we want a slightly different story?

- Exact inference is intractable.
- Use an approximate algorithm.

Fitting an LDA model to our data

But how are the tools implemented?

And what if we want a slightly different story?

- Exact inference is intractable.
- Use an approximate algorithm.
- Current tools use modern complex algorithms:
  - Fast
  - Scale well to huge number of topics and documents
  - Beyond the scope of this course

Fitting an LDA model to our data

But how are the tools implemented?

And what if we want a slightly different story?

- Exact inference is intractable.
- Use an approximate algorithm.
- Current tools use modern complex algorithms:
  - Fast
  - Scale well to huge number of topics and documents
  - Beyond the scope of this course
- but for fitting a small to medium data, we can use Gibbs sampling.
  - (Gibbs is also our best bet for implementing modifications of LDA)

LDA Gibbs Sampler

Recall:

- Inputs: α, η, K
- Obeserved variables: words, $W = w_{d,n}$
- Unobserved: $\theta = \theta^1, \ldots, \theta^D$, $\beta = \beta^1, \ldots, \beta^K$, $Z = z_{d,n}$

We need to sample from

$p(Z, \theta, \beta | W, \alpha, \eta)$

In Gibbs:

Initialize random z

Then, repeatedly:

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- Sample θ^d based on Z, W, α
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- Sample θ^d based on Z, W, α
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- Sample θ^d based on Z, W, α
- **Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W**
- **Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W**
- ...

These lines are easy:

$$p(z_{d,i} = k \mid Z^{-\mathbf{d},\mathbf{i}}, \theta^d, \beta, W) = \theta^d_k \cdot \beta^k_{W_{d,i}}$$

θ^d_k probability of generating topic k in doc d

$\beta^k_{W_{d,i}}$ probability of generating word $W_{d,i}$ from topic k

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?
We need to sample θ^d from $p(\theta|Z, \alpha)$.

- Given Z, we can derive an MLE estimate of θ^d:

$$\theta_k^d = \frac{count(z_{d,i} = k)}{n_d}$$

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?
We need to sample θ^d from $p(\theta|Z, \alpha)$.

- Given Z, we can derive an MLE estimate of θ^d:

$$\theta_k^d = \frac{count(z_{d,i} = k)}{n_d}$$

- But no. We need to sample. What does it mean to *sample* θ?

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?
We need to sample θ^d from $p(\theta|Z, \alpha)$.

- Given Z, we can derive an MLE estimate of θ^d:

$$\theta_k^d = \frac{count(z_{d,i} = k)}{n_d}$$

- But no. We need to sample. What does it mean to *sample* θ?
- Under the *Bayesian* philosophy, we do not commit to a single estimate of θ. Instead, we have a distribution $p(\theta^d|Z, \alpha)$ of possible θ^d, based on our prior belief α and the data we saw Z.

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?
We need to sample θ^d from $p(\theta|Z, \alpha)$.

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

What does this line mean?
We need to sample θ^d from $p(\theta|Z, \alpha)$.

Because $\theta^d \sim \text{DIRICHLET}(\alpha, K)$, and because dirichlet is *conjugate* to multinomial, we have:

$$\theta^d|Z, \alpha \sim \text{DIRICHLET}(\alpha + c^d)$$

where c^d is s K-dim vector based on counts from Z, with c^d_k is the number of items in document d with topic k.

LDA Gibbs Sampler

- For each $k \in \{1, \cdots, K\}$, sample β^k based on Z, W, η
- **Sample θ^d based on Z, W, α**
- Sample $z_{d,1}$ based on $Z^{-\mathbf{d},\mathbf{1}}$, θ^d, β, W
- Sample $z_{d,2}$ based on $Z^{-\mathbf{d},\mathbf{2}}$, θ^d, β, W
- ...

We need to sample θ^d from $p(\theta|Z, \alpha)$.

$$\theta^d|Z, \alpha \sim \textsc{Dirichlet}(\alpha + c^d)$$

- There are algorithms for sampling from Dirichlet, but we don't need to actualy used them.
- Instead, we will use the the **collapsed** Gibbs sampler.

Collapsed Gibbs sampler

Recall:

- Inputs: α, η, K
- Obeserved variables: words, $W = w_{d,n}$
- Unobserved: $\theta = \theta^1, \ldots, \theta^D$, $\beta = \beta^1, \ldots, \beta^K$, $Z = z_{d,n}$

We need to sample from

$p(Z, \theta, \beta | W, \alpha, \eta)$

But actually, we are ok with just Z. Can we get rid of θ, β?

- If θ, β were discrete, we could marginalize over them.

Collapsed Gibbs sampler

Recall:

- Inputs: α, η, K
- Obeserved variables: words, $W = w_{d,n}$
- Unobserved: $\theta = \theta^1, \ldots, \theta^D$, $\beta = \beta^1, \ldots, \beta^K$, $Z = z_{d,n}$

We need to sample from

$p(Z, \theta, \beta | W, \alpha, \eta)$

But actually, we are ok with just Z. Can we get rid of θ, β?

- If θ, β were discrete, we could marginalize over them.
- But they are continuous, so instead we need to *integrate*

$$p(Z|W, \alpha, \eta) = \int\int p(Z, \theta, \beta | W, \alpha, \eta) \mathrm{d}\theta \mathrm{d}\beta$$

Collapsed Gibbs sampler

Recall:

- Inputs: α, η, K
- Obeserved variables: words, $W = w_{d,n}$
- Unobserved: $\theta = \theta^1, \ldots, \theta^D$, $\beta = \beta^1, \ldots, \beta^K$, $Z = z_{d,n}$

We need to sample from

$p(Z, \theta, \beta | W, \alpha, \eta)$

But actually, we are ok with just Z. Can we get rid of θ, β?

- If θ, β were discrete, we could marginalize over them.
- But they are continuous, so instead we need to *integrate*

$$p(Z|W, \alpha, \eta) = \int\int p(Z, \theta, \beta | W, \alpha, \eta) \mathrm{d}\theta \mathrm{d}\beta$$

Collapsed Gibbs sampler

$$p(z_{d,i} = k|W, Z^{-\mathbf{d},\mathbf{i}}, \alpha, \eta) = \int\int p(z_{d,i} = k, \theta, \beta \mid W, Z^{-\mathbf{d},\mathbf{i}}, \alpha, \eta)\mathrm{d}\theta\mathrm{d}\beta$$

$$= \int p(z_{d,i} = k|\theta)p(\theta|\alpha)\mathrm{d}\theta \int p(w_{d,i} = v|W^{-\mathbf{d},\mathbf{i}}, z_{d,i} = k, Z^{-\mathbf{d},\mathbf{i}}, \beta)p(\beta|\eta)\mathrm{d}\eta$$

You don't really need to know how to integrate!
Just remember that for Dirichlet:

$$\int p(x|data, \theta)p(\theta|\alpha)\mathrm{d}\theta = \frac{c_x + \alpha}{|data| + K\alpha}$$

Where c_x is the count of event x in the data, and $|data| = \sum_{x'} c'_x$ is the number of samples in the data.

Collapsed Gibbs Sampler

Just remember that for Dirichlet:

$$\int p(x|data,\theta)p(\theta|\alpha)\mathrm{d}\theta = \frac{c_x + \alpha}{|data| + K\alpha}$$

Where c_x is the count of event x in the data, and $|data| = \sum_{x'} c'_x$ is the number of samples in the data.
Use this rule twice (once for each $\int$), and get:

$$p(z_{d,i} = k|Z^{-\mathbf{d},\mathbf{i}}, \alpha, \eta, w_i) = \frac{c_k^d + \alpha}{\sum_{k'} c_{k'}^d + K\alpha} \frac{v_{w_i}^k + \eta}{\sum_{i'} v_{w_{i'}}^k + |V|\eta}$$

- c_k^d number of words in doc d with topic k in $Z^{-\mathbf{d},\mathbf{i}}$
- $v_{w_i}^k$ number of times word w_i is assigned to topic k in $Z^{-\mathbf{d},\mathbf{i}}$
- K number of topics
- $|V|$ vocabulary size

Collapsed Gibbs Sampler

Rule of Thumb

In MLE land:

$$p(x_n = k | x_1, x_2, \ldots, x_{n-1}) = \frac{count(k)}{n-1}$$

In Dirichlet-prior α land:

$$p(x_n = k | x_1, x_2, \ldots, x_{n-1}, \alpha) = \frac{count(k) + \alpha}{n - 1 + K\alpha}$$

Derivation in MacKay and Peto (1994)

Collapsed Gibbs Sampler

- Initialize random topics Z
- For many iterations, for each document d, for each word i:
  - forget $z_{d,i}$ getting $Z^{-\mathbf{d},\mathbf{i}}$
  - sample new assignment for $z_{d,i}$ based on equation below.

$$p(z_{d,i} = k | Z^{-\mathbf{d},\mathbf{i}}, \alpha, \eta, w_i) = \frac{c_k^d + \alpha}{\sum_{k'} c_{k'}^d + K\alpha} \frac{v_{w_i}^k + \eta}{\sum_{i'} v_{w_{i'}}^k + |V|\eta}$$

c_k^d number of words in doc d with topic k in $Z^{-\mathbf{d},\mathbf{i}}$

$v_{w_i}^k$ number of times word w_i is assigned to topic k in $Z^{-\mathbf{d},\mathbf{i}}$

K number of topics

$|V|$ vocabulary size

LDA Evaluation

We have topics, are they good?

LDA Evaluation

Internal Evaluation

If we want to compare two different LDA models on the same data:

- Compare the Probability that is assigned to the data by each model.
- Higher probability $\rightarrow$ better model

LDA Evaluation

Internal Evaluation

If we want to compare two different LDA models on the same data:

- Compare the Probability that is assigned to the data by each model.
- Higher probability $\rightarrow$ better model
- But this does not tell us much about how useful the topics are...

LDA Evaluation

Internal Evaluation

If we want to compare two different LDA models on the same data:

- Compare the Probability that is assigned to the data by each model.
- Higher probability $\rightarrow$ better model
- But this does not tell us much about how useful the topics are...

External (task-based) Evaluation

- Use the LDA topics as features in another task
- Measure the accuracy of the other task

LDA Evaluation

Internal Evaluation

If we want to compare two different LDA models on the same data:

- Compare the Probability that is assigned to the data by each model.
- Higher probability $\rightarrow$ better model
- But this does not tell us much about how useful the topics are...

External (task-based) Evaluation

- Use the LDA topics as features in another task
- Measure the accuracy of the other task
- Good! But we need to have a task that we can automatically measure.

LDA Evaluation

Human Evaluation

If we just want to know if our topics are “good” we can ask people.

- But what is a good topic?

LDA Evaluation

Human Evaluation

If we just want to know if our topics are “good” we can ask people.

- But what is a good topic?
- “Intruder Detection”
  - Take top words from a topic.
  - Insert a random word which is high in another topic.
  - Can a human identify the random word?
  - Yes $\rightarrow$ good topic

Other Applications of LDA

Change the definition of Document

Selectional Preferences

Take parsed corpus:

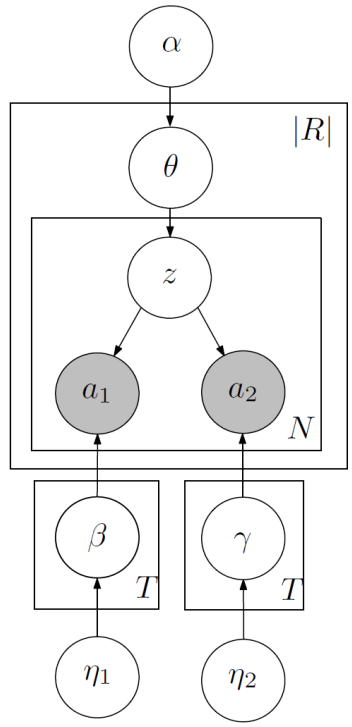
Documents each Verb is a document

Words each subject of a verb is a “word” in the document

Topics each topic is one “kind” of arguments

Topic t	Arg1	Relations which assign highest probability to t	Arg2
18	The residue - The mixture - The reaction mixture - The solution - the mixture - the reaction mixture - the residue - The reaction - the solution - The filtrate - the reaction - The product - The crude product - The pellet - The organic layer - Thereto - This solution - The resulting solution - Next - The organic phase - The resulting mixture - C.)	was treated with, is treated with, was poured into, was extracted with, was purified by, was diluted with, was filtered through, is disolved in, is washed with	EtOAc - CH2Cl2 - H2O - CH.sub.2Cl.sub.2 - H.sub.2O - water - MeOH - NaHCO3 - Et2O - NHCl - CHCl.sub.3 - NHCl - dropwise - CH2Cl.sub.2 - Celite - Et.sub.2O - Cl.sub.2 - NaOH - AcOEt - CH2C12 - the mixture - saturated NaHCO3 - SiO2 - H2O - N hydrochloric acid - NHCl - preparative HPLC - to0 C
151	the Court - The Court - the Supreme Court - The Supreme Court - this Court - Court - The US Supreme Court - the court - This Court - the US Supreme Court - The court - Supreme Court - Judge - the Court of Appeals - A federal judge	will hear, ruled in, decides, upholds, struck down, overturned, sided with, affirms	the case - the appeal - arguments - a case - evidence - this case - the decision - the law - testimony - the State - an interview - an appeal - cases - the Court - that decision - Congress - a decision - the complaint - oral arguments - a law - the statute
211	President Bush - Bush - The President - Clinton - the President - President Clinton - President George W. Bush - Mr. Bush - The Governor - the Governor - Romney - McCain - The White House - President - Schwarzenegger - Obama	hailed, vetoed, promoted, will deliver, favors, denounced, defended	the bill - a bill - the decision - the war - the idea - the plan - the move - the legislation - legislation - the measure - the proposal - the deal - this bill - a measure - the program - the law - the resolution - efforts - the agreement - gay marriage - the report - abortion
224	Google - Software - the CPU - Clicking - Excel - the user - Firefox - System - The CPU - Internet Explorer - the ability - Program - users - Option - SQL Server - Code - the OS - the BIOS	will display, to store, to load, processes, cannot find, invokes, to search for, to delete	data - files - the data - the file - the URL - information - the files - images - a URL - the information - the IP address - the user - text - the code - a file - the page - IP addresses - PDF files - messages - pages - an IP address

Table 1: Example argument lists from the inferred topics. For each topic number t we list the most



Model is slightly different - topic generates two groups of things.

(how would you change the Gibbs sampler?)

Change the definition of Document

Beyond NLP

Dataset of users who watched movies

Documents each user is a document

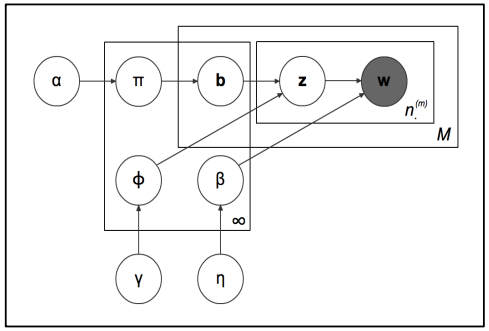
Words each movie is a word

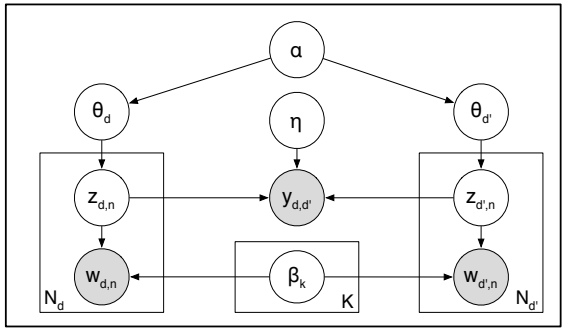
Topics each topic is a “taste” or “genre”

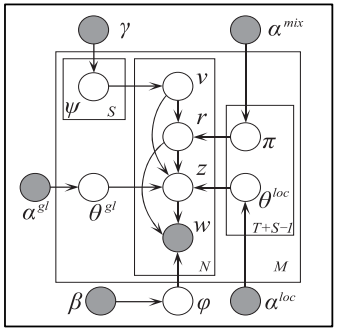
- High topic-word prob: movie belong to genre
- High topic-doc prob: user likes genre

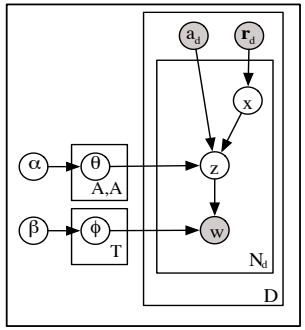
Can recommend new movies to users

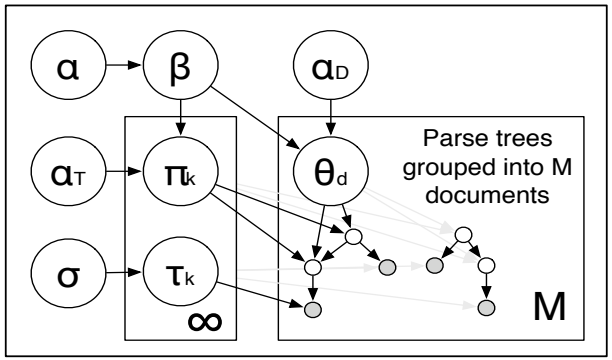
Extending LDA

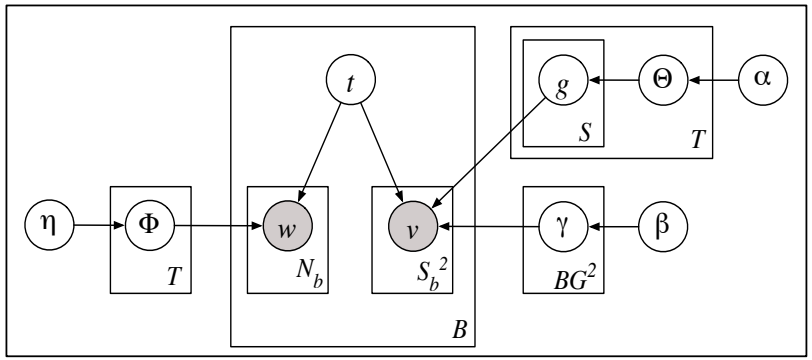


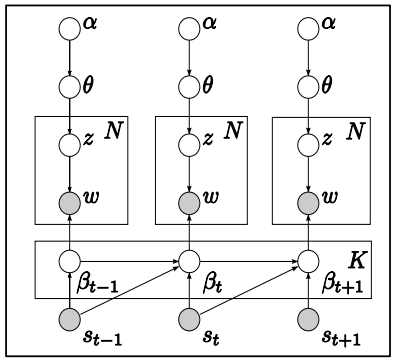












- LDA can be **embedded in more complicated models**, embodying further intuitions about the structure of the texts.
- E.g., it can be used in models that account for syntax, authorship, word sense, dynamics, correlation, hierarchies, and other structure.

Summary

Unsupervised Learning

- Define generative story
- Include hidden (“latent”) variables
- Find probable assignments to latent variables
- Can use Gibbs sampling

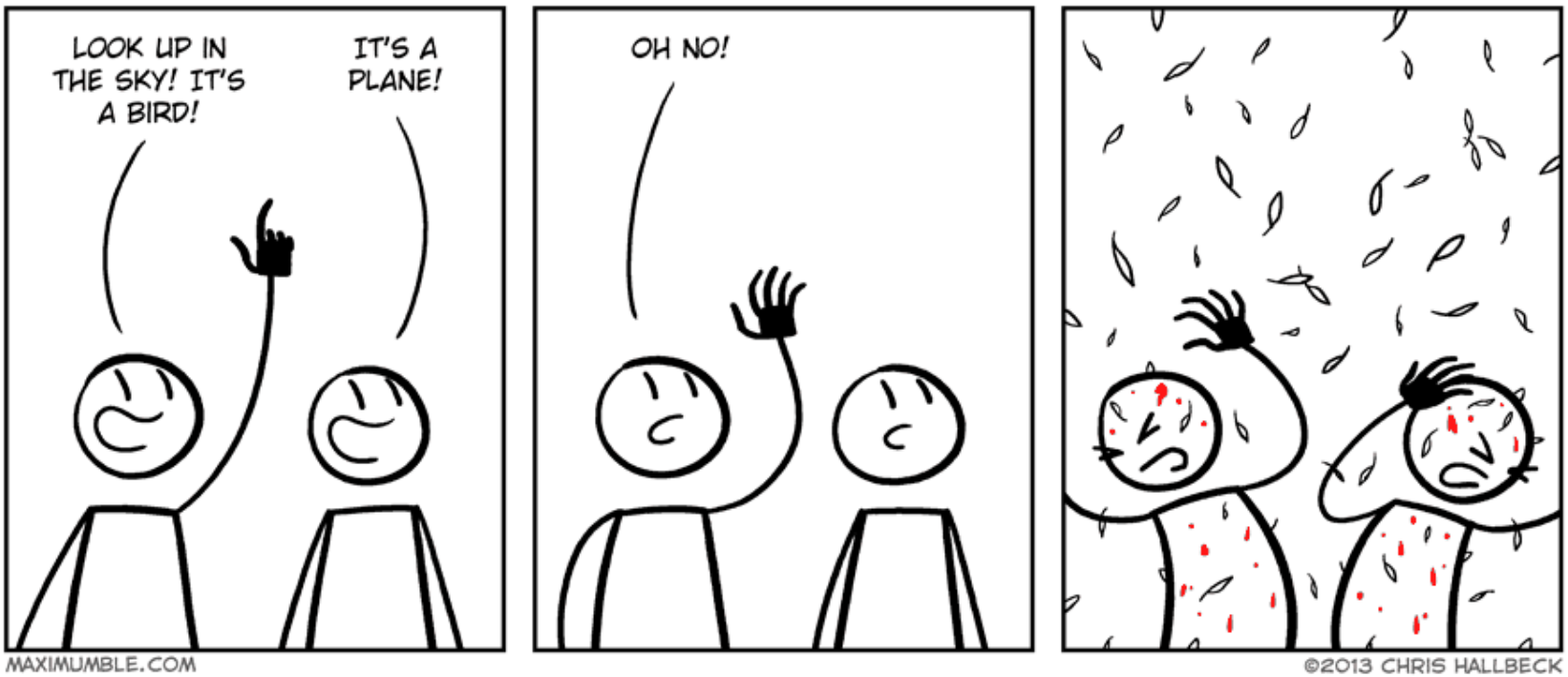
Unsupervised Learning

- Define generative story
- Include hidden ("latent") variables
- Find probable assignments to latent variables
- Can use Gibbs sampling

Topic Modeling / LDA

- A very powerful and useful model. **Use it**
- Generative story for LDA
- Dirichlet distributions $\rightarrow$ can encourage sparsity
- Examples of LDA usage
- Gibbs sampler for LDA (briefly)
  - relevant for every model with dirichlet
- Evaluation: quantify human judgement ("intruder detection")
- Creative definition of documents

Next Time


LOOK UP IN THE SKY! IT'S A BIRD!
IT'S A PLANE!
OH NO!
MAXIMUMBLE.COM
©2013 CHRIS HALLBECK

Next Time

Benedict Evans
@BenedictEvans

WHAT DO WE WANT?
Natural language processing!
WHEN DO WE WANT IT?
Sorry, when do we want what?