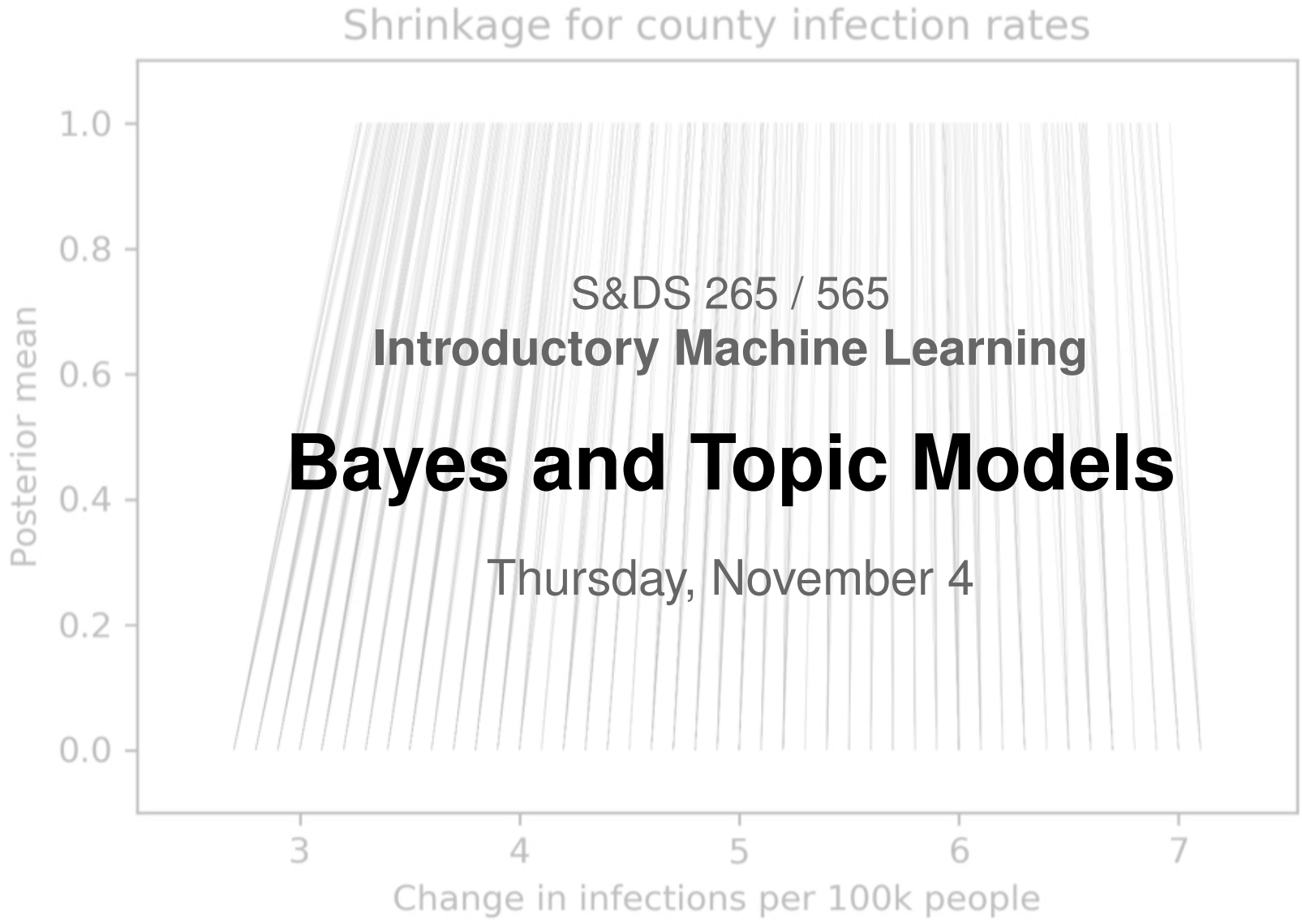


S&DS 265 / 565

Introductory Machine Learning

Bayes and Topic Models

Thursday, November 4

Yale

Quick notes

- Assignment 5 is out; due week from today (Nov 11)
- Two more to go!
- Midterm solutions posted

For Today

- Bayes (continued)
- Overview of topic models

Bayesian Inference

The parameter θ of a model is viewed as a random variable. Inference usually carried out as follows:

- Choose a *generative model* $p(x \mid \theta)$ for the data.
- Choose a *prior distribution* $\pi(\theta)$ that expresses beliefs about the parameter before seeing any data.
- After observing data $\mathcal{D}_n = \{x_1, \ldots, x_n\}$, update beliefs and calculate the *posterior distribution* $p(\theta \mid \mathcal{D}_n)$.

Bayes' Theorem

A simple consequence of conditional probability:

$$\begin{aligned}\mathbb{P}(A \mid B) &= \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \\ &= \frac{\mathbb{P}(B \mid A)\, \mathbb{P}(A)}{\mathbb{P}(B)}\end{aligned}$$

Bayes' Theorem

The posterior distribution can be written as

$$p(\theta \mid x_1, \ldots, x_n) = \frac{p(x_1, \ldots, x_n \mid \theta)\pi(\theta)}{p(x_1, \ldots, x_n)} = \frac{\mathcal{L}_n(\theta)\pi(\theta)}{c_n} \propto \mathcal{L}_n(\theta)\pi(\theta)$$

where $\mathcal{L}_n(\theta)$ is the *likelihood function* and

$$c_n = p(x_1, \ldots, x_n) = \int p(x_1, \ldots, x_n \mid \theta)\pi(\theta)d\theta = \int \mathcal{L}_n(\theta)\pi(\theta)d\theta$$

is the normalizing constant, which is also called *evidence*.

Important Example

Take model $X \sim \text{Bernoulli}(\theta)$.

This is a "coin flip": $X = 1$ means "heads" and $X = 0$ means "tails."

Natural prior is Beta(α, β) distribution

$$\pi_{\alpha,\beta}(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

Important Example

Take model $X \sim \text{Bernoulli}(\theta)$.

Natural prior is Beta(α, β) distribution

$$\pi_{\alpha,\beta}(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

The scaling constant is a little scary:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

where $\Gamma(\cdot)$ is the "Gamma function"

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x}\, dx$$

---

https://en.wikipedia.org/wiki/Gamma_function

Important Example

$X \sim$ Bernoulli(θ) with data $\mathcal{D}_n = \{x_1, \ldots, x_n\}$. Prior Beta($\alpha, \beta$) distribution

$$\pi_{\alpha,\beta}(\theta) \;\propto\; \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

Let $s = \sum_{i=1}^{n} x_i$ be the number of "heads"

Posterior distribution $\theta \,|\, \mathcal{D}_n$ is another beta distribution!

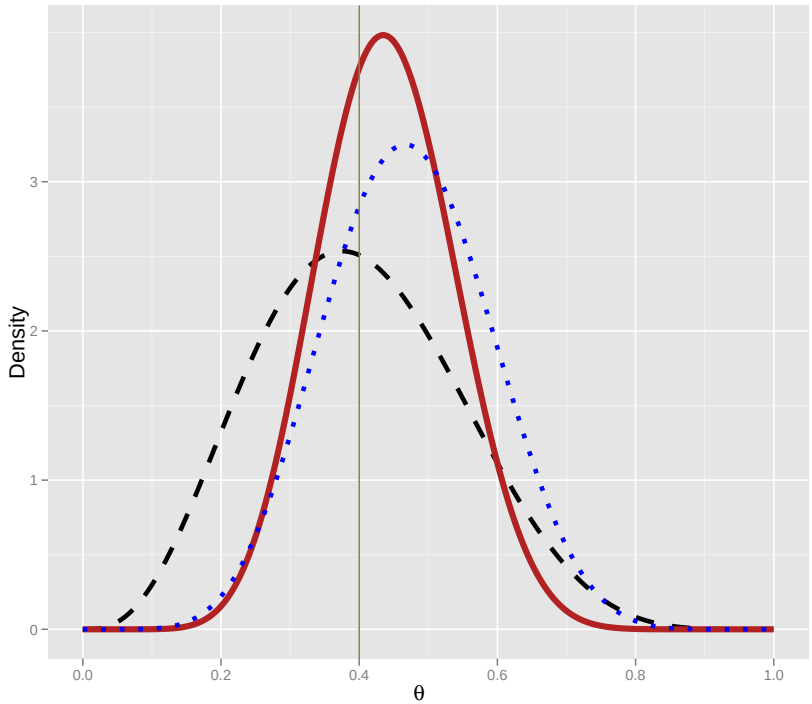
Specifically, with

$$\widetilde{\alpha} = \alpha + \text{number of heads} = \alpha + s$$
$$\widetilde{\beta} = \beta + \text{number of tails} = \beta + n - s$$

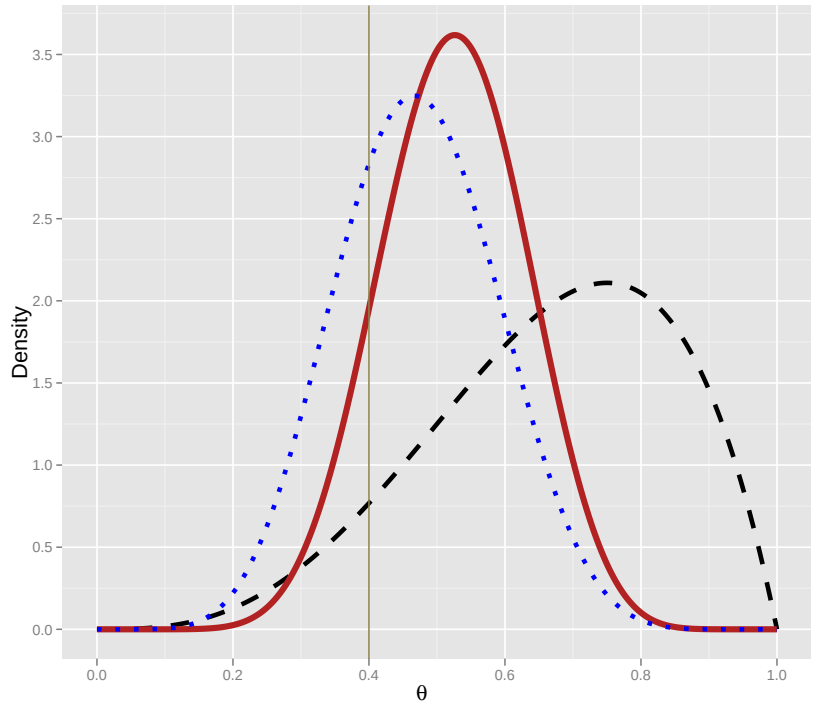
Showing this just uses the simple fact that $\theta^{\alpha-1}\theta^{x} = \theta^{x+\alpha-1}$

Example

$n = 15$ points sampled as $X \sim \text{Bernoulli}(\theta = 0.4)$, with $s = 7$ heads.



Prior A



Prior B

Prior distribution (black-dashed), likelihood function (blue-dotted), posterior distribution (red-solid).

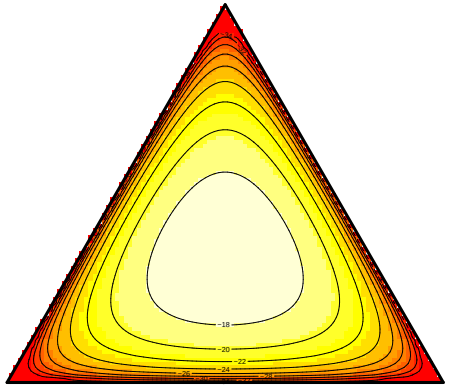
Dirichlet

Multinomial model with Dirichlet prior is generalization of the Bernoulli/Beta model.

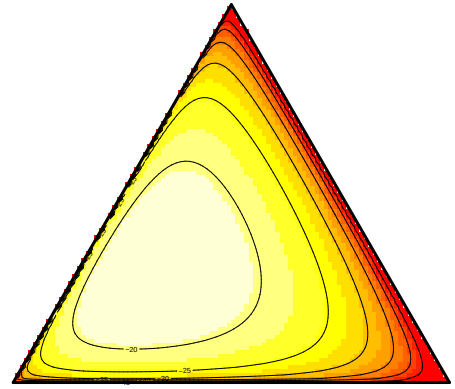
$$\text{Dirichlet}_\alpha(\theta) \;\propto\; \theta_1^{\alpha_1 - 1}\, \theta_2^{\alpha_2 - 1} \;\cdots \theta_K^{\alpha_K - 1}$$

where $\alpha = (\alpha_1, \ldots, \alpha_K) \in \mathbb{R}_+^K$ is a non-negative vector.

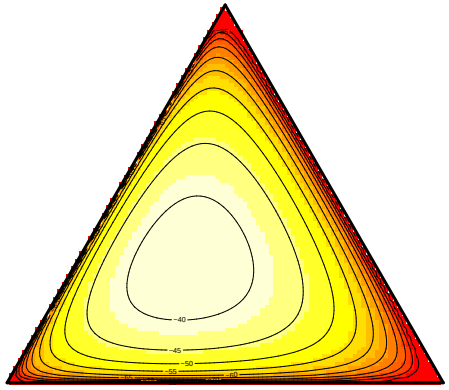
Example



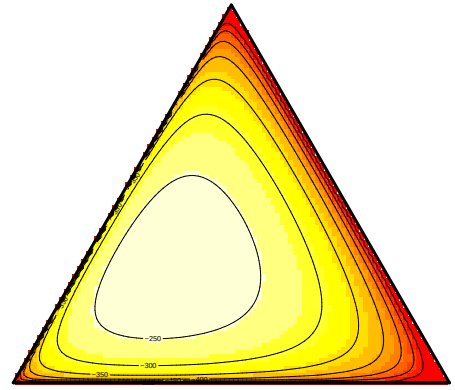
prior with Dirichlet(6,6,6)



likelihood function with $n = 20$



posterior distribution with $n = 20$



posterior distribution with $n = 200$

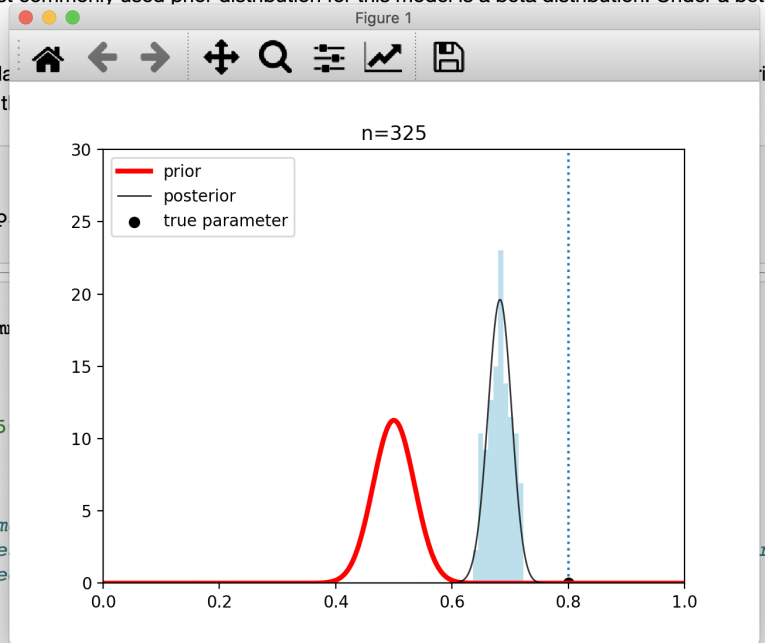
Notebook

Demo code for Bayesian analysis

In this notebook we illustrate some of the basic models and priors for Bayesian inference. These concepts will be important for our discussions about "topic models."

First, we illustrate the situation where the parameter θ that we are modeling is a Bernoulli parameter. This can be thought of as the probability that flipping a certain coin comes up heads. The most commonly used prior distribution for this model is a beta distribution. Under a beta prior, the posterior distribution is again a beta distribution.

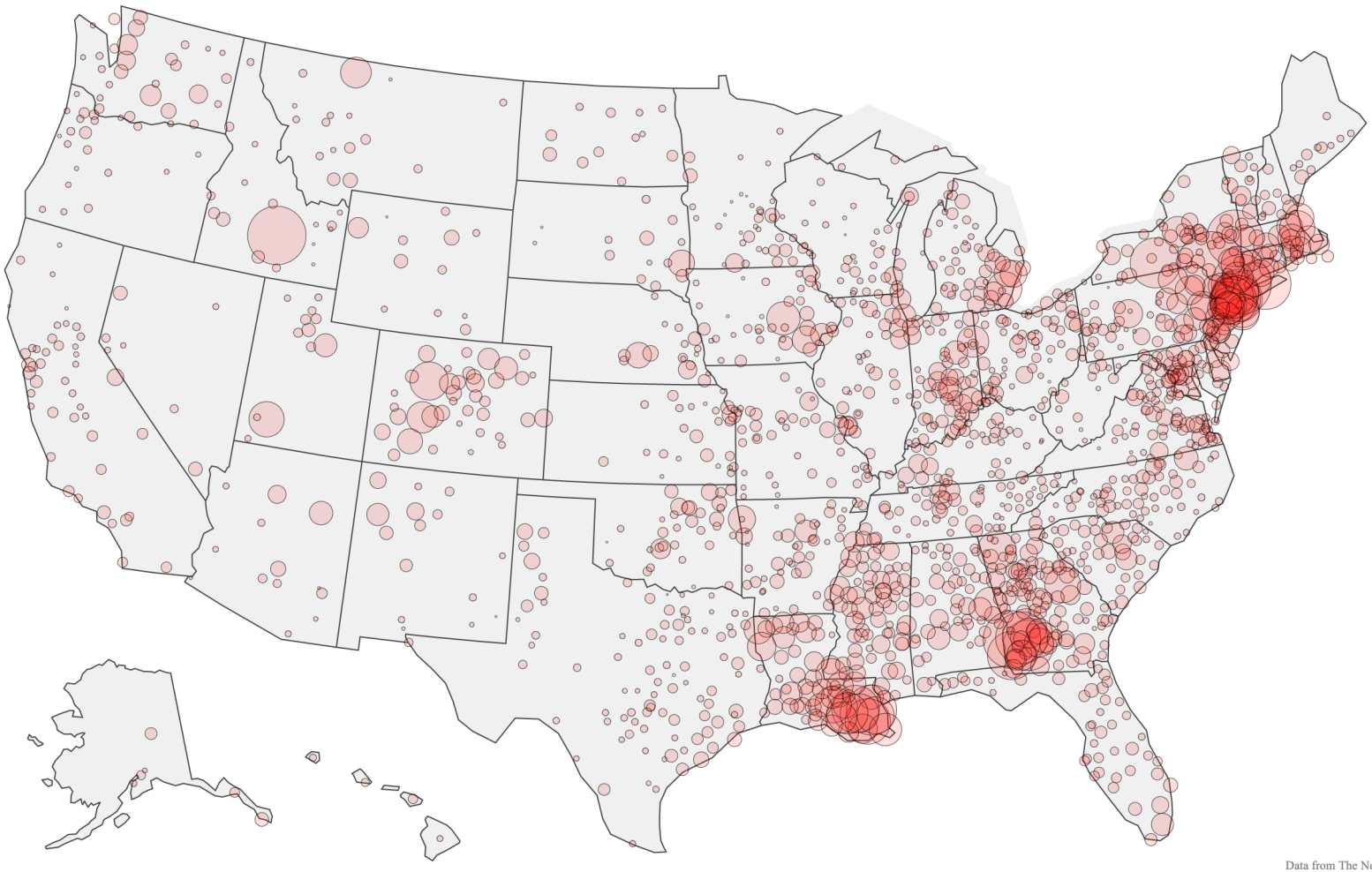
This is illustrated in the following simula ior becomes concentrated on the true parameter. But as the variance of t



In [5]:
```
import os, gzip
import numpy as np
import matplotlib.pyplot as p
```

In [*]:
```
%matplotlib qt
from scipy.special import gam
from scipy import random
from scipy.stats import beta

theta = np.linspace(0,1,num=5
fig = plt.figure(1)
plt.ion()

# The following are the param                    the
# variance of the prior decre                    required for
# the posterior to be centere

scale = 100
a0 = scale*1
b0 = scale*1

sample_size = 100
```

Covid Cases per 100,000 people (in April 2020)



Hierarchical models

We are interested in modeling the infection rates across counties

In a Bayesian hierarchical model, we tie together our inferences across regions

In each county c we test n_c people (at random) and observe how many people Y_c have Covid-19.

$Y_c \sim \text{Binomial}(n_c, \theta_c)$. This is equivalent to n_c coin flips (Bernoulli) each with probability of heads θ_c.

A simple hierarchical model

For each county c,

$$\theta_c \sim F$$
$$Y_c \,|\, \theta_c \sim \text{Binomial}(n_c, \theta_c)$$

This ties together the parameters and makes better use of the data

A simple hierarchical model

A simple transformation makes the use of Gaussians more appropriate.

Transform by $\psi_c \equiv \log\left(\frac{\theta_c}{1-\theta_c}\right)$ to use Gaussian approximation:

$$\psi_c \sim N(\mu, \tau^2)$$
$$Z_c \,|\, \psi_c \sim N(\psi_c, \sigma_c^2)$$

- Posterior distribution may not be written down explicitly
- But we can sample from it
- This approximates the posterior as a mixture of Gaussians

A run from April 2020

File Edit View Insert Cell Kernel Widgets Help Trusted

Run Code Dashboard View:

In [1]:
```
import covid19 as cvd
import covid19_predict as cvd_predict
```

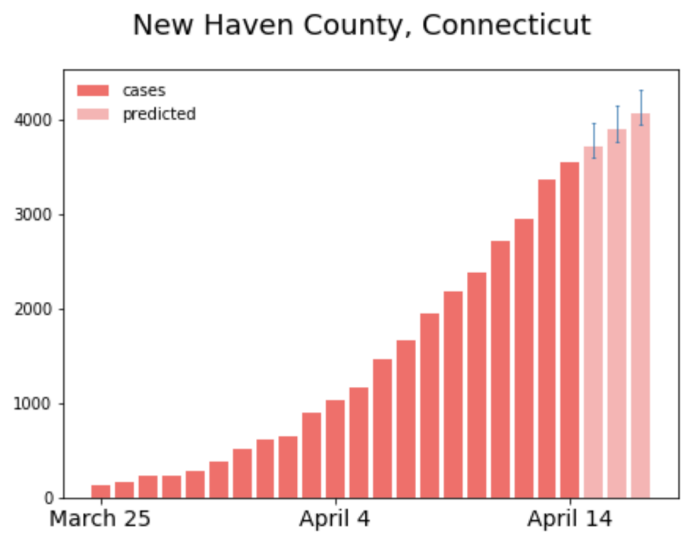
```
covid19: Most recent NY Times data: Tuesday April 14, 2020
covid19_predict: initializing for simulation
covid19_predict: running Gibbs sampler: n=3193, B=10000
covid19_predict: making predictions
```

In [2]:
```
cvd_predict.df[cvd_predict.df['county']=='New Haven']
```

Out[2]:

	date	county	state	cases	deaths	population	cases_per_100k	delta	delta_bar	delta_95	delta_05	cases_predicted	cases_95_credible
83	2020-04-14	New Haven	Connecticut	3543	151	854757	414.5	21.6	20.36	28.79	14.42	3716	3789

In [3]:
```
_ = cvd_predict.plot_predictions_for_addr('New Haven, CT', show=True)
```



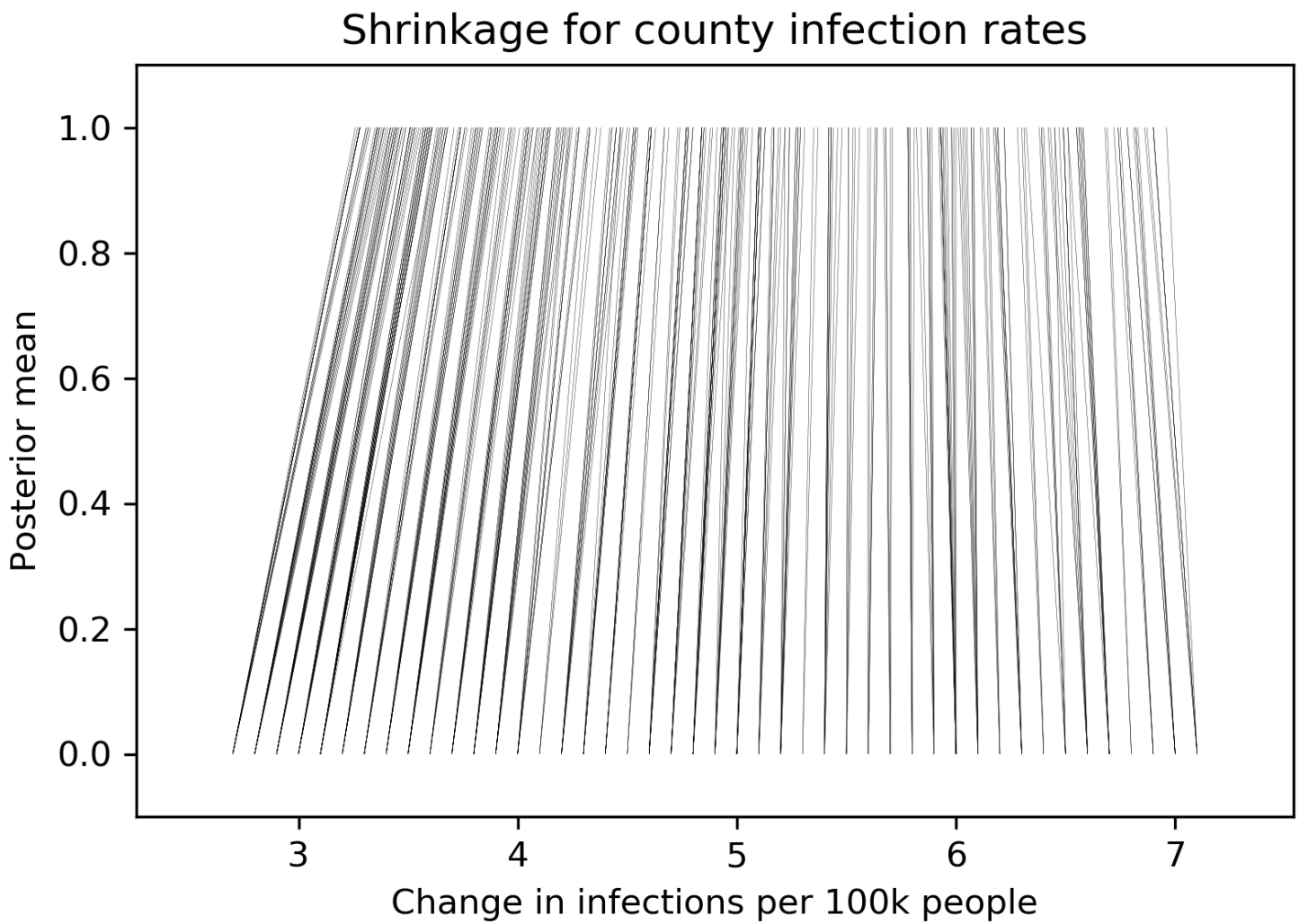
Shrinkage

Tying the parameters together results in the small probabilities being increased and the larger probabilities being decreased.

This is called “shrinkage.” The posterior estimates are closer together than the raw frequencies.

This is different than but conceptually related to shrinkage from regularization

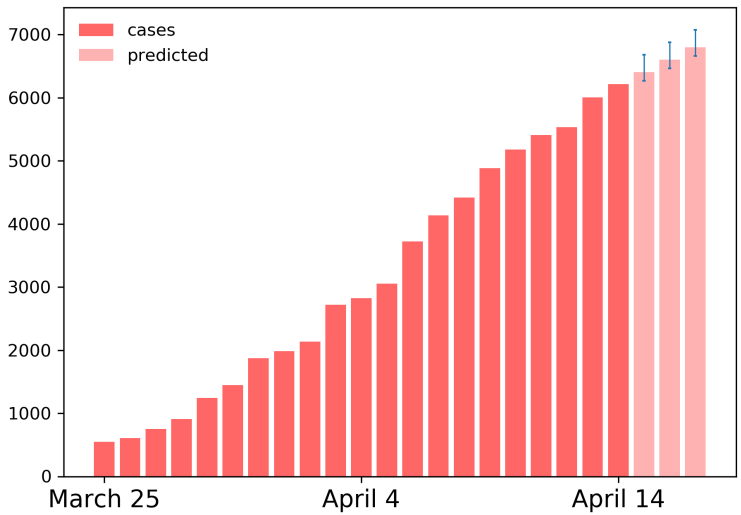
Shrinkage



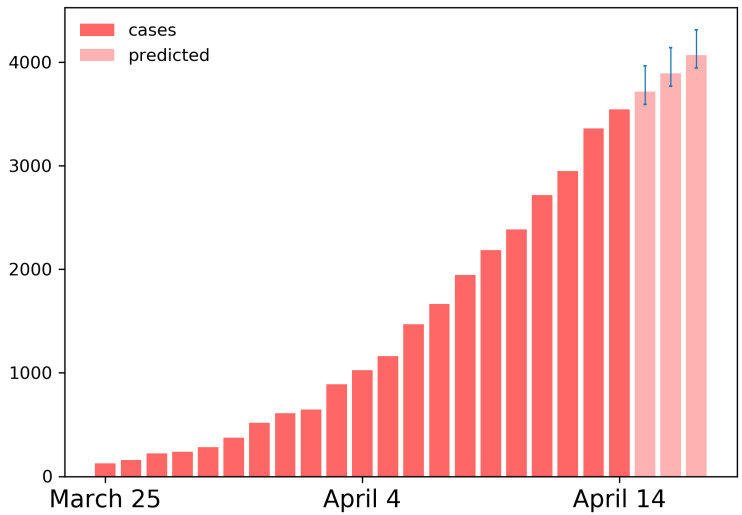
This is different than but conceptually related to shrinkage from regularization

Examples

Fairfield County, Connecticut



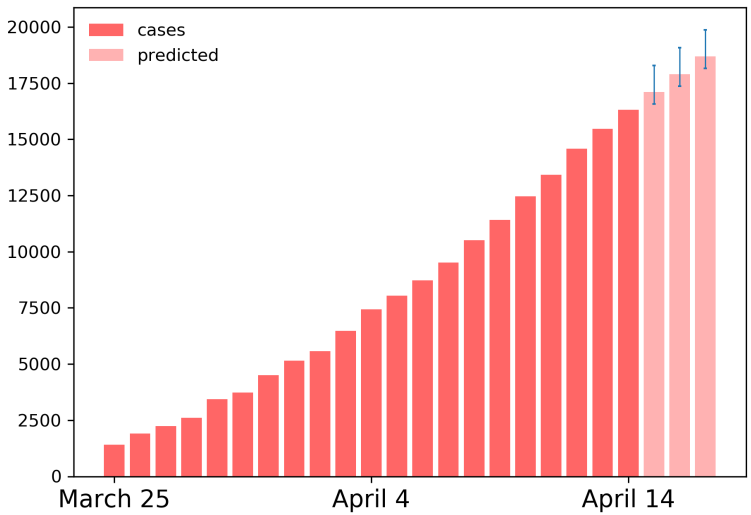
New Haven County, Connecticut



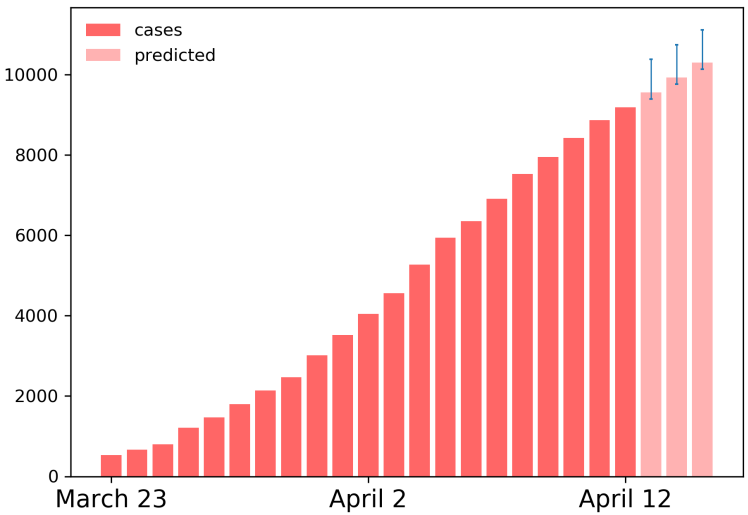
The “credible intervals” trap 95% of the sampled parameters under the simulation.

Examples

Cook County, Illinois



Los Angeles County, California



The “credible intervals” trap 95% of the sampled parameters under the simulation.

Another Example: Election Forecasting

https://projects.economist.com/us-2020-forecast/president/how-this-works

The Economist | Today | Weekly edition | ≡ Menu

Forecasting the US elections

The Economist is analysing polling, economic and demographic data to predict America's elections in 2020

→ **Read more of our election coverage**

President | Senate | House

National forecast
How this works

COMPETITIVE STATES
Arizona
Florida
Georgia
Iowa
Michigan
Nevada
New Hampshire
North Carolina
Ohio
Pennsylvania
Texas
Wisconsin

ALL STATES
Alabama

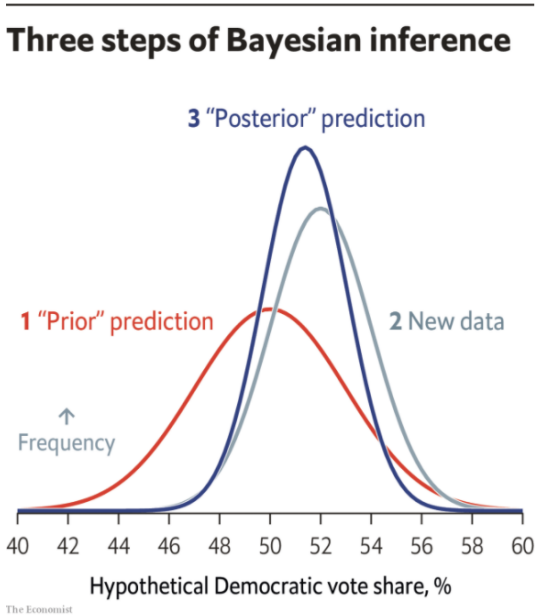
How The Economist presidential forecast works

THIS YEAR, *The Economist* is publishing its first-ever statistical forecast of an American presidential election. Developed with the assistance of Andrew Gelman and Merlin Heidemanns, political scientists at Columbia University, our model calculates Joe Biden's and Donald Trump's probabilities of winning each individual state and the election overall. Its projections will be updated every day at https://projects.economist.com/us-2020-forecast/president.

In another first, we are publishing the source code for what we believe to be the most innovative section of the model. All readers are welcome to download it, explore how it works, tweak its parameters and run it

Another Example: Election Forecasting

https://projects.economist.com/us-2020-forecast/president/how-this-works



Back to Bayes-ics

Readers acquainted with the workings of similar forecasting models may be surprised that the phrase "state polls" has not yet entered the equation. This exclusion is by design. Our model follows a logical structure first developed by Thomas Bayes, an 18th-century reverend whose ideas have shaped a large and growing family of statistical techniques. His approach works in two stages. First, before conducting a study, researchers explicitly state what they believe to be true, and how confident they are in that belief. This is called a "prior". Next, after acquiring data, they update this prior to reflect the new information—gaining more confidence if it confirms the prior, and generally becoming more uncertain if it refutes the prior (though not if the new numbers are so definitive that leave little room for doubt). In this framework, the expected distribution of potential vote shares in each state derived above is the prior, and state polls that trickle in during the course of the campaign are the new data. The result—a "posterior", in Bayesian lingo—is our forecast.

Reading

- See notes on Bayesian inference on iML site (`bayes-notes.pdf`)
- These are more advanced, from S&DS 365.
- Not necessary to understand everything—but you should be able to do some of the basic "coin flipping" calculations
- We'll build on this when discussing topic models

Summary: Bayes

- Computing with mixtures uses basic probabilistic reasoning
- Bayesian inference is popular in ML
- In a Bayesian approach, the parameters are random, and the data are fixed.
- Bayesian hierarchical models tie together data using latent variables

Next up: Topic models

- High level intro to topic models
- Use of latent variables, mixtures
- Work through a Jupyter notebook
- More details and examples on Thursday

Intro to Topic Modeling

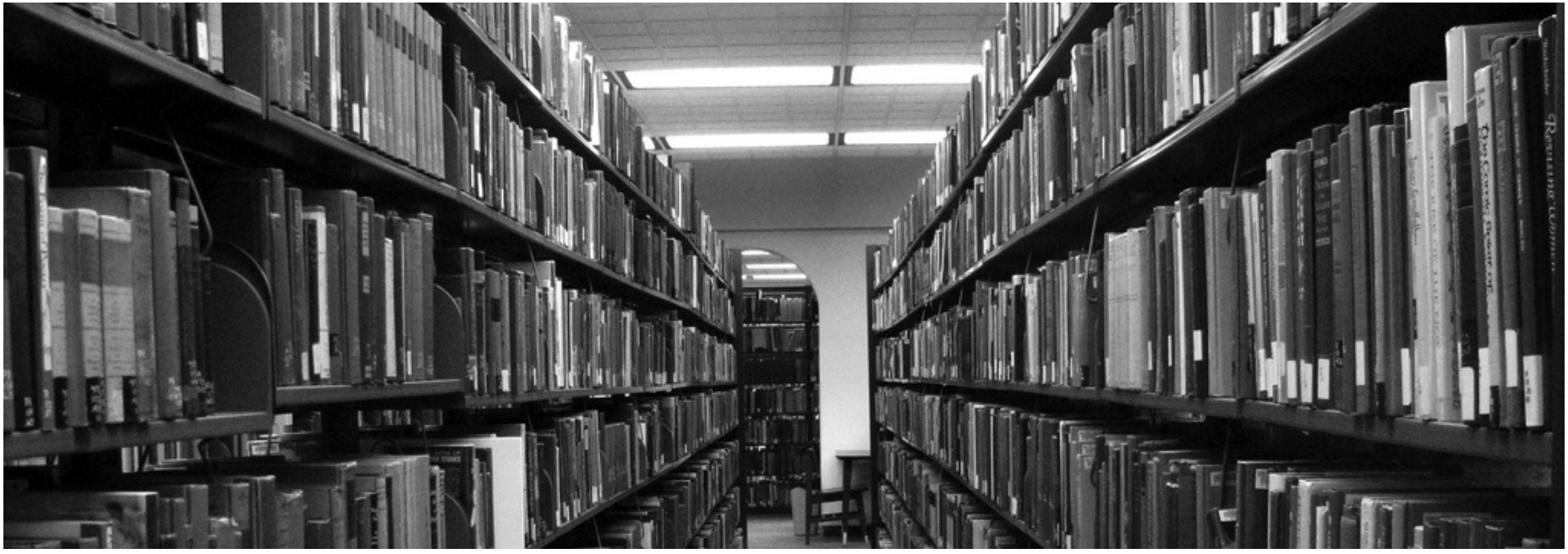
Readings:

`http://www.cs.columbia.edu/~blei/topicmodeling.html`

A survey paper describing many of these ideas in more detail is here:

`https://cacm.acm.org/magazines/2012/4/147361-probabilistic-topic-models/fulltext`

Topic modeling



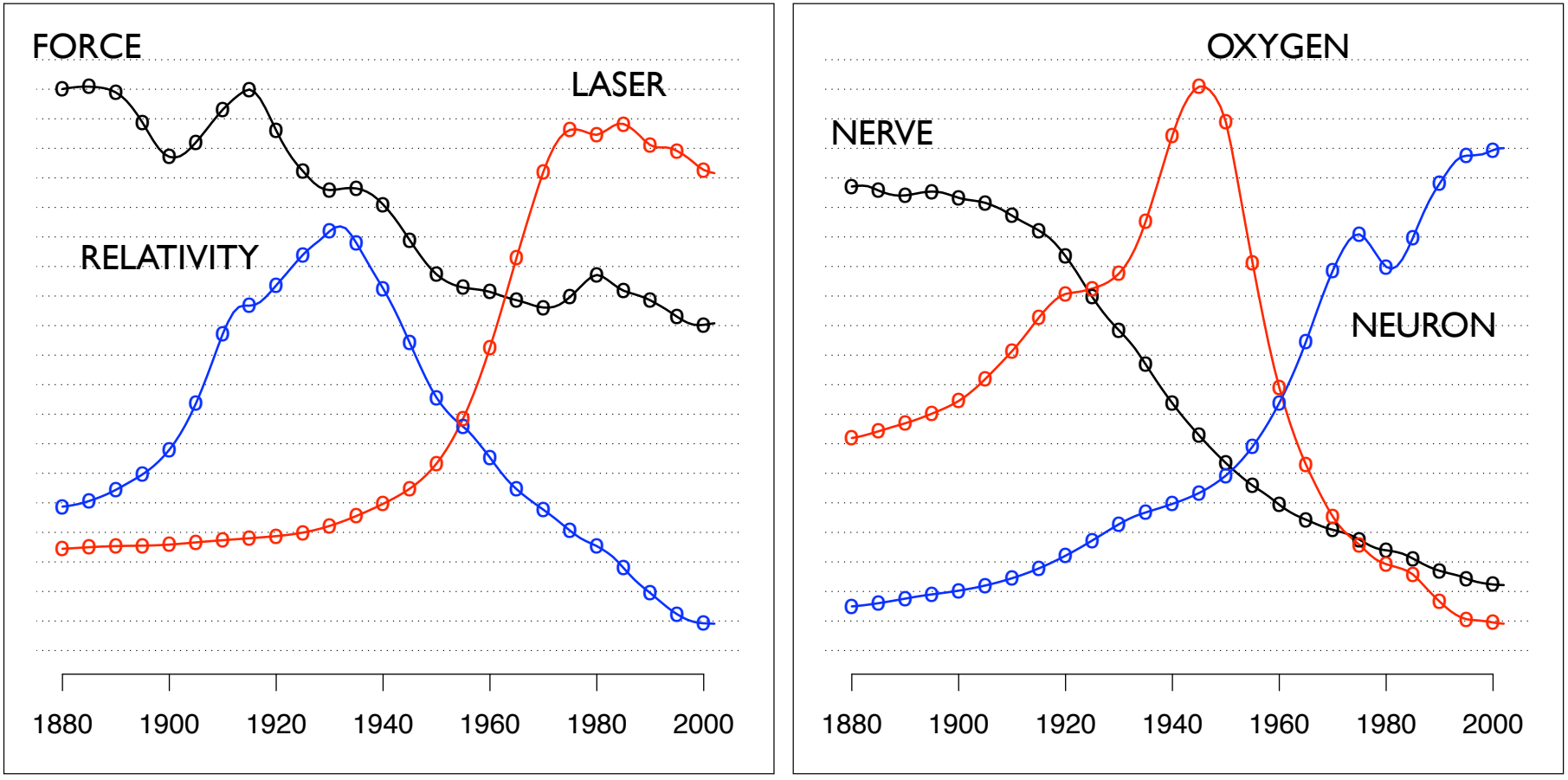
Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives.

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.

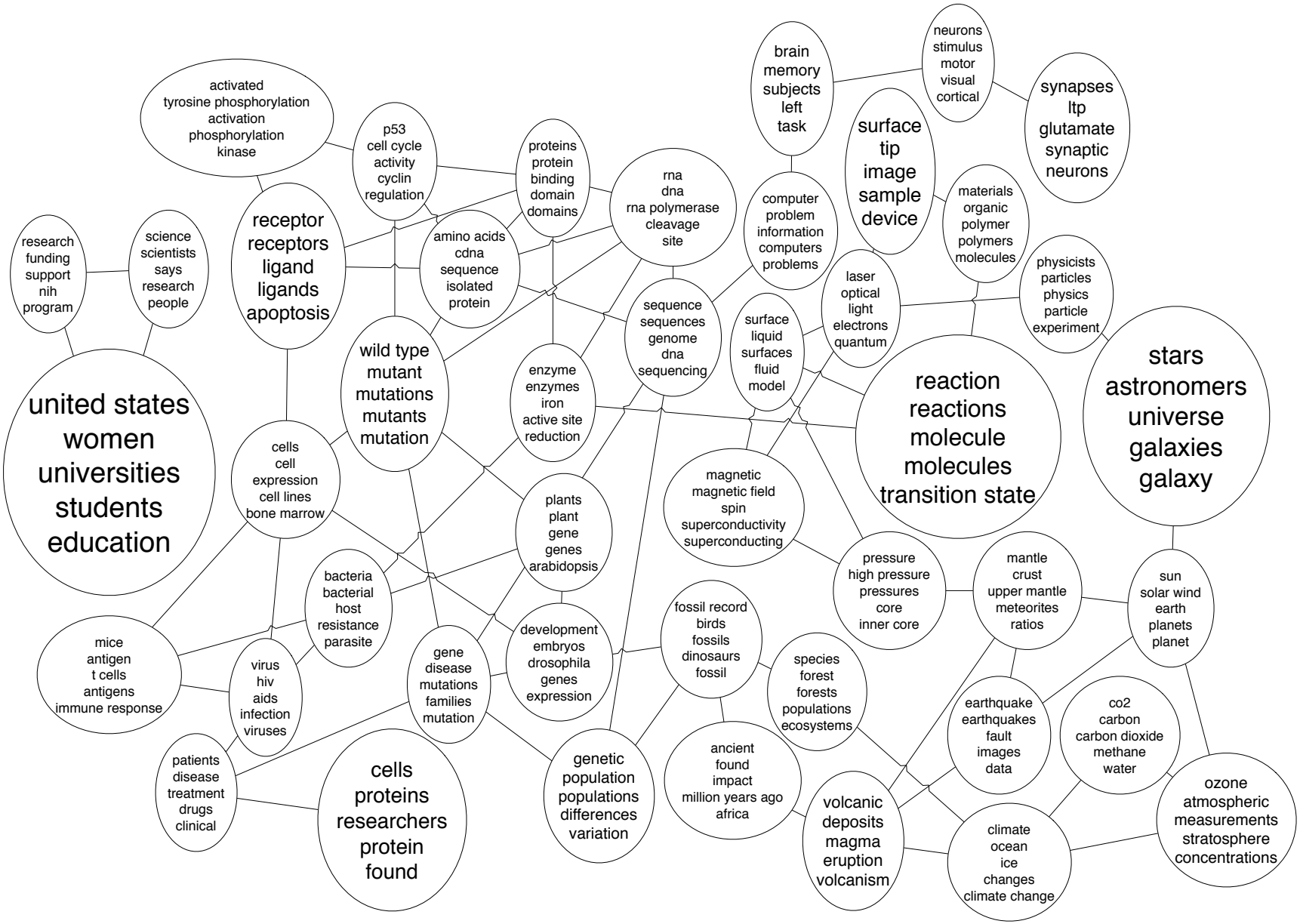
Discover topics from a corpus

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

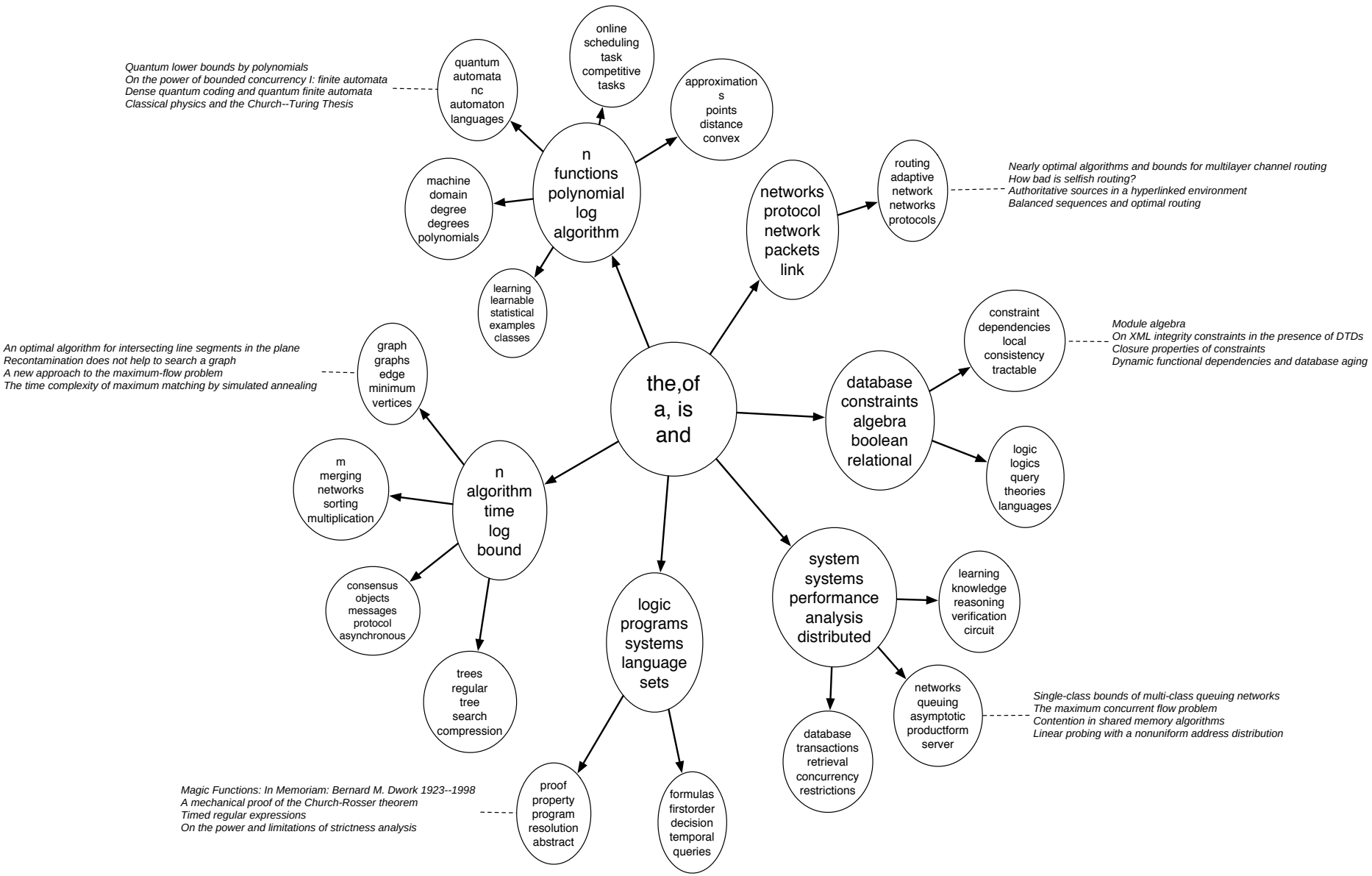
Model the evolution of topics over time



Model connections between topics



Find hierarchies of topics



Annotate images

SKY WATER TREE
MOUNTAIN PEOPLE

SCOTLAND WATER
FLOWER HILLS TREE

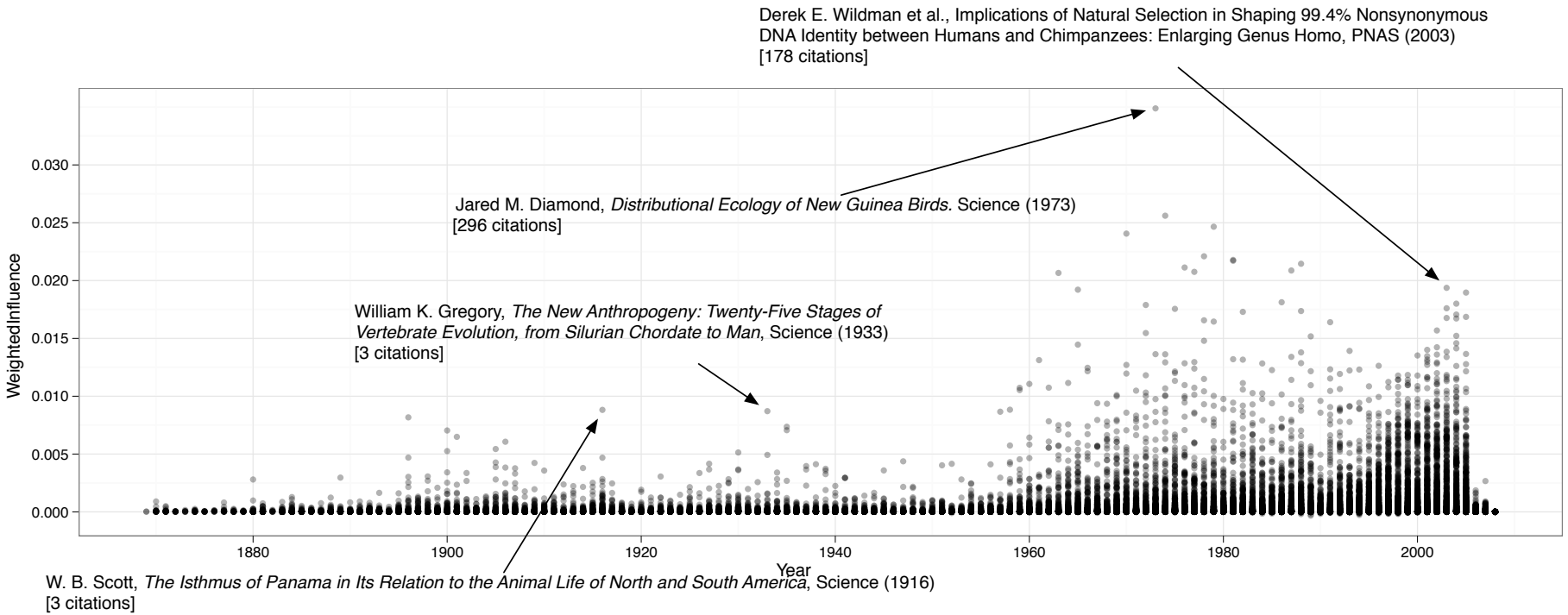
SKY WATER BUILDING
PEOPLE WATER

FISH WATER OCEAN
TREE CORAL

PEOPLE MARKET PATTERN
TEXTILE DISPLAY

BIRDS NEST TREE
BRANCH LEAVES

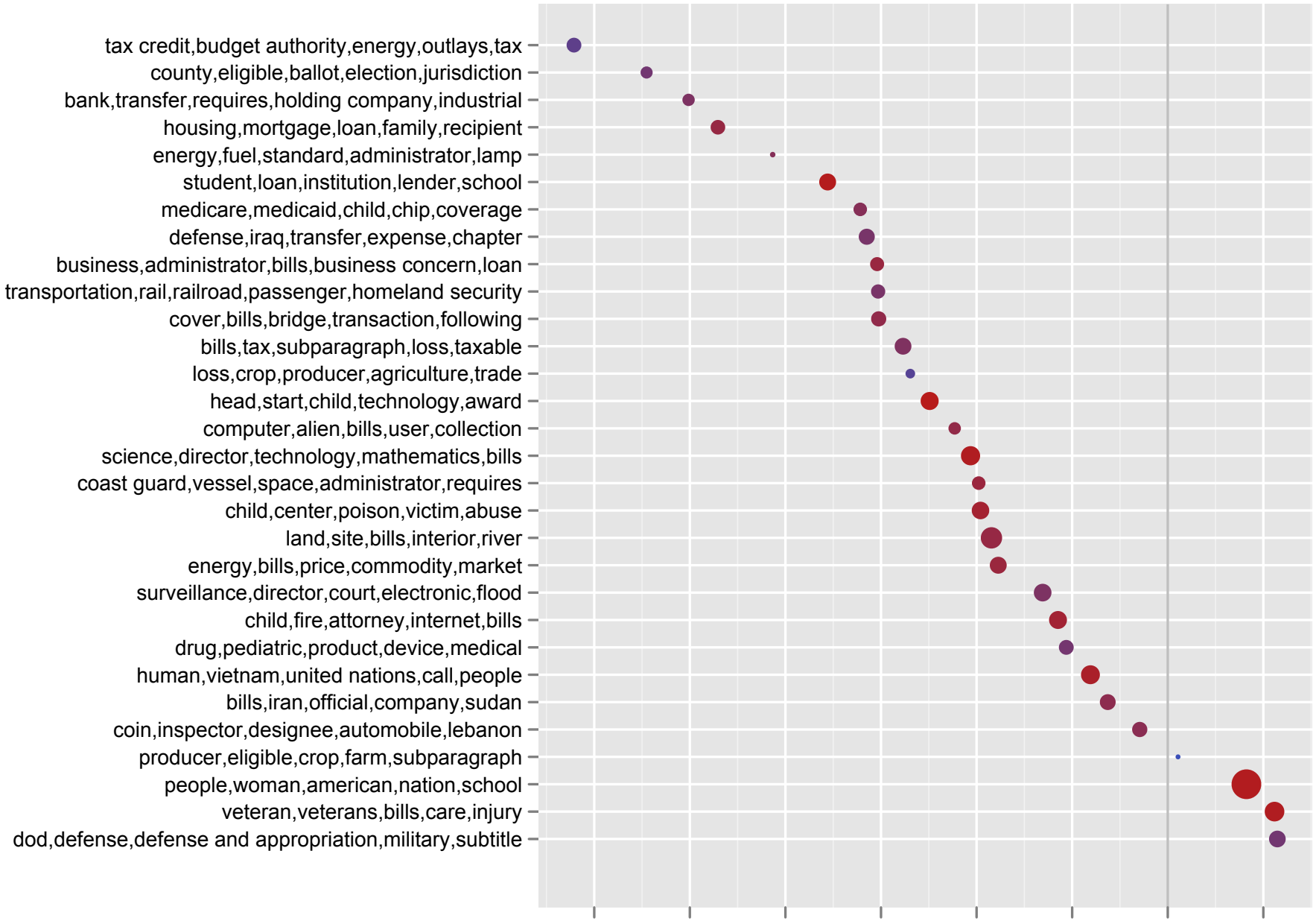
Discover influential articles



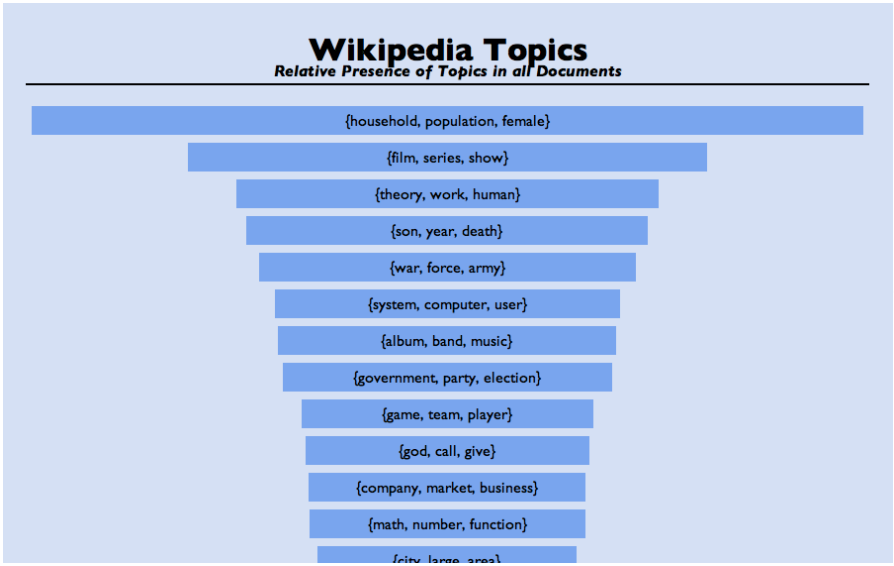
Predict links between articles

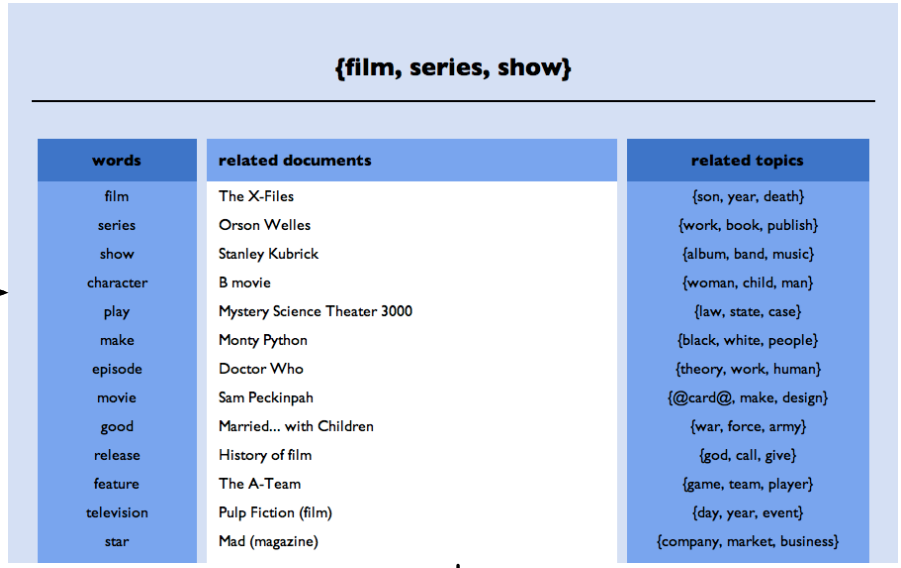
Markov chain Monte Carlo convergence diagnostics: A comparative review	
Minorization conditions and convergence rates for Markov chain Monte Carlo Rates of convergence of the Hastings and Metropolis algorithms **Possible biases induced by MCMC convergence diagnostics** Bounding convergence time of the Gibbs sampler in Bayesian image restoration Self regenerative Markov chain Monte Carlo Auxiliary variable methods for Markov chain Monte Carlo with applications **Rate of Convergence of the Gibbs Sampler by Gaussian Approximation** Diagnosing convergence of Markov chain Monte Carlo algorithms	**RTM (ψ_e)**
Exact Bound for the Convergence of Metropolis Chains Self regenerative Markov chain Monte Carlo **Minorization conditions and convergence rates for Markov chain Monte Carlo** Gibbs-markov models Auxiliary variable methods for Markov chain Monte Carlo with applications Markov Chain Monte Carlo Model Determination for Hierarchical and Graphical Models Mediating instrumental variables A qualitative framework for probabilistic inference Adaptation for Self Regenerative MCMC	**LDA + Regression**

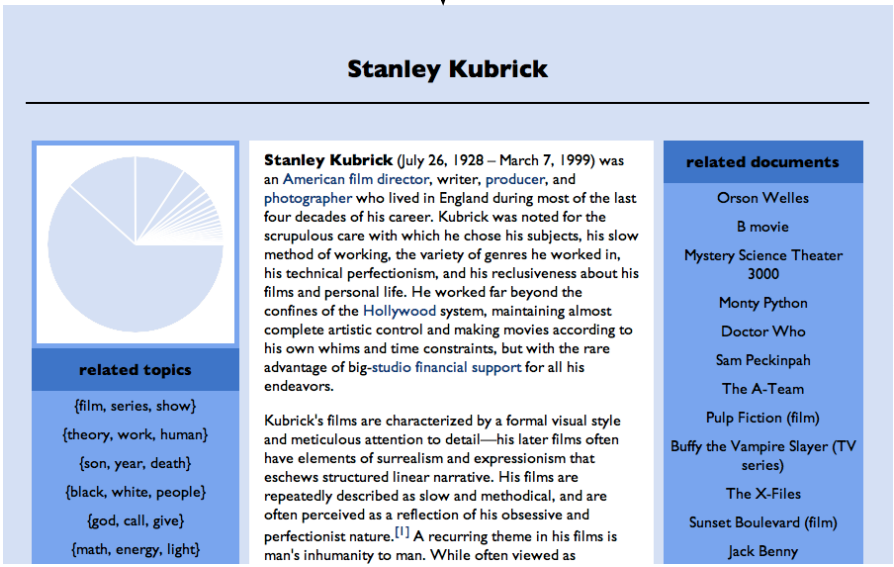
Characterize political decisions

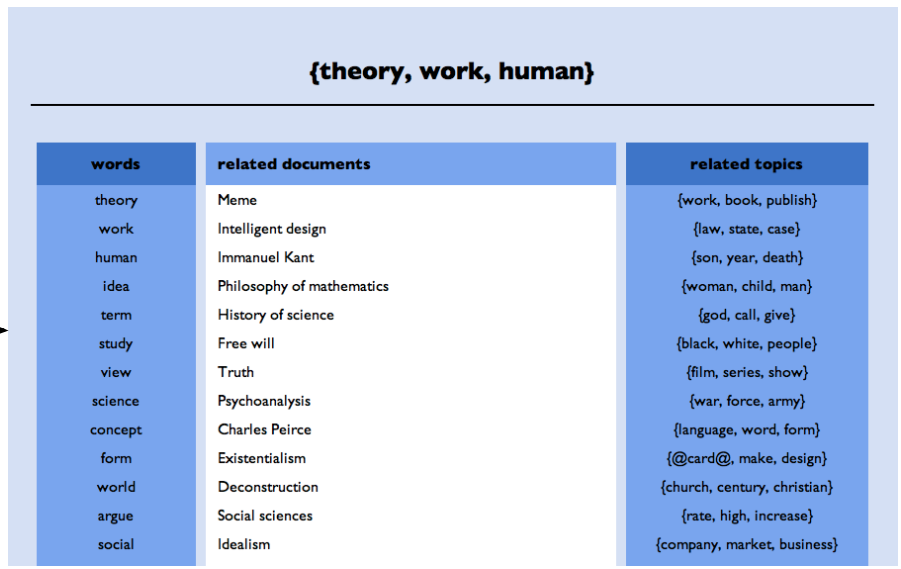


Organize and browse large corpora









Topics in scientific texts

Quantum physics	spin energy field electron magnetic state states hamiltonian
Particle physics	higgs neutrino coupling decay scale masses mixing quark
Astrophysics	mass gas star stellar galaxies disk halo radius luminosity
Relativity	black metric hole schwarzschild gravity holes einstein
Number theory	prime integer numbers conjecture integers degree modulo
Graph theory	graph vertex vertices edges node edge number set tree
Linear algebra	matrix matrices vector basis vectors diagonal rank linear
Optimization	problem optimization algorithm function solution gradient
Probability	random probability distribution process measure time
Machine learning	layer word image feature sentence model cnn lstm training

Topic modeling for equations

Topic	Generated Equations
Quantum physics	• $E = \hbar \frac{\partial^2 S}{\partial t^2} (\frac{\partial \varphi}{\partial c}) - \frac{k}{\hbar^2} \frac{\partial B}{\partial t} (t + \partial_t \delta)$. • $\Psi_{\text{pr}} = \sum_{\mathbf{l}} (\psi_{\mathbf{r}+\uparrow} - \psi^{\dagger}_{\mathbf{r}\downarrow}) + \sum_{\mathbf{r}'} (\psi^{\dagger}_{\mathbf{r}\downarrow,\uparrow} - \psi_{\mathbf{r}\downarrow} \sigma^{\dagger})$.
Particle physics	• $\mathcal{H} = \frac{2}{4} (\partial_\mu \phi)^2 + 2m\phi_\nu(\phi) + \frac{1}{2} m^2 (\phi)(1 - \phi^2)^2$. • $m_{\text{eff}}(M) = 1.4 \cdot 10^{-13}\,\text{GeV}$.
Relativity	• $\mathcal{M} = \frac{2}{2} g^{\mu\nu} (f_{\mu\nu,\mu} - g_{\mu\nu,\nu} + g_{\nu\nu,b} f_{\mu,\nu}) + \frac{1}{2} g^{\mu\nu}$. • $T_{\mu\nu} = \int_0^\infty ds_{\mu\nu} ds^2 + a_\mu^2 dr^2 + r^2 d\Omega^2$.

https://seas.yale.edu/news-events/news/michihiro-yasunaga-wins-cra-outstanding-undergraduate-researcher-award

Uber Topics

From a former student:

> *We're using topic models at Uber to discover topics in rider feedback – when riders write comments about their driver after the trip. We're trying to find topics such as 'unprofessional driver', 'driver no-show', 'sexual harassment', etc. LDA has worked really well with this...*

Introduction to Topic Modeling

Probabilistic modeling

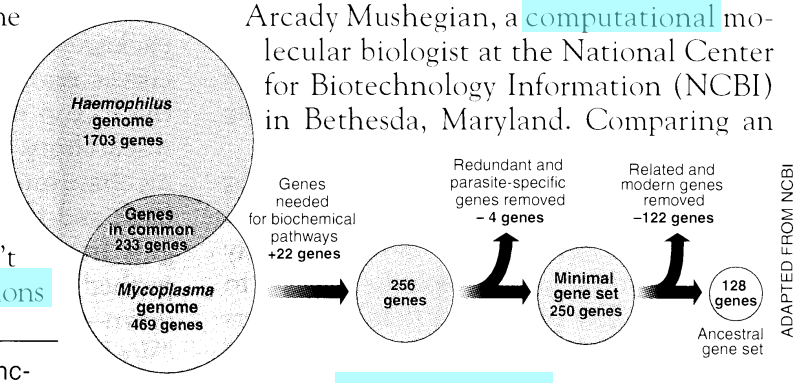
1. Data are assumed to be observed from a generative probabilistic process that includes hidden variables.
   - *In text, the hidden variables are the thematic structure.*
2. Infer the hidden structure using posterior inference
   - *What are the topics that describe this collection?*
3. Situate new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



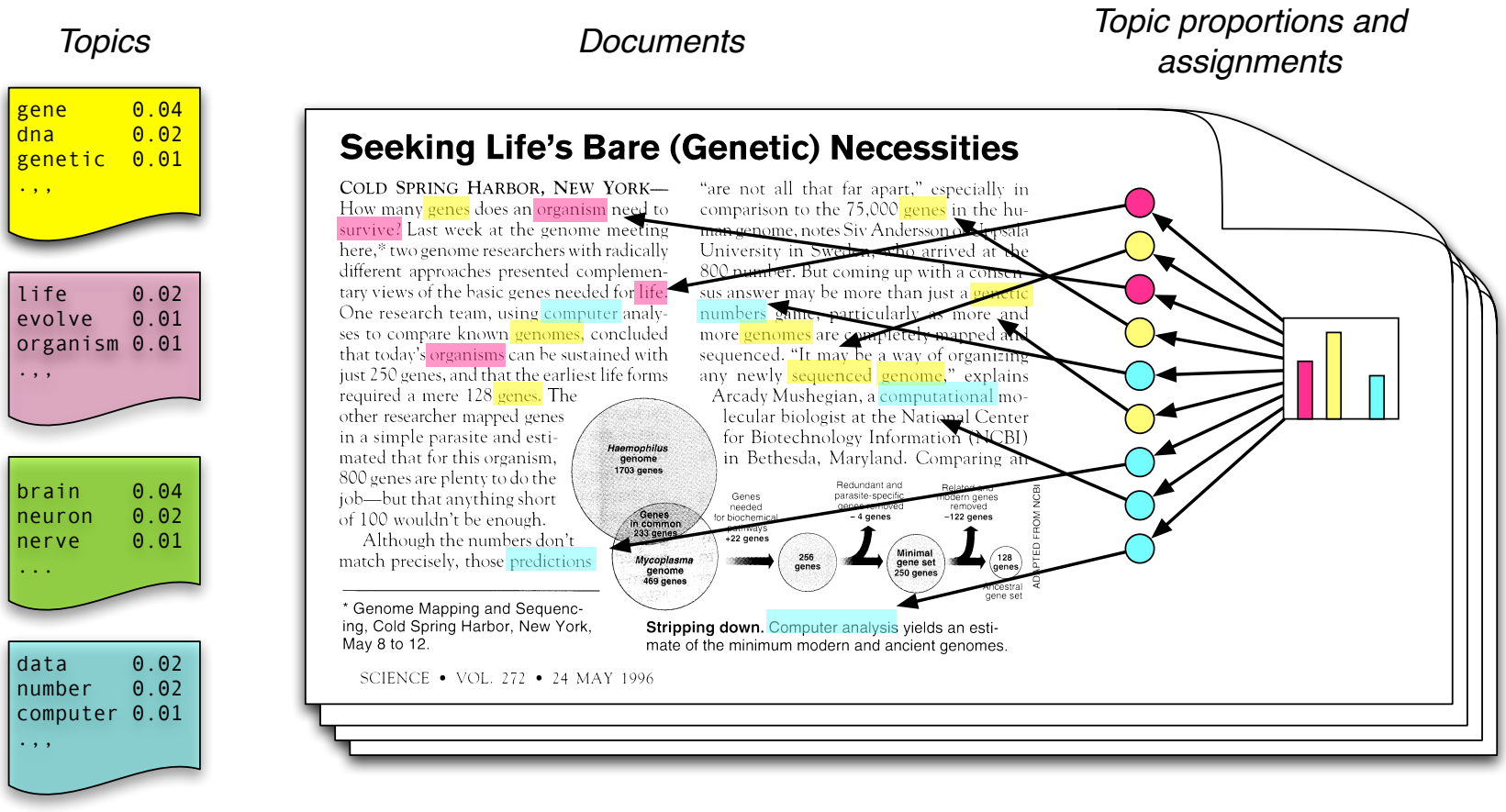
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

SCIENCE • VOL. 272 • 24 MAY 1996

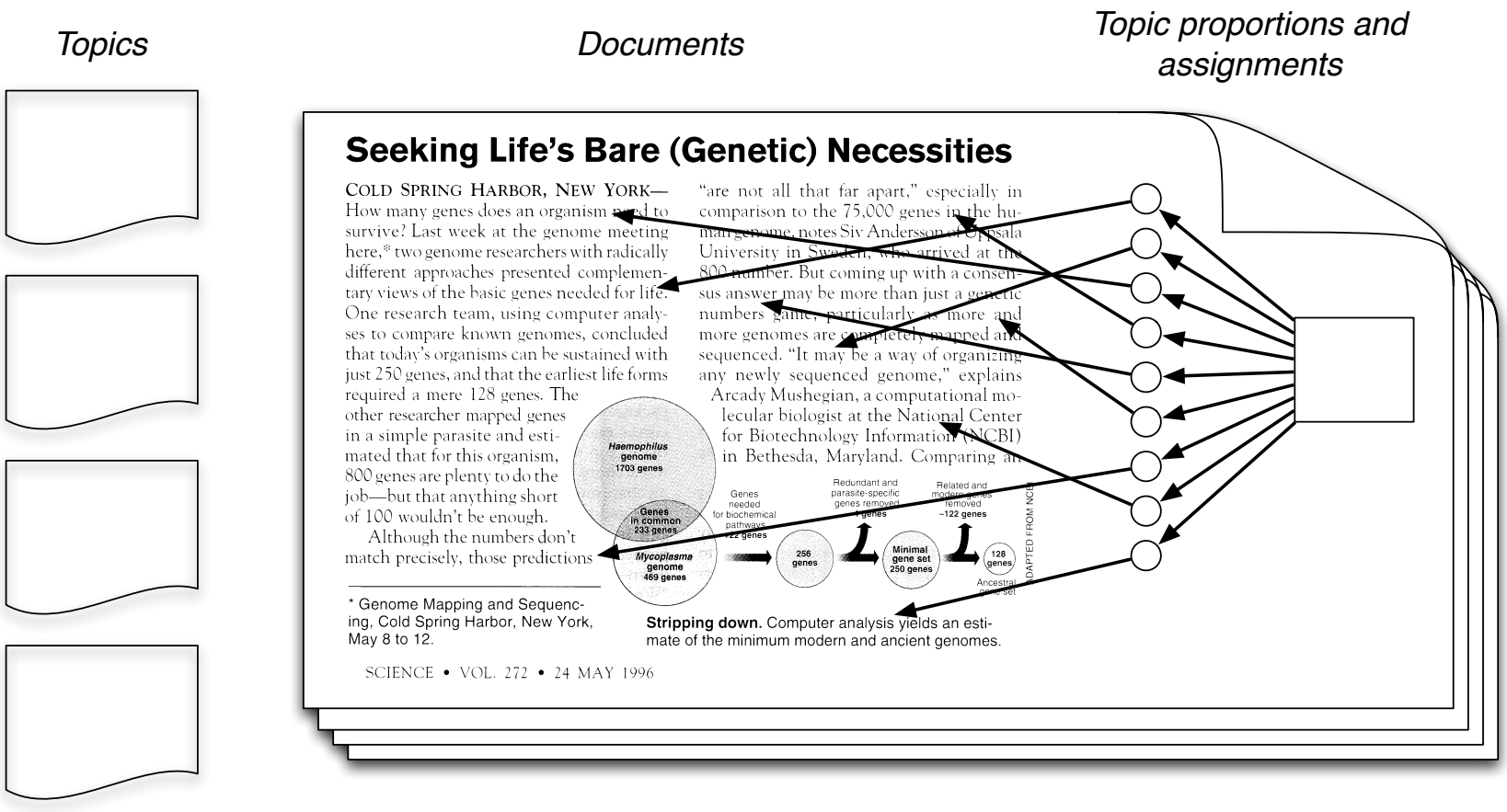
Simple intuition: Documents exhibit multiple topics.

Generative model for LDA



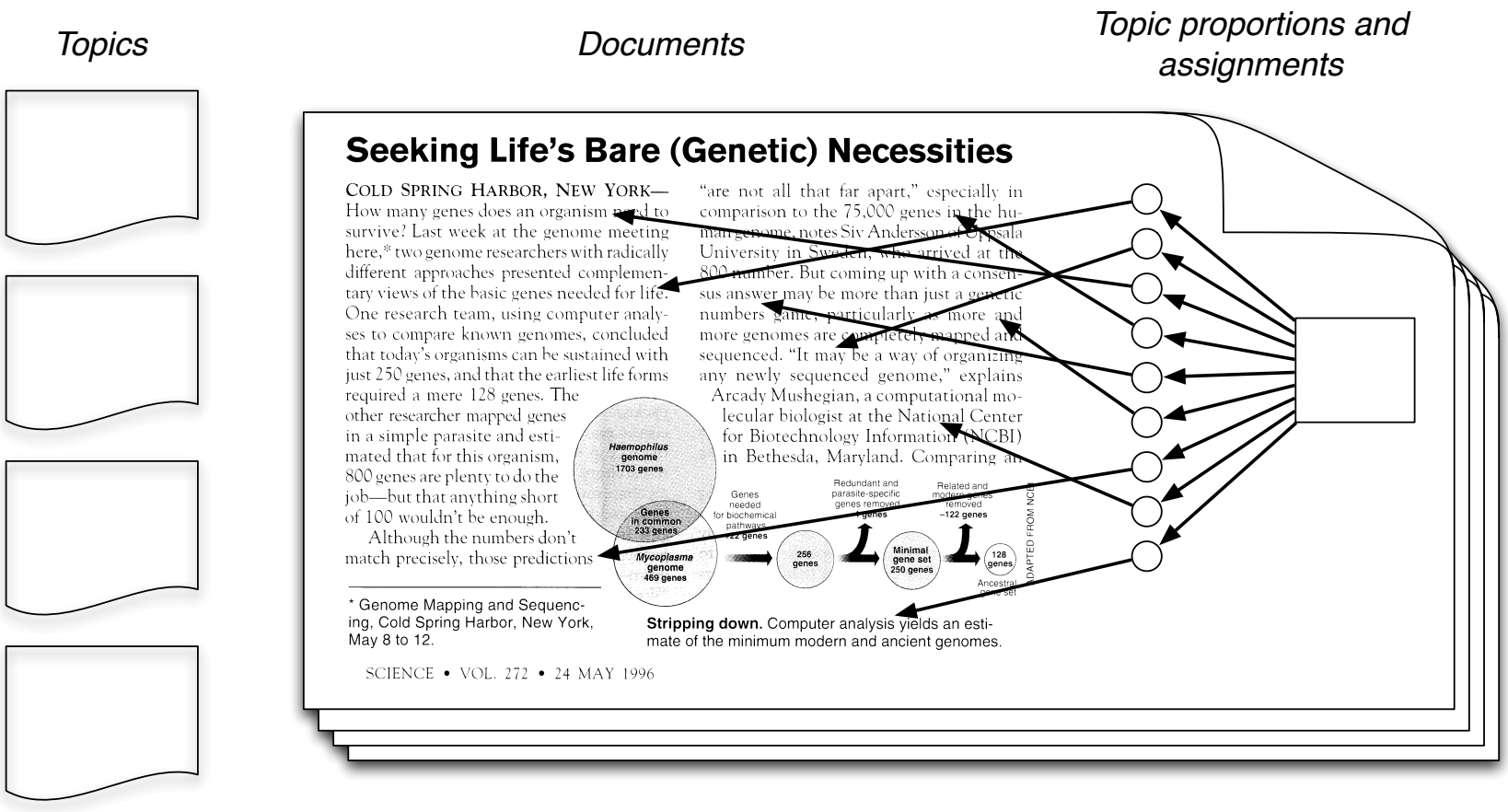
- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of those topics

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

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

Summary: Topic models

- Topic models automatically extract “semantic themes” from large document collections
- Based on latent variables, mixtures, and Bayesian inference
- Can be useful for a wide variety of data