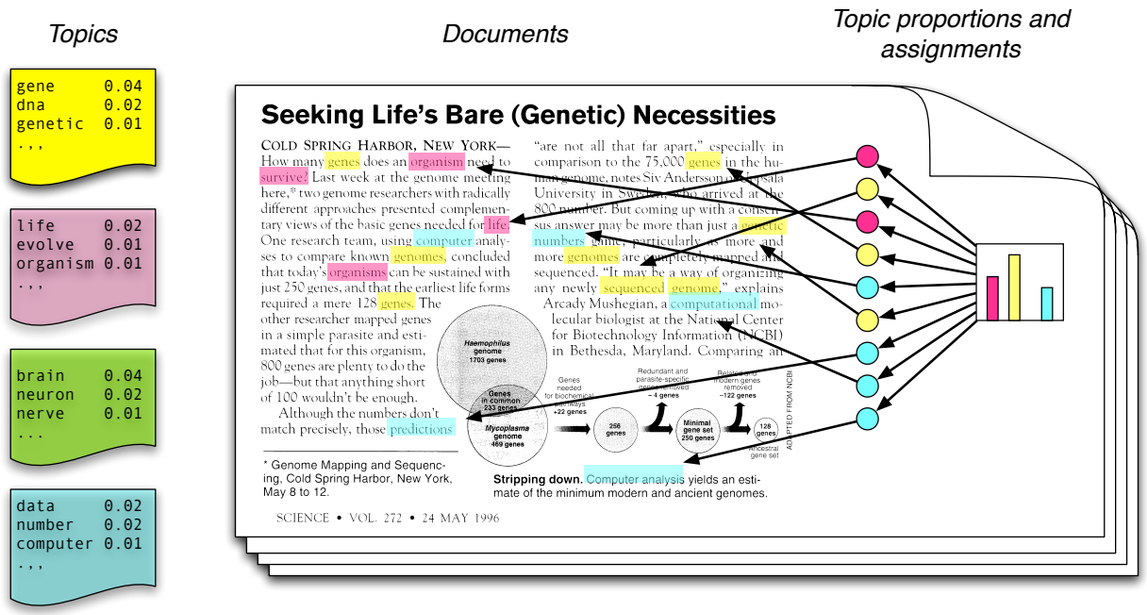
CS109/Stat121/AC209/E-109
Data Science
Bayesian Methods Continued, Text Data

Hanspeter Pfister, Joe Blitzstein, Verena Kaynig



Blei, https://www.cs.princeton.edu/~blei/papers/Blei2011.pdf

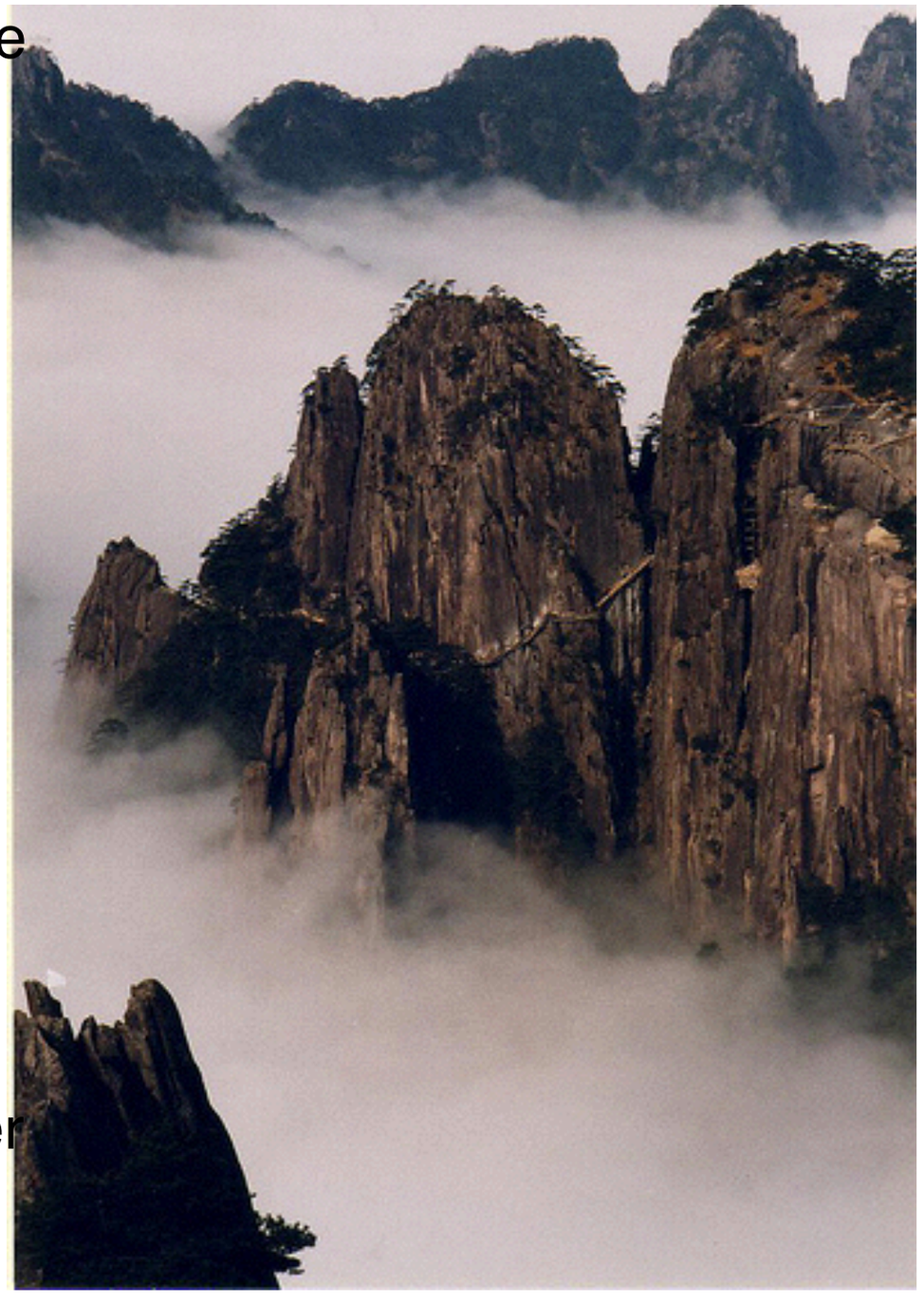
This Week

- Project team info is due tonight at 11:59 pm via the Google form: http://goo.gl/forms/CzVRIuCZk6
- HW4 is due this Thursday (Nov 5) at 11:59 pm
- Before this Thursday's lecture on interactive visualizations:
  - Download/install Tableau Public at https://public.tableau.com/
  - Download data file (.zip) from http://bit.ly/cs109data

MCMC as mountain exploration

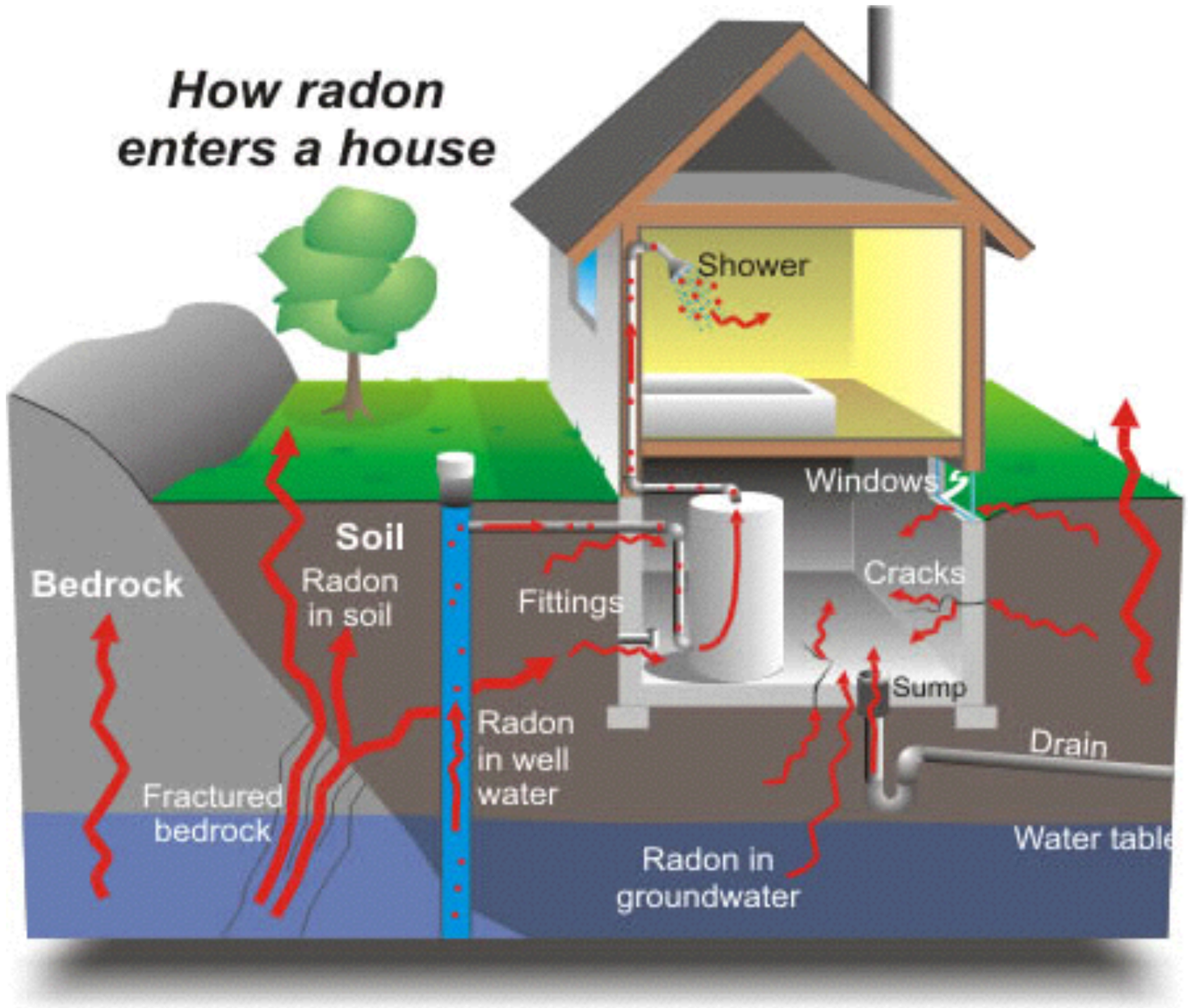
Bayesian: space explored is posterior
Often hard to do posterior computation w/o conjuagte prior: ie, if conjugate prior unrealistic

VS.



strength of Bayes; get full distribution, not just number
MCMC: you do a random walk: don't care about history of path to explore —> most MCMC are near-sighted/local. Trade-off, tiny steps, big jumps
http://healthyalgorithms.com/2010/03/12/a-useful-metaphor-for-explaining-mcmc/
speed vs local exploration/more information

Bayesian Hierarchical Models: Radon Example



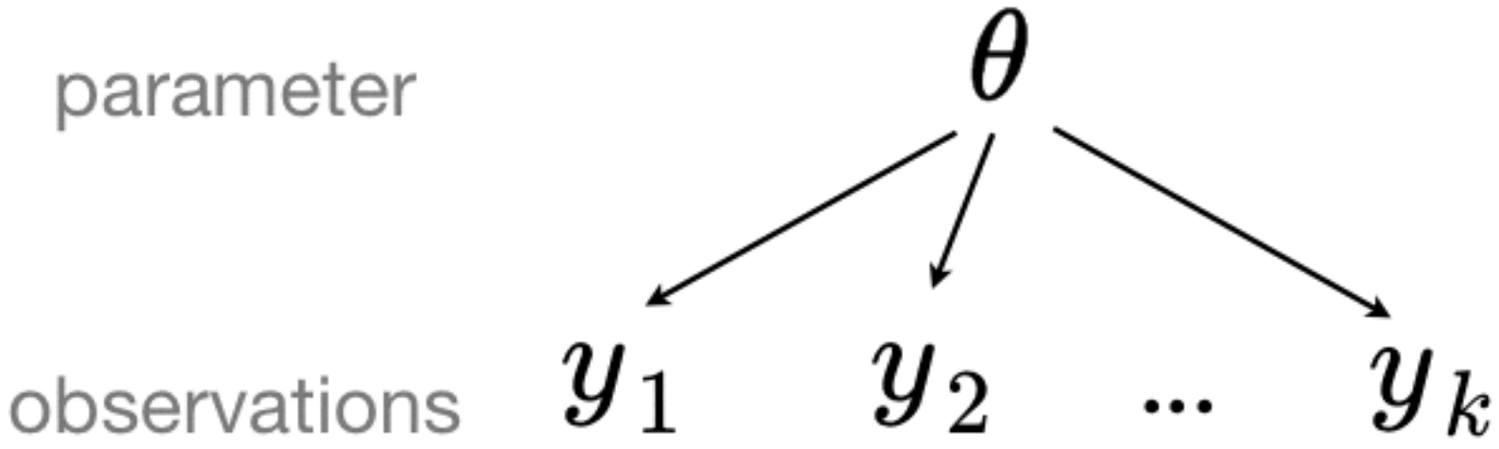
Example from Gelman http://www.eecs.berkeley.edu/~russell/classes/cs294/f05/papers/gelman-2005.pdf

Python-based exposition at http://twiecki.github.io/blog/2014/03/17/bayesian-glms-3/

Complete Pooling vs. No pooling

complete pooling: $\text{radon}_{i,c} = \alpha + \beta \cdot \text{floor}_{i,c} + \epsilon$

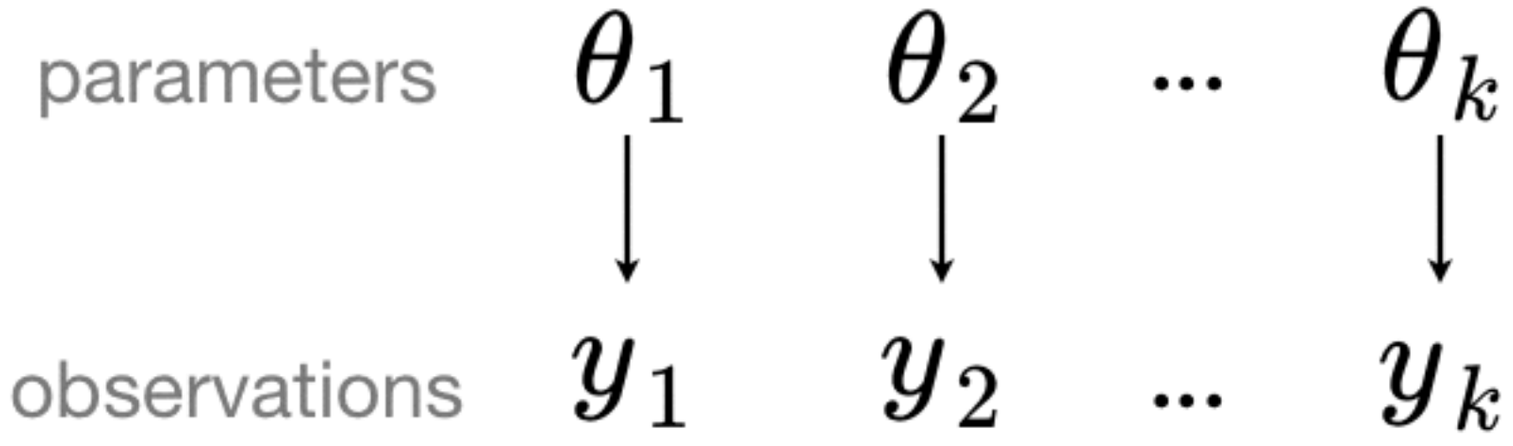
crude: don't consider fixed effects due to county, c or clustering.



alpha_c captures county fixed-effects

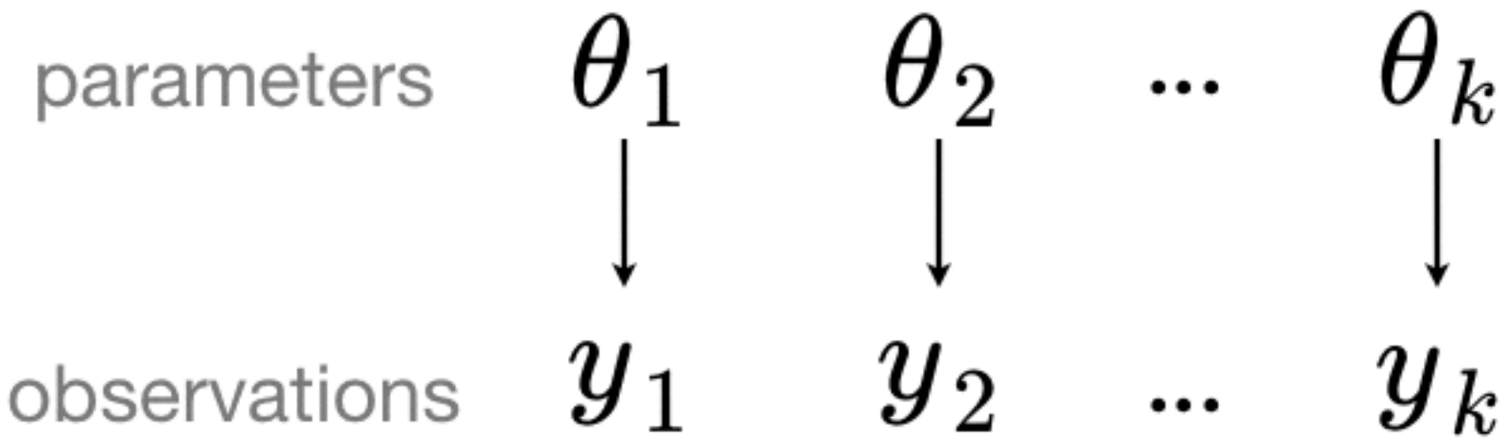
no pooling: $\text{radon}_{i,c} = \alpha_c + \beta_c \cdot \text{floor}_{i,c} + \epsilon_c$

but, with pooling you may lose sample size in each regression: less precession in beta_c.



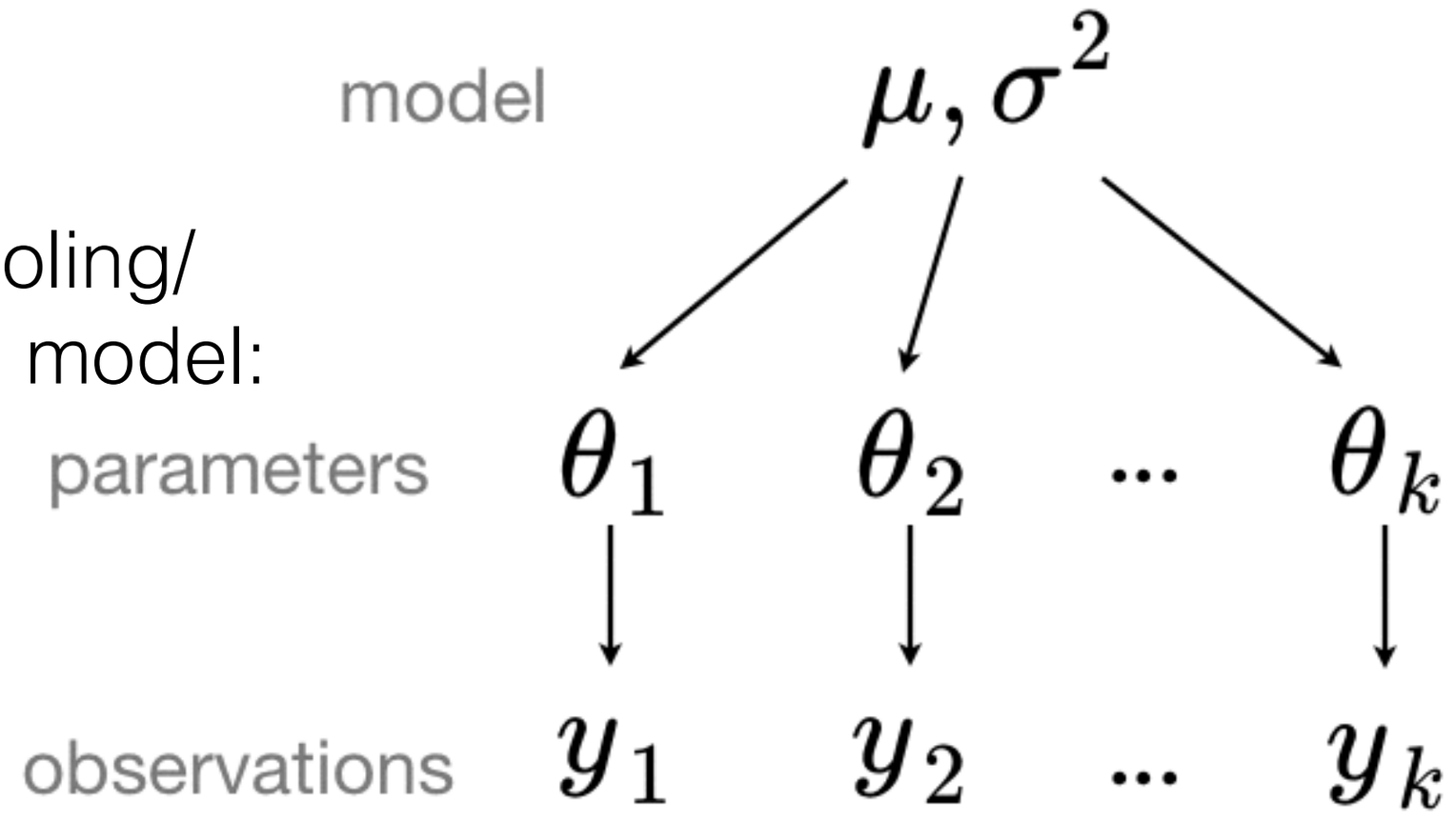
Partial Pooling

no pooling:

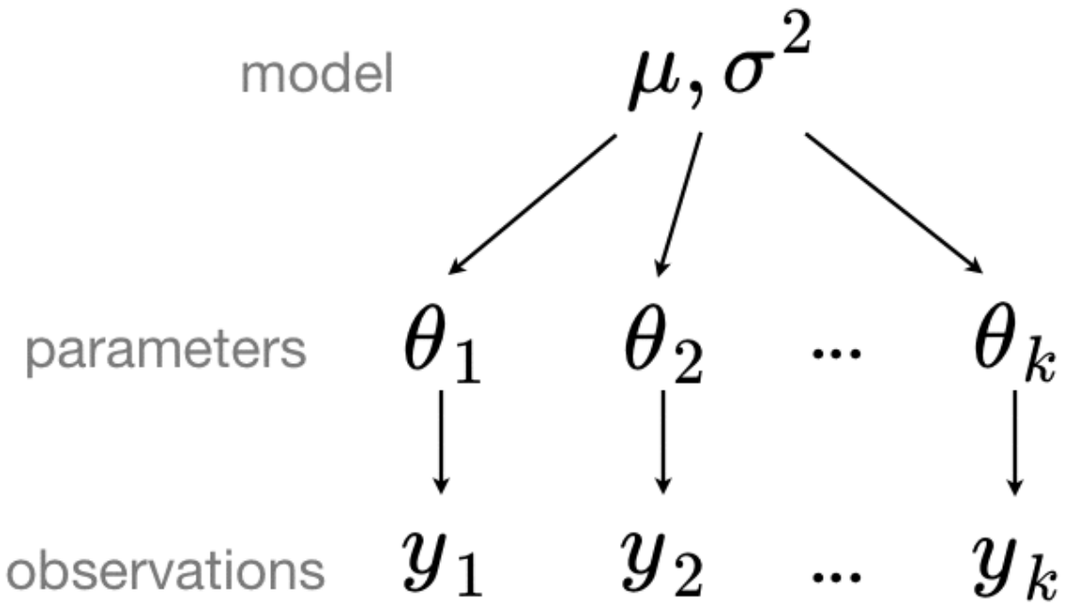


compromise

partial pooling/
hierarchical model:



Partial Pooling



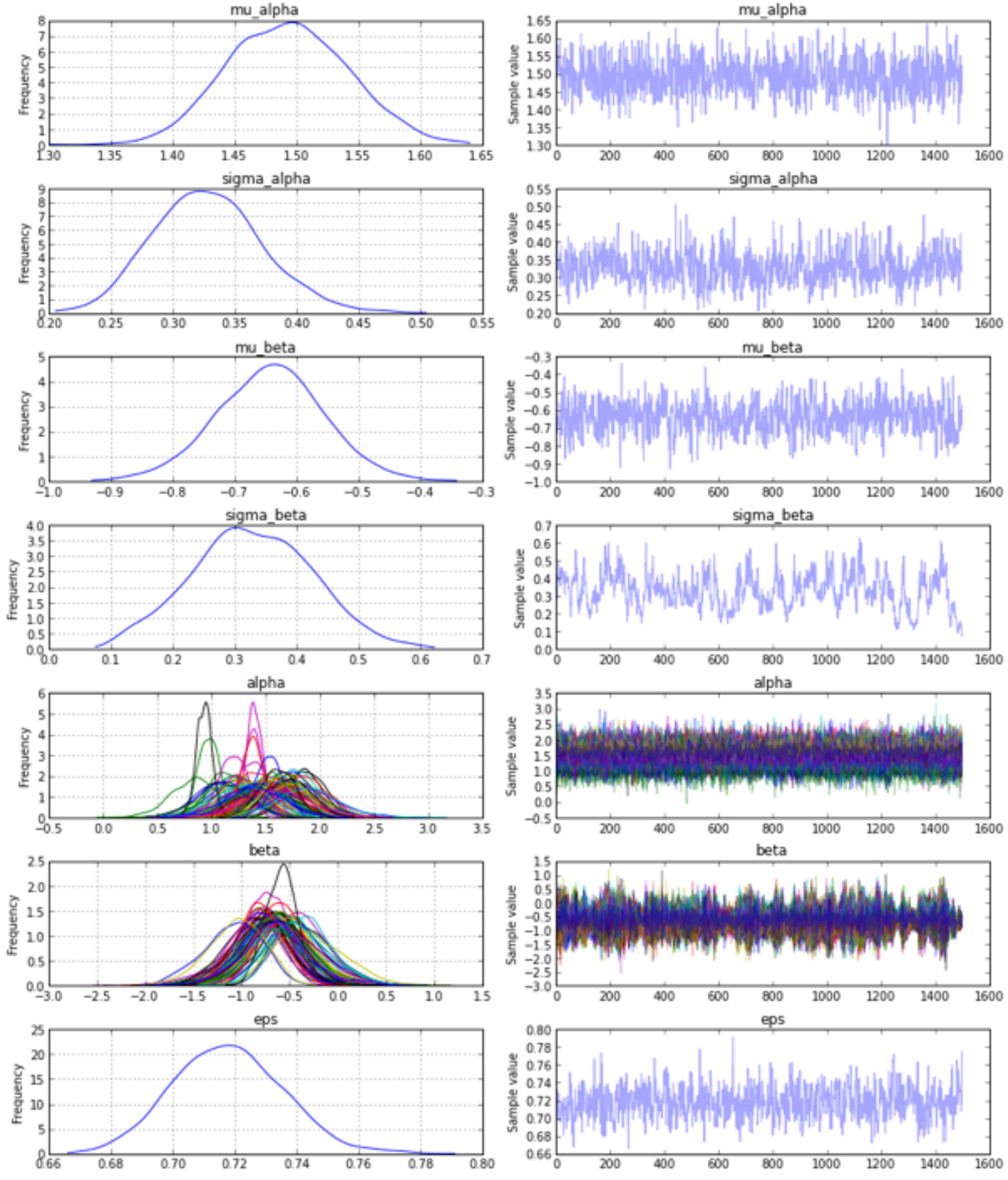
$$\text{radon}_{i,c} = \alpha_c + \beta_c \cdot \text{floor}_{i,c} + \epsilon_c$$

The counties are independent draws from same probability distribution: this means the alpha_c have common structure which means we avoid issue of complete distribution

$$\alpha_c \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$$

$$\beta_c \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2)$$

mu_alpha, sigma_alpha are hyper parameters
How do we know?
We can pick plausible values
Or create an above hierarchy, dist. of mu_alpha —> how many levels?
Or empirical Bayes —> find params with full data set (validation?)



http://twiecki.github.io/blog/2014/03/17/bayesian-glms-3/

Hierarchical Models Provide:

- a compromise between no pooling and complete pooling
- regularization and shrinkage
- give sensible estimates even for small groups
- organize the parameters in an interpretable way
- incorporate information at different levels in the hierarchy (e.g., individual level, county level, state level)
- predictions at various levels of the hierarchy (e.g., for new house or for new county)

organize parameters into hierarchy like a tree: easier to organize and interpret

good assuming you can do the computation for it.

Gibbs Sampler

MCMC computation algorithm
MCMC explores posterior

Explore space by updating *one coordinate at a time*.

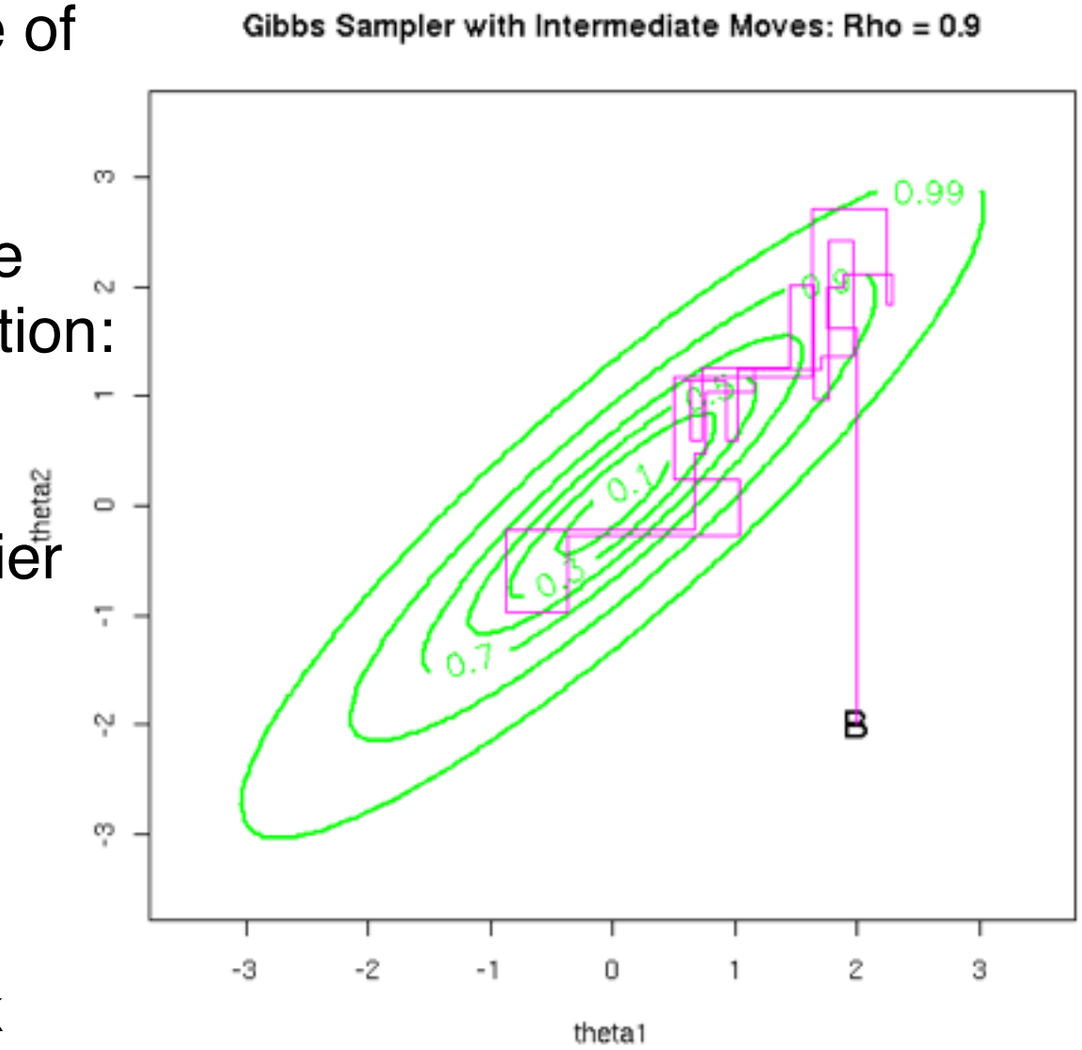
2D parameter space version:

$$\text{Draw new } \theta_1 \text{ from conditional distribution of } \theta_1|\theta_2$$
$$\text{Draw new } \theta_2 \text{ from conditional distribution of } \theta_2|\theta_1$$

MCMC explores space of possible posteriors based on parameters
Requires math to figure out conditional distribution: ie theta_1 I theta_2

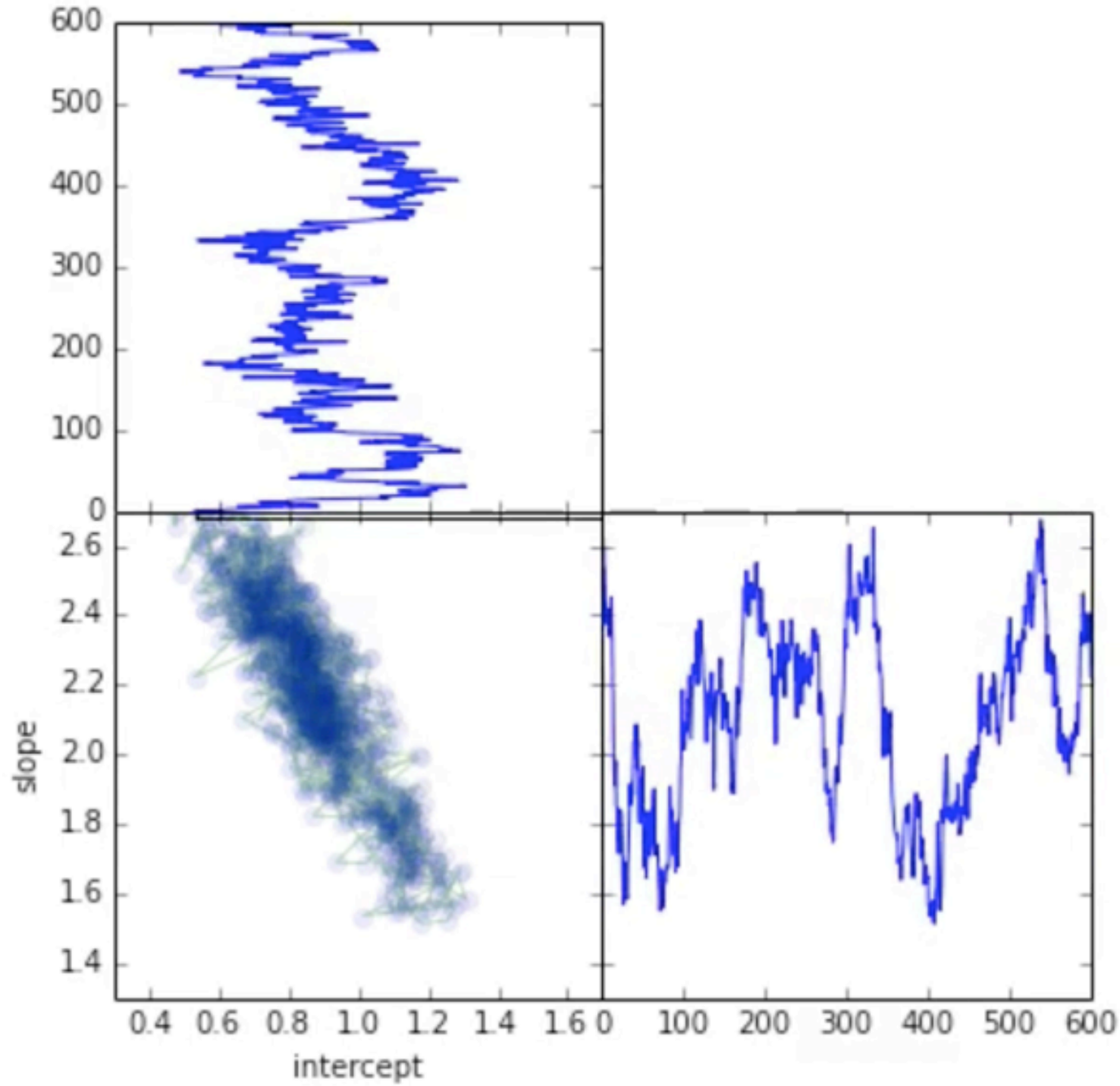
but they are much easier than the posterior

CAN ONLY MOVE horizontal or vertical: NOT DIAGONALLY based on random walk



http://zoonek.free.fr/blosxom//R/2006-06-22_useR2006_rbiNormGiggs.png

Gibbs sampler animation



http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

Metropolis-Hastings Algorithm

Start with any Markov Chain: some random walk through space (proposal chain)
stationary distribution: what chain converges to (ie., posterior in Bayesian)
proposal is independent of stationary —> use anyone).

Modify a Markov chain on a state space of interest to obtain a new chain with *any* desired stationary distribution!

1. If $X_n = i$, propose a new state j using the transition probabilities p_{ij} of the original Markov chain.

2. Compute an *acceptance probability*,

$$a_{ij} = \min\left(\frac{s_j p_{ji}}{s_i p_{ij}}, 1\right).$$

3. Flip a coin that lands Heads with probability a_{ij}, independently of the Markov chain. Eventually it WILL converge.

4. If the coin lands Heads, accept the proposal and set $X_{n+1} = j$. Otherwise, stay in state i; set $X_{n+1} = i$.

from *SIAM News*, Volume 33, Number 4

The Best of the 20th Century: Editors Name Top 10 Algorithms

By Barry A. Cipra

Algos is the Greek word for pain. *Algor* is Latin, to be cold. Neither is the root for *algorithm*, which stems instead from al-Khwarizmi, the name of the ninth-century Arab scholar whose book *al-jabr wa'l muqabalah* devolved into today's high school algebra textbooks. Al-Khwarizmi stressed the importance of methodical procedures for solving problems. Were he around today, he'd no doubt be impressed by the advances in his eponymous approach.

Some of the very best algorithms of the computer age are highlighted in the January/February 2000 issue of *Computing in Science & Engineering*, a joint publication of the American Institute of Physics and the IEEE Computer Society. Guest editors Jack Don-garra of the University of Tennessee and Oak Ridge National Laboratory and Fran-cis Sullivan of the Center for Comput-ing Sciences at the Institute for Defense Analyses put togeth-er a list they call the "Top Ten Algorithms of the Century."

"We tried to assemble the 10 al-gorithms with the greatest influence on the development and practice of science and engineering in the 20th century," Dongarra and Sullivan write. As with any top-10 list, their selections—and non-selections—are bound to be controversial, they acknowledge. When it comes to picking the algorithmic best, there seems to be no best algorithm.

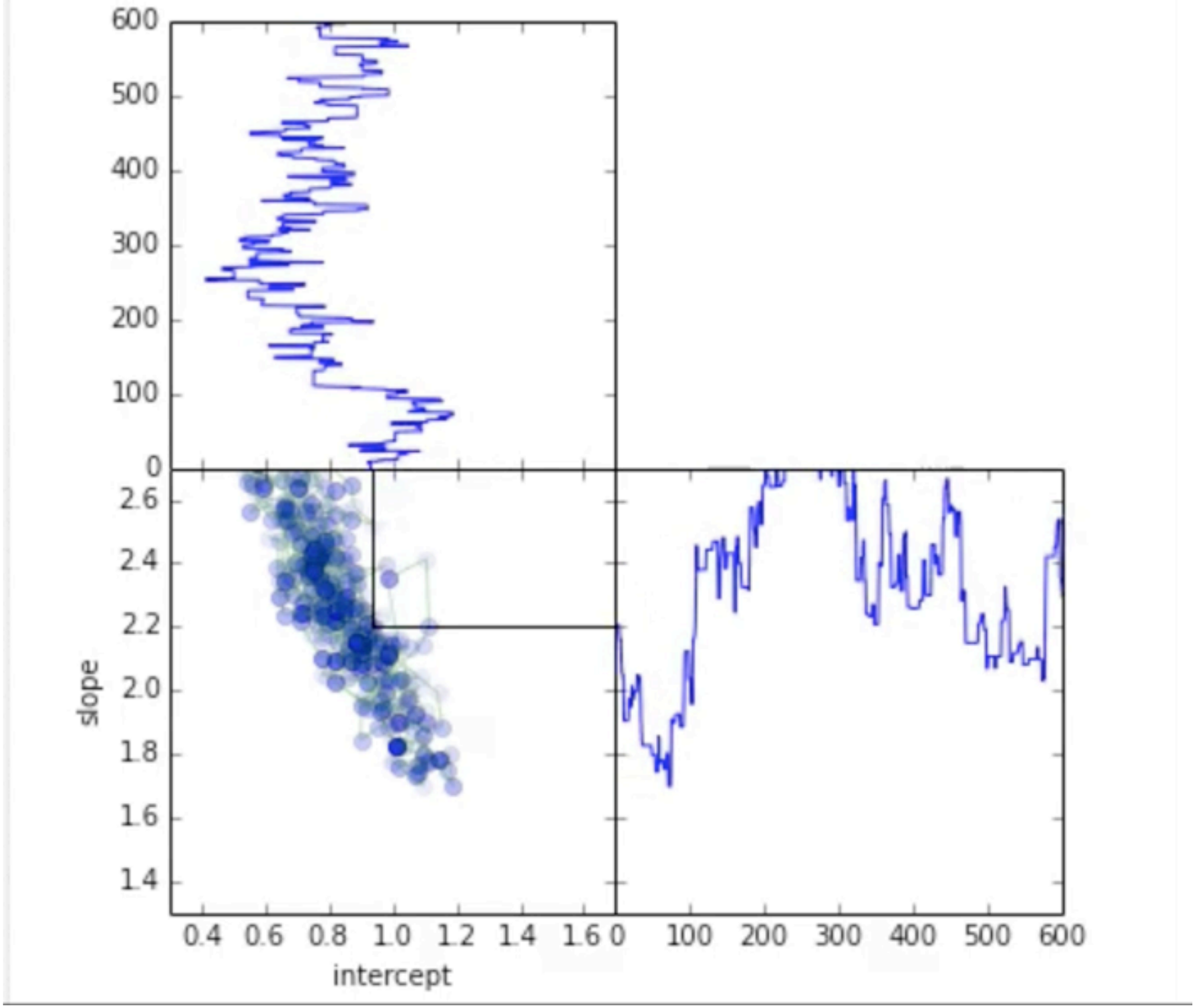
Without further ado, here's the CiSE top-10 list, in chronological order. (Dates and names associated with the algorithms should be read as first-order approximations. Most algorithms take shape over time, with many contributors.)

1946: John von Neumann, Stan Ulam, and Nick Metropolis, all at the Los Alamos Scientific Laboratory, cook up the Metropolis algorithm, also known as the **Monte Carlo method**.

The Metropolis algorithm aims to obtain approximate solutions to numerical problems with unmanageably many degrees of freedom and to combinatorial problems of factorial size, by mimicking a random process. Given the digital computer's reputation for deterministic calculation, it's fitting that one of its earliest applications was the generation of random numbers.

Metropolis-Hastings animation

jumpier than Gibbs: beginning, have rejects w/ tail, then it doesn't go anywhere, can make jumps —> it's less rigid than Gibbs, can do diagonal instead of horizon/vertical.



http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

MCMC in Python

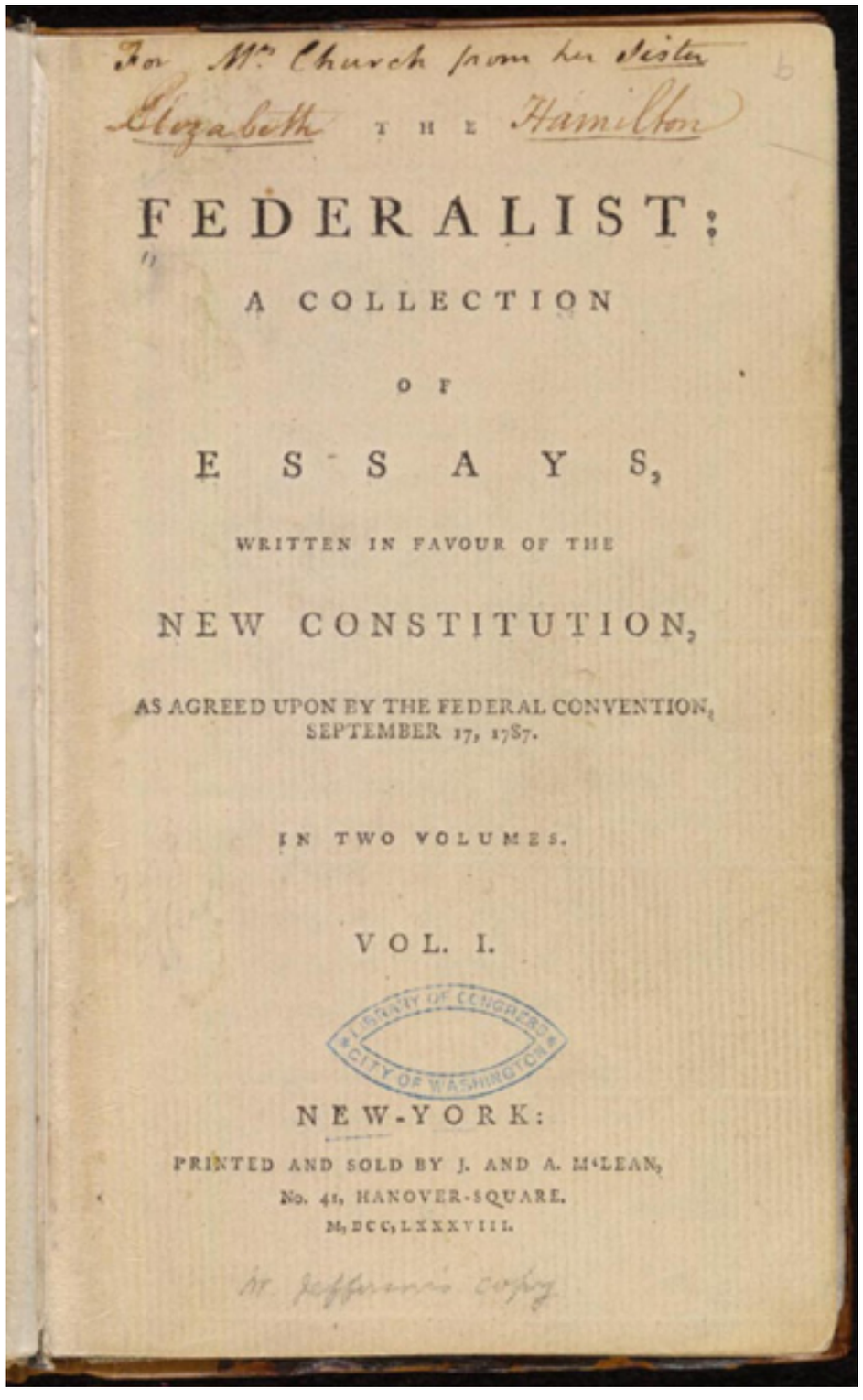
- Stan: http://mc-stan.org
- PyMC: https://pymc-devs.github.io/pymc/

Issue with metropolis: how to choose the proposal chain (it converges EVENTUALLY: want to minimize run time (especially large data).

Mosteller-Wallace, Federalist Papers Authorship

Test analysis:
data science applied to history

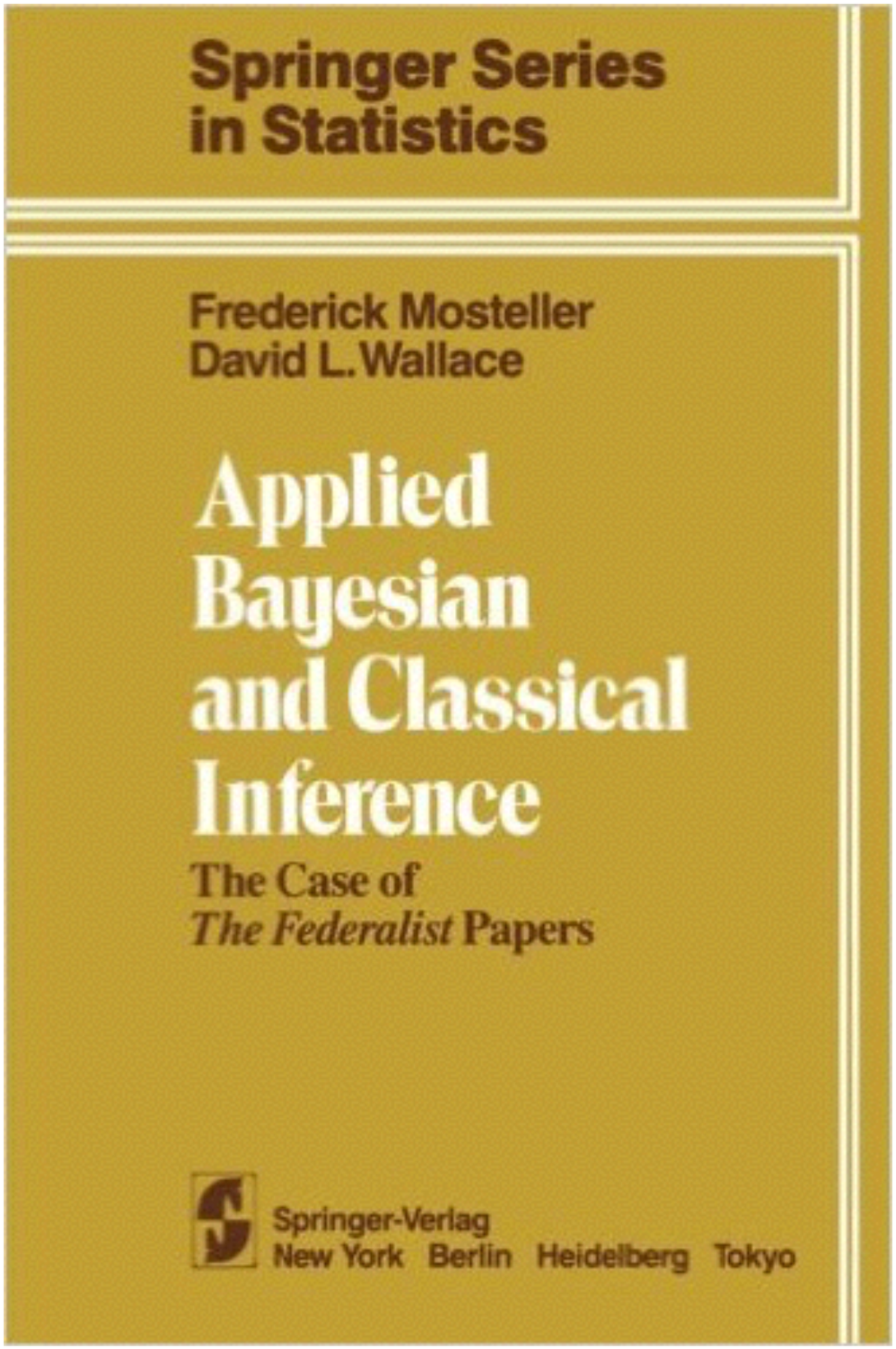
Question: Who authored some of the Federalist Papers.



Mosteller-Wallace, Federalist Papers Authorship

word frequency by diff authors
training data: essays we know.
Data cleaning: break up text into token:
roots or plural forms of words?

Done w/o computers
manual word counting …
Used entire Stats dept at Harvard



https://www.stat.cmu.edu/Exams/mosteller.pdf

Use of "upon" by Hamilton vs. Madison

But, sample size: not probability/confidence of conclusion for Madison: strength of Bayes.

Rate/1000 Words	Authored by Hamilton	Authored by Madison	12 Disputed Papers
Exactly 0	0	41	11
(0.0, 0.4)	0	2	0
$[0.4, 0.8)$	0	4	0
$[0.8, 1.2)$	2	1	1
$[1.2, 1.6)$	3	2	0
$[1.6, 2.0)$	6	0	0
$[2.0, 3.0)$	11	0	0
$[3.0, 4.0)$	11	0	0
$[4.0, 5.0)$	10	0	0
$[5.0, 6.0)$	3	0	0
$[6.0, 7.0)$	1	0	0
$[7.0, 8.0)$	1	0	0
Totals:	48	50	12

Table from Samaniego, Stochastic Modeling and Mathematical Statistics

But what is the *probability* that Madison authored a particular disputed document, and how *confident* should we be about our answer?

Poisson Model

$$f(y|\lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!}$$

y is the number of occurrences of a specific word

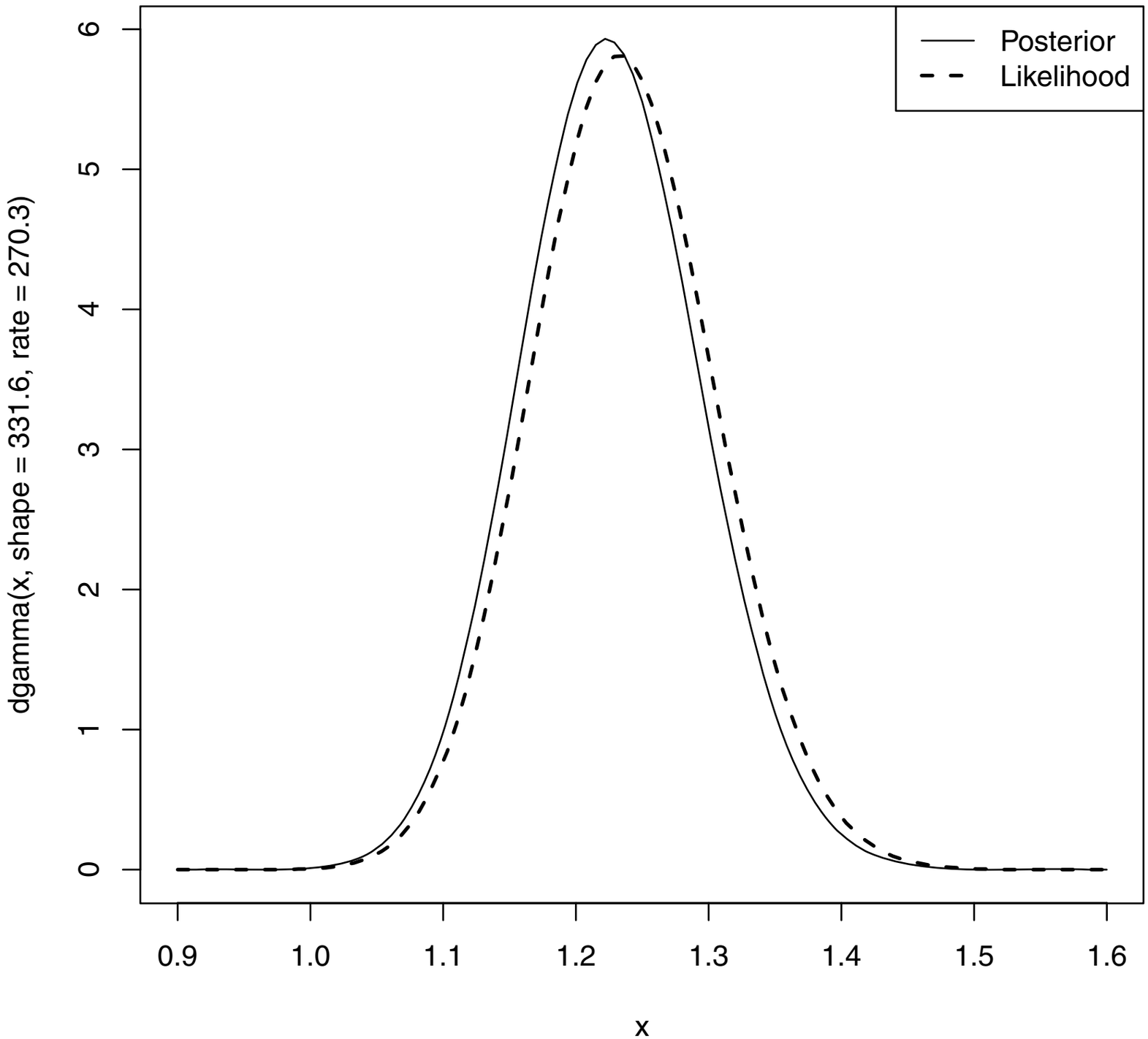
λ is the rate parameter

Gamma prior is conjugate: $p(\lambda) \propto \lambda^{a-1}e^{-b\lambda}$

ie, lambda is the rate of word appearance
applies to document: many words, any individual word is unlikely to be "upon"

How a and b? Empirical Bayes approach

Likelihood and Posterior for Madison's use of "from"



n-grams

Data science is fun.

Unigrams: look at *individual words*.
“data”, “science”, “is”, “fun”

Bigrams: look at *word pairs.*
“data science”, “science is”, “is fun”

Trigrams: look at *word triplets.*
“data science is”, “science is fun”

now: lots of pages: lost more combinations: 5 grams or more, you get common setences.

n-grams: Randomized Hobbit

Find the n-grams: then you can generate text.
Get gibberish, but works better for larger n.

into trees, and then bore to the Mountain to go through?" groaned the hobbit. "Well, are you doing, And where are you doing, And where are you?" it squeaked, as it was no answer. They were surly and angry and puzzled at finding them here in their holes

Karl Broman, Randomized Hobbit
http://www.r-bloggers.com/randomized-hobbit/

n-grams: Hobbit/Cat in the Hat Mixture

"I am Gandalf," said the fish. This is no way at all! already off his horse and among the goblin and the dragon, who had remained behind to guard the door. "Something is outside!" Bilbo's heart jumped into his boat on to sharp rocks below; but there was a good game, Said our fish No! No! Those Things should not fly.

Karl Broman, Randomized Hobbit
http://www.r-bloggers.com/randomized-hobbit/

n-grams

If you may know which are you want to data sort the data feeds web friend someone on trending topics as the data in Hadoop is the data science requires a book demonstrates why visualizations are but we do massive correlations across many commercial disk drives in Python language and creates more tractable form making connections then use and uses it to solve a data.

—Bigram Model

In hindsight MapReduce seems like an epidemic and if so does that give us new insights into how economies work That's not a question we could even have asked a few years there has been instrumented.

—Trigram Model

n-gram useful for exploration: but not stats inference or a model.

Joel Grus, *Data Science from Scratch*

each word comes from topics: some words appear more commonly depending on topic: overlap tolerated

Topic Modeling

Topics

Documents

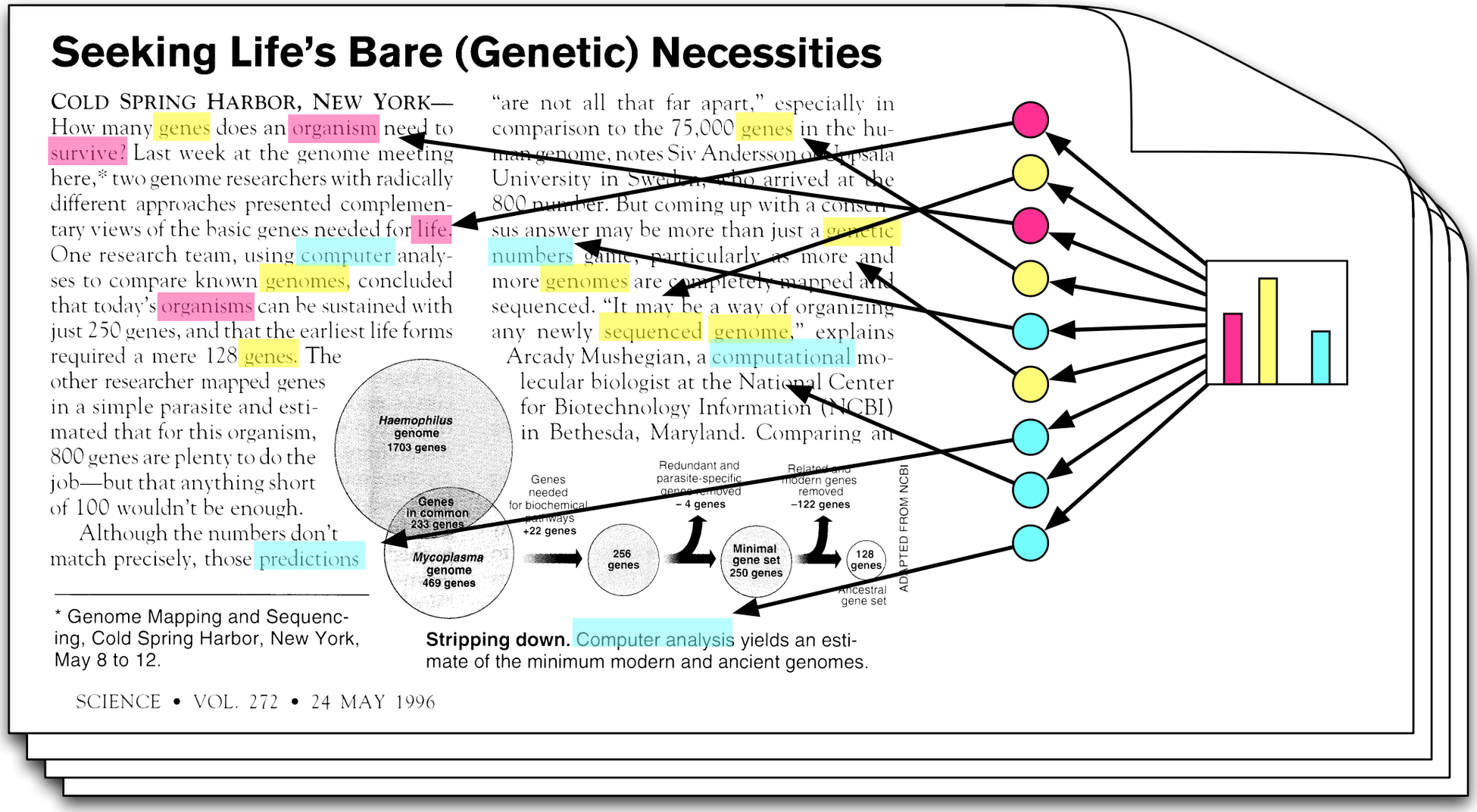
Topic proportions and assignments

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
...	



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.

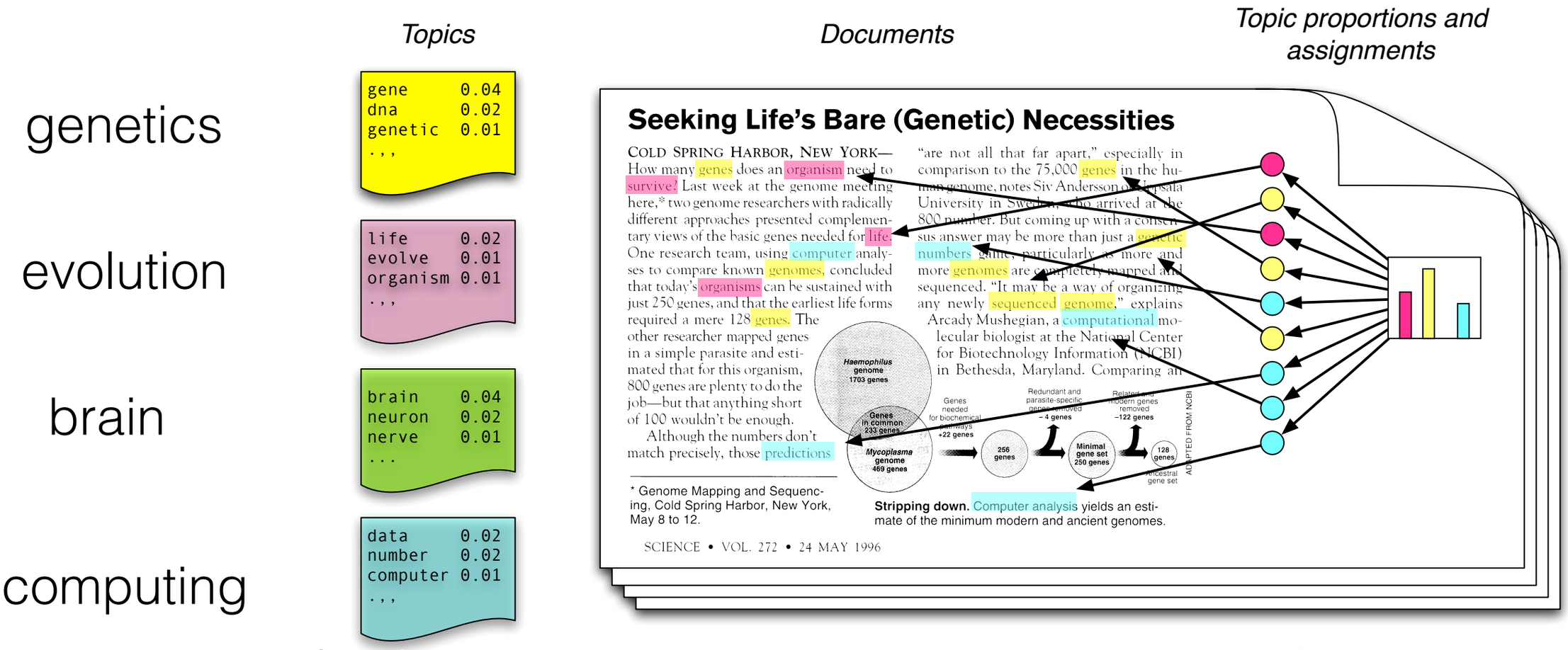
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just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

Blei, https://www.cs.princeton.edu/~blei/papers/Blei2011.pdf

Topic Modeling



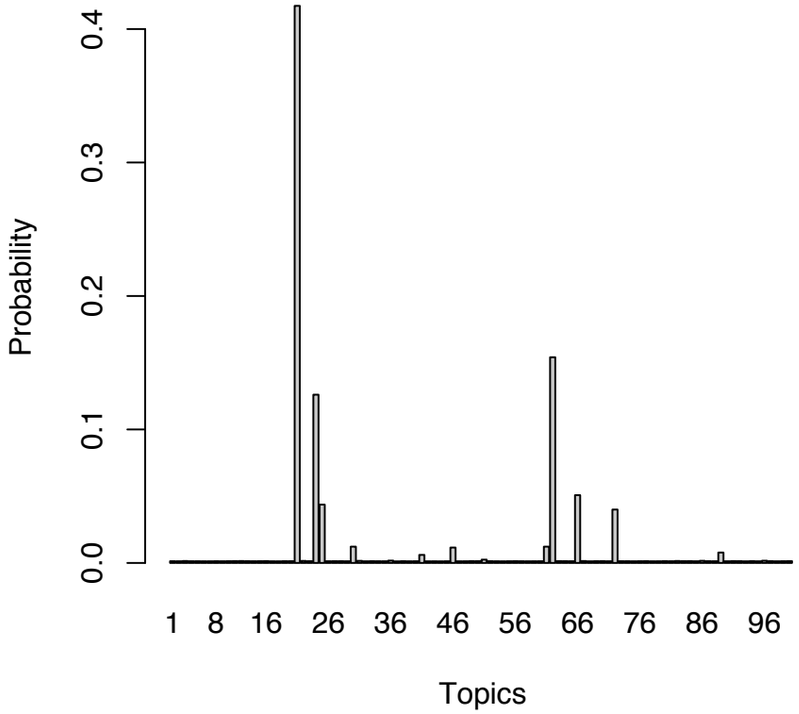
can get mixture of topics: but it's unsupervised: trying to automatically discover the topics

lots of parameters in model: want a large collection of articles
(more train/test/cross validation potential)

Blei, https://www.cs.princeton.edu/~blei/papers/Blei2011.pdf

17,000 articles from *Science*, 100 topics

Mixture of topics: probability of being topic as the % of mixture assigned to topic.

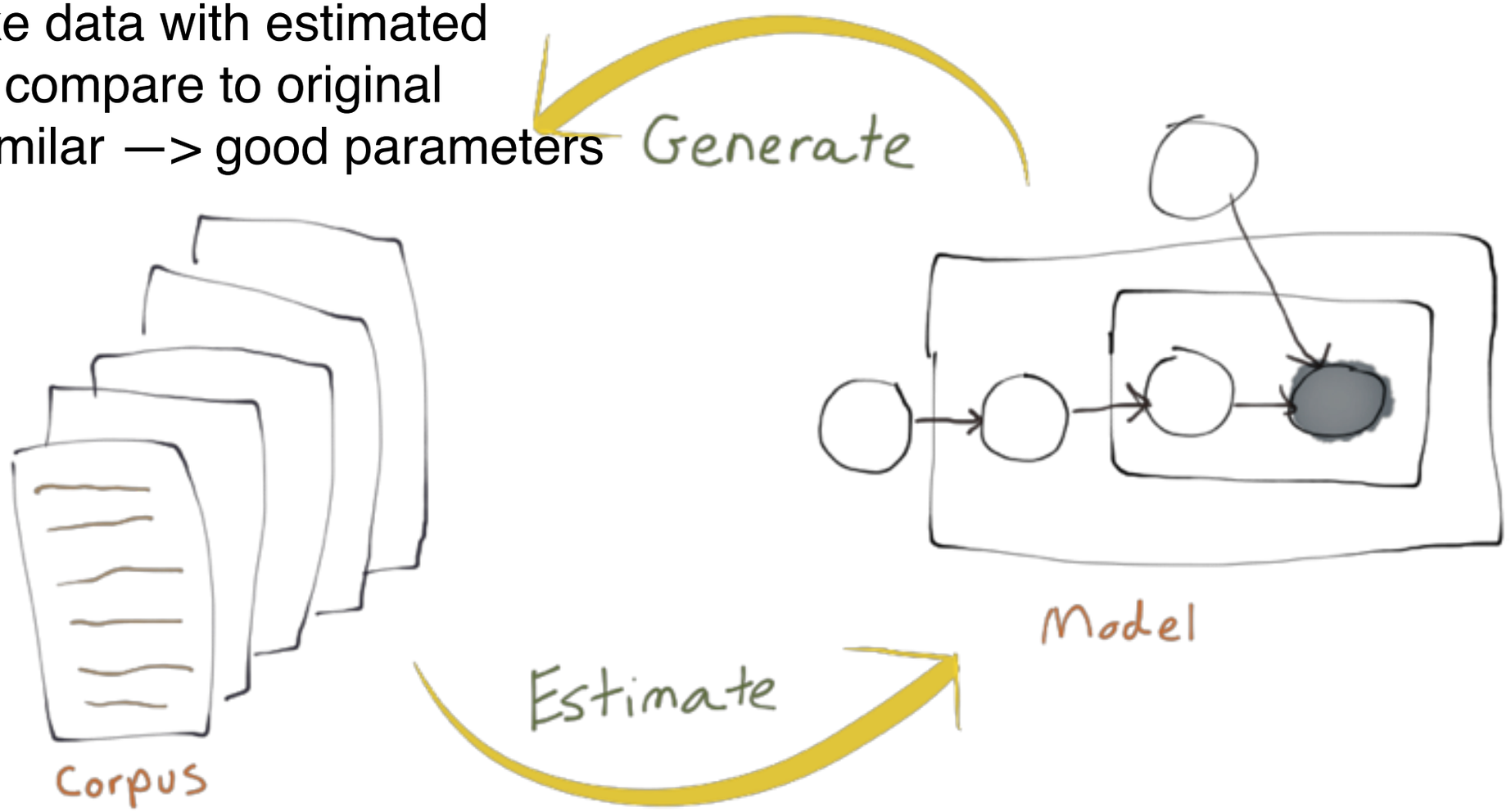


"Genetics"	**"Evolution"**	**"Disease"**	**"Computers"**
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	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

Blei, https://www.cs.princeton.edu/~blei/papers/Blei2011.pdf

Latent Dirichlet Allocation (LDA): Generation and Estimation

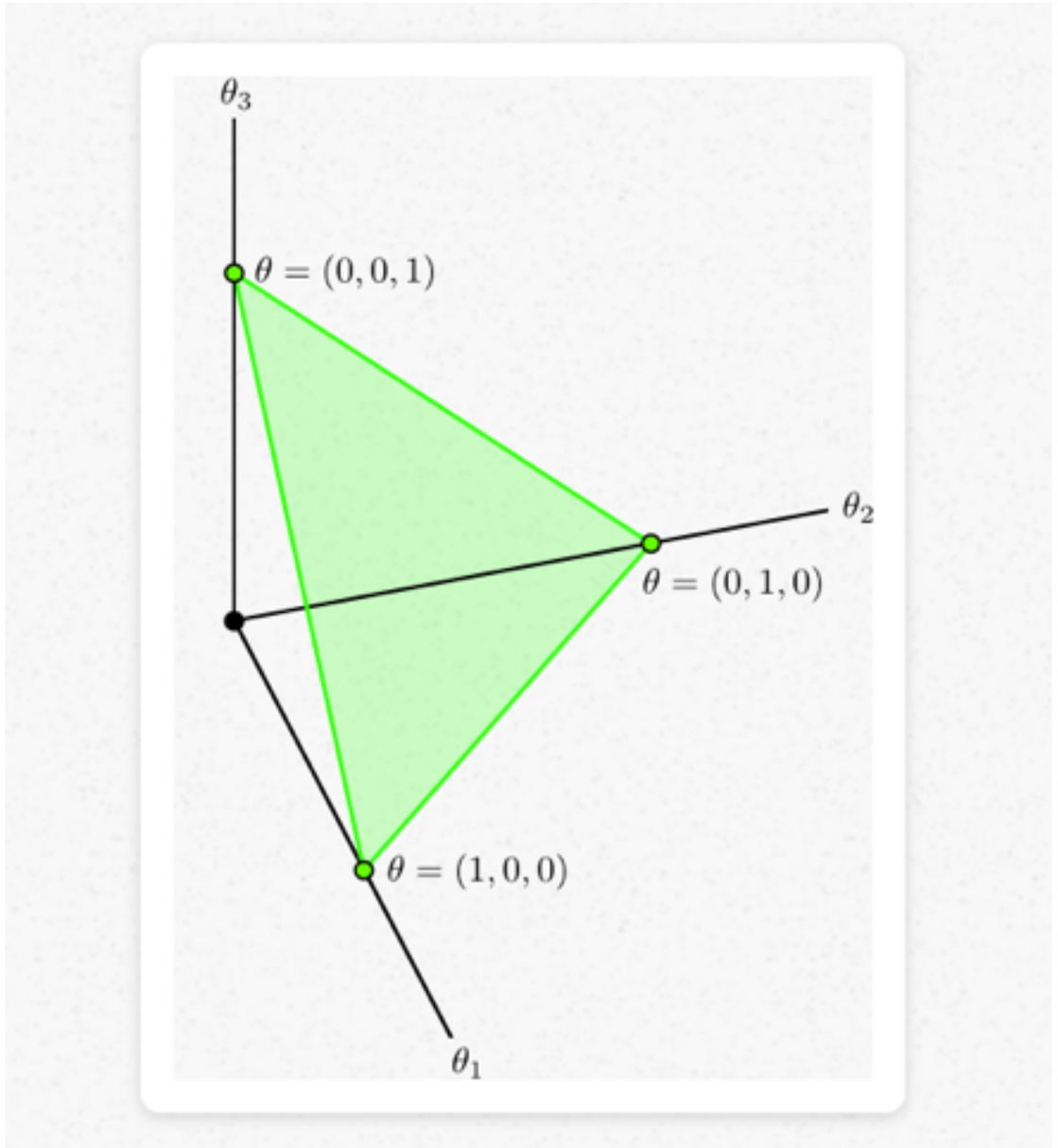
generate fake data with estimated parameters, compare to original
if they are similar —> good parameters



http://mcburton.net/blog/joy-of-tm/

Dirichlet Distribution

looks like a generalized beta.



http://blog.bogatron.net/blog/2014/02/02/visualizing-dirichlet-distributions/

Latent Dirichlet Allocation (LDA): Generative Model

α is the parameter of the Dirichlet prior on the per-document topic distributions,

β is the parameter of the Dirichlet prior on the per-topic word distribution,

θ_i is the topic distribution for document *i*,

φ_k is the word distribution for topic *k*,

z_{ij} is the topic for the *j*th word in document *i*, and

w_{ij} is the specific word.

The generative process is as follows. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes the following generative process for a corpus D consisting of M documents each of length N_i:

1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$, where $i \in \{1, \ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is the Dirichlet distribution for parameter α
2. Choose $\varphi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1, \ldots, K\}$
3. For each of the word positions i, j, where $j \in \{1, \ldots, N_i\}$, and $i \in \{1, \ldots, M\}$
   (a) Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.
   (b) Choose a word $w_{i,j} \sim \mathrm{Multinomial}(\varphi_{z_{i,j}})$.

multinomial: generalized binomial

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

Latent Dirichlet Allocation (LDA): Generative Model Example

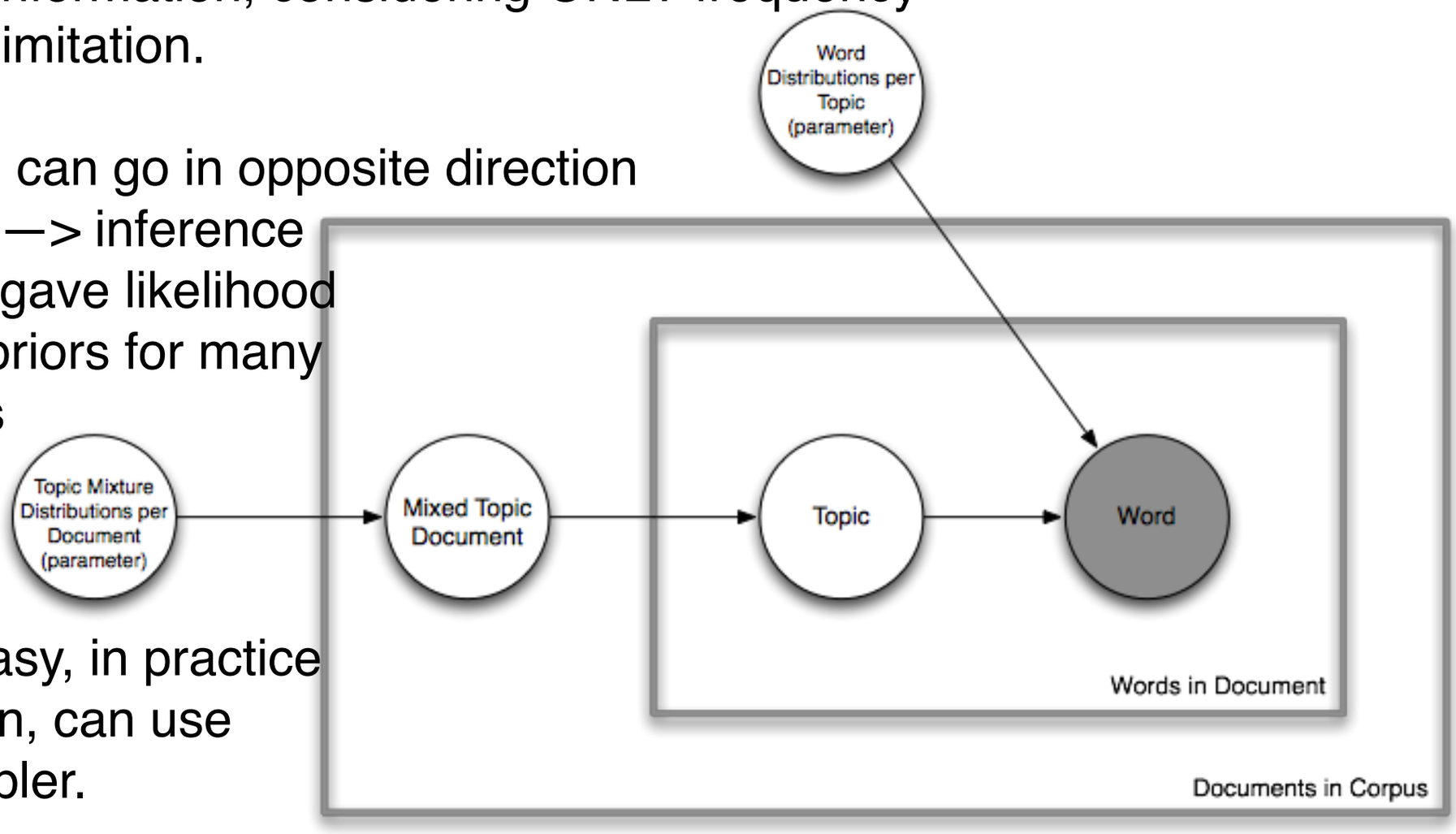
- Pick 5 to be the number of words in D.
- Decide that D will be 1/2 about food and 1/2 about cute animals.
- Pick the first word to come from the food topic, which then gives you the word “broccoli”.
- Pick the second word to come from the cute animals topic, which gives you “panda”.
- Pick the third word to come from the cute animals topic, giving you “adorable”.
- Pick the fourth word to come from the food topic, giving you “cherries”.
- Pick the fifth word to come from the food topic, giving you “eating”.

http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/

Latent Dirichlet Allocation (LDA): Generative Model

Bag of words: ignore order of words
—> losing information, considering ONLY frequency
yeah it's a limitation.

Bayes rule: can go in opposite direction
generative —> inference
generative gave likelihood
now need priors for many parameters



in theory easy, in practice computation, can use Gibbs sampler.

http://mcburton.net/blog/joy-of-tm/

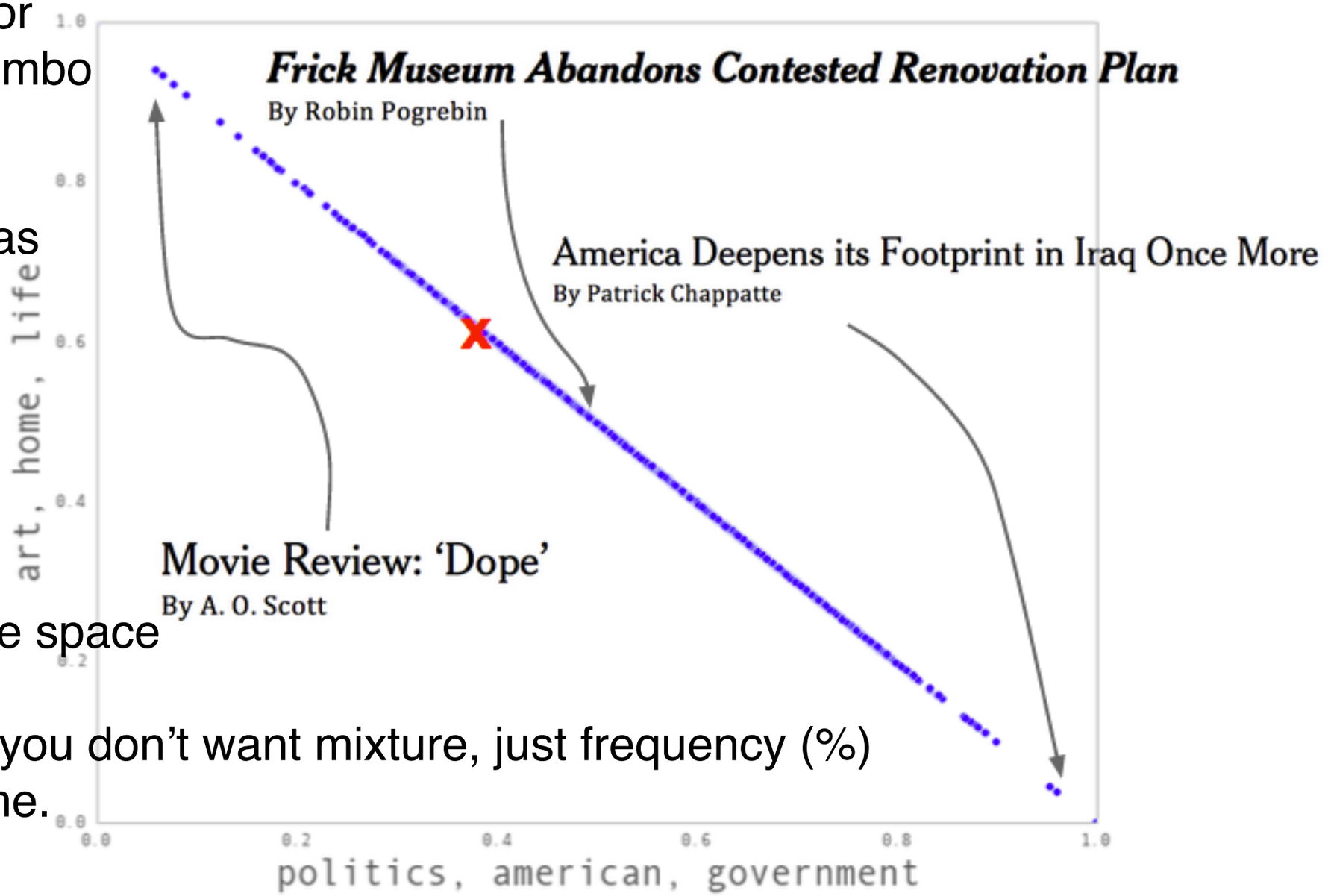
Recommendation Systems and LDA in the NY Times

blue dots: individual article LDA describe for each article, combo of topics.

X: user, NYT has data on what reader reads

ppl on same axes as article recommend nearby on same space

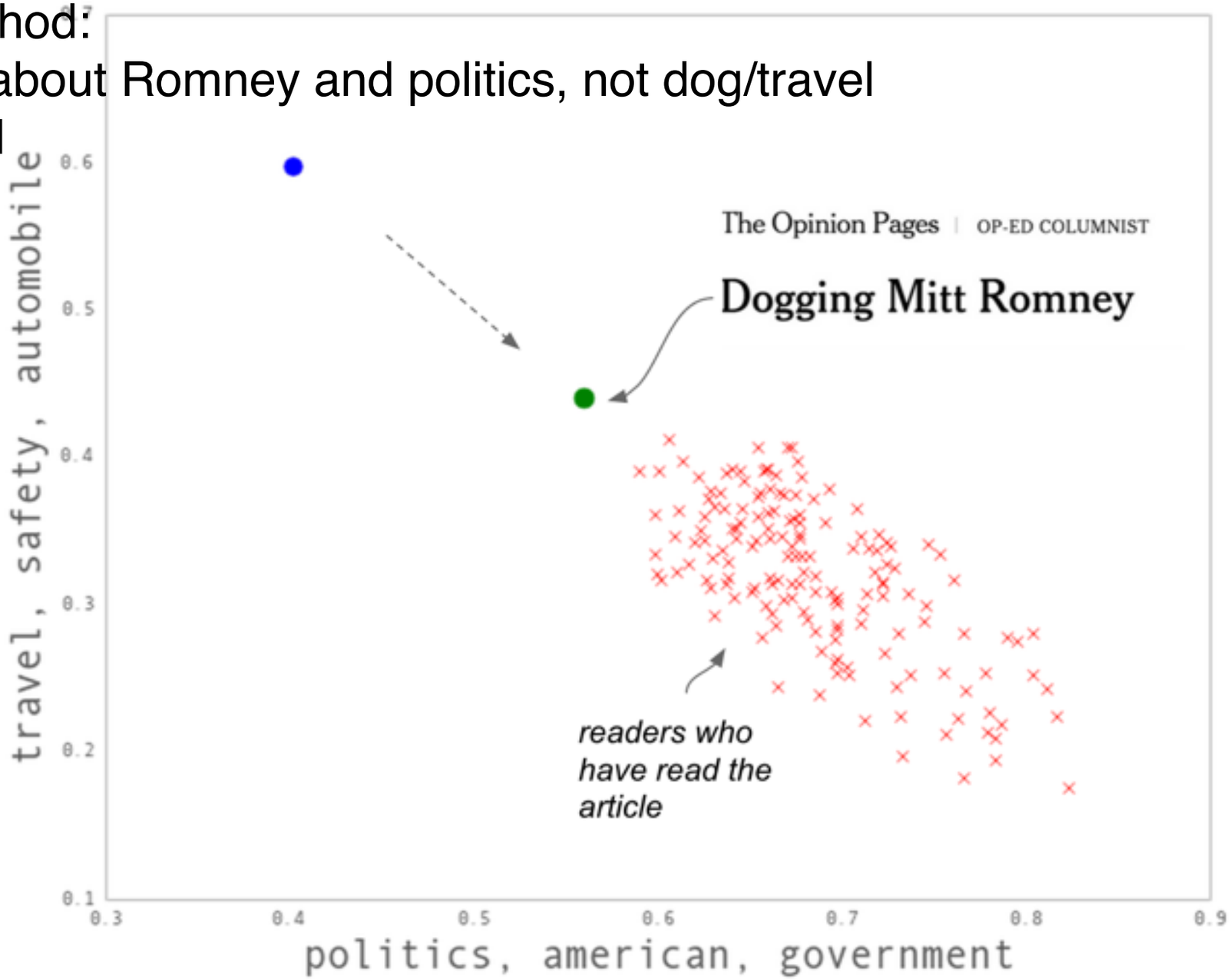
issues: maybe you don't want mixture, just frequency (%) at either extreme.



http://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/?_r=2

Recommendation Systems and LDA in the NY Times

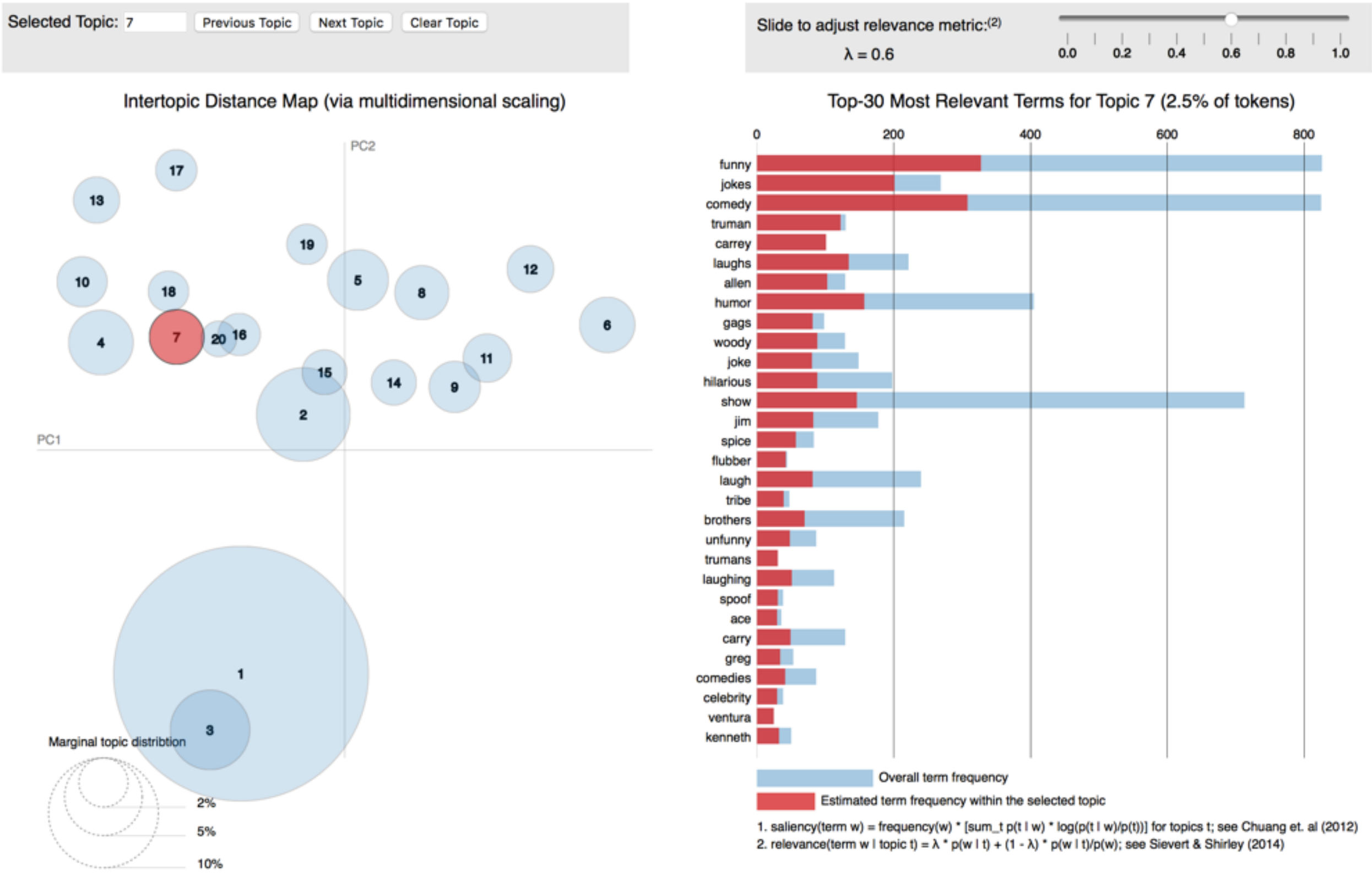
issues with method:
article is really about Romney and politics, not dog/travel
condition based on preferences of readers.



http://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/?_r=2

always get rid of stop words first, common and useless: the, this, that, a, an

LDA Visualization



http://cpsievert.github.io/LDAvis/reviews/reviews.html
pyLDAvis: https://pypi.python.org/pypi/pyLDAvis