Topic Models: Sampling, Inference, and Applications

CS224N // 13 November 2013

Spence Green

Stanford University

(with material from David Blei, Bob Carpenter, Matt Gormley, and Dan Ramage)

Plan for today

Topic model applications

Preliminaries

Sampling

Bayesian Naive Bayes

Latent Dirichlet Allocation

Caveat emptor

Some math today. But this is an NLP class!

Caveat emptor

Some math today. But this is an NLP class!

Preliminaries

- Start with maximum likelihood
- Build from CS224N material

Caveat emptor

Some math today. But this is an NLP class!

Preliminaries

- Start with maximum likelihood
- Build from CS224N material

Theory, derivations: CS228 / CS228T / STATS213

Problem: I got this new numeric/categorical data set. Now what?

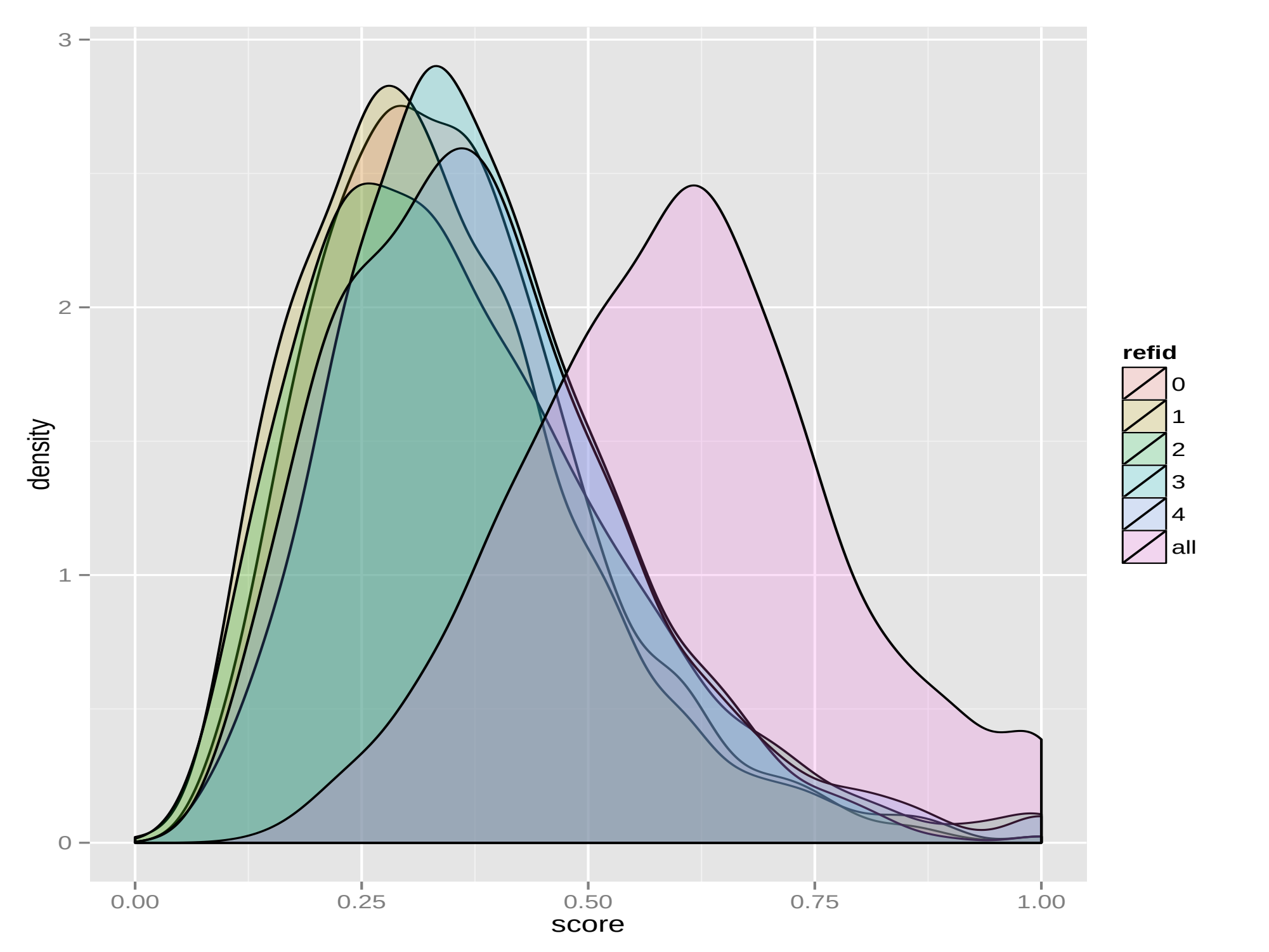
Problem: I got this new numeric/categorical data set. Now what?

Idea: Exploratory data analysis (look at the data)

Problem: I got this new numeric/categorical data set. Now what?

Idea: Exploratory data analysis (look at the data)

Concrete Idea: visualize it


3
2
1
0
density
0.00
0.25
0.50
0.75
1.00
score
refid
0
1
2
3
4
all

Problem: I got this new **text corpus**. Now what?

Problem: I got this new **text corpus**. Now what?

Idea #1: Compute gross statistics (#tokens, #word types, etc.)

Problem: I got this new **text corpus**. Now what?

Idea #1: Compute gross statistics (#tokens, #word types, etc.)

Idea #2: Label words. Visualize distributions of labels.

Problem: I got this new **text corpus**. Now what?

Idea #1: Compute gross statistics (#tokens, #word types, etc.)

Idea #2: Label words. Visualize distributions of labels.

Problem: How to set the labels? POS tagger? NER? Coref?

What's in my text corpus?

Assume that text corpus is generated from some set of **topics** in the world (sports, business, politics in Luxembourg, etc.)

What's in my text corpus?

Assume that text corpus is generated from some set of **topics** in the world (sports, business, politics in Luxembourg, etc.)

$\text{Reagan} \in \{\text{politics}, \text{movies}\}$

$\text{petroleum} \approx \text{gasoline}$

$\text{browser} \notin \text{woodworking}$

What's in my text corpus?

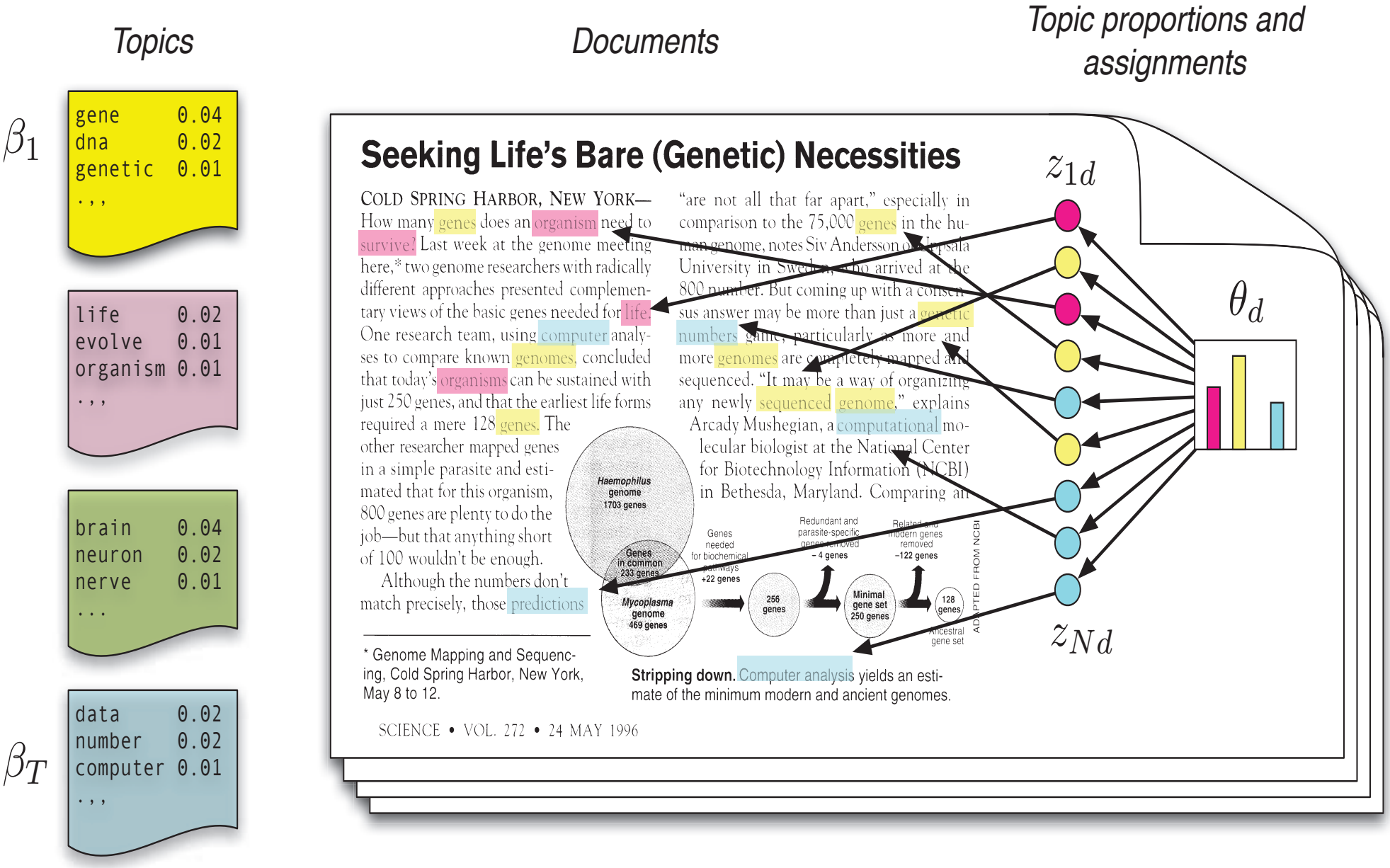
Assume that text corpus is generated from some set of **topics** in the world (sports, business, politics in Luxembourg, etc.)

$\text{Reagan} \in \{\text{politics}, \text{movies}\}$

$\text{petroleum} \approx \text{gasoline}$

$\text{browser} \notin \text{woodworking}$

Infer the topics/labels from data: **Latent Dirichlet Allocation** (LDA)



(Blei, *Introduction to Probabilistic Topic Models*, 2011)

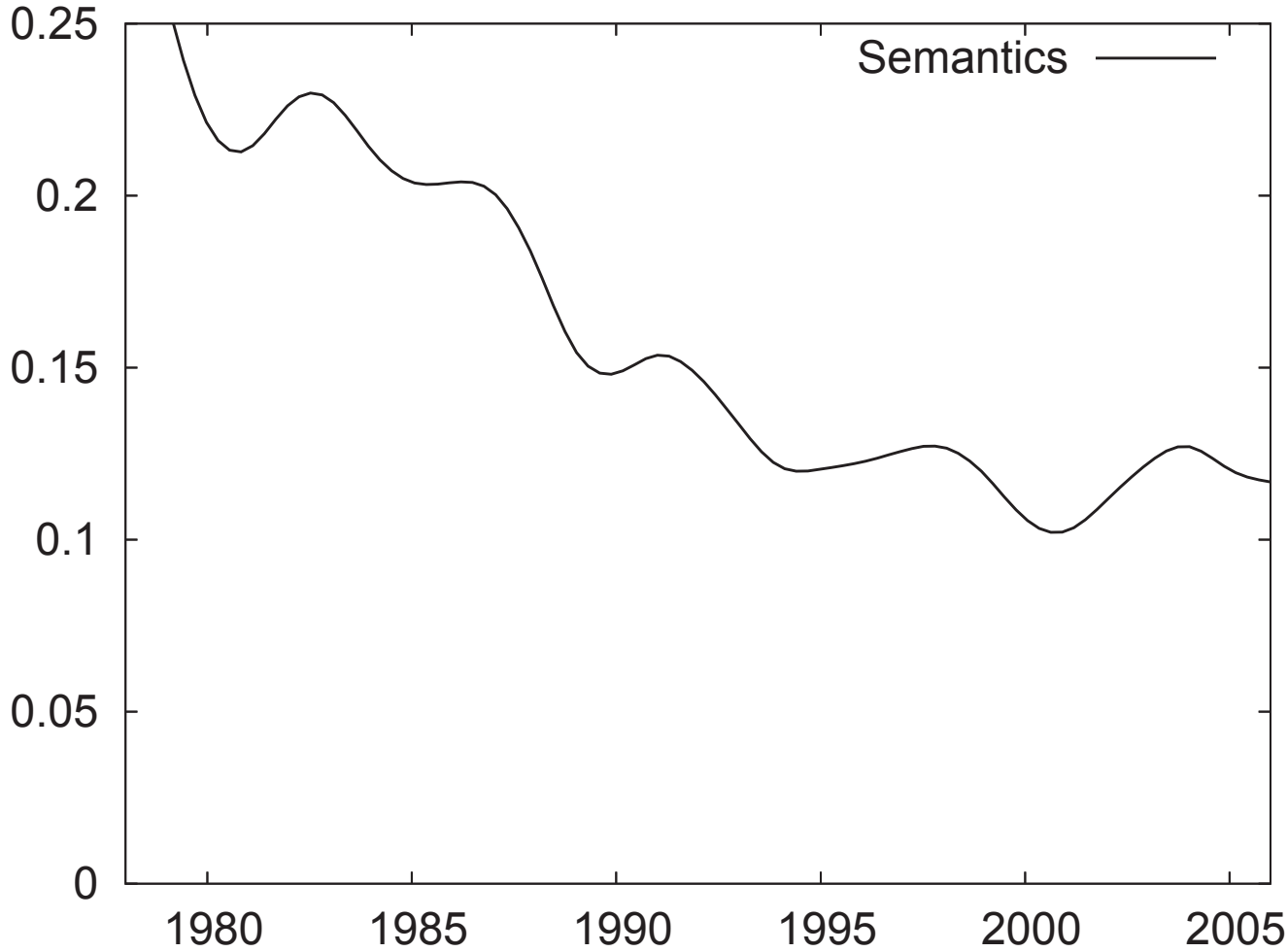
Application #1: History of Ideas

Hall et al. (2008) trained LDA on large corpus of NLP papers

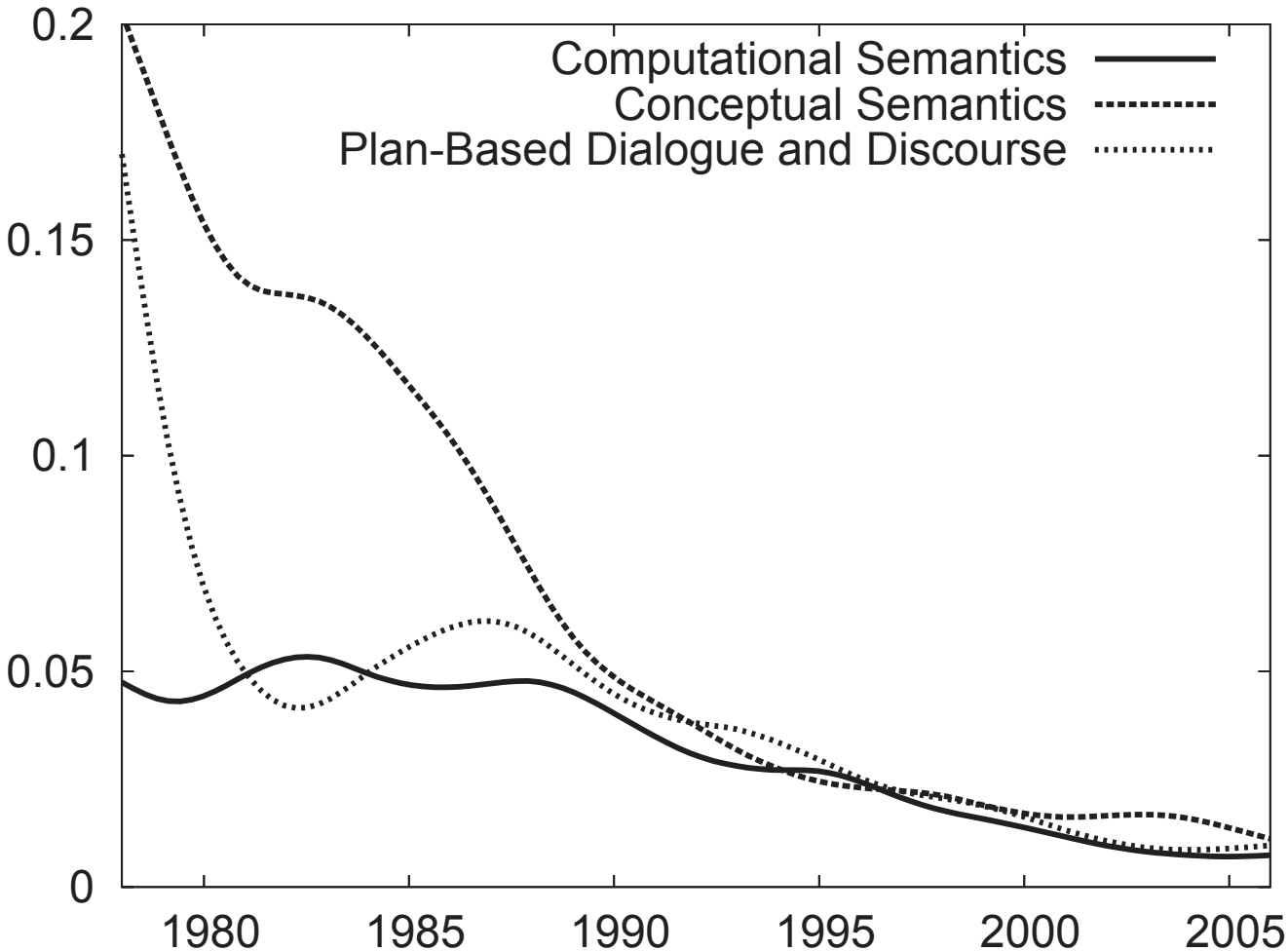
Hand-labeled some learned topics

Plotted mass over time for topics

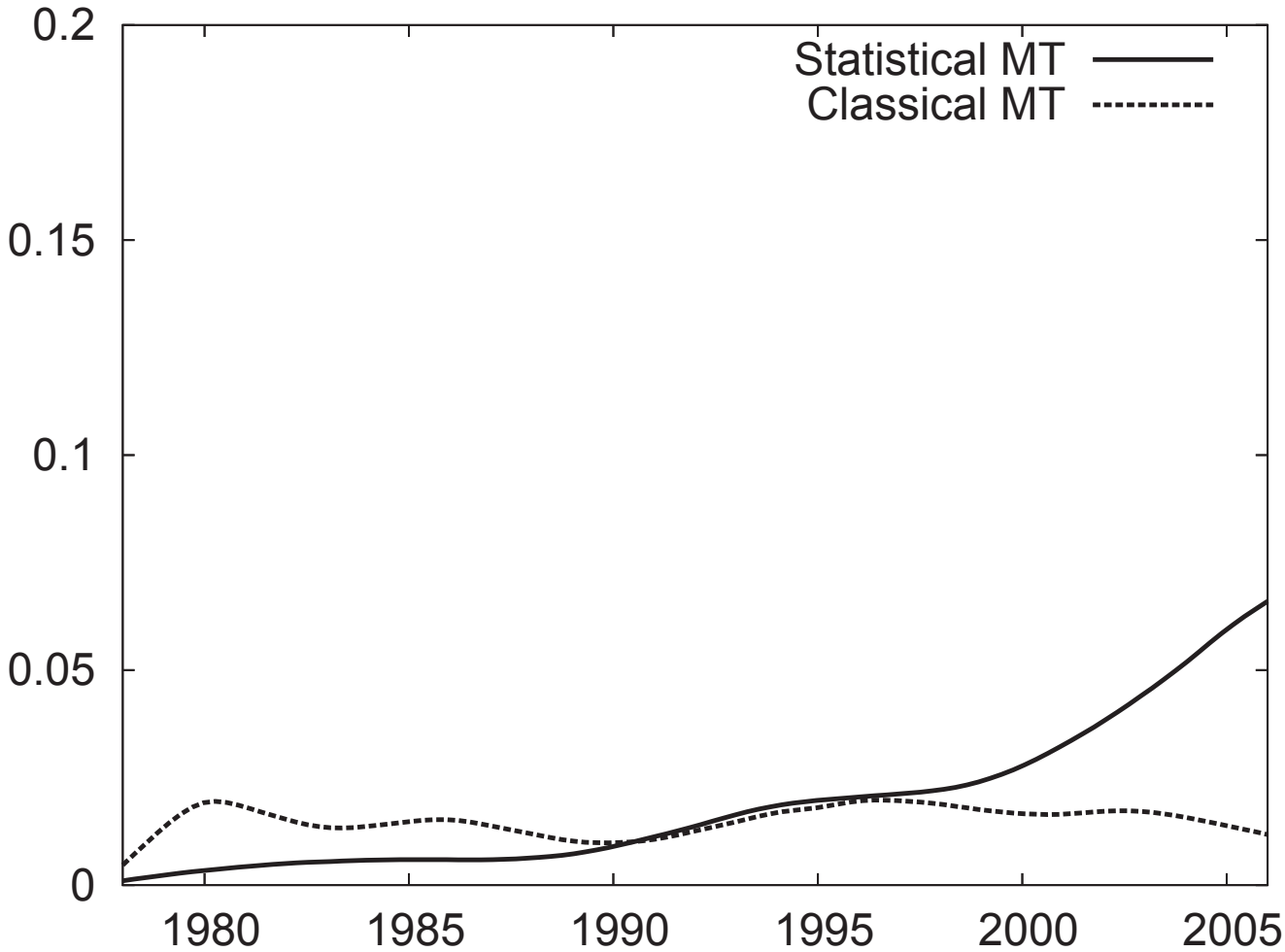
Hall et al. (2008): Semantics :-(



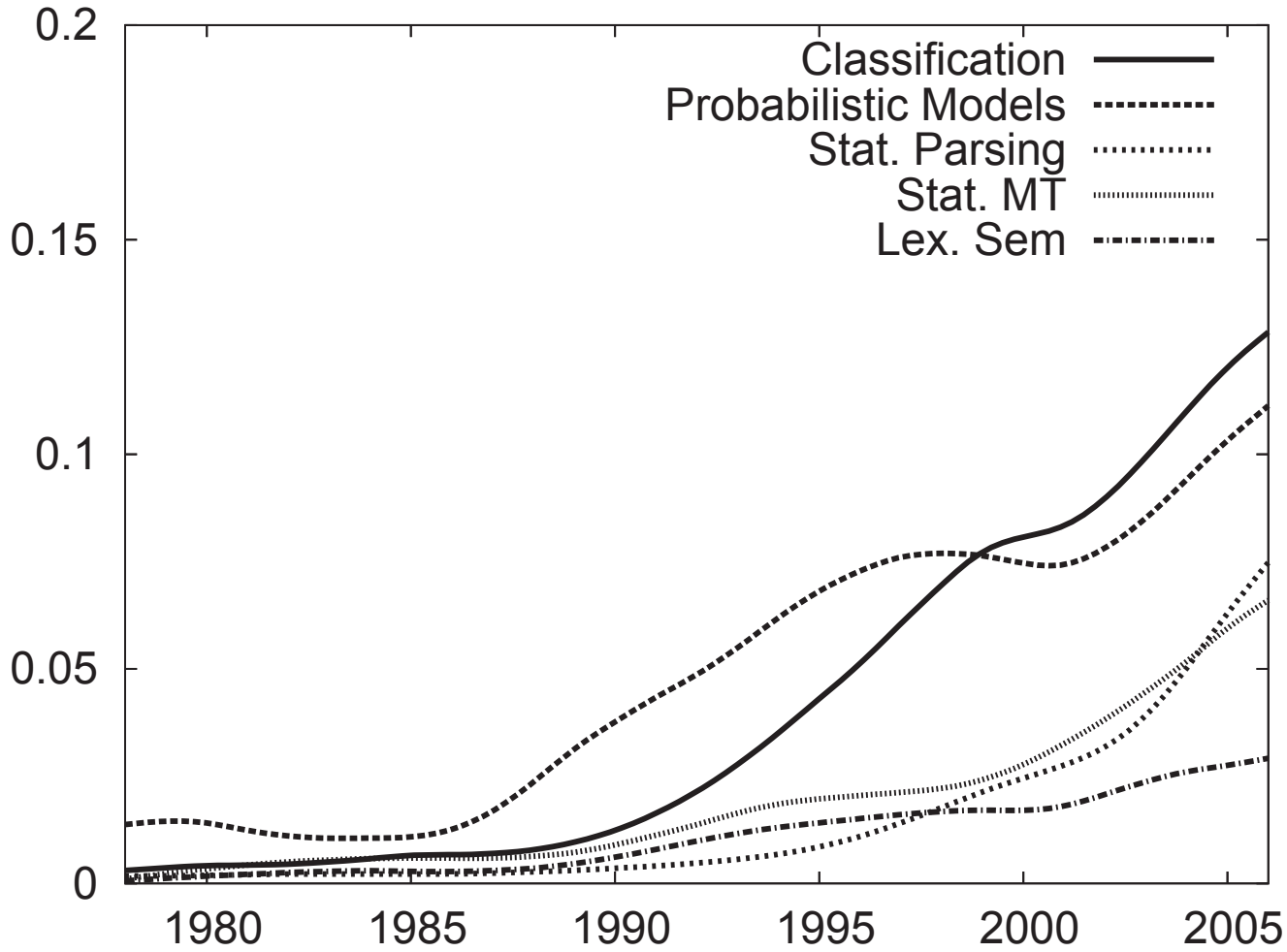
Hall et al. (2008): Semantics :-(:-(



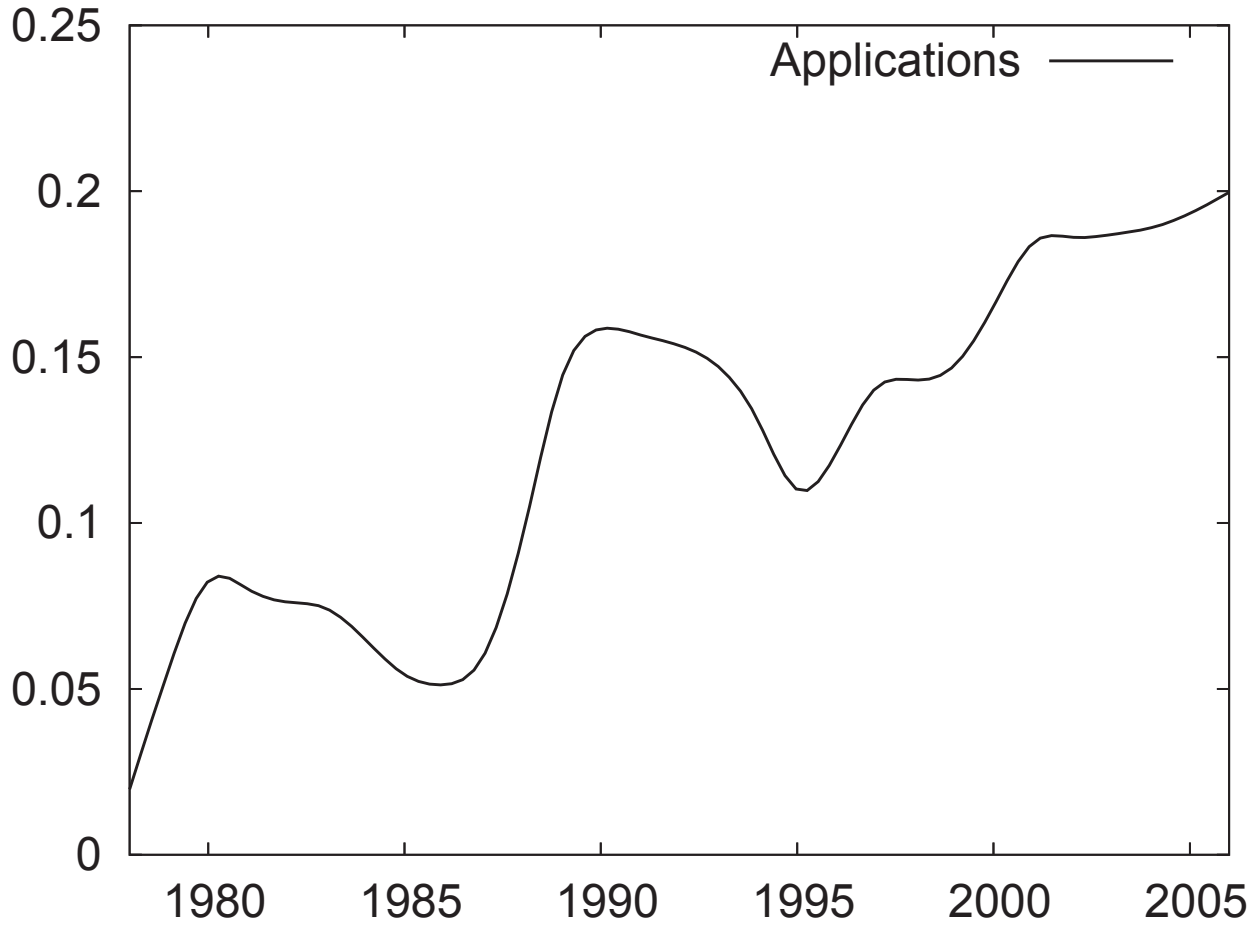
Hall et al. (2008): Machine Translation :-)



Hall et al. (2008): The Statistical Revolution



Hall et al. (2008): NLP is more applied



Application #2: Stanford Dissertation Browser

Stanford Ph.D theses **catalogued by department**

Chuang et al. (2012) used a supervised variant of LDA to visualize topic trends over time

- Cosine similarity between department topic vectors
- Topics abstract over e.g., 'heat' and 'thermodynamics'

Stanford Dissertation Browser

http://www-nlp.stanford.edu/projects/dissertations/

Working up to LDA

Parameter estimation

Generative stories

Notation

Sampling

Parameter Estimation

Method of maximum likelihood (MLE) for discrete models:

$$p(y|x) = \frac{c(y, x)}{c(x)}$$

Parameter Estimation

Method of maximum likelihood (MLE) for discrete models:

$$p(y|x) = \frac{c(y,x)}{c(x)}$$

Maybe we smooth:

$$p(y|x) = \frac{c(y,x) + \alpha}{c(x) + |y|\alpha}$$

Parameter Estimation

Method of maximum likelihood (MLE) for discrete models:

$$p(y|x) = \frac{c(y,x)}{c(x)}$$

Maybe we smooth:

$$p(y|x) = \frac{c(y,x) + \alpha}{c(x) + |y|\alpha}$$

Closed-form estimate is exact

Parameter Estimation: MLE

We've seen this before:

PA#1 IBM Model 1:

$$t(e|f) = \frac{c(e,f)}{\sum_{f'} c(e,f')}$$

Parameter Estimation: MLE

We've seen this before:

PA#1 IBM Model 1:

$$t(e|f) = \frac{c(e, f)}{\sum_{f'} c(e, f')}$$

PA#2 PCFGs:

$$p(A \rightarrow \alpha) = \frac{c(A \rightarrow \alpha)}{\sum_{\alpha'} c(A \rightarrow \alpha')}$$

Parameter Estimation: Maximum a posteriori

Prior over parameters instead of smoothing?

PA#3 Maxent $p(y|x;\theta)$ with Gaussian prior:

$$\theta_{MAP} = \arg\max_{\theta} p(\theta|x)$$

Parameter Estimation: Maximum a posteriori

Prior over parameters instead of smoothing?

PA#3 Maxent $p(y|x;\theta)$ with Gaussian prior:

$$\theta_{MAP} = \arg\max_{\theta} p(\theta|x)$$
$$= \arg\max_{\theta} \frac{p(x|\theta)p(\theta)}{p(x)}$$

Parameter Estimation: Maximum a posteriori

Prior over parameters instead of smoothing?

PA#3 Maxent $p(y|x;\theta)$ with Gaussian prior:

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} p(\theta|x) \\ &= \arg\max_{\theta} \frac{p(x|\theta)p(\theta)}{p(x)} \\ &= \arg\max_{\theta} p(x|\theta)p(\theta)\end{aligned}$$

Parameter Estimation: MAP

θ_{MAP} is still a **point estimate**

Idea: Keep a distribution over the parameters?

Parameter Estimation: Bayesian Model Selection

Don't take the argmax, keep the **posterior**

Apply Bayes rule (like in MAP):

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

$$posterior = \frac{likelihood \cdot prior}{evidence}$$

Parameter Estimation: Bayesian Model Selection

MAP: we discarded the normalizer $p(x)$

Now it's important (and often hard to compute):

$$p(x) = \int_{\theta \in \Theta} p(x|\theta)p(\theta)d\theta$$

When the joint/likelihood is intractable, sample!

Preliminaries: Generative stories

Generative story: Account of how observed data is generated under the model

We've seen this before also...

Generative stories

Generate target sequence e from source sequence f

Generative stories

Generate target sequence e from source sequence f

Choose a length n for the target sequence e

For each position i in e, choose a position j in source sequence f, and generate e_i from f_j:

$$p(e, a|f, n) = \prod_{i=1}^{n} q(a_i|i, n, m)t(e_i|f_{a_i})$$

Generative stories

Generate target sequence e from source sequence f

Choose a length n for the target sequence e

For each position i in e, choose a position j in source sequence f, and generate e_i from f_j:

$$p(e, a|f, n) = \prod_{i=1}^{n} q(a_i|i, n, m)t(e_i|f_{a_i})$$

IBM Model 2: generative word alignments

Generative stories

Generate tree t from grammar G

Generative stories

Generate tree t from grammar G

Choose a start symbol $\Diamond \in V$ with probability 1.0

Recursively re-write $\Diamond$ with rules $A \rightarrow \alpha$ in grammar G

Generative stories

Generate tree t from grammar G

Choose a start symbol $\Diamond \in V$ with probability 1.0

Recursively re-write $\Diamond$ with rules $A \rightarrow \alpha$ in grammar G

PCFG: generative parse trees

Preliminaries: Plate notation

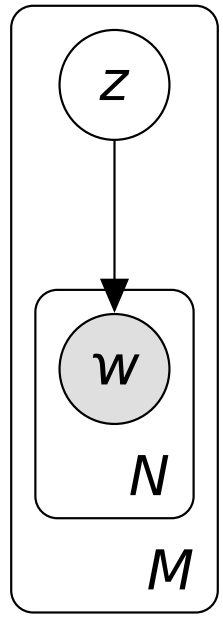
Plate notation: Compact representation of variable networks

Circles: random variables (shaded for observed variables)

Plates: replicate a block of variables

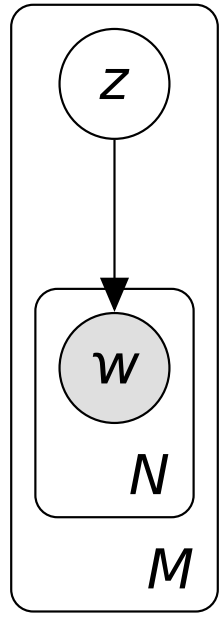
Directed edges: dependencies between variables

Who am I?



M docs each with *N* **independent** words; class *z* per doc

Who am I?



M docs each with *N* **independent** words; class *z* per doc

Naive Bayes

- z_1 = "Spam"
- z_2 = "Not Spam"

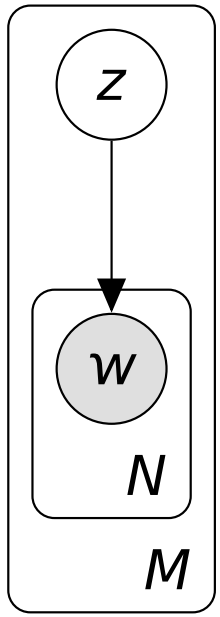
Aside: Draws from distributions

$w \sim Discrete(\theta)$ means w "is drawn from"

w is a **random variate**

How to draw w (without Matlab)?

Idea: if we can generate uniform pseudo-random variates, we can draw from other distributions



Draws from distributions

$$X \sim U(0, 1)$$

Draws from distributions

$$X \sim U(0, 1)$$

```
x = Math.random()
```

Draws from distributions

$$X \sim Bernoulli(p)$$

Draws from distributions

$$X \sim Bernoulli(p)$$

```
x = Math.random() < p ?  true :  false
```

Draws from distributions

$$X \sim Bernoulli(p)$$

```
x = Math.random() < p ?  true :  false
```

Binomial is a generalization of Bernoulli

Draws from distributions

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

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
}
```

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
}
```

Multinomial is a generalization of Discrete

Draws from distributions

$$X \sim Dirichlet(\alpha)$$

Draws from distributions

$$X \sim Dirichlet(\alpha)$$

```
dir = double[k];

for i in 0...k-1
  dir[i] = drawGamma();

normalize(dir);
return dir;
```

Gamma sampling also based on uniform variates (see Knuth)

Bayesian Naive Bayes

Notation

m	document with N_m words per document
$w_{m,n}$	nth word in the mth document (W in total)
z	latent topic (Z in total with support K)
z_m	topic/class of document m

I'll use "topic" and "class" interchangeably

The truth about Naive Bayes

Naive Bayes is not Bayesian

1. "Bayes" part: $p(Z|W) = \frac{p(W|Z)p(Z)}{p(W)}$

The truth about Naive Bayes

Naive Bayes is not Bayesian

1. "Bayes" part: $p(Z|W) = \frac{p(W|Z)p(Z)}{p(W)}$
2. "Naive" part: $p(Z|W) = \frac{p(Z)\prod_i p(W_i|Z)}{p(W)}$

   Words are generated independently

   Not true…come to lecture on Monday!

Bayesian Naive Bayes

Two changes:

Bayesian Naive Bayes

Two changes:

1. Dirichlet prior over document-topic distributions

Bayesian Naive Bayes

Two changes:

1. Dirichlet prior over document-topic distributions
2. Dirichlet prior over topic-word distributions

Bayesian Naive Bayes

Draw $\theta_m \sim Dirichlet(\alpha)$

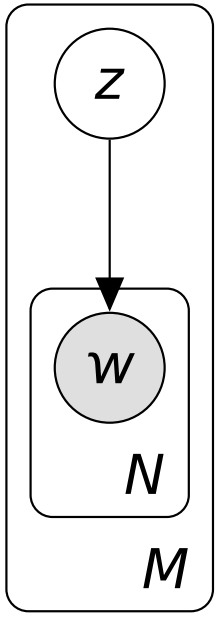
Draw $\phi_{z_m} \sim Dirichlet(\beta)$

For each document m:

Draw class $z_m \sim Discrete(\theta_m)$

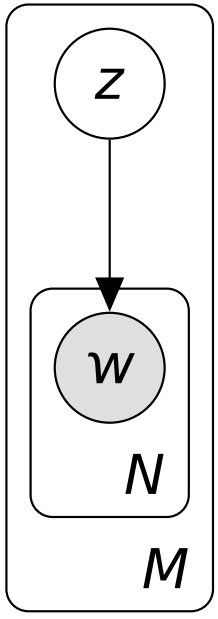
Draw each word $w_{m,n} \sim Discrete(\phi_{z_m})$

Bayesian Naive Bayes

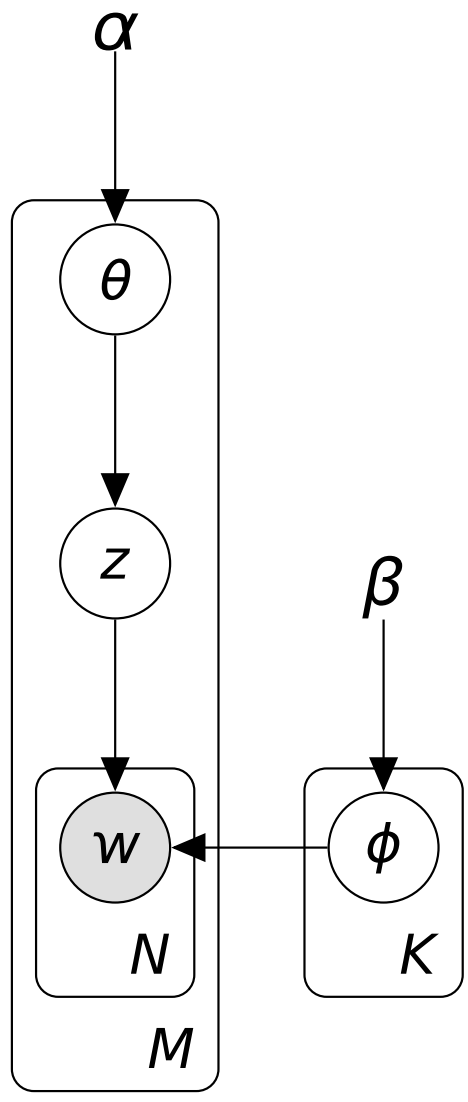


Old school

Bayesian Naive Bayes



Old school



New school

Bayesian NB: Learning

Step 1: Write out the joint density

Plate model shows the factorization

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

Bayesian NB: Learning

Step 2: Derive a conditional distribution of one z_m in terms of all other classes Z_{-m}

Iterate over all documents and words setting z_m

Theory: converges to joint

This is called a **Gibbs sampler**

Z is discrete...we know how to draw z_m

Aside: Gibbs sampling

Say you have variables a, b, c

Here is a Gibbs sampler:

1. Draw a conditioned on b, c
2. Draw b conditioned on a, c
3. Draw c conditioned on a, b
4. Repeat until convergence

Bayesian NB: Learning

Step 2: Derive a conditional distribution of one z_m in terms of all other variables Z_{-m}

$$p(z_m|Z_{-m}, W, \alpha, \beta) = \frac{p(z_m, Z_{-m}, W; \alpha, \beta)}{p(Z_{-m}, W; \alpha, \beta)}$$

Bayesian NB: Learning

Step 2: Derive a conditional distribution of one z_m in terms of all other variables Z_{-m}

$$p(z_m|Z_{-m}, W, \alpha, \beta) = \frac{p(z_m, Z_{-m}, W; \alpha, \beta)}{p(Z_{-m}, W; \alpha, \beta)}$$
$$= \frac{p(Z, W; \alpha, \beta)}{p(Z_{-m}, W; \alpha, \beta)}$$

Bayesian NB: Learning

Discrete conditional distribution:

$$p(z_m = k | Z_{-m}, W, \alpha, \beta)$$

Bayesian NB: Learning

Discrete conditional distribution:

$$p(z_m = k | Z_{-m}, W, \alpha, \beta)$$

Run our Gibbs sampler for awhile

Bayesian NB: Learning

Discrete conditional distribution:

$$p(z_m = k | Z_{-m}, W, \alpha, \beta)$$

Run our Gibbs sampler for awhile

Converge to document-topic and word-topic distributions

Latent Dirichlet Allocation

LDA

Just one change:

Associate a topic with each word

LDA

Just one change:

Associate a topic with each word

Recall our original motivation: documents comprised of topics

LDA: Generative Story

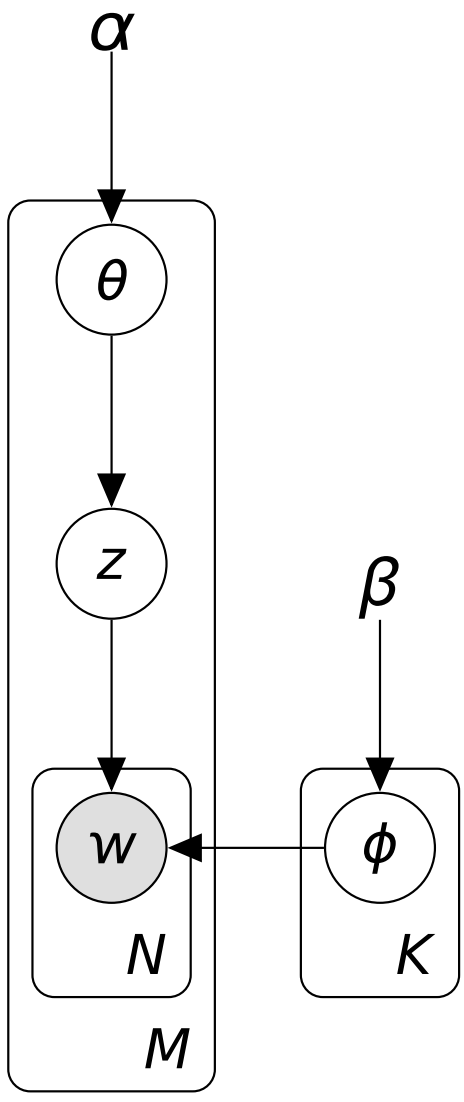
Draw θ_m from $Dirichlet(\alpha)$

Draw ϕ_k from $Dirichlet(\beta)$

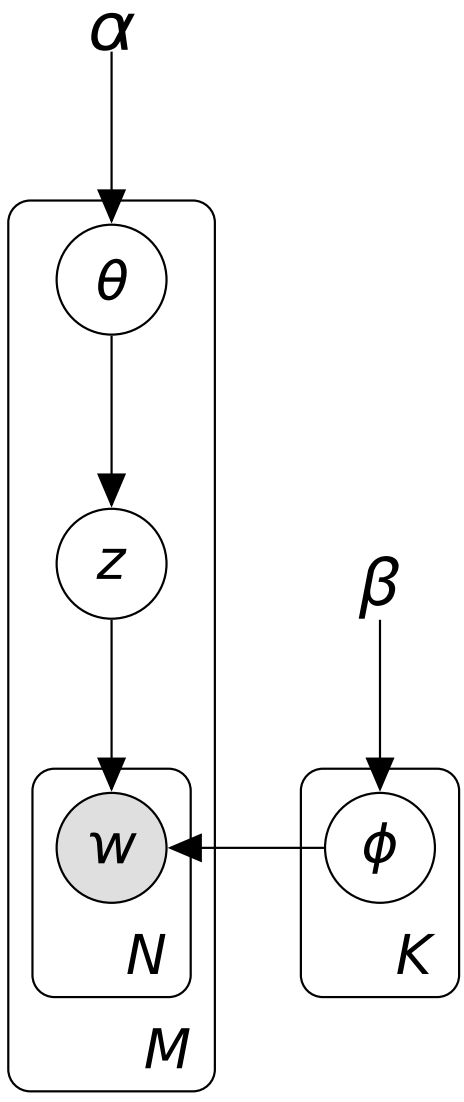
- for each word $w_{m,n}$ in document m:

  Draw $z_{m,n} \sim Discrete(\theta_m)$

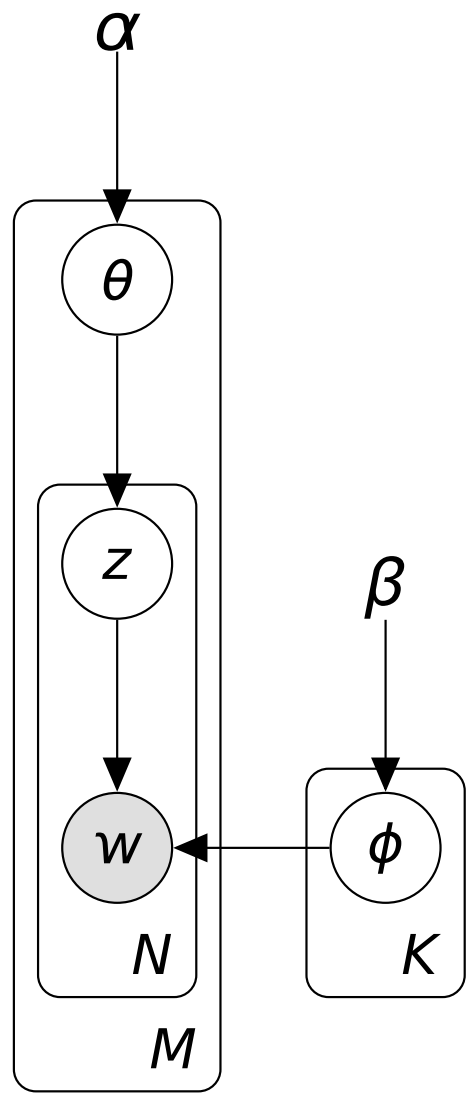
  Draw $w_{m,n} \sim Discrete(\phi_{z_{m,n}})$



New school NB



New school NB

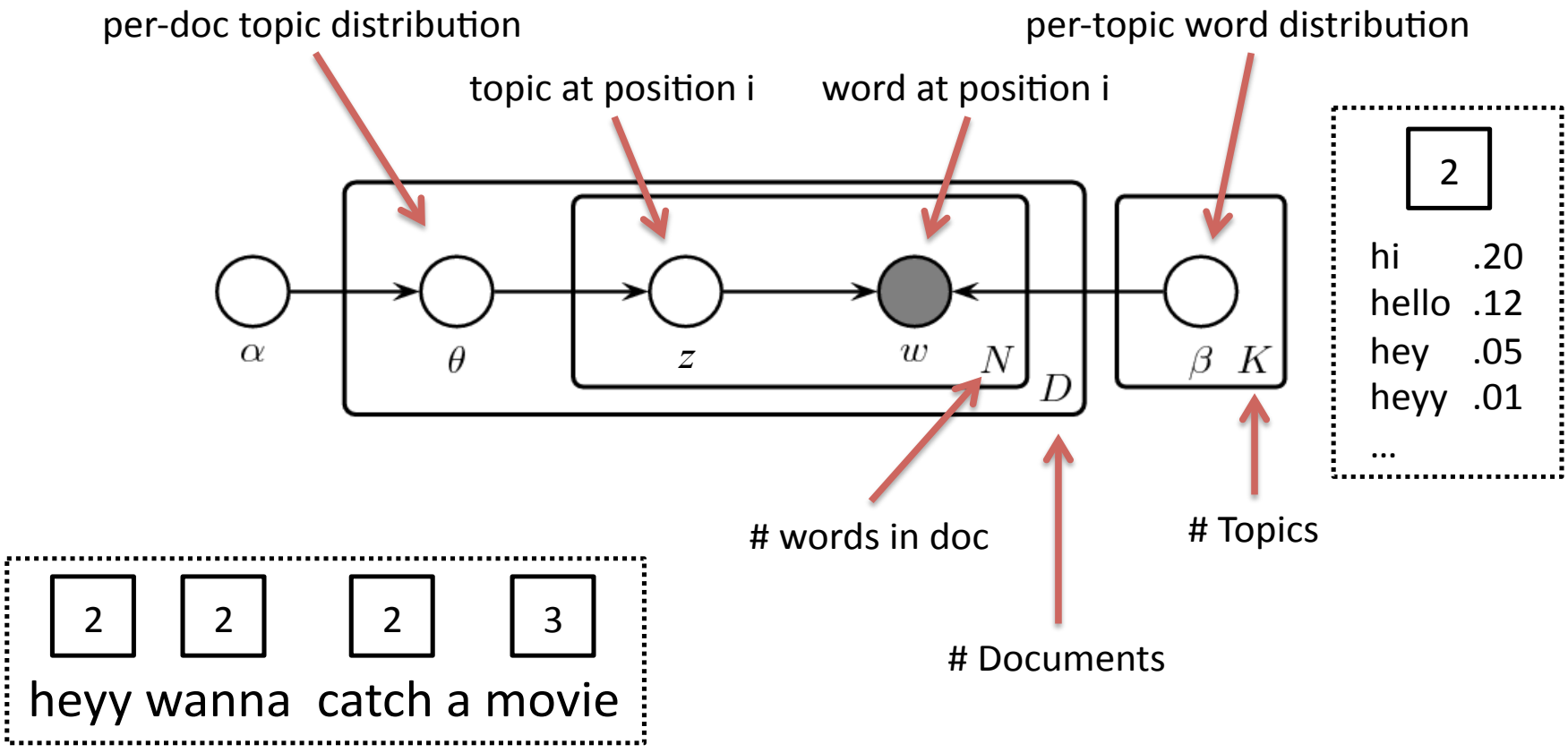


LDA

Content modeling with LDA

Blei, Ng, Jordan 2003

Latent Dirichlet Allocation



LDA: Learning

Step 1: Write joint density

Bayesian NB:

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

LDA: Learning

Step 1: Write joint density

Bayesian NB:

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

LDA:

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

LDA: Learning

Step 1: Write joint density

Bayesian NB:

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

LDA:

$$p(W, Z, \theta, \phi; \alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(Z|\theta)p(W|Z, \phi)$$

Only difference: Dimension of Z (latent structure)

LDA: Learning

Step 2: Derive full conditionals

$$p(z_{m,n} = k | Z_{-m,n}, W, \alpha, \beta) = \frac{e_{k,w}^{-m,n}}{\sum_w (e_{k,w}^{-m,n} + \beta_w)} \times \frac{d_{m,k}^{-m,n} + \alpha_k}{\sum_k (d_{m,k}^{-m,n} + \alpha_k)}$$

LDA: Full conditional

Topic-word counts term:

$$\frac{e_{k,w}^{-m,n}}{\sum_w (e_{k,w}^{-m,n} + \beta_w)}$$

LDA: Full conditional

Document-topic counts term:

$$\frac{d_{m,k}^{-m,n} + \alpha_k}{\sum_k (d_{m,k}^{-m,n} + \alpha_k)}$$

LDA: Learning

Step 2: Derive full conditionals

$$p(z_{m,n} = k | Z_{-m,n}, W, \alpha, \beta)$$

This is discrete. We know how to sample

"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.

More interesting application