AN INTRODUCTION TO TOPIC MODELS

Michael Paul

December 4, 2013

600.465 Natural Language Processing

Johns Hopkins University

Prof. Jason Eisner

Making sense of text

Suppose you want to learn something about a corpus that's too big to read

- What topics are trending today on Twitter?
- What research topics receive grant funding (and from whom)?
- What issues are considered by Congress (and which politicians are interested in which topic)?
- Are certain topics discussed more in certain languages on Wikipedia?

need to make sense of…

- **half a billion** tweets daily
- **80,000** active NIH grants
- **hundreds** of bills each year
- **Wikipedia** (it's big)

Making sense of text

Suppose you want to learn something about a corpus that's too big to read

Why don't we just throw all these documents at the computer and see what interesting patterns it finds?

need to make sense of…

- **half a billion** tweets daily
- **80,000** active NIH grants
- **hundreds** of bills each year
- **Wikipedia** (it's big)

Preview

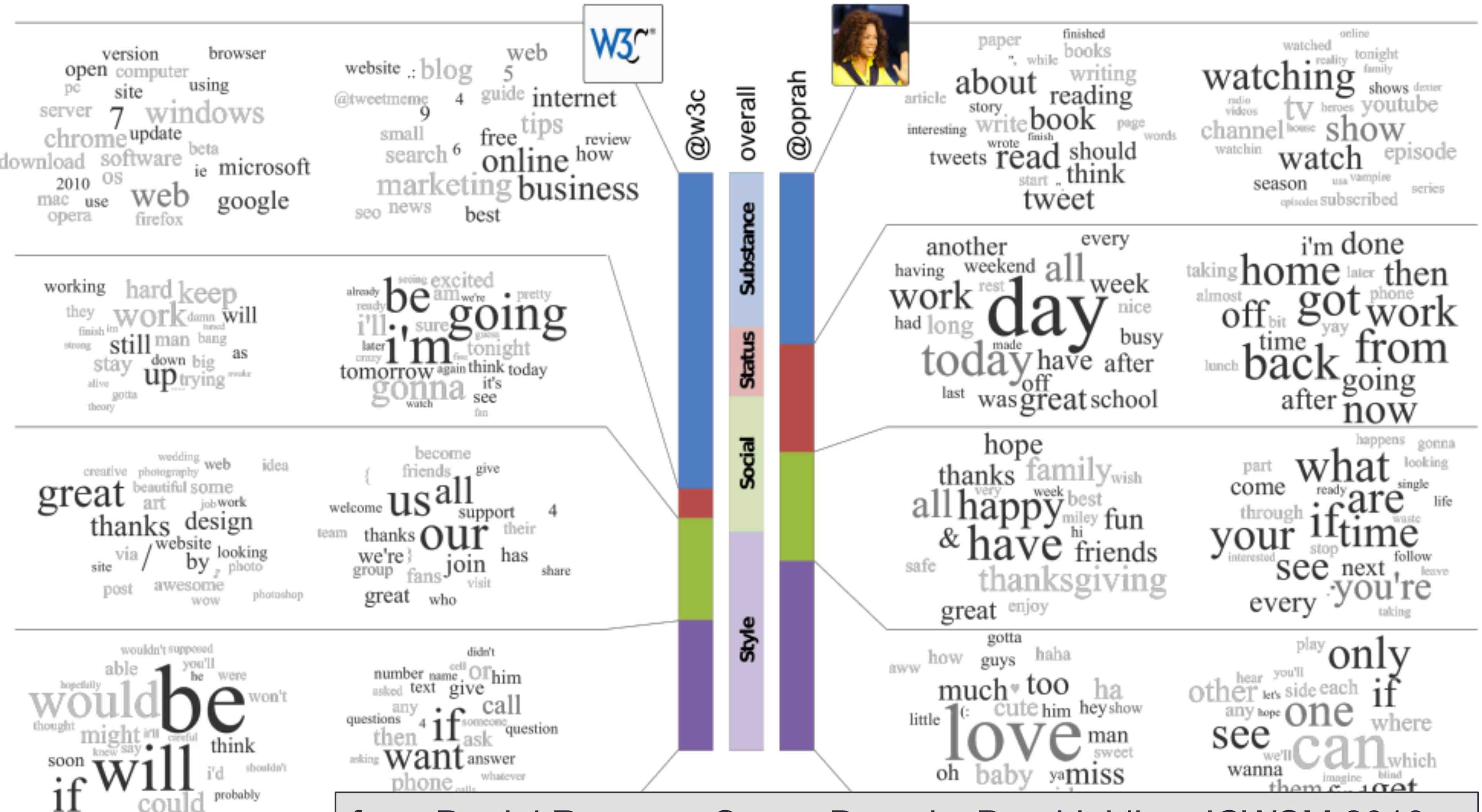
- Topic models can help you automatically discover patterns in a corpus
  - **unsupervised** learning

- Topic models automatically…
  - group topically-related words in “topics”
  - associate tokens and documents with those topics

Twitter topics

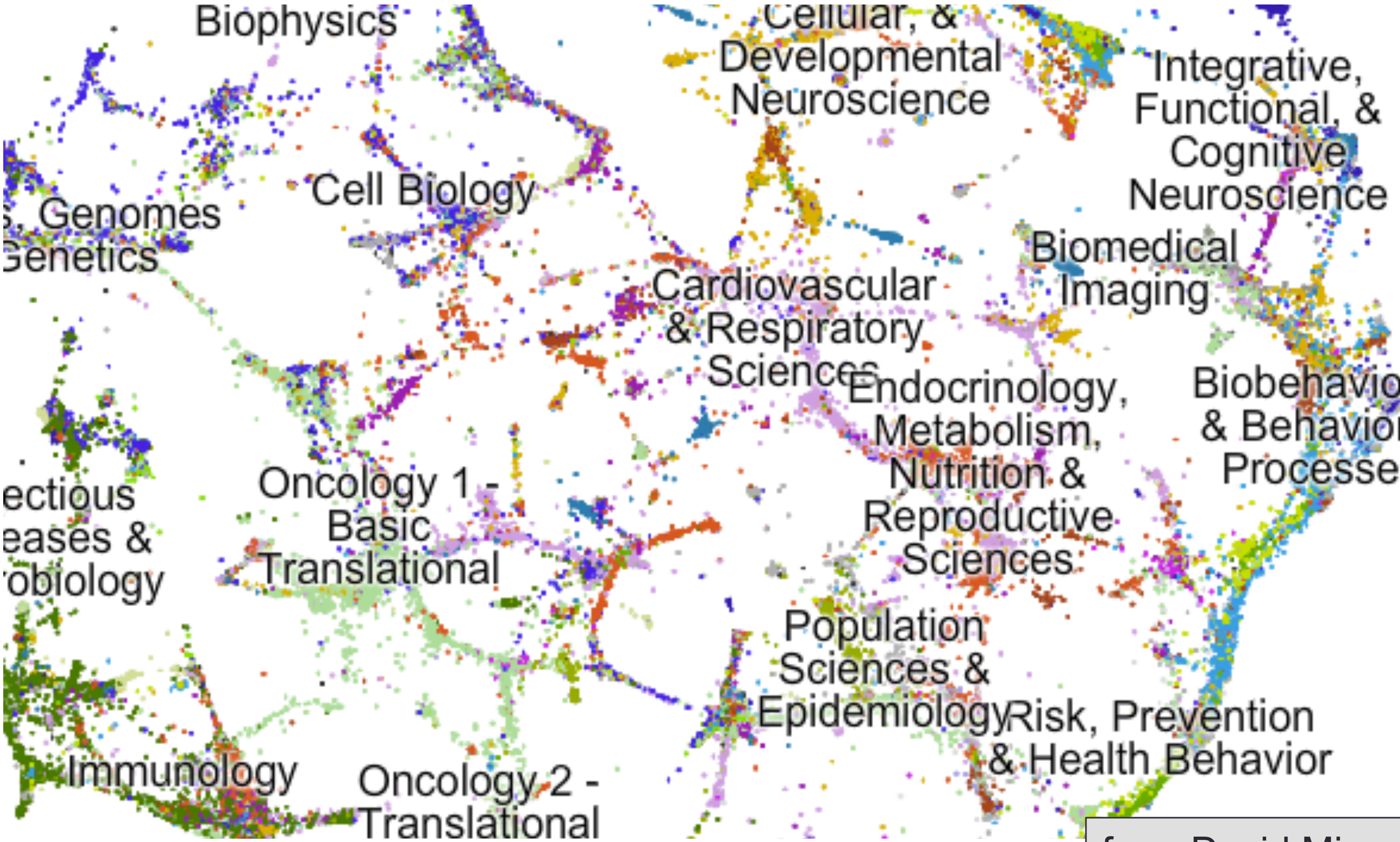
from Daniel Ramage, Susan Dumais, Dan Liebling. ICWSM 2010.

Twitter topics



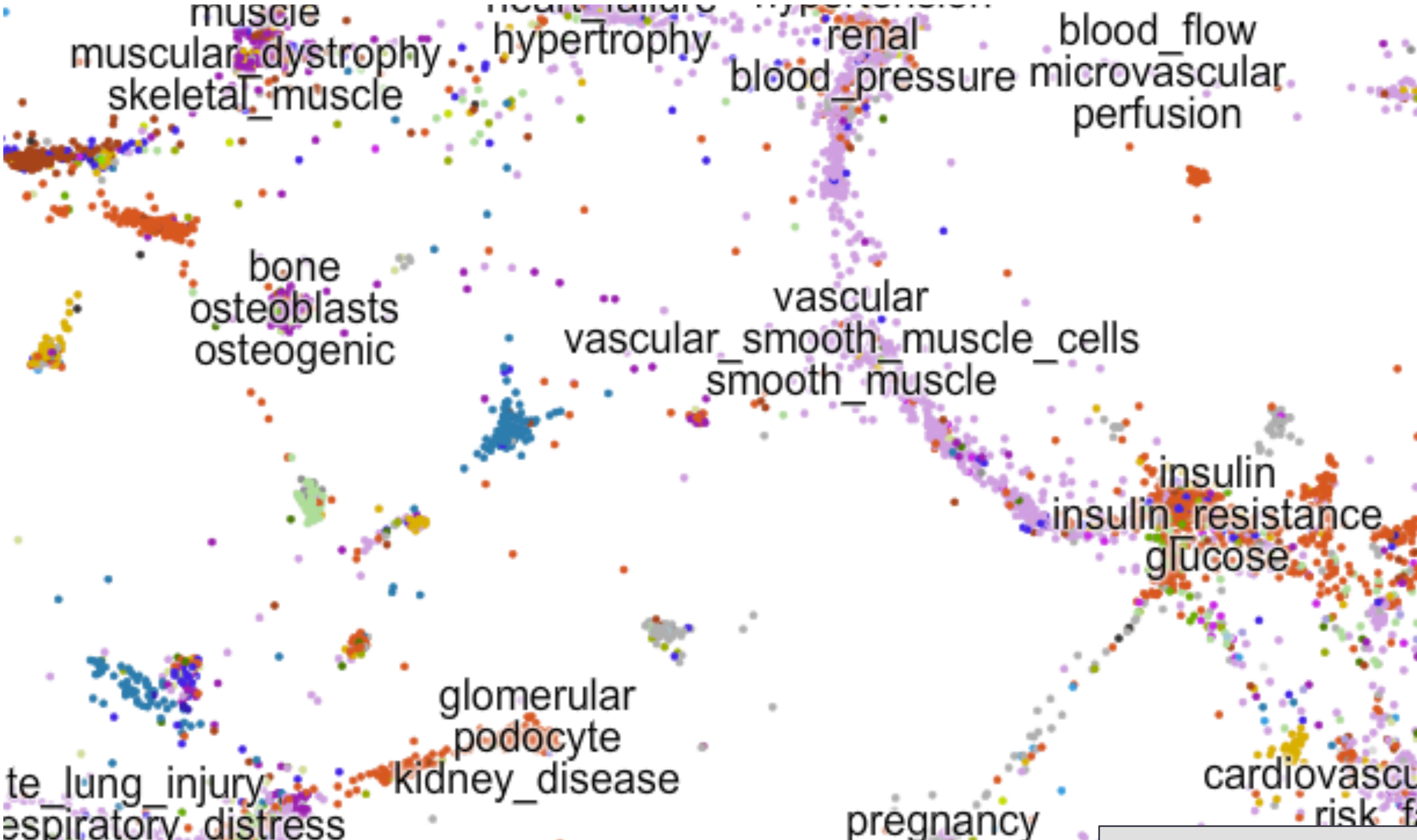
from Daniel Ramage, Susan Dumais, Dan Liebling. ICWSM 2010.

Research grants



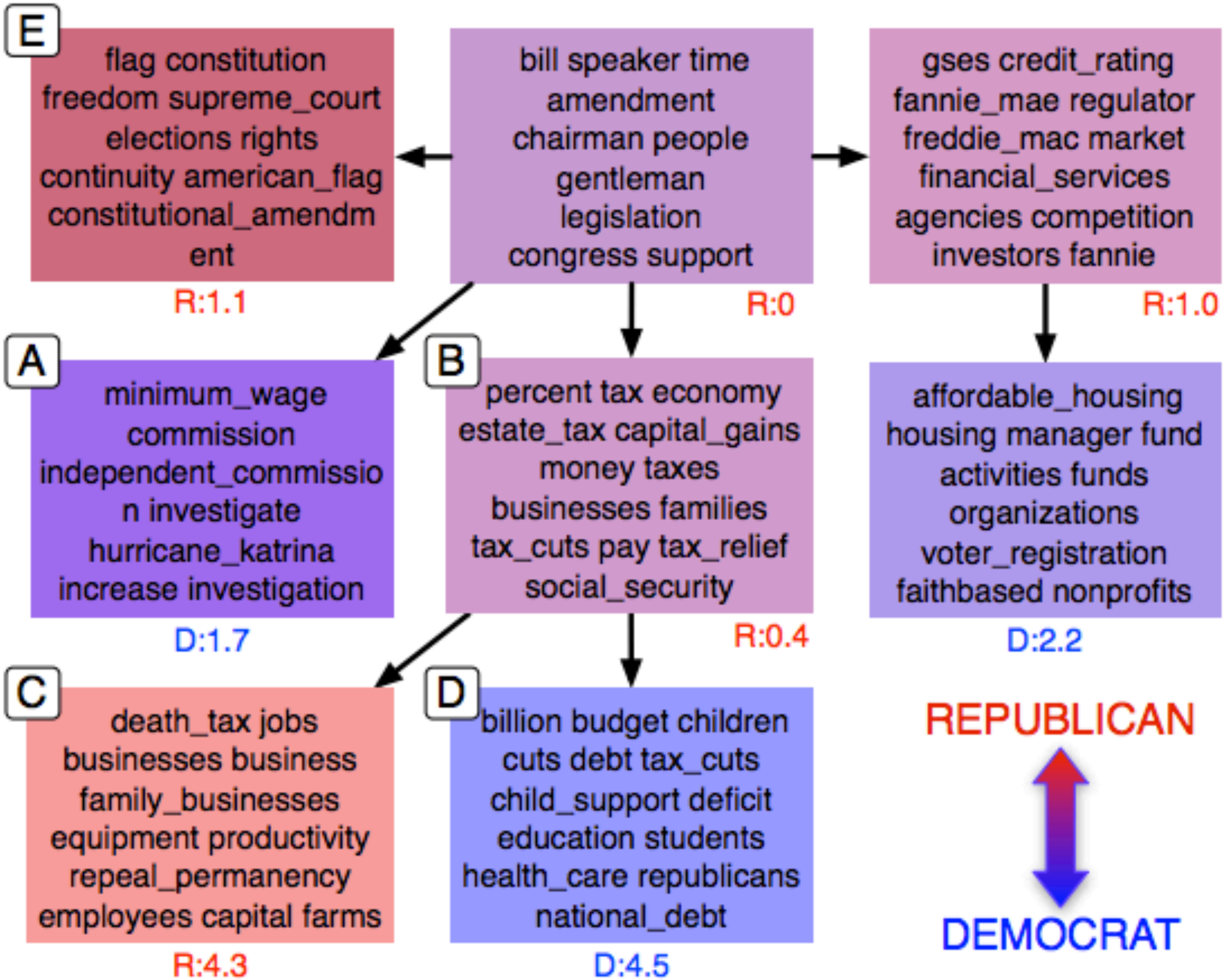
from David Mimno

Research grants



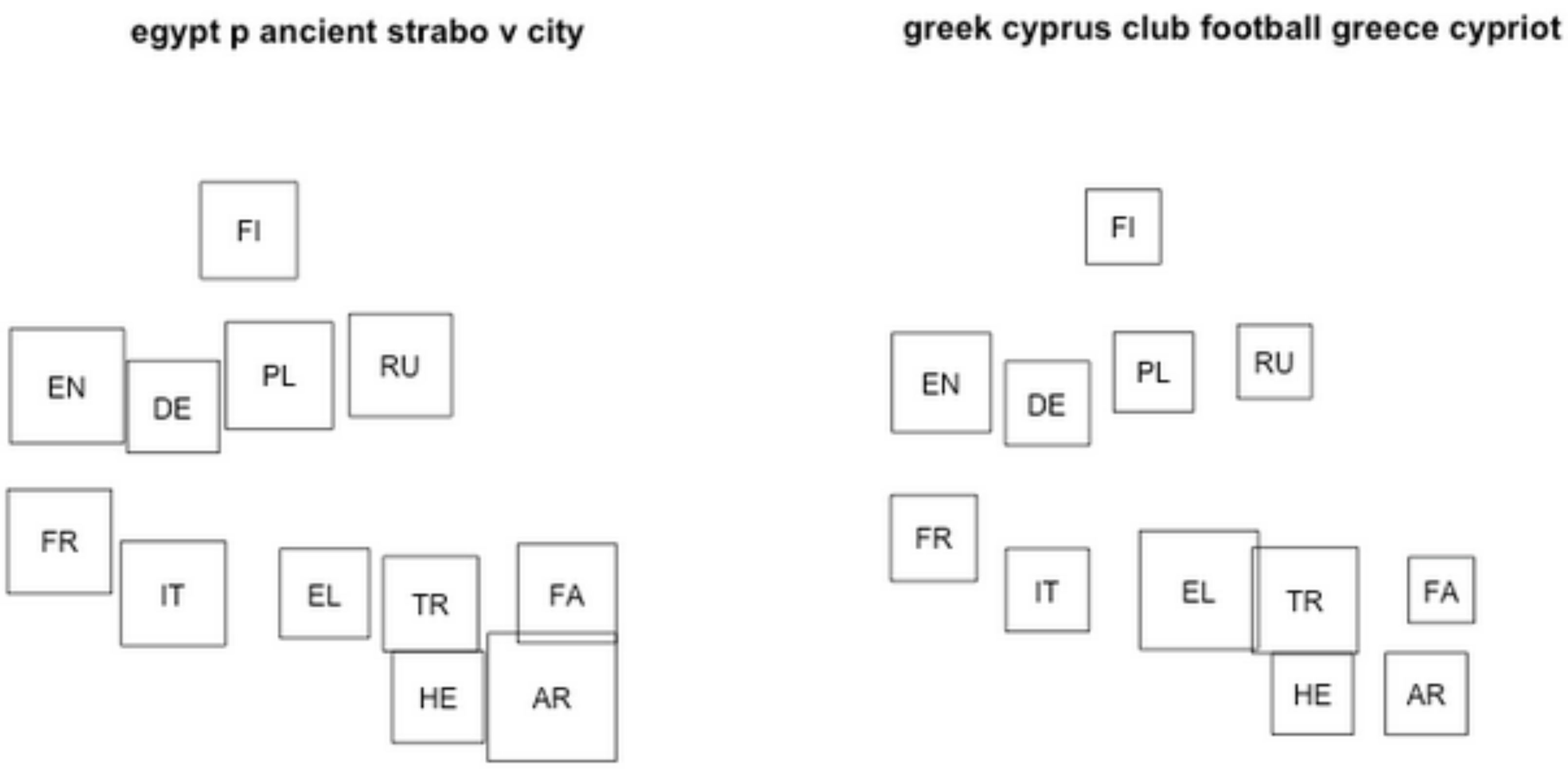
from David Mimno

Political issues



from Viet-An Nguyen, Jordan Boyd-Graber, Phillip Resnik. NIPS 2013.

Classics and “digital humanities”



from David Mimno

So what is “topic”?

- Loose idea: a grouping of words that are likely to appear in the same **context**

- A hidden structure that helps determine what words are likely to appear in a corpus
  - but the underlying structure is different from what you’ve seen before – it’s not syntax
  - e.g. if “war” and “military” appear in a document, you probably won’t be surprised to find that “troops” appears later on

    why? it’s not because they’re all nouns

    …though you might say they all belong to the same *topic*
  - long-range context (cf. local dependencies like n-grams, syntax)

This lecture

1. ~~Topic models: informal definition~~

2. Topic models: formal definition

3. Smoothing, EM, and Bayesian inference

You’ve seen these ideas before

Most of NLP is about inferring hidden structures that we assume are behind the observed text

- parts of speech, syntax trees

You’ve already seen a model that can capture topic

- let’s look at HMMs again

Hidden Markov models

Every token is associated with some hidden state

- the probability of the word token depends on the state
- the probability of that token's state depends on the state of the previous token (in a 1st order model)

- The states are not observed, but you can infer them using the forward-backward algorithm

Hidden Markov models

HMM is a reasonable model of part-of-speech:

Stocks mixed after long holiday weekend

Microsoft codename 'Threshold': The next major Windows

Apple iPads beat early holiday expectations

- coloring corresponds to value of hidden state (POS)

Hidden Markov models

HMM is a reasonable model of part-of-speech:

Stocks mixed after long holiday weekend

Microsoft codename 'Threshold': The next major Windows

Apple iPads beat early holiday expectations

but you might imagine modeling topic associations instead:

Stocks mixed after long holiday weekend

Microsoft codename 'Threshold': The next major Windows

Apple iPads beat early holiday expectations

Topic models

Take an HMM, but give every document its own transition probabilities (rather than a global parameter of the corpus)

- This let's you specify that certain topics are more common in certain documents
  - whereas with parts of speech, you probably assume this doesn't depend on the specific document

Topic models

Take an HMM, but give every document its own transition probabilities (rather than a global parameter of the corpus)

- This let's you specify that certain topics are more common in certain documents
  - whereas with parts of speech, you probably assume this doesn't depend on the specific document
- We'll also assume the hidden state of a token doesn't actually depend on the previous tokens
  - "0th order"
  - individual documents probably don't have enough data to estimate full transitions
  - plus our notion of "topic" doesn't care about local interactions

Topic models

- The probability of a token is the joint probability of the word and the topic label

P(word=Apple, topic=1 | θ_d , β_1)

= P(word=Apple | topic=1, β_1) P(topic=1 | θ_d)

Topic models

- The probability of a token is the joint probability of the word and the topic label

$$P(\text{word}=\text{Apple}, \text{topic}=1 \mid \theta_d, \beta_1)$$
$$= P(\text{word}=\text{Apple} \mid \text{topic}=1, \beta_1)\, P(\text{topic}=1 \mid \theta_d)$$

each topic has distribution over *words*
(the emission probabilities)

- **global** across all documents

each document has distribution over *topics*
(the 0th order "transition" probabilities)

- **local** to each document

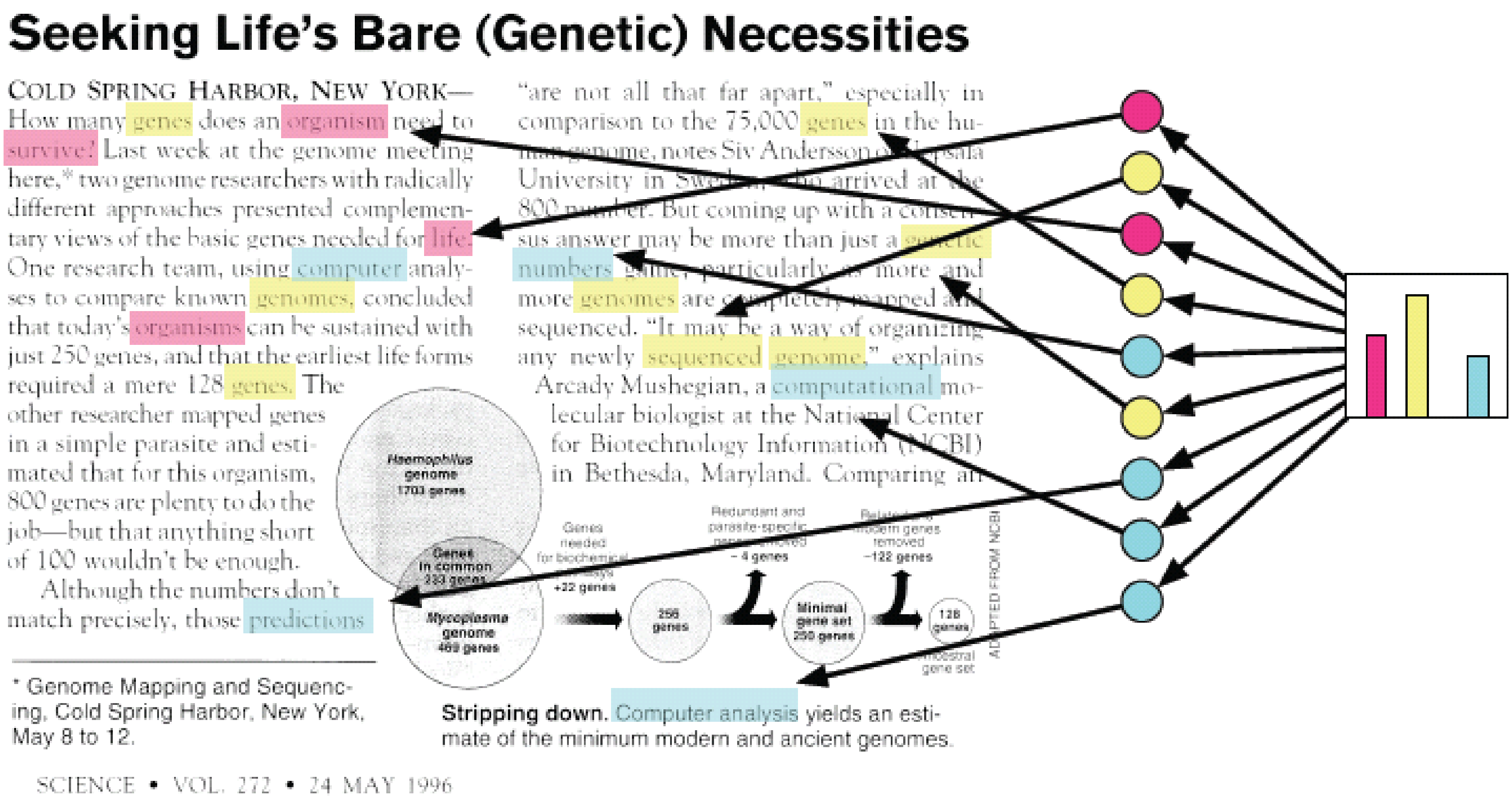
Topic models

- The probability of a token is the joint probability of the word and the topic label

P(word=Apple, topic=1 | θ_d , β_1)
= P(word=Apple | topic=1, β_1) P(topic=1 | θ_d)

- The probability of a document is the product of all of its token probabilities
  - the tokens are independent because it's a 0th order model
- The probability of a corpus is the product of all of its document probabilities

Topic models



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

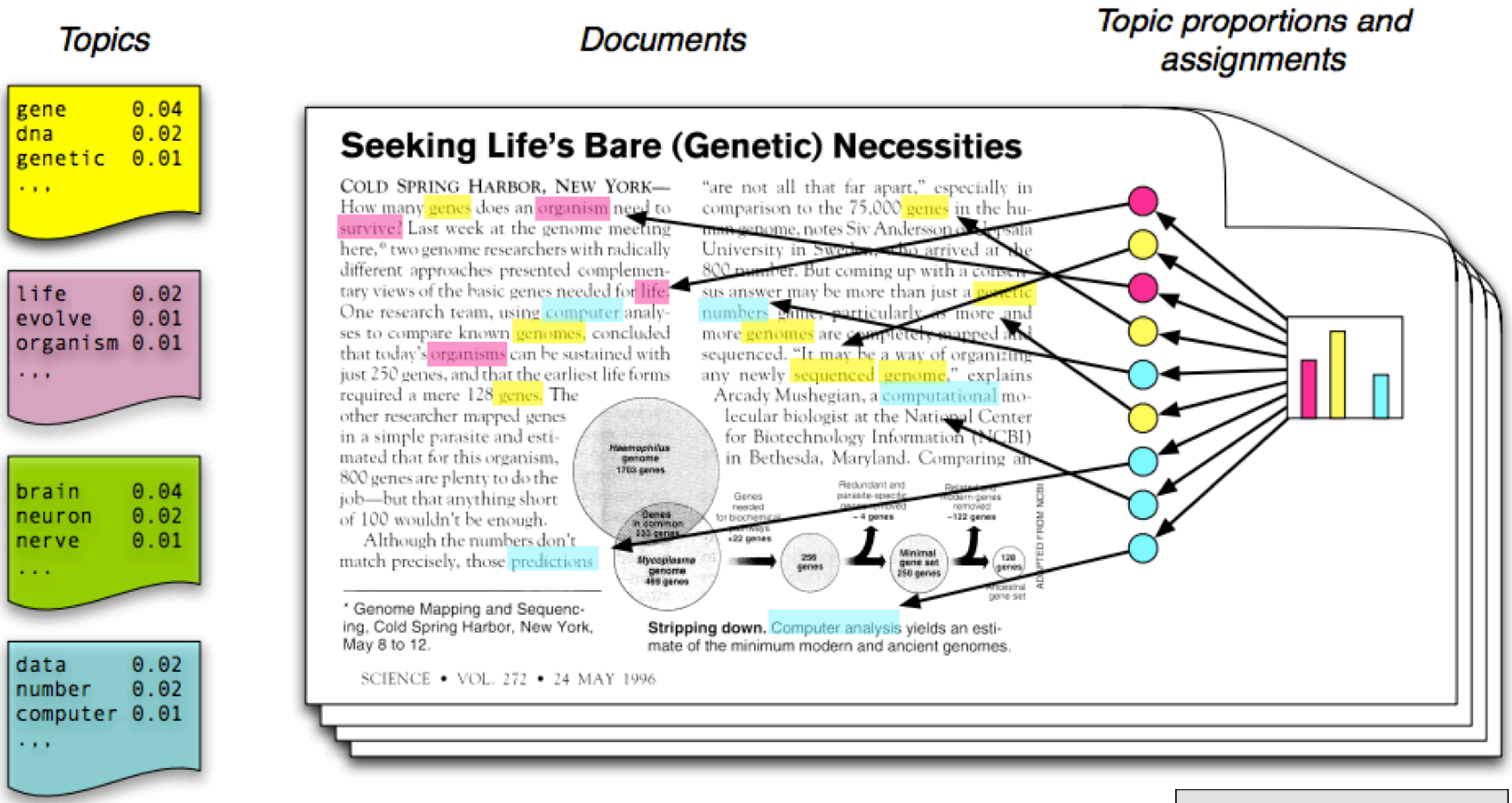
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

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

SCIENCE • VOL. 272 • 24 MAY 1996

from David Blei

Topic models



from David Blei

Estimating the parameters

- Need to estimate the parameters θ, β
  - want to pick parameters that maximize the likelihood of the observed data
- This is easy if all the tokens were labeled with topics (observed variables)

Data: Apple iPads beat early holiday expectations

  - just counting
- But we don't actually know the (hidden) topic assignments

Data: Apple iPads beat early holiday expectations

  - sound familiar?

Estimating the parameters

Expectation Maximization (EM) to the rescue!

1. Compute the expected value of the variables, given the current model parameters

2. Pretend these *expected* counts are real and update the parameters based on these
   - now parameter estimation is back to "just counting"

3. Repeat until convergence

Estimating the parameters

Expectation Maximization (EM) to the rescue!

E-step

P(topic=1 | word=Apple, θ_d , β_1)

$$= \frac{\text{P(word=Apple, topic=1} \mid \theta_d , \beta_1)}{\Sigma_k \text{ P(word=Apple, topic=}k \mid \theta_d , \beta_k)}$$

Estimating the parameters

Expectation Maximization (EM) to the rescue!

M-step

new θ_{d1}

$$= \frac{\text{\# tokens in } d \text{ with topic label 1}}{\text{\# tokens in } d}$$

if *the topic labels were observed!*

- just counting

Estimating the parameters

Expectation Maximization (EM) to the rescue!

M-step

new θ_{d1}

$$= \frac{\sum_{i \in d} P(\text{topic } i=1 \mid \text{word } i, \theta_d, \beta_1)}{\sum_k \sum_{i \in d} P(\text{topic } i=k \mid \text{word } i, \theta_d, \beta_k)}$$

just the number of tokens in the document

sum over each token *i* in document *d*

- numerator: “the expected number of tokens with topic 1”
- denominator: “the (expected) number of tokens”

Estimating the parameters

Expectation Maximization (EM) to the rescue!

M-step

$$\text{new } \beta_{1w} = \frac{\#\text{ tokens with topic label 1 and word type } w}{\#\text{ tokens with topic label 1}}$$

if *the topic labels were observed!*

- just counting

Estimating the parameters

Expectation Maximization (EM) to the rescue!

M-step

new β_{1w}

$$= \frac{\sum_i \overbrace{\mathrm{I}(\text{word } i\text{=w})}^{\text{1 if word=w, 0 otherwise}}\, P(\text{topic } i=1 \mid \text{word } i=w,\, \theta_d,\, \beta_1)}{\sum_v \sum_i \mathrm{I}(\text{word } i\text{=v})\, P(\text{topic } i=1 \mid \text{word } i=v,\, \theta_d,\, \beta_1)}$$

sum over vocabulary

sum over each token *i* in the entire corpus

- numerator: “the expected number of times word *w* belongs to topic 1”
- denominator: “the expected number of all tokens belonging to topic 1”

Smoothing revisited

- Topics are just language models
- Can use standard smoothing techniques for the topic parameters (the word distributions)
  - most commonly add-lambda smoothing
- Can also smooth the topic proportions in each document

Smoothing: A Bayesian perspective

- The parameters themselves are random variables
  - $P(\theta \mid \alpha)$
  - $P(\beta \mid \eta)$

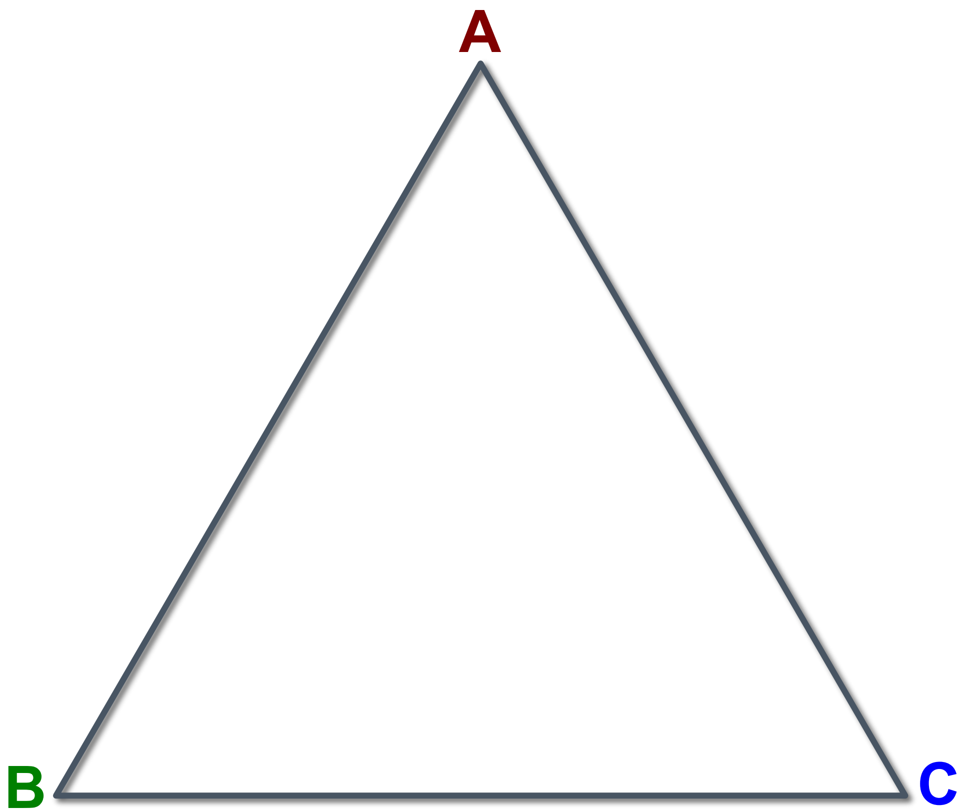
- Some parameters are more likely than others
  - as defined by a **prior** distribution

- You'll see that add-lambda smoothing is the result when the parameters have a prior distribution called the **Dirichlet** distribution
  - (in fact, add-lambda is called "Dirichlet prior smoothing" in some circles)

Geometry of probability distributions

A distribution over K elements is a point on a K-1 **simplex**

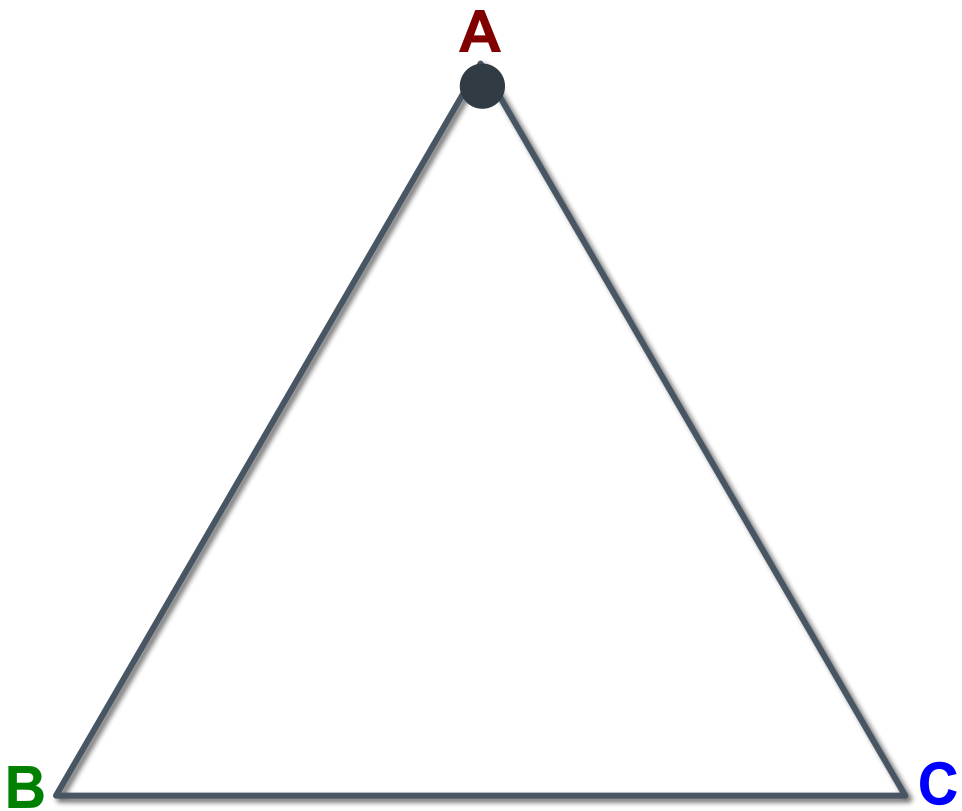
- a 2-simplex is called a triangle



Geometry of probability distributions

A distribution over K elements is a point on a K-1 **simplex**

- a 2-simplex is called a triangle

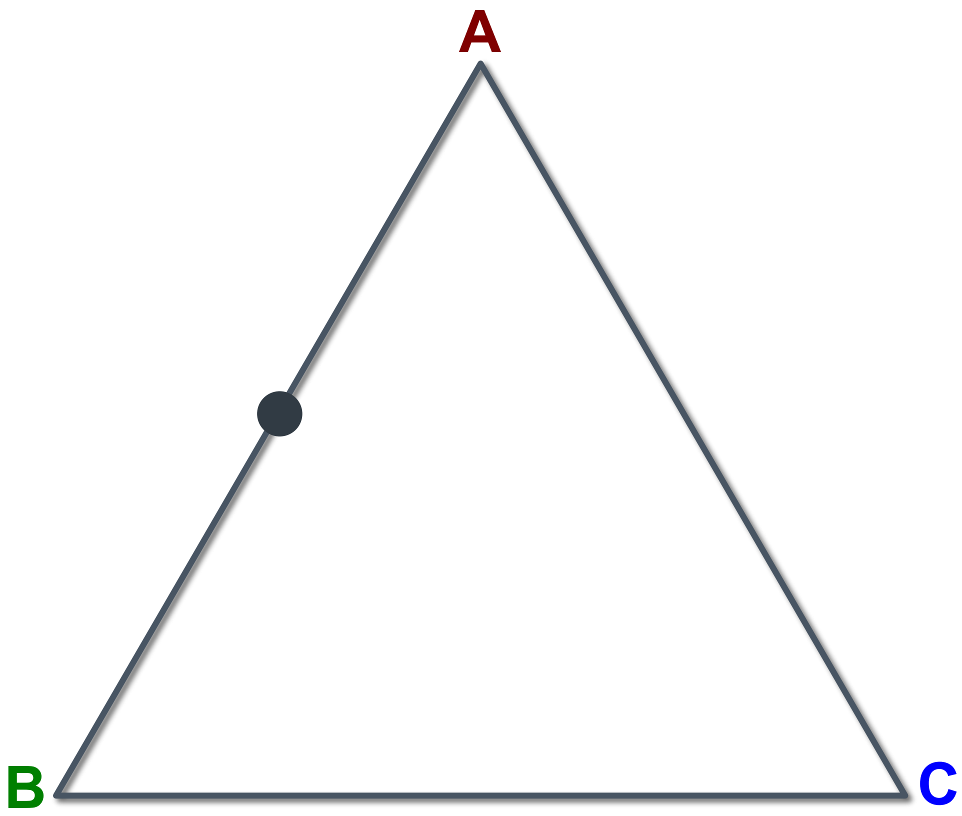


P(**A**) = 1
P(**B**) = 0
P(**C**) = 0

Geometry of probability distributions

A distribution over K elements is a point on a K-1 **simplex**

- a 2-simplex is called a triangle

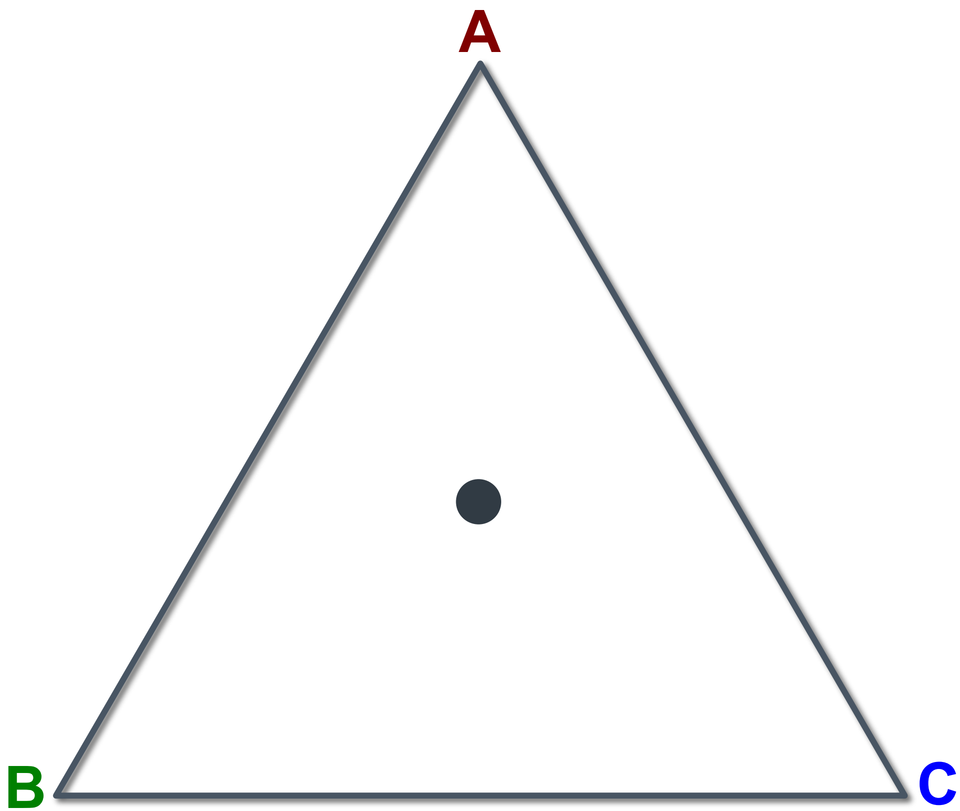


P(**A**) = 1/2
P(**B**) = 1/2
P(**C**) = 0

Geometry of probability distributions

A distribution over K elements is a point on a K-1 **simplex**

- a 2-simplex is called a triangle

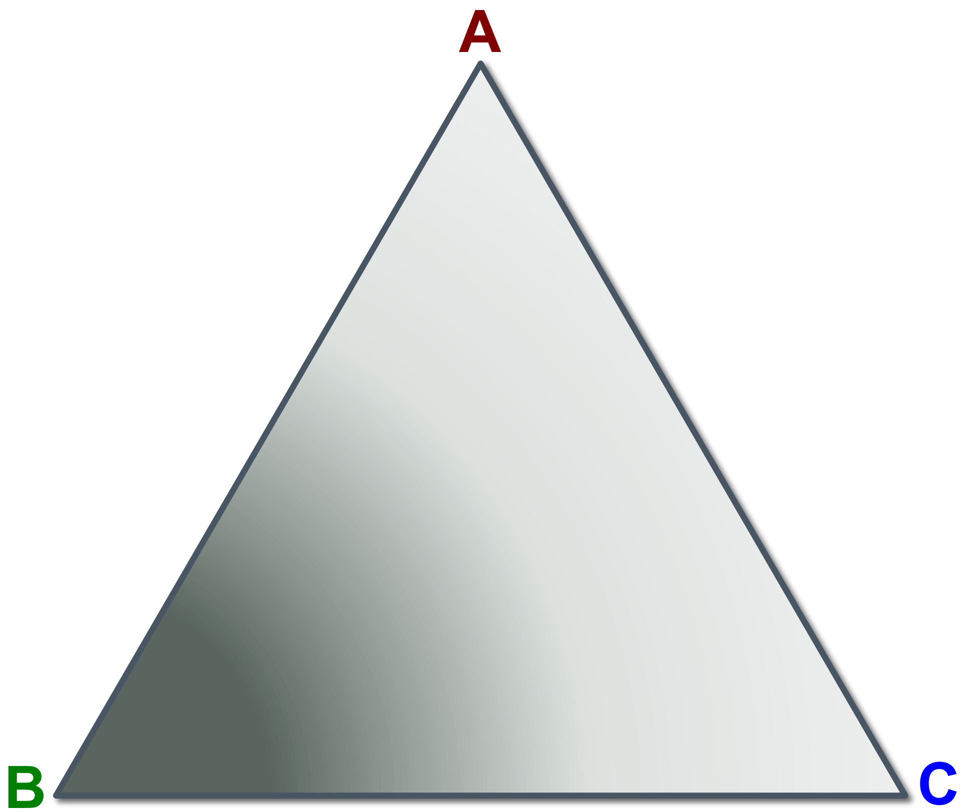


P(**A**) = 1/3
P(**B**) = 1/3
P(**C**) = 1/3

The Dirichlet distribution

Continuous distribution (probability density) over points in the simplex

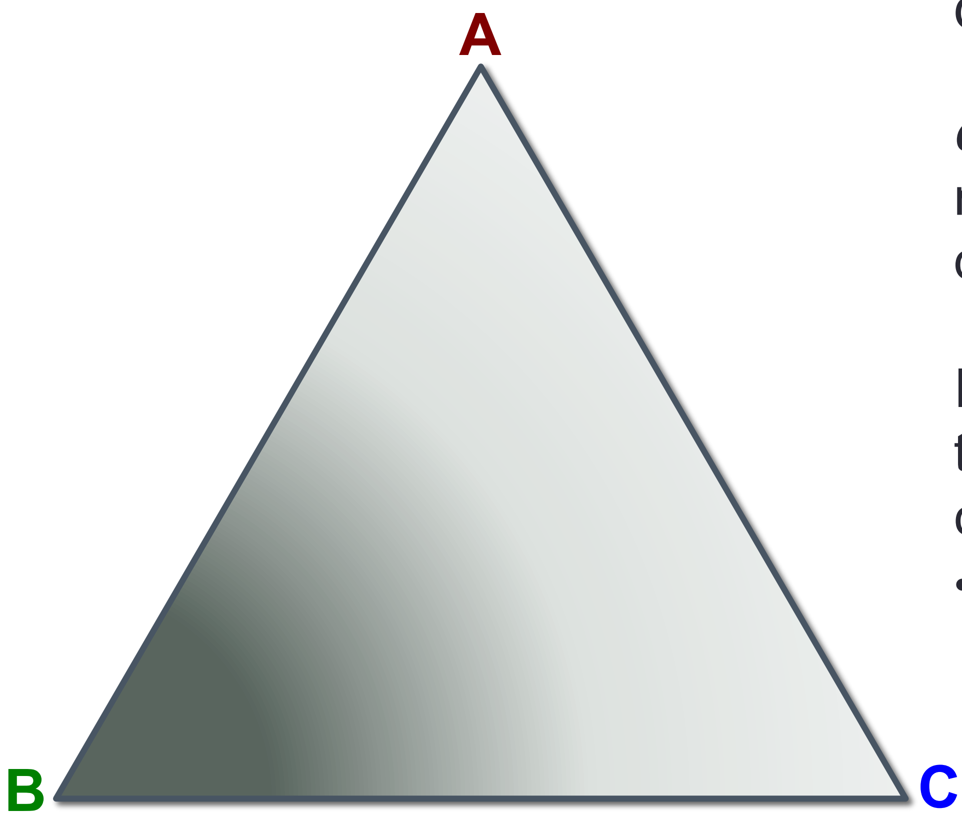
- "distribution of distributions"



The Dirichlet distribution

Continuous distribution (probability density) over points in the simplex

- “distribution of distributions”



denoted Dirichlet($\boldsymbol{\alpha}$)

$\boldsymbol{\alpha}$ is a vector that gives the mean/variance of the distribution

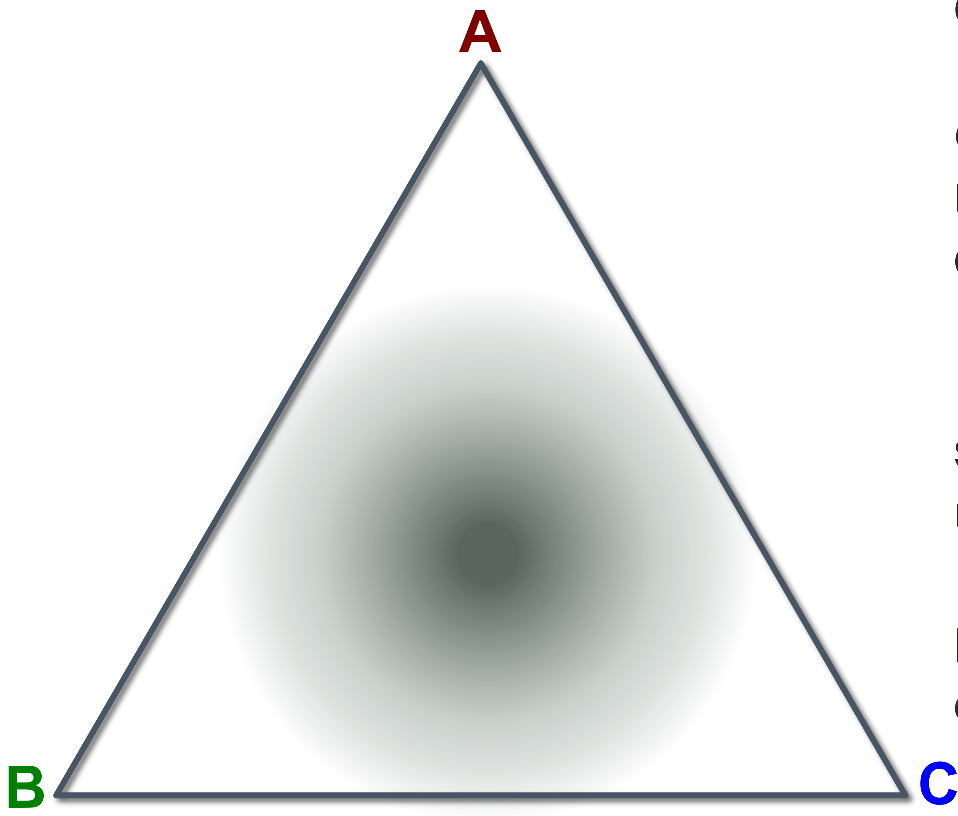
In this example, α_B is larger than the others, so points closer to B are more likely

- distributions that give **B** high probability are more likely than distributions that don’t

The Dirichlet distribution

Continuous distribution (probability density) over points in the simplex

- "distribution of distributions"



denoted Dirichlet(α)

α is a vector that gives the mean/variance of the distribution

In this example, $\alpha_A=\alpha_B=\alpha_C$, so distributions close to uniform are more likely

Larger values of α mean higher density around mean

(lower variance)

Latent Dirichlet allocation (LDA)

LDA is the basic topic model you saw earlier, but with Dirichlet priors on the parameters θ and β

- P(θ | α) = Dirichlet(α)
- P(β | η) = Dirichlet(η)

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

The posterior distribution

- Now we can reason about the probability of the hidden variables and parameters, given the observed data

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

MAP estimation

- Earlier we saw how to use EM to find parameters that maximize the likelihood of the data, given the parameters
- EM can also find the **maximum a posteriori (MAP)** value
  - the parameters that maximum the posterior probability

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

constant

- This is basically maximum likelihood estimation, but with additional terms for the probability of θ and β

MAP estimation

- E-step is the same
- M-step is modified

pseudocounts

new θ_{d1}

$$= \frac{\boxed{\alpha_1 - 1} + \Sigma_{i \in d} \mathrm{P}(\text{topic } i = 1 \mid \text{word } i, \theta_d, \beta_1)}{\Sigma_k \left(\alpha_k - 1 + \Sigma_{i \in d} \mathrm{P}(\text{topic } i = k \mid \text{word } i, \theta_d, \beta_k)\right)}$$

This amounts to add-lambda smoothing!

“add-alpha-minus-one smoothing”

Where do the pseudocounts come from?

The probability of observing the *k*th topic *n* times given the parameter θ_k is proportional to:

$$\theta_k^{\,n}$$

The probability density of the parameter θ_k given the Dirichlet parameter α_k is proportional to:

$$\theta_k^{\,\alpha_k - 1}$$

So the product of these probabilities is proportional to:

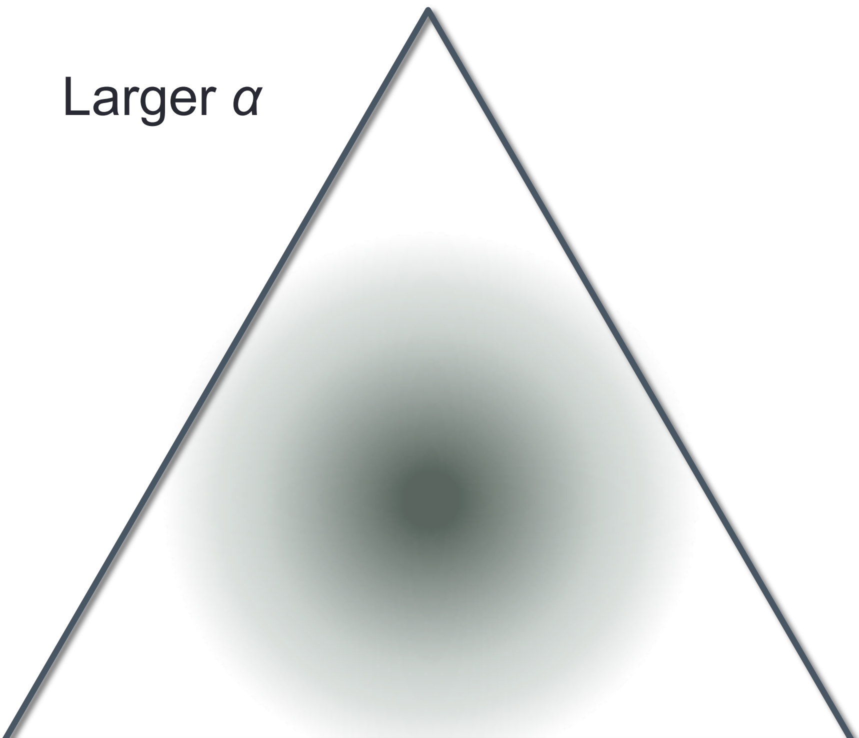
$$\theta_k^{\,n + \alpha_k - 1}$$

Smoothing: A Bayesian perspective

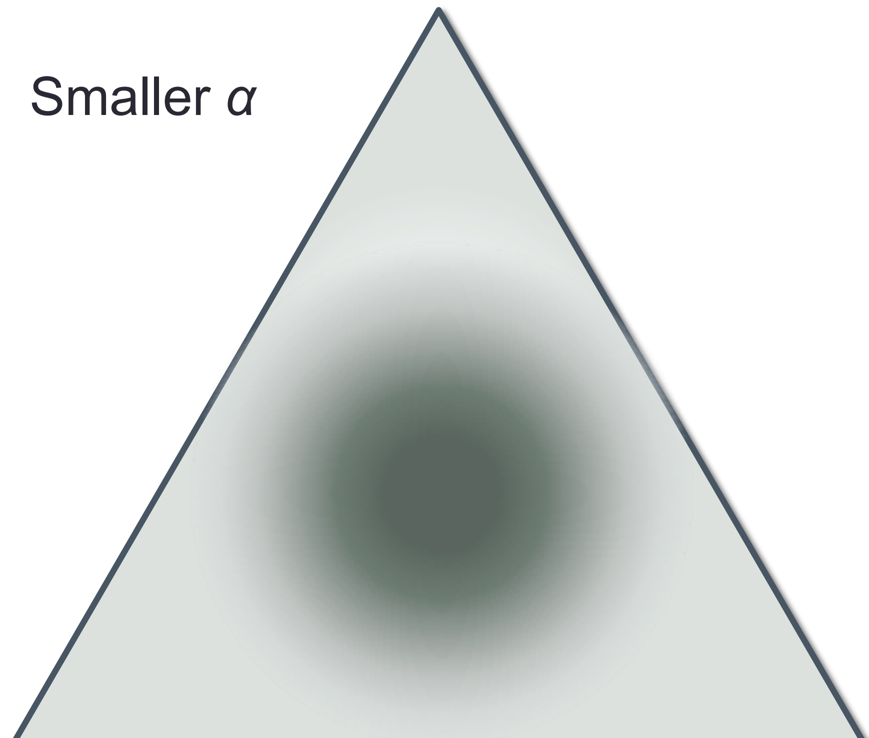
Larger pseudocounts will bias the MAP estimate more heavily

Larger Dirichlet parameters concentrate the density around the mean

Larger α

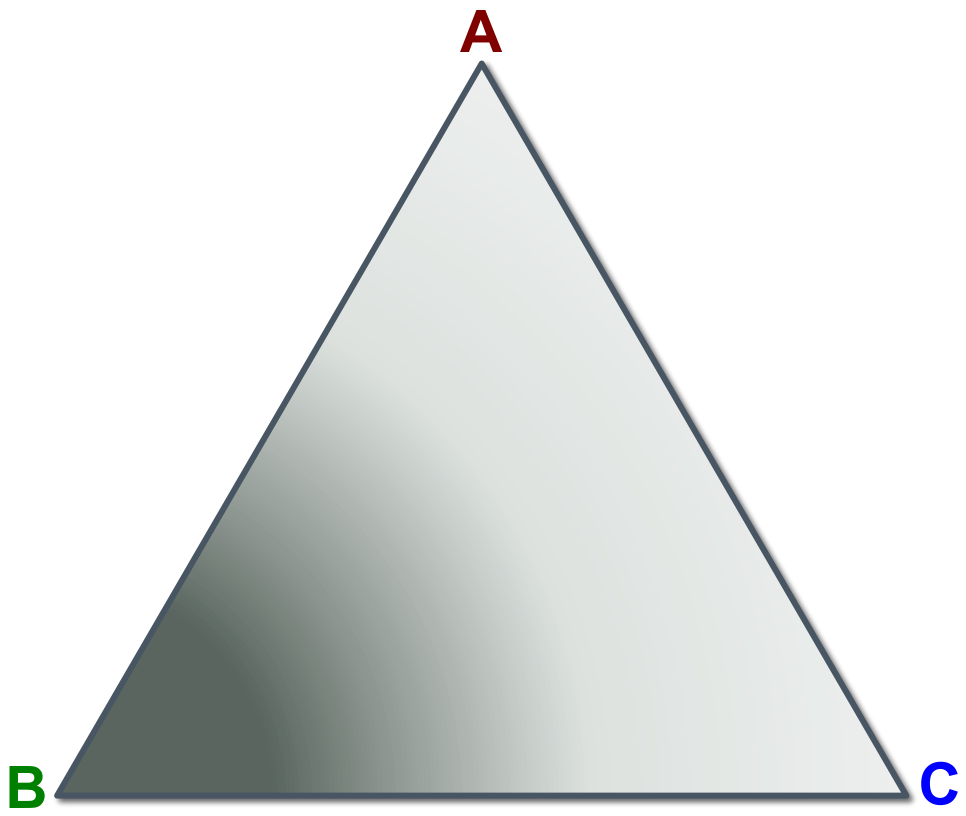


Smaller α



Asymmetric smoothing

We don't have to smooth toward the uniform distribution



Asymmetric smoothing

We don't have to smooth toward the uniform distribution

- You might expect one topic to be very common in all documents

Symmetric α	0.080	a **field emission** an **electron** the
	0.080	a the **carbon** and **gas** to an
	0.080	the of a to and about at
	0.080	of a **surface** the with in **contact**
	0.080	the a and to is of **liquid**
Asymmetric α	0.895	the a of to and is in
	0.187	**carbon nanotubes nanotube catalyst**
	0.043	sub is c or and n sup
	0.061	**fullerene compound fullerenes**
	0.044	**material particles coating inorganic**

from Hanna Wallach, David Mimno, Andrew McCallum. NIPS 2009.

“Negative” smoothing

- Dirichlet prior MAP estimation yields “$\alpha - 1$” smoothing
  - So what happens if $\alpha < 1$?

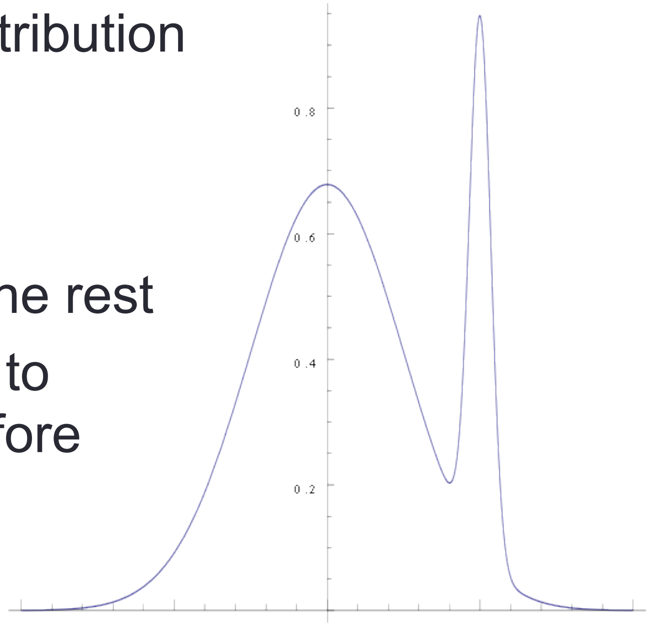
Posterior inference

What if we don't just want the parameters that maximize the posterior?

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

What if we care about the entire posterior distribution?

- or at least the mean of the posterior distribution



Why?

- maybe the maximum doesn't look like the rest
- other points of the posterior more likely to generalize to data you haven't seen before

Posterior inference

What if we don't just want the parameters that maximize the posterior?

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

This is harder

- Computing the denominator involves marginalizing over all possible configurations of the hidden variables/parameters

Posterior inference: approximations

- Random sampling
  - Monte Carlo methods

- Variational inference
  - Optimization using EM-like procedure
  - MAP estimation is a simple case of this

I didn’t tell you…

- where the number of topics K comes from
- where the Dirichlet parameters α and η come from

Extensions

- n-grams
- topic hierarchies
- supervision

- can you think of other ideas?