Web-Mining Agents

Topic Analysis: pLSI and LDA

Tanya Braun

Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Acknowledgments

Pilfered from:

Ramesh M. Nallapati

Machine Learning applied to Natural Language Processing

Thomas J. Watson Research Center, Yorktown Heights, NY USA

from his presentation on

Generative Topic Models for Community Analysis

&

David M. Blei

MLSS 2012

from his presentation on Probabilistic Topic Models

&

Sina Miran

CS 290D Paper Presentation on

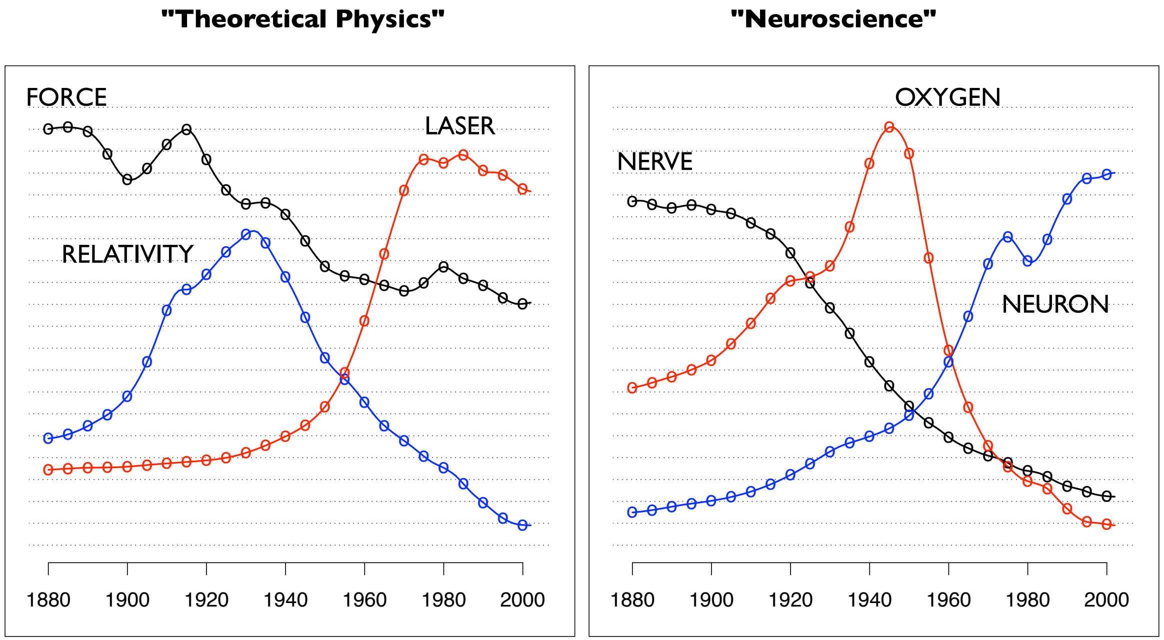
Probabilistic Latent Semantic Indexing (PLSI)

Recap

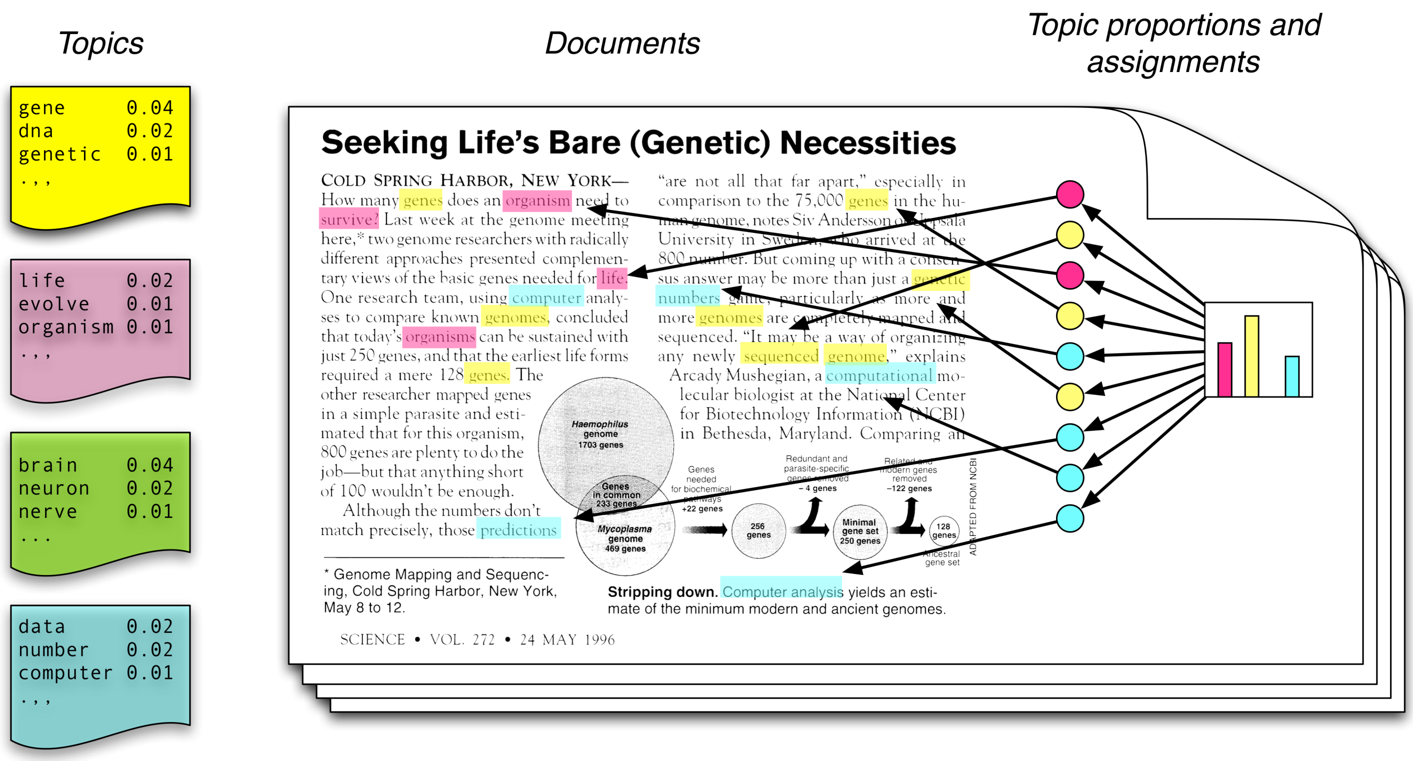
- Agents
  - Task/goal: Information retrieval
  - Environment: Document repository
  - Means:
    - Vector space (bag-of-words)
      - Dimension reduction (LSI)
    - Probability based retrieval (binary)
      - Language models
- Today: Topic models as special language models
  - Probabilistic LSI (pLSI)
  - Latent Dirichlet Allocation (LDA)
- Soon: What agents can take with them
  - What agents can leave at the repository (win-win)

Objectives

- Topic Models: statistical methods that analyze the words of texts in order to:
  - Discover the themes that run through them (topics)
  - How those themes are connected to each other
  - How they change over time

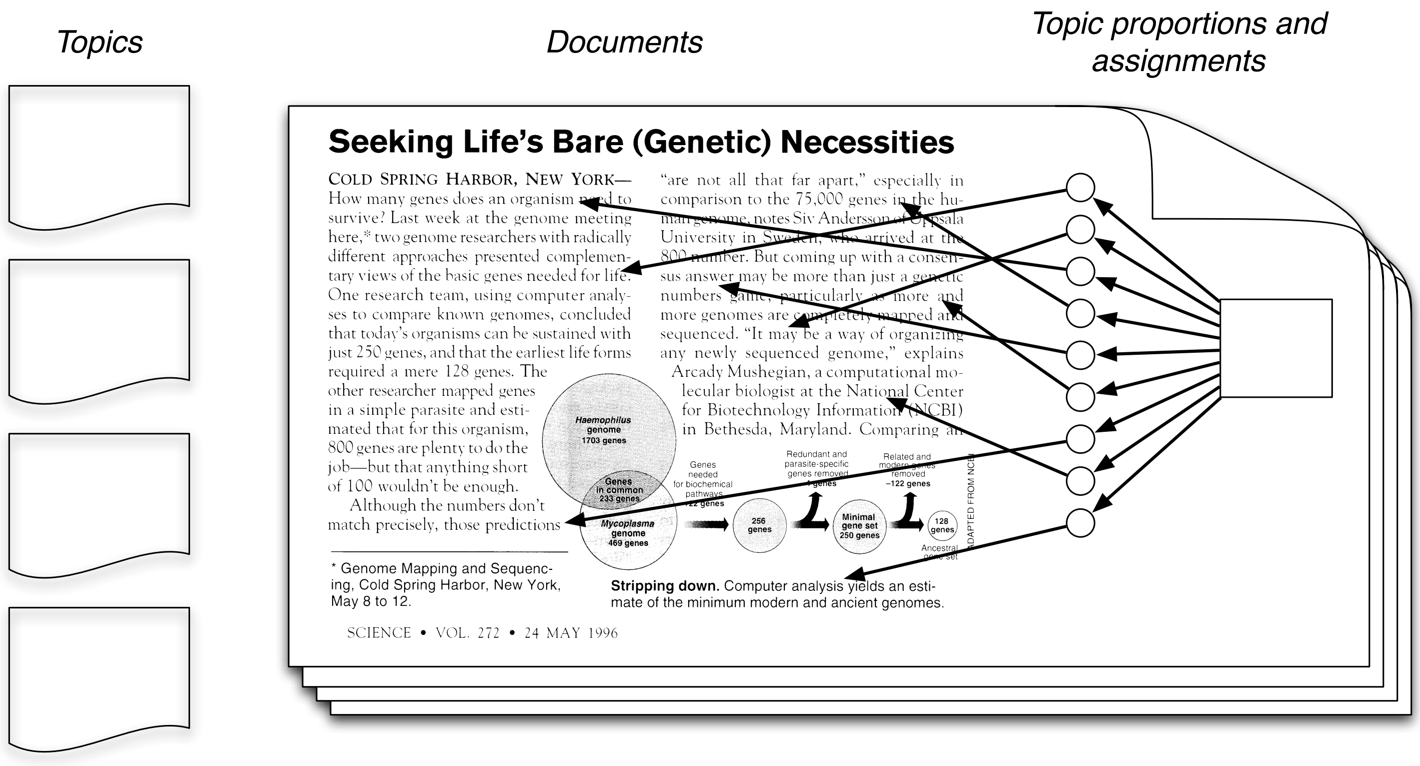


Topic Modeling Scenario



- 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

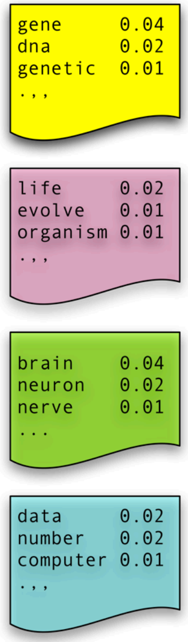
Topic Modeling Scenario

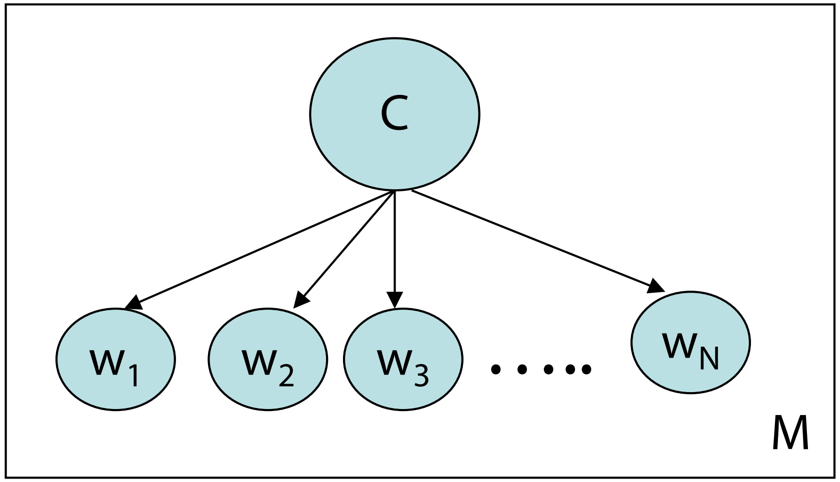


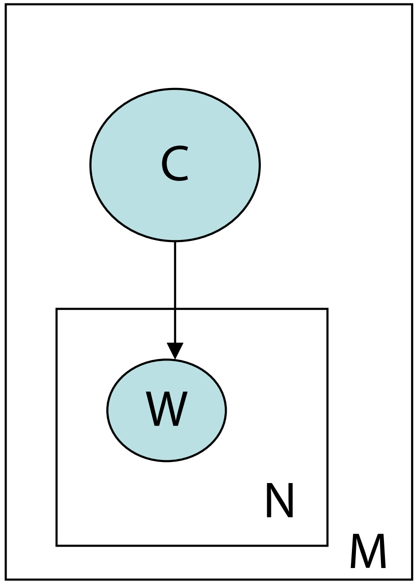
- In reality, we only observe the documents
- The other structures are hidden variables
- Topic modeling algorithms infer these variables from data

Plate Notation

- Naïve Bayes Model: Compact representation
  - C = topic/class (name for a word distribution)
  - N = number of words in considered document
  - W_i one specific word in corpus
  - M documents, W now words in document





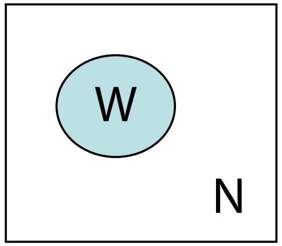


  - Idea: Generate doc from P(W, C)

Generative vs. Descriptive Models

- Generative models: Learn P(x, y)
  - Tasks:
    - Transform P(x,y) into P(y | x) for classification
    - Use the model to predict (infer) new data
  - Advantages
    - Assumptions and model are explicit
    - Use well-known algorithms (e.g., MCMC, EM)
- Descriptive models: Learn P(y | x)
  - Task: Classify data
  - Advantages
    - Fewer parameters to learn
    - Better performance for classification

Earlier Topic Models: Topics Known



- Unigram
  - No context information

$$P(w_1, \dots, w_N) = \prod_i P(w_i)$$

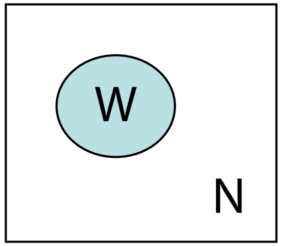
```
fifth, an, of, futures, the, an, incorporated, a,
a, the, inflation, most, dollars, quarter, in, is,
mass

thrift, did, eighty, said, hard, 'm, july, bullish

that, or, limited, the
```

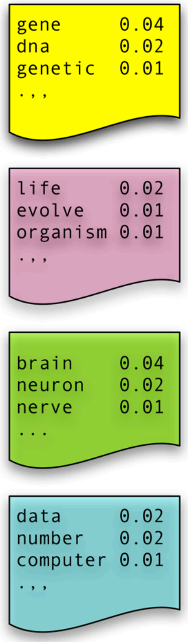
Automatically generated sentences from a unigram model

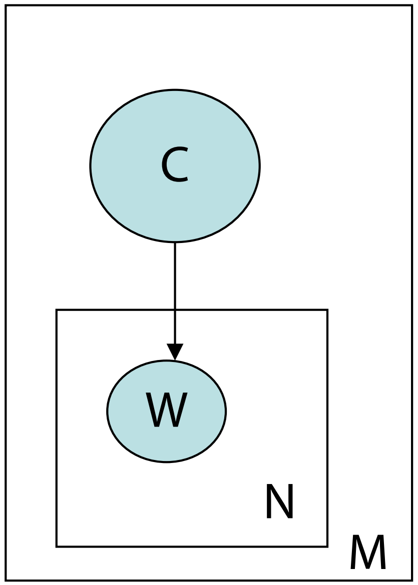
Earlier Topic Models: Topics Known



- Unigram
  - No context information

$$P(w_1, \ldots, w_N) = \prod_i P(w_i)$$



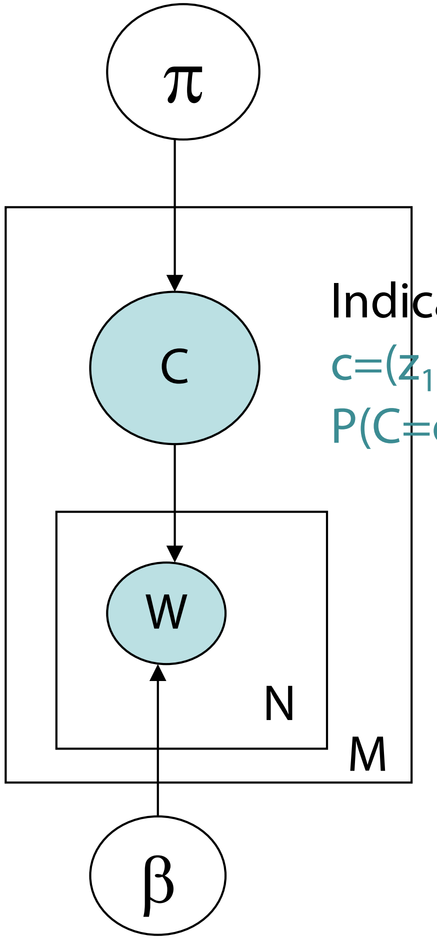


- Mixture of Unigrams
  - One topic per document

$$\prod_{d=1}^{M} P(w_1, \ldots, w_N, c_d) = \prod_{d=1}^{M} P(c_d) \prod_i P(w_i | c_d)$$

  - How to specify $P(c_d)$?

Mixture of Unigrams: Known Topics



Indication vector

$c=(z_1,\ldots z_k)^\top \in \{0, 1\}^K$

$P(C=c)= \prod_{k=1}^{K} \pi_k{}^{zk}$

- Multinomial Naïve Bayes
  - For each document $d = 1, \ldots, M$
    - Generate $c_d \sim \text{Mult}(\,.\mid \pi)$
    - For each position $i = 1, \ldots, N_d$
      - Generate $w_i \sim \text{Mult}(\,.\mid \beta, c_d)$

$$\prod_{d=1}^{M} P\left(w_1, \ldots, w_{N_d}, c_d \mid \beta, \pi\right)$$

$$= \prod_{d=1}^{M} P(c_d|\pi) \prod_{i=1}^{N_d} P(w_i|\beta, c_d) = \prod_{d=1}^{M} \pi_{c_d} \prod_{i=1}^{N_d} \beta_{c_d,w_i}$$

$$\pi_{c_d} := P(c_d|\pi)$$

$$\beta_{c_d,w_i} := P(w_i|\beta, c_d)$$

multinomial

Multinomial Distribution

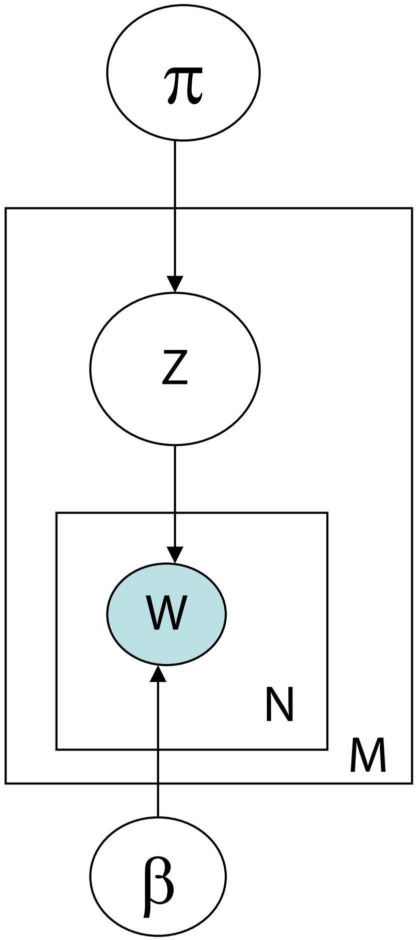
- Generalization of binomial distribution
  - K possible outcomes instead of 2
  - Probability mass function
    - n = number of trials
    - x_j = count for how often class j occurring $\sum_{i=1}^{k} x_i = n$
    - p_j = probability of class j occurring

$$f(x_1, \dots, x_K; p_1, \dots, p_K) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{K} p_i^{x_i}$$

- Here, the input to $\Gamma(\cdot)$ is a positive integer, so

$$\Gamma(n) = (n-1)!$$

Mixture of Unigrams: Unknown Topics



- Topics/classes are hidden
  - Joint probability of words and classes

$$\prod_{d=1}^{M} P\left(w_1, \dots, w_{N_d}, c_d \mid \beta, \pi\right) = \prod_{d=1}^{M} \pi_{c_d} \prod_{i=1}^{N_d} \beta_{c_d, w_i}$$

  - Sum over topics (K = number of topics)

$$\prod_{d=1}^{M} P\left(w_1, \dots, w_{N_d} \mid \beta, \pi\right) = \prod_{d=1}^{M} \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

$$\pi_{z_k} \coloneqq P(z_k \mid \pi)$$
$$\beta_{z_k, w_i} \coloneqq P(w_i \mid \beta, z_k)$$

Mixture of Unigrams: Learning

- Learn parameters π and β

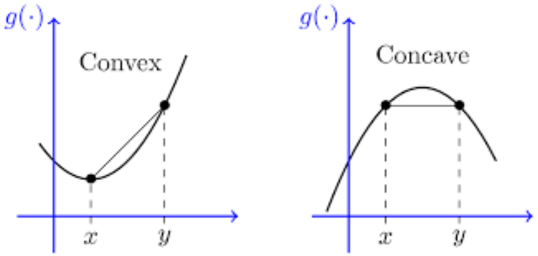
$$\prod_{d=1}^{M} P(w_1, \dots, w_{N_d} | \beta, \pi) \quad = \prod_{d=1}^{M} \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

- Use likelihood

$$\sum_{d=1}^{M} \log P(w_1, \dots, w_{N_d} | \beta, \pi) = \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$

- Solve

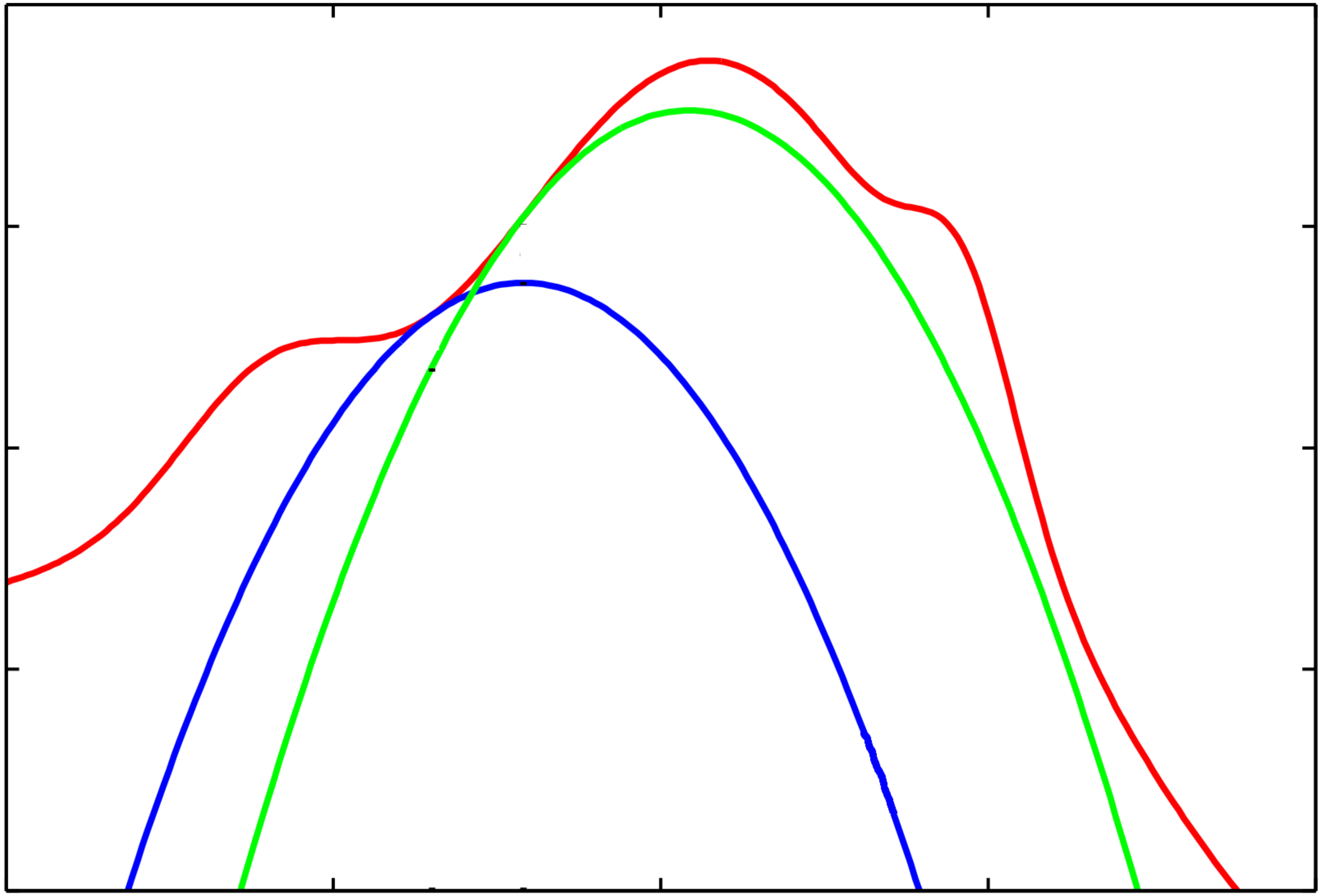
$$argmax_{\beta\pi} \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k, w_i}$$



  - Not a concave/convex function
  - Note: a non-concave/non-convex function is not necessarily convex/concave
  - Possibly no unique max, many saddle or turning points

    No easy way to find roots of derivative

Trick: Optimize Lower Bound



Mixture of Unigrams: Learning

- The problem $$argmax_{\beta\pi} \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k,w_i}$$

- Optimize w.r.t. each document
- Derive lower bound

$$\log \sum_i \gamma_i x_i \geq \sum_i \gamma_i \log x_i \text{ where } \gamma_i \geq 0 \wedge \sum_i \gamma_i = 1$$ Jensen's inequality

$$\log \sum_i x_i = \log \sum_i \gamma_i \frac{x_i}{\gamma_i} \geq \sum_i (\gamma_i \log x_i - \gamma_i \log \gamma_i)$$

H(γ)

$$argmax_{\beta\pi} \sum_{d=1}^{M} \log \sum_{k=1}^{K} \pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k,w_i}$$ The problem again

$$\log \sum_{k=1}^{K} \pi_k \prod_{i=1}^{N_d} \beta_{k,w_i} \geq \sum_{k=1}^{K} \left(\gamma_k \log(\pi_k \prod_{i=1}^{N_d} \beta_{k,w_i}) \right) + H(\gamma)$$

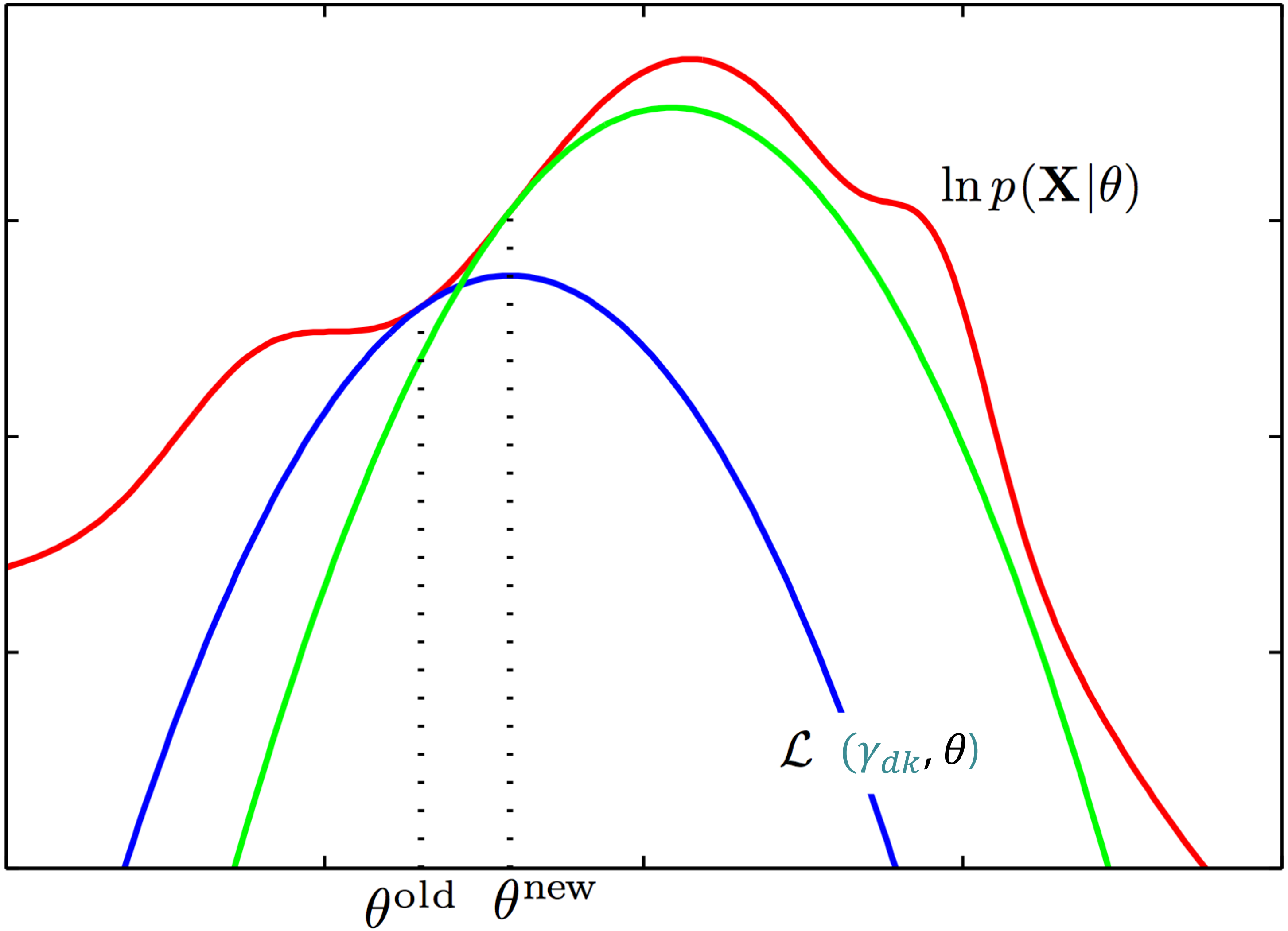
Mixture of Unigrams: Learning

- For each document d

$$\log \sum_{k=1}^{K} \pi_k \prod_{i=1}^{N_d} \beta_{k,w_i} \geq \sum_{k=1}^{K} \left(\gamma_{dk} \log(\pi_k \prod_{i=1}^{N_d} \beta_{k,w_i}) \right) + H(\gamma)$$

- Chicken-and-egg problem:
  - If we knew γ_{dk} we could find π_k and β_{k,w_i} with ML
  - If we knew π_k and β_{k,w_i} we could find γ_{dk} with ML
- Finally we need π_{z_k} and β_{z_k,w_i}
- Solution: Expectation Maximization
  - Iterative algorithm to find local optimum
  - Guaranteed to maximize a lower bound on the log-likelihood of the observed data

Graphical Idea of the EM Algorithm



$\theta = (\pi_k, \beta_{k,w_i})$

Mixture of Unigrams: Learning

- For each document d

$$\log \sum_{k=1}^{K} \pi_k \prod_{i=1}^{N_d} \beta_{k,w_i} \geq \sum_{k=1}^{K} \left(\gamma_{dk} \log(\pi_k \prod_{i=1}^{N_d} \beta_{k,w_i}) \right) + H(\gamma)$$

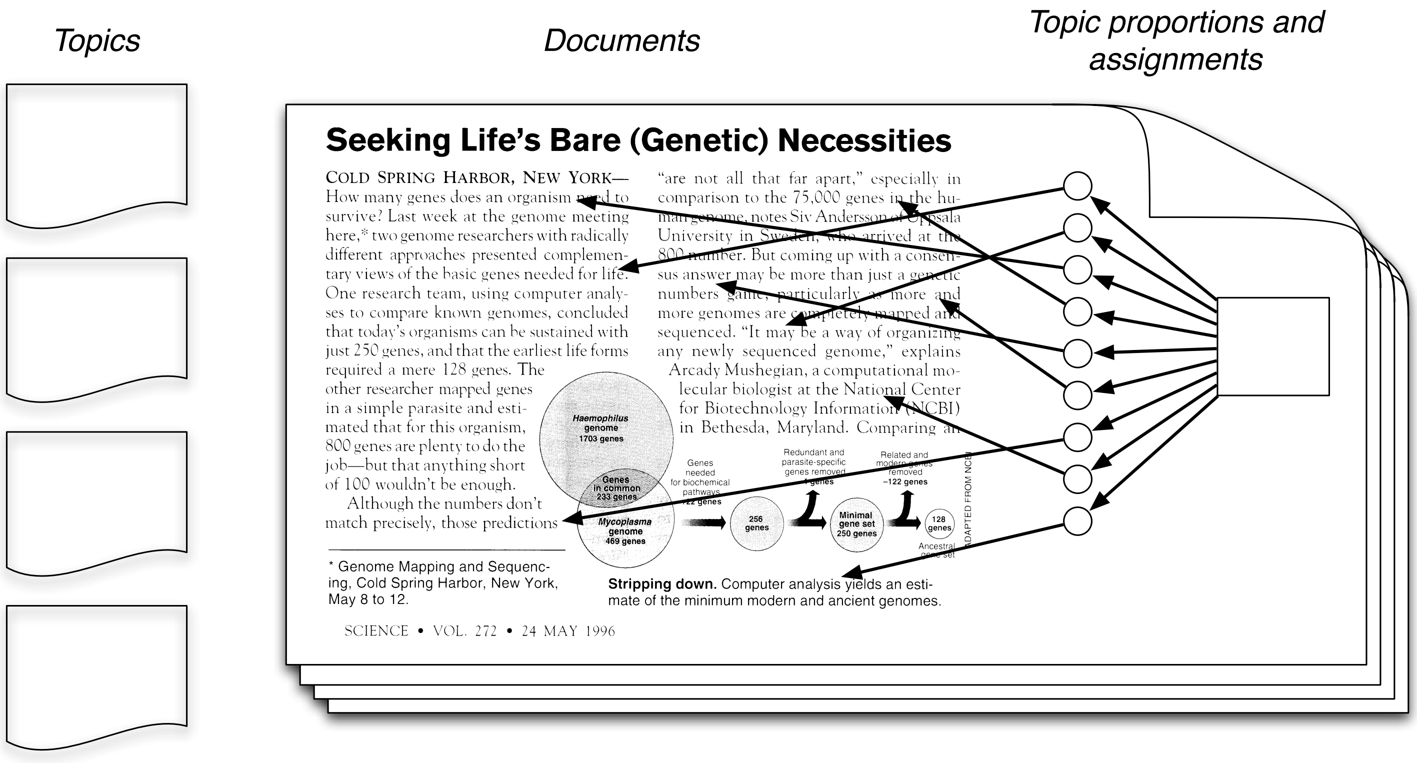
- EM solution
  - E step

$$\gamma_{dk}^{(t+1)} = \frac{\pi_k^{(t)} + \sum_{i=1}^{N_d} \beta_{k,w_i}^{(t)}}{\sum_{j=1}^{K} \pi_j^{(t)} \prod_{i=1}^{N_d} \beta_{k,w_i}^{(t)}}$$

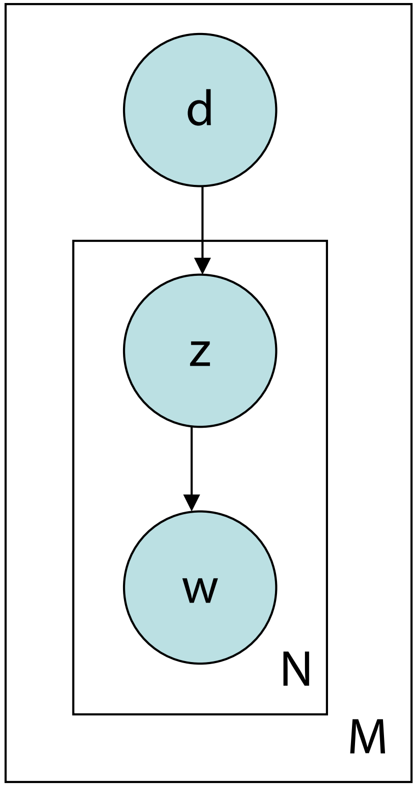
  - M step

$$\pi_k^{(t+1)} = \frac{\sum_{d=1}^{M} \gamma_{dk}^{(t)}}{M} \qquad \beta_{k,w_i}^{(t+1)} = \frac{\sum_{d=1}^{M} \gamma_{dk}^{(t)} n(d, w_i)}{\sum_{d=1}^{M} \gamma_{dk}^{(t)} \sum_{j=1}^{N_d} n(d, w_j)}$$

Back to Topic Modeling Scenario



Probabilistic LSI



- Select a document d with probability P(d)
- For each word of d in the training set
  - Choose a topic z with probability P(z | d)
  - Generate a word with probability P(w | z)

$$P(d, w_i) = P(d) \sum_{k=1}^{K} P(w_i | z_k) P(z_k | d)$$

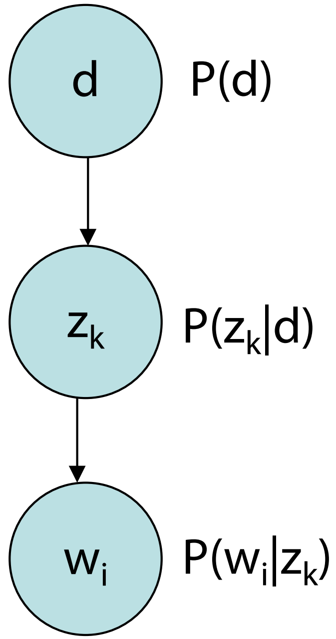
- Documents can have multiple topics

Thomas Hofmann, Probabilistic Latent Semantic Indexing, Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99), **1999**

pLSI

- Joint probability for all documents, words

$$\prod_{d=1}^{M}\prod_{i=1}^{N_d} P(d,w_i)^{n(d,w_i)}$$

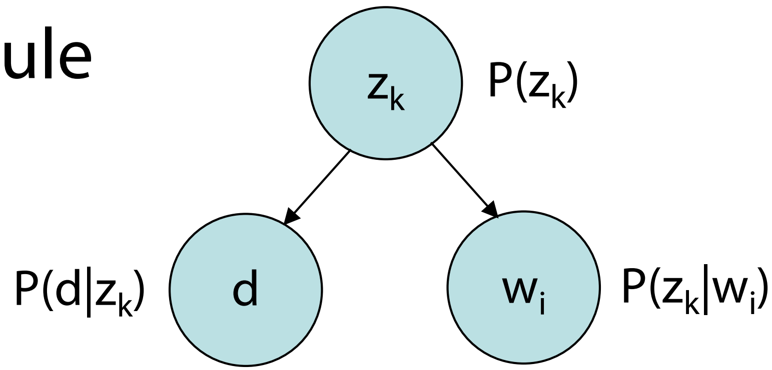


- Distribution for document d, word w_i

$$P(d,w_i) = P(d)\sum_{k=1}^{K} P(w_i|z_k)P(z_k|d)$$

- Reformulate $P(z_k|d)$ with Bayes' Rule

$$P(d,w_i) = \sum_{k=1}^{K} P(d|z_k)P(z_k)P(w_i|z_k)$$



pLSI: Learning Using EM

- Model

$$\prod_{d=1}^{M}\prod_{i=1}^{N_d} P(d,w_i)^{n(d,w_i)} \qquad P(d,w_i)=\sum_{k=1}^{K} P(d|z_k)P(z_k)P(w_i|z_k)$$

- Likelihood

$$L=\sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)\log P(d,w_i)=\sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)\log\sum_{k=1}^{K} P(d|z_k)P(z_k)P(w_i|z_k)$$

- Parameters to learn (M step)

$$P(d|z_k) \qquad P(z_k) \qquad P(w_i|z_k)$$

- (E step)

$$P(z_k|d,w_i)$$

pLSI: Learning Using EM

- EM solution
  - E step

$$P(z_k|d,w_i) = \frac{P(z_k)P(d|z_k)P(w_i|z_k)}{\sum_{m=1}^{K} P(z_m)P(d|z_m)P(w_i|z_m)}$$

  - M step

$$P(w_i|z_k) \quad = \frac{\sum_{d=1}^{M} n(d,w_i)P(z_k|d,w_i)}{\sum_{d=1}^{M}\sum_{j=1}^{N_d} n\left(d,w_j\right)P\left(z_k\middle|d,w_j\right)}$$

$$P(d|z_k) \quad = \frac{\sum_{i=1}^{N_d} n(d,w_i)P(z_k|d,w_i)}{\sum_{c=1}^{M}\sum_{i=1}^{N_c} n(c,w_i)P(z_k|c,w_i)}$$

$$P(z_k) \quad = \frac{1}{R}\sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)P(z_k|d,w_i),\ R = \sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)$$

pLSI: Overview

- More realistic than mixture model
  - Documents can discuss multiple topics!
- Problems
  - Very many parameters
  - Danger of overfitting

pLSI Testrun

- PLSI topics (TDT-1 corpus)
  - Approx. 7 million words, 15863 documents, K = 128

The two most probable topics that generate the term “flight” (left) and “love” (right).

List of most probable words per topic, with decreasing probability going down the list.

“plane”	“space shuttle”	“family”	“Hollywood”
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

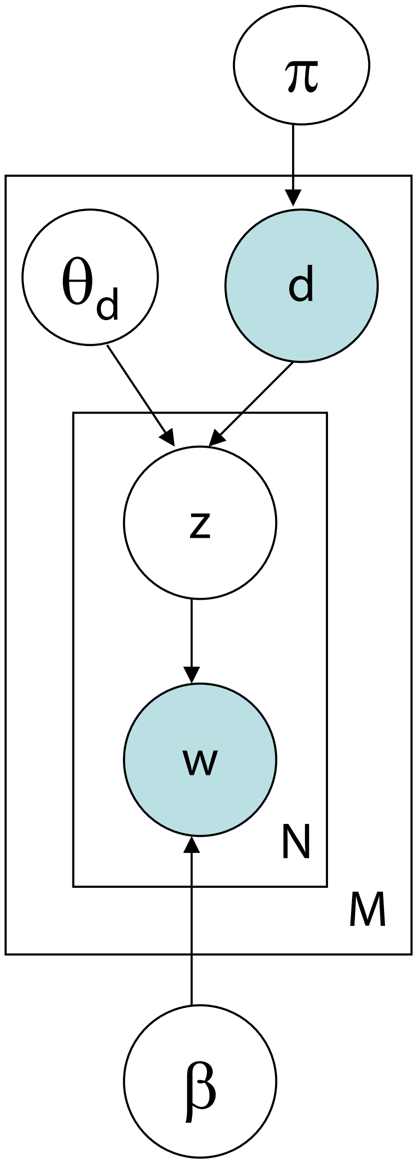
Relation with LSI

$$P = U_k \Sigma_k V_k^T \qquad P(d, w_i) = \sum_{k=1}^{K} P(d|z_k)P(z_k)P(w_i|z_k)$$

$$U_k = \left(P(d|z_k)\right)_{d,k} \qquad \Sigma_k = \mathrm{diag}\left(P(z_k)\right)_k \qquad V_k = \left(P(w_i|z_k)\right)_{i,k}$$

- Difference:
  - LSI: minimize Frobenius (L-2) norm
  - pLSI: log-likelihood of training data

pLSI with Multinomials



- Multinomial Naïve Bayes
  - Select document $d \sim \text{Mult}(\cdot \mid \pi)$
    - For each position $i = 1, \dots, N_d$
      - Generate $z_i \sim \text{Mult}(\cdot \mid d, \theta_d)$
      - Generate $w_i \sim \text{Mult}(\cdot \mid z_i, \beta_k)$

$$\prod_{d=1}^{M} P\left(w_1, \dots, w_{N_d}, d \mid \beta, \theta, \pi\right)$$

$$= \prod_{d=1}^{M} P(d \mid \pi) \prod_{i=1}^{N_d} \sum_{k=1}^{K} P(z_i = k \mid d, \theta_d) P(w_i \mid \beta_k)$$

$$= \prod_{d=1}^{M} \pi_d \prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{d,k} \beta_{k,w_i}$$

Prior Distribution for Topic Mixture

- Goal: *topic mixture proportions* for each document are drawn from some distribution.
  - Distribution on multinomials (k-tuples of non-negative numbers that sum to one)
- The space is of all of these multinomials can be interpreted geometrically as a (k-1)-*simplex*
  - Generalization of a triangle to (k-1) dimensions
- Criteria for selecting our prior:
  - It needs to be defined for a (k-1)-simplex
  - Should have nice properties

In Bayesian probability theory, if the posterior distributions $p(\theta|x)$ are in the same family as the prior probability distribution $p(\theta)$, the prior and posterior are then called **conjugate distributions,** and the prior is called a **conjugate prior** for the likelihood function. [Wikipedia]

Latent Dirichlet Allocation

- Document = mixture of topics (as in pLSI), but according to a Dirichlet prior
  - When we use a uniform Dirichlet prior, pLSI=LDA

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation.
Journal of Machine Learning Research, 3:993-1022, January **2003**

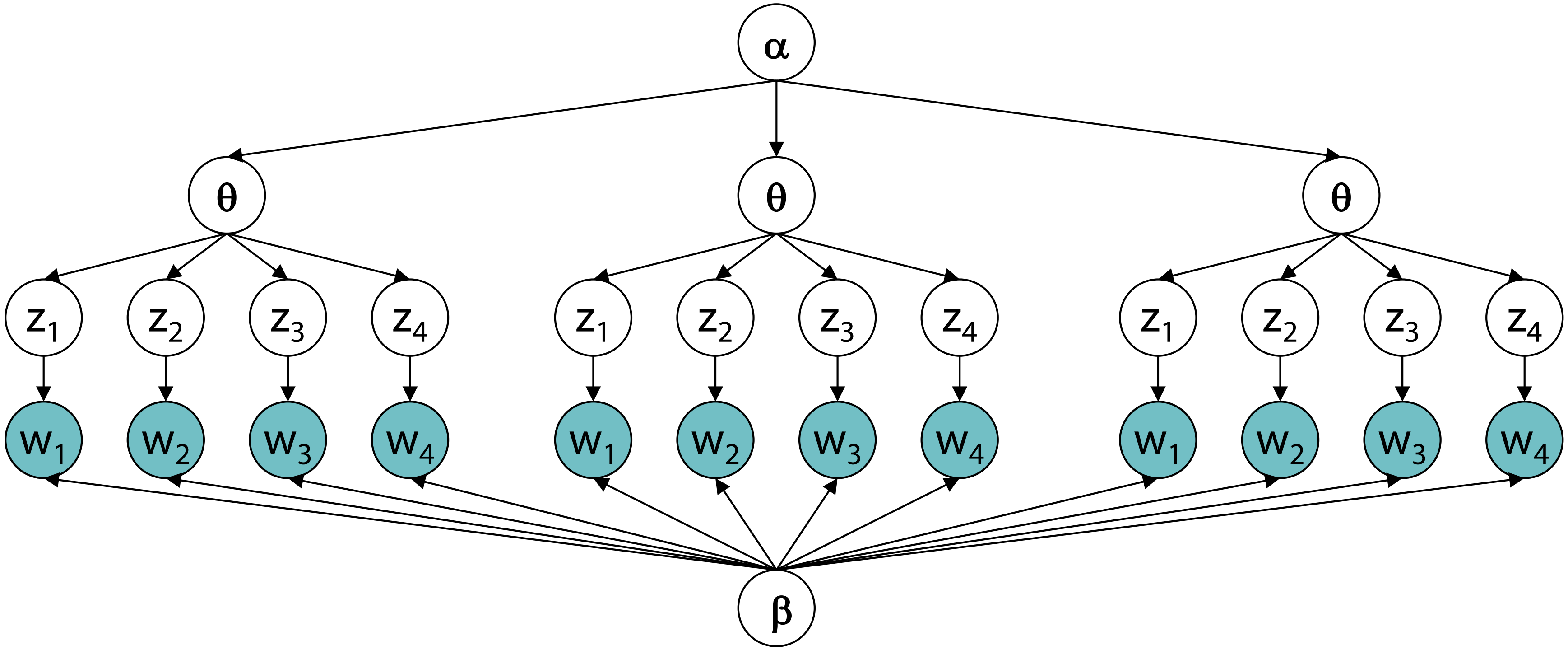
Dirichlet Distributions

$$p(\theta|\alpha) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

- Defined over a (k-1)-simplex
  - Takes K non-negative arguments which sum to one.
  - Consequently it is a natural distribution to use over multinomial distributions.
- The Dirichlet parameter α_i can be thought of as a prior count of the i^{th} class
- Conjugate prior to the multinomial distribution
  - Conjugate prior: if our likelihood is multinomial with a Dirichlet prior, then the posterior is also a Dirichlet

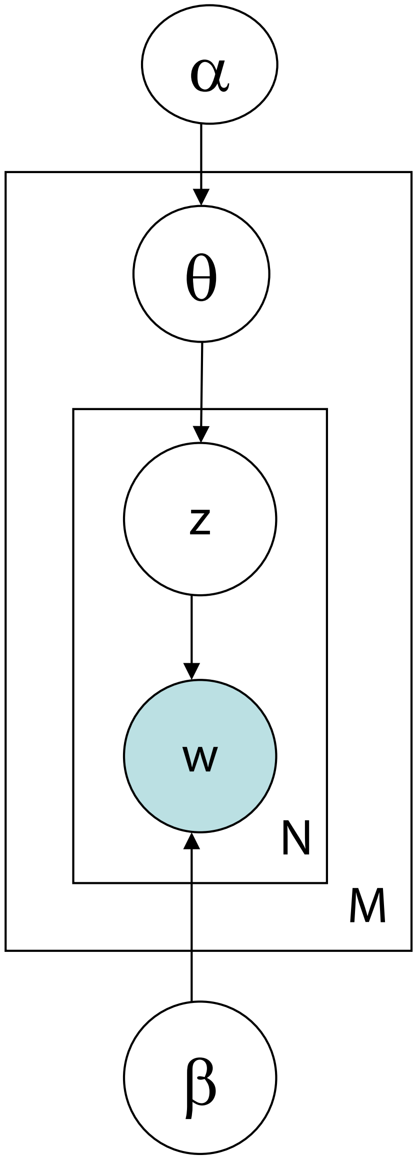
$$f(x_1, \ldots, x_K; p_1, \ldots, p_K) = \frac{\Gamma(\sum_i x_i + 1)}{\prod_i \Gamma(x_i + 1)} \prod_{i=1}^{K} p_i^{x_i}$$

LDA Model



α
θ
θ
θ
z1 z2 z3 z4
z1 z2 z3 z4
z1 z2 z3 z4
w1 w2 w3 w4
w1 w2 w3 w4
w1 w2 w3 w4
β

LDA Model – Parameters



←Proportions parameter
(k-dimensional vector of real numbers)

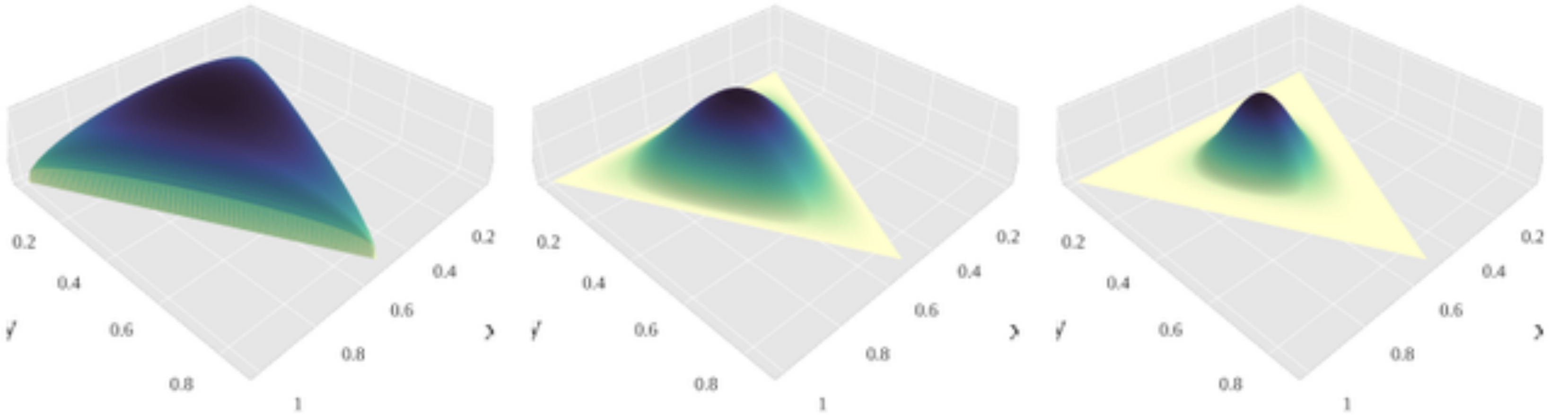
←Per-document topic distribution
(k-dimensional vector of
probabilities summing up to 1)

←Per-word topic assignment
(number from 1 to k)

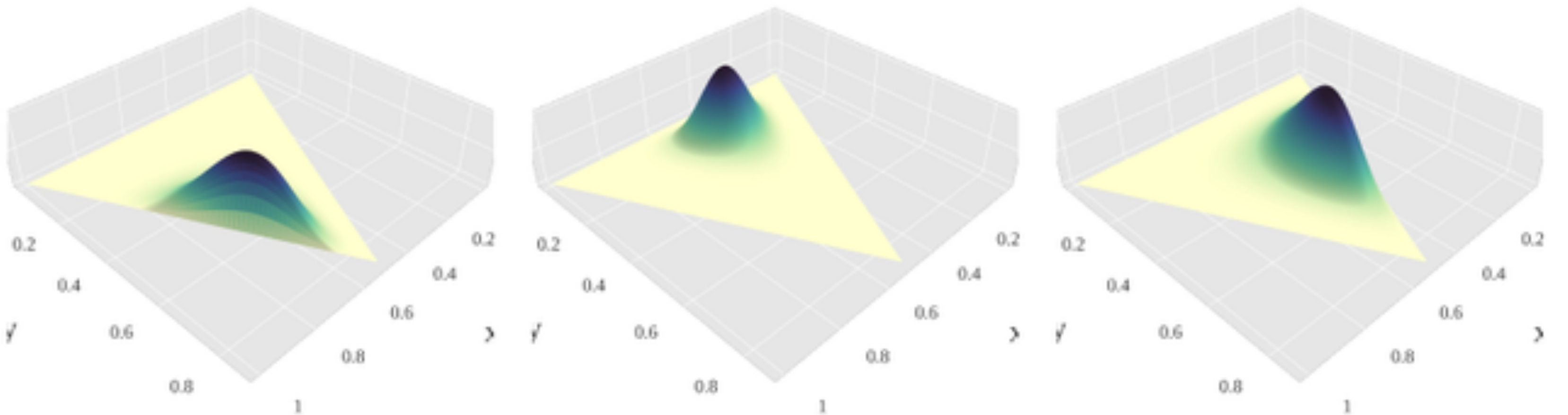
←Observed word
(number from 1 to v, where v is the
number of words in the vocabulary)

←Word "prior"
(v-dimensional)

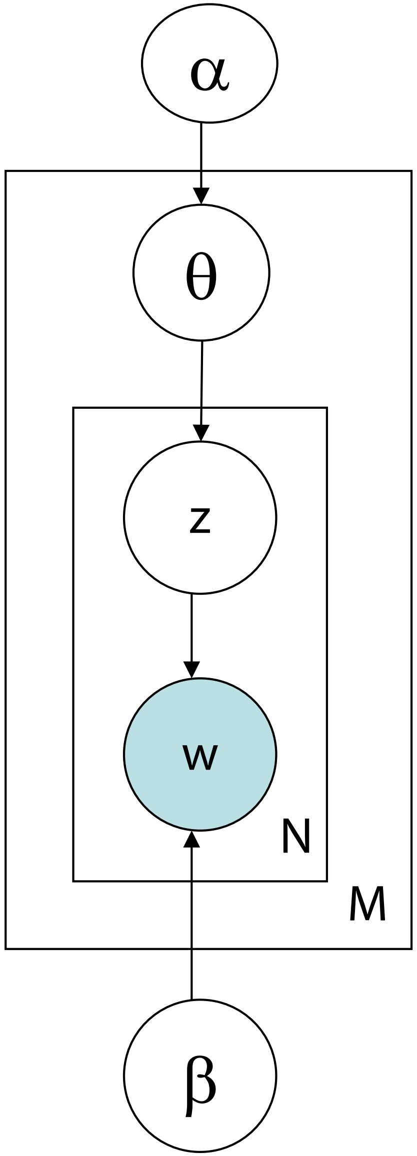
Dirichlet Distribution over a 2-Simplex



A panel illustrating probability density functions of a few Dirichlet distributions over a 2-simplex, for the following α vectors (clockwise, starting from the upper left corner): (1.3, 1.3, 1.3), (3,3,3), (7,7,7), (2,6,11), (14, 9, 5), (6,2,6). [Wikipedia]



LDA Model – Plate Notation



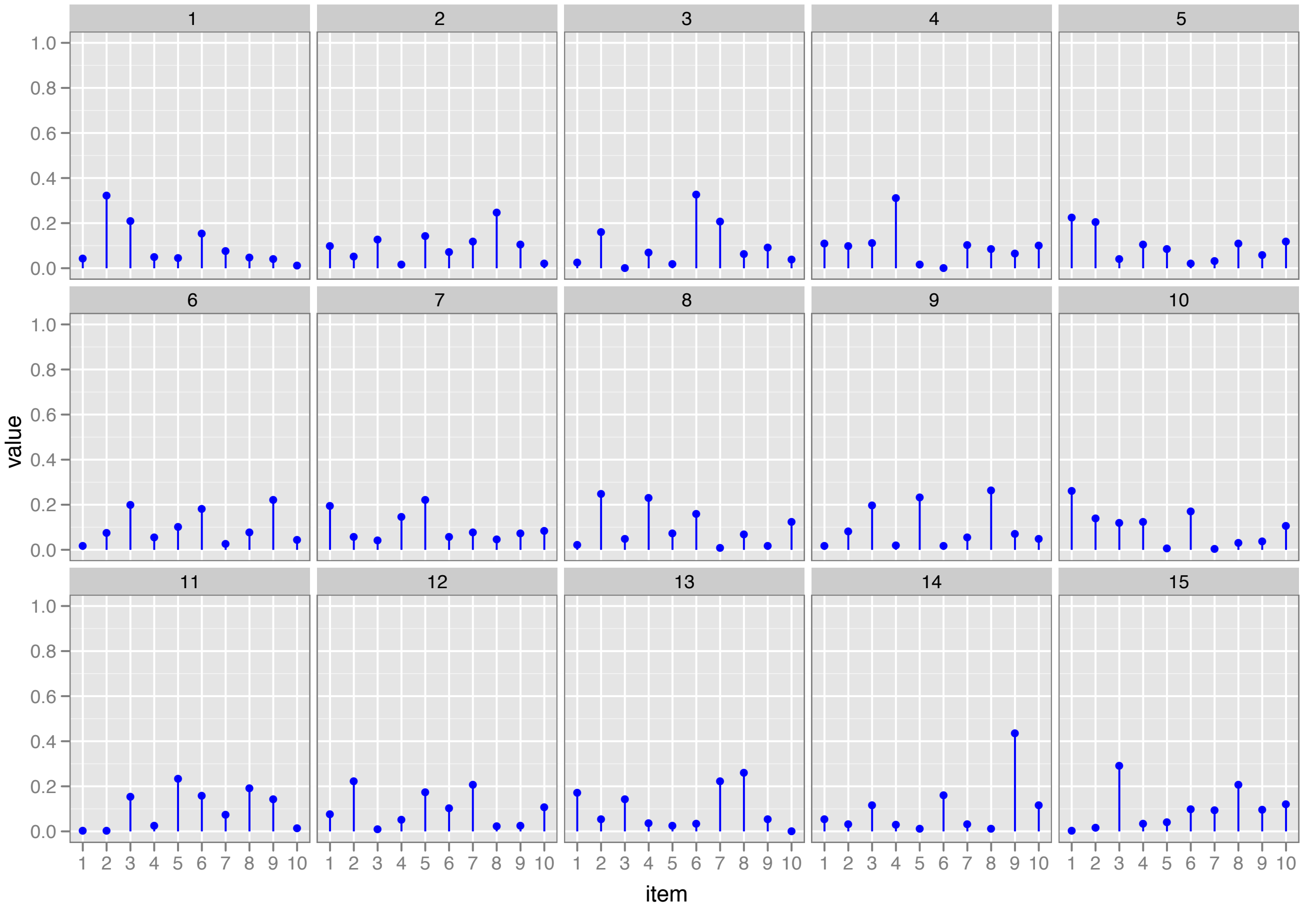
- For each document d,
  - Generate $\theta_d \sim \text{Dirichlet}(\cdot \mid \alpha)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot \mid \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot \mid z_i, \beta)$

$$P\left(\beta, \theta, z_1, \ldots, z_{N_d}, w_1, \ldots, w_{N_d}\right)$$

$$= \prod_{d=1}^{M} P(\theta_d \mid \alpha) \prod_{i=1}^{N_d} P(z_i \mid \theta_d) P(w_i \mid \beta, z_i)$$

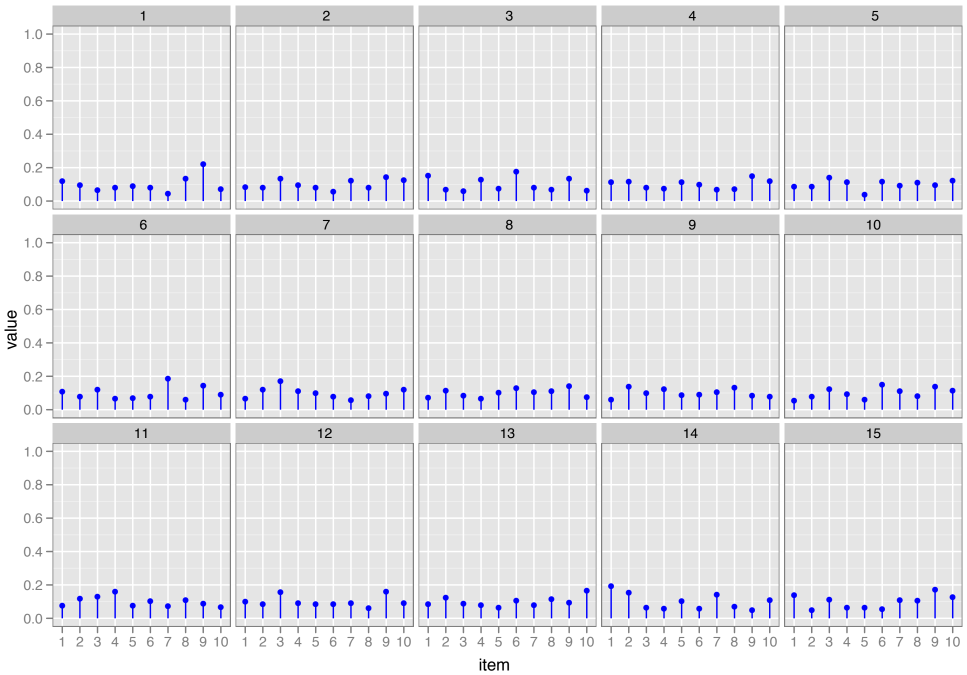
Corpus-level Parameter $\alpha = K^{-1} \Sigma_i \alpha_i$

- Let $\alpha = 1$
- Per-document topic distribution: K = 10, D = 15

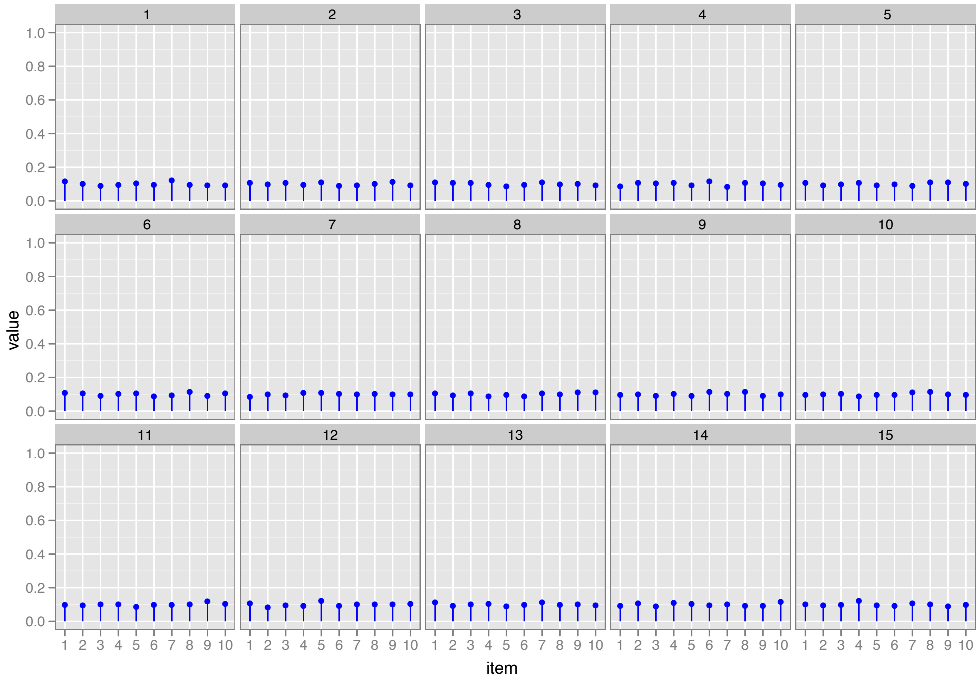


Corpus-level Parameter α

- $\alpha = 10$

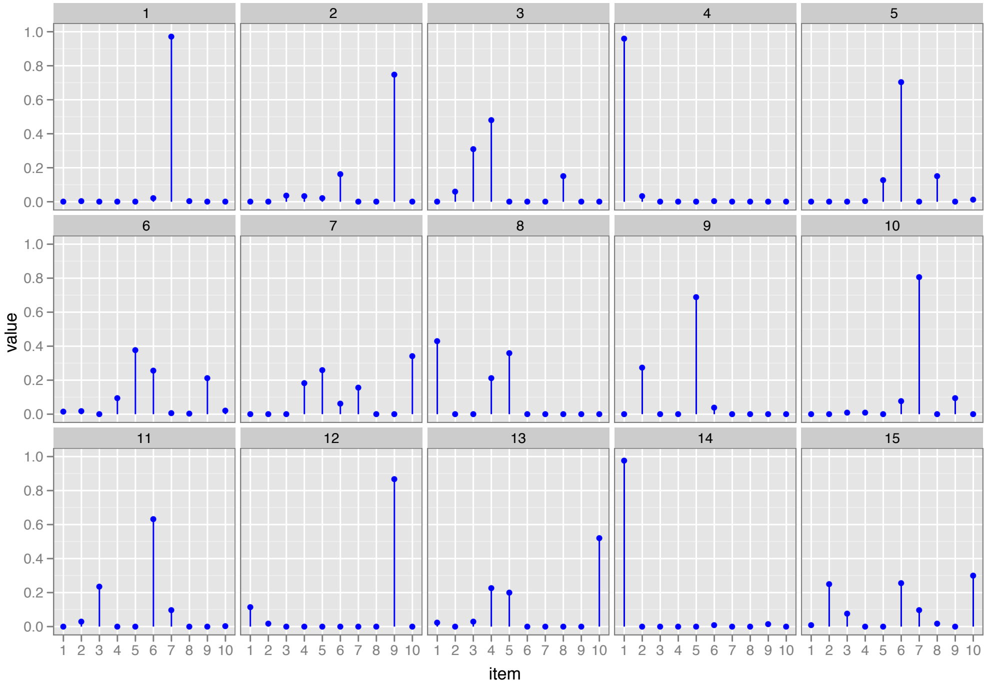


- $\alpha = 100$

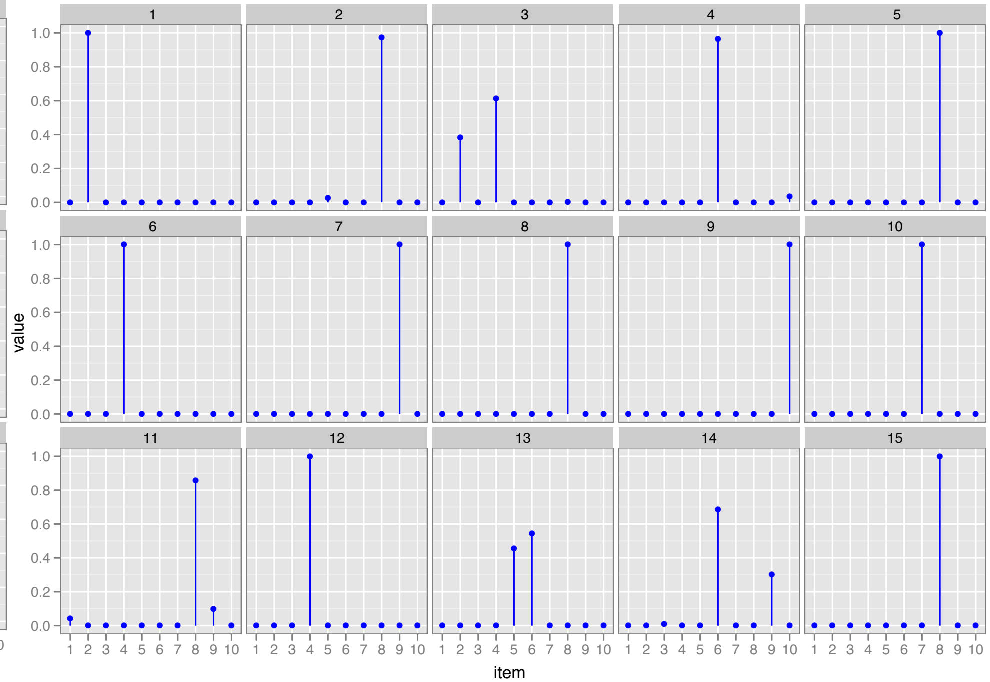


Corpus-level Parameter α

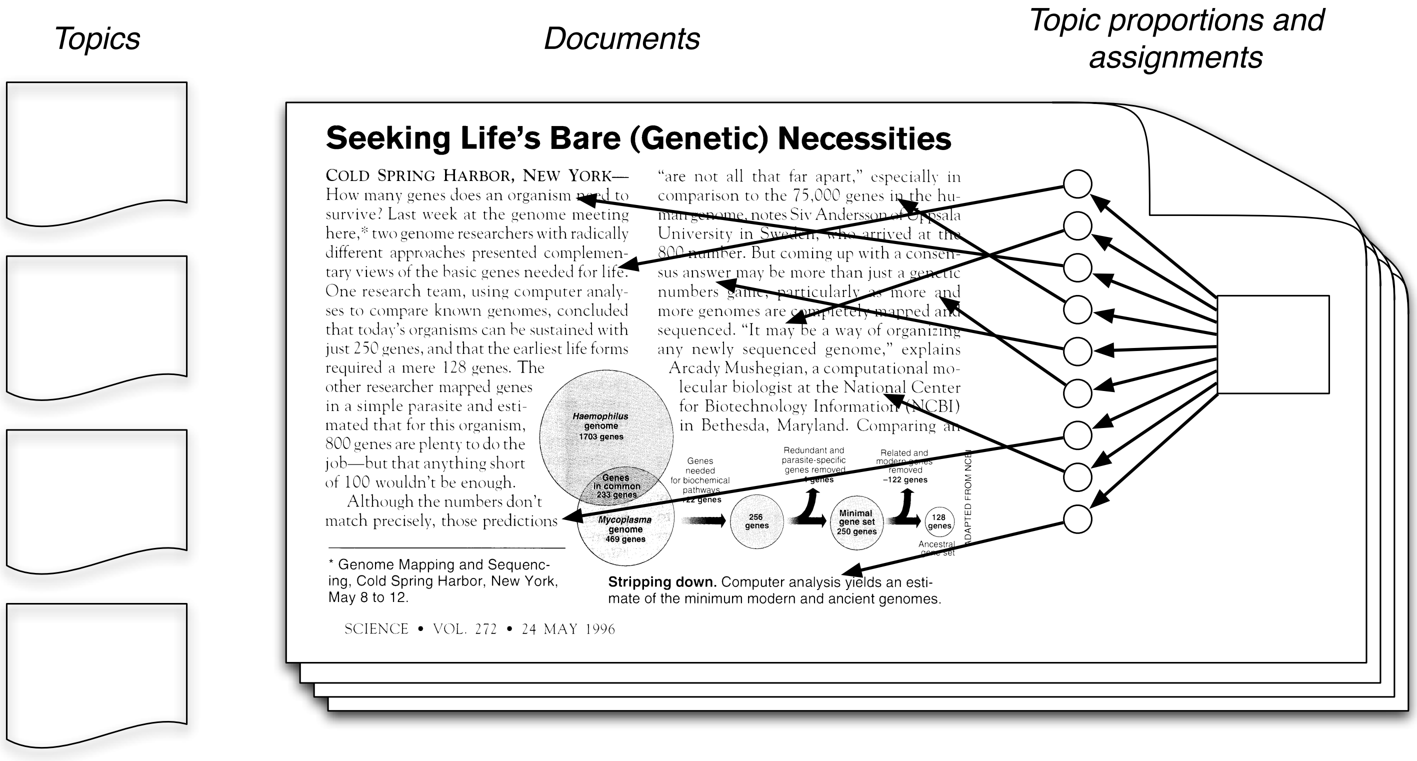
- $\alpha = 0.1$



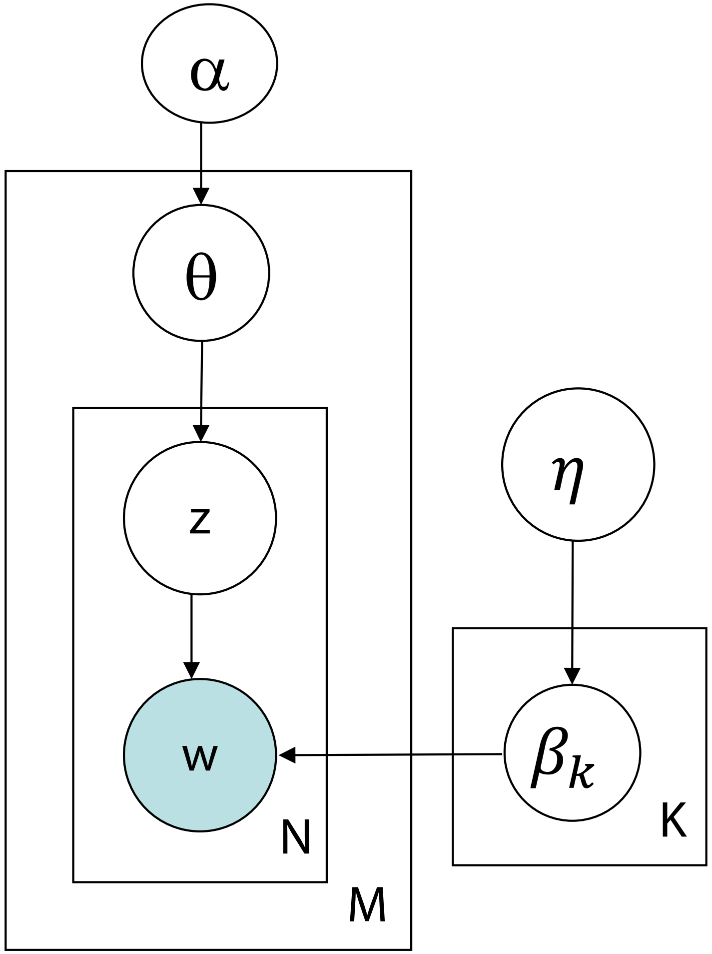
- $\alpha = 0.01$



Back to Topic Modeling Scenario

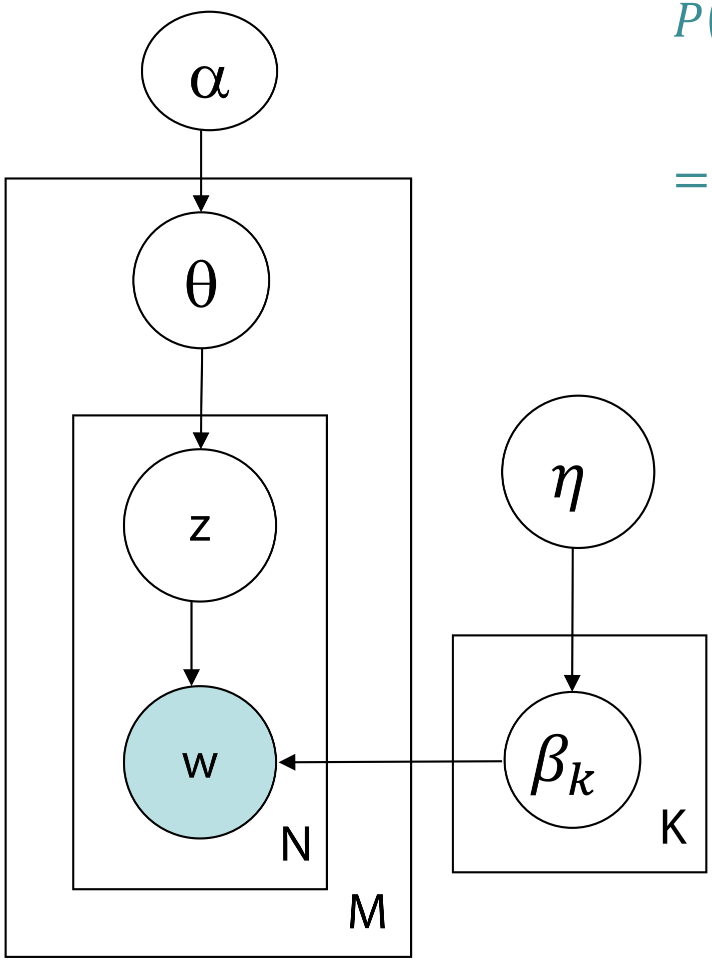


Smoothed LDA Model



- Give a different word distribution to each topic
  - β is $K \times V$ matrix (V vocabulary size), each row denotes word distribution of a topic
- For each document d
  - Choose $\theta_d \sim \text{Dirichlet}(\cdot \mid \alpha)$
  - Choose $\beta_k \sim \text{Dirichlet}(\eta \cdot \mid)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot \mid \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot \mid z_i, \beta_k)$

Smoothed LDA Model

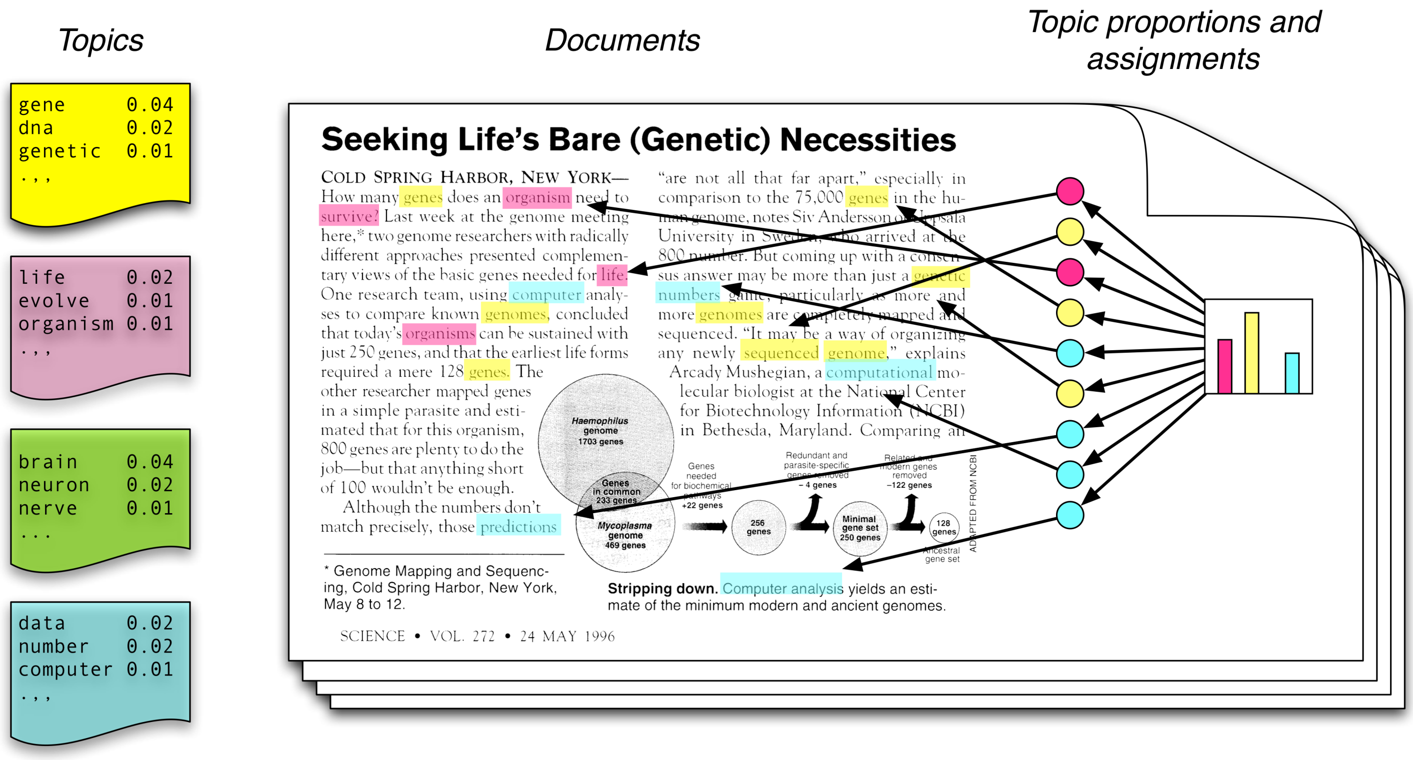


$$P\left(\beta, \theta, z_1, \ldots, z_{N_d}, w_1, \ldots, w_{N_d}\right)$$

$$= \left(\prod_{k=1}^{K} P(\beta_k | \eta)\right)$$

$$\cdot \left(\prod_{d=1}^{M} P(\theta_d | \alpha) \prod_{i=1}^{N_d} P\left(z_{d,i} \middle| \theta_d\right) P(w_{d,i} | \beta_{1:K}, z_{d,i}) \right)$$

Back to Topic Modeling Scenario



- But…

Why does LDA “work”?

- Trade-off between 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

Inference: The Problem (non-smoothed version)

- To which topics does a given document belong?
  - Compute the posterior distribution of the hidden variables given a document:

$$P(\theta, \mathbf{z} | \mathbf{w}, \alpha, \beta) = \frac{P(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta)}{P(\mathbf{w} | \alpha, \beta)}$$

$$P(\theta, \mathbf{z}, \mathbf{w} | \alpha, \beta) = P(\theta | \alpha) \prod_{i=1}^{N} P(z_i | \theta) P(w_i | z_i, \beta)$$

$$P(\mathbf{w} | \alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{k=1}^{K} \theta_k^{\alpha_k - 1} \right) \left(\prod_{i=1}^{N} \sum_{k=1}^{K} \prod_{j=1}^{V} (\theta_k \beta_{kj})^{w_i^j} \right) d\theta$$

This not only looks awkward, but is as well *computationally intractable* in general. Coupling between θ and β_{ij}. Solution: *Approximations*.

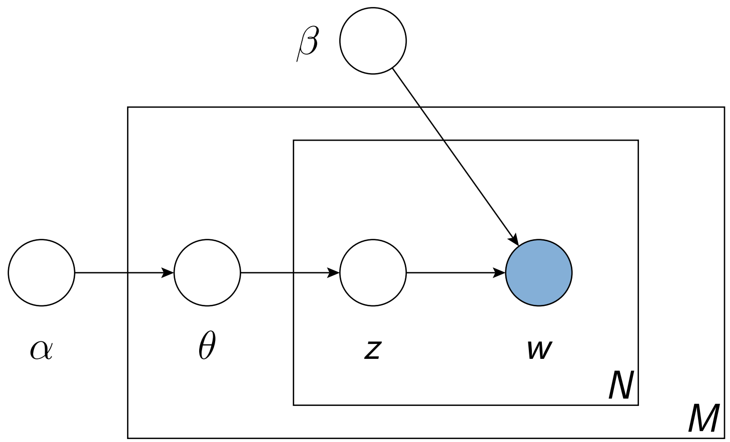
LDA Learning

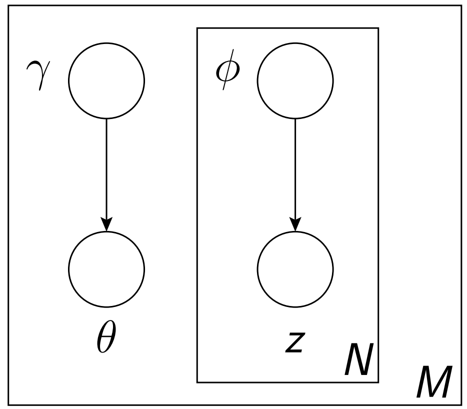
- Parameter learning:
  - Variational EM
    - Numerical approximation using lower-bounds
    - Results in biased solutions
    - Convergence has numerical guarantees
  - Gibbs Sampling
    - Stochastic simulation
    - Unbiased solutions
    - Stochastic convergence

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation.
Journal of Machine Learning Research, 3:993-1022, January **2003**

LDA: Variational Inference

- Replace LDA model with simpler one





- Minimize the Kullback-Leibler divergence between the two distributions.

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} KL(q(\theta, z|\gamma, \phi)||p(\theta, z|w, \alpha, \beta))$$

$$KL(P\|Q) = -\sum_i P(i) \log \frac{Q(i)}{P(i)},$$

LDA: Gibbs Sampling

- MCMC algorithm
  - Fix all current values but one and sample that value, e.g, for z

$$P(z_i = k|\boldsymbol{z}_{-i}, \boldsymbol{w})$$

  - Eventually converges to true posterior

Variational Inference vs. Gibbs Sampling

- Gibbs sampling is slower (takes days for mod.-sized datasets), variational inference takes a few hours.
- Gibbs sampling is more accurate.
- Gibbs sampling convergence is difficult to test, although quite a few machine learning approximate inference techniques also have the same problem.

LDA Application: Reuters Data

- Setup
  - 100-topic LDA trained on a 16,000 documents corpus of news articles by Reuters
  - Some standard stop words removed
- Top words from some of the P(w|z)

"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

LDA Application: Reuters Data

- Result

Again: "Arts", "Budgets", "Children", "Education".

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.

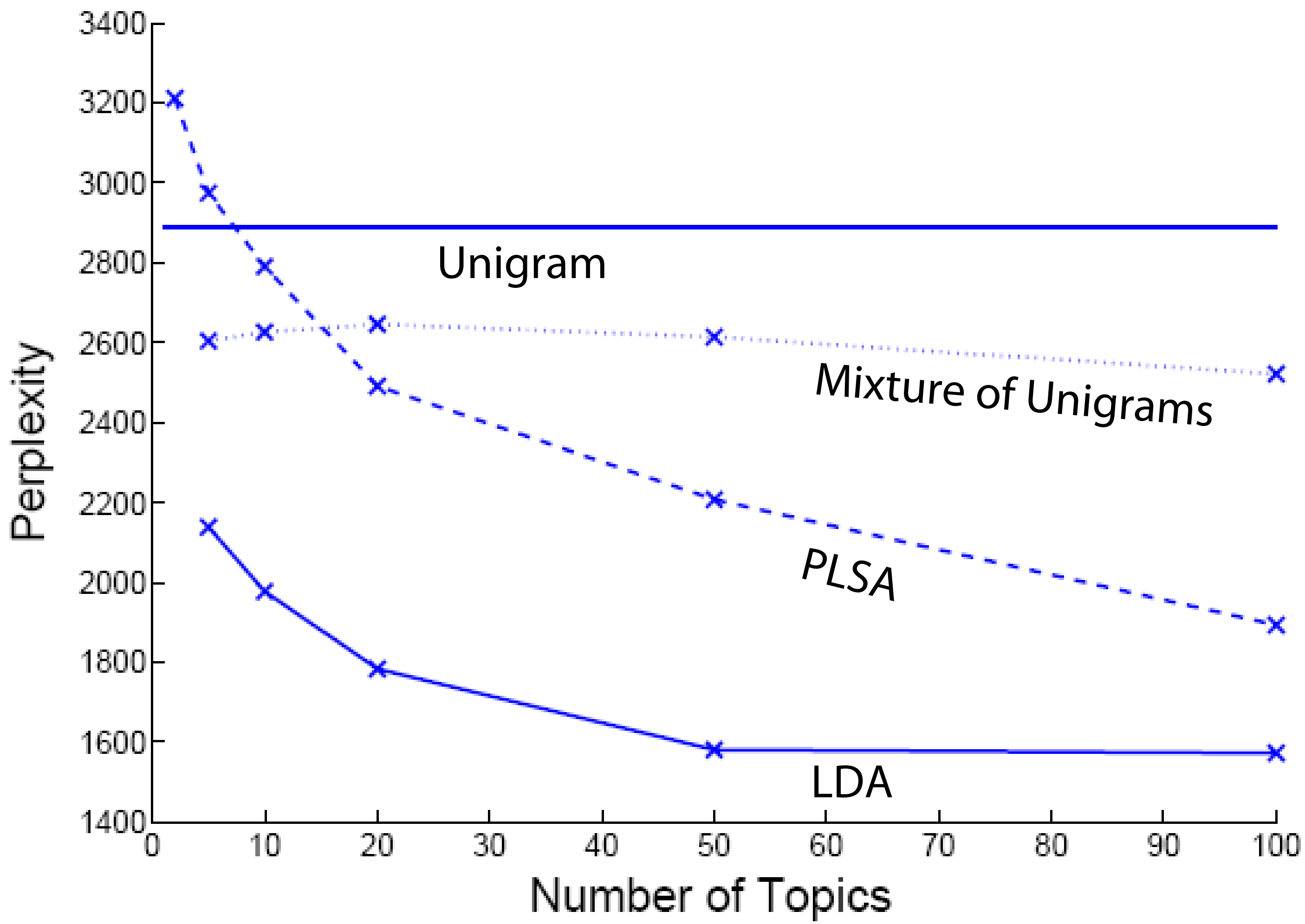
Measuring Performance

- Perplexity of a probability model
- How well a probability distribution or probability model predicts a sample
  - q: Model of an unknown probability distribution p based on a training sample drawn from p
  - Evaluate q by asking how well it predicts a separate test sample $x_1, \ldots, x_N$ also drawn from p
  - Perplexity of q w.r.t. sample $x_1, \ldots, x_N$ defined as

$$2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 q(x_i)}$$

  - A better model q will tend to assign higher probabilities to $q(x_i)$ → lower perplexity ("less surprised by sample")

Perplexity of Various Models



Use of LDA

- A widely used topic model (Griffiths, Steyvers, 04)
- Complexity is an issue
- Use in IR:
  - Ad hoc retrieval (Wei and Croft, SIGIR 06: TREC benchmarks)
  - Improvements over traditional LM (e.g., LSI techniques)
  - But no consensus on whether there is any improvement over Relevance model, i.e., model with relevance feedback (relevance feedback part of the TREC tests)

T. Griffiths, M. Steyvers, Finding Scientific Topics. Proceedings of the National Academy of Sciences, 101 (suppl. 1), 5228-5235. **2004**

Xing Wei and W. Bruce Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (SIGIR '06). ACM, New York, NY, USA, 178-185. **2006**.

TREC=Text REtrieval Conference

Generative Topic Models for Community Analysis

Pilfered from: Ramesh Nallapati

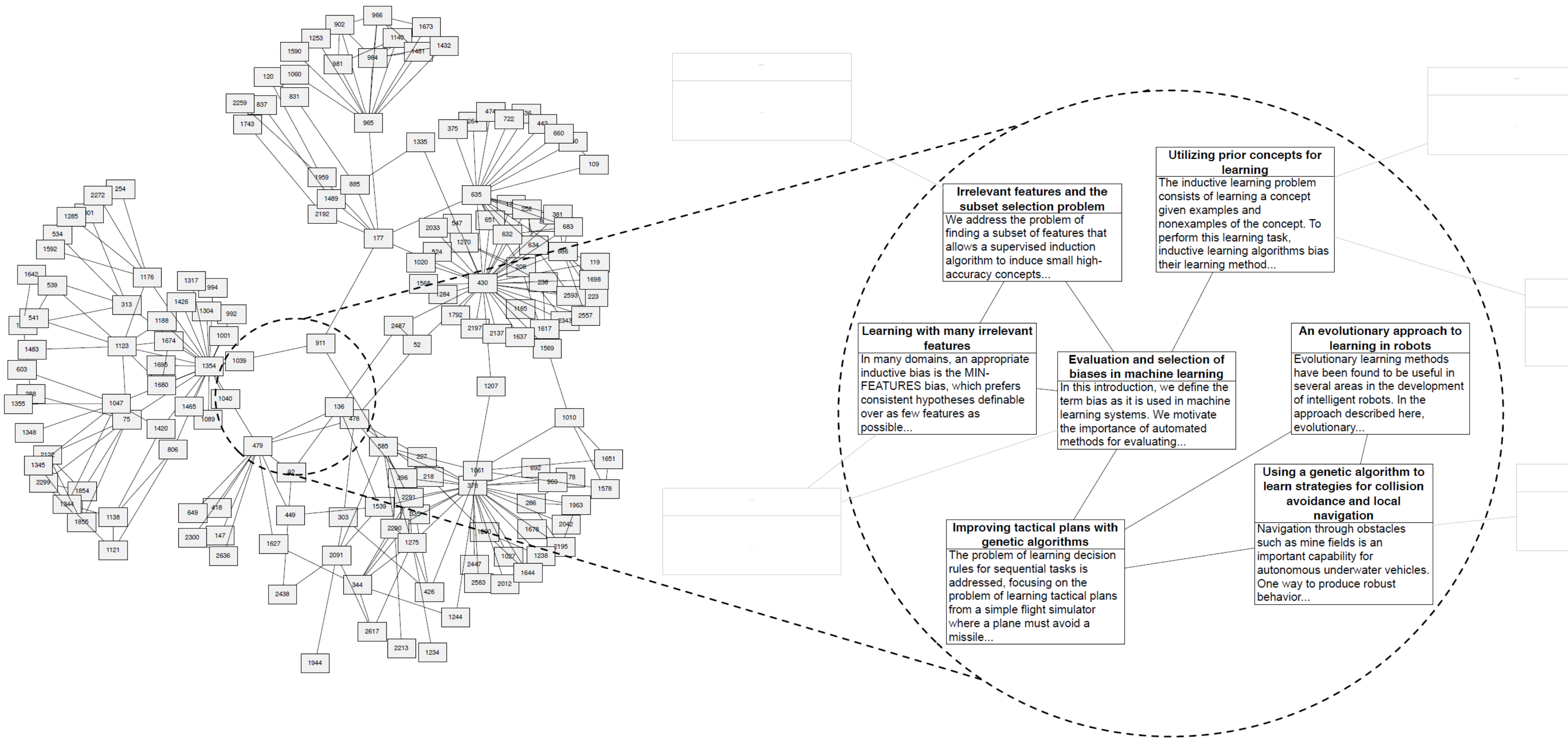
http://www.cs.cmu.edu/~wcohen/10-802/lda-sep-18.ppt

&

Arthur Asuncion, Qiang Liu, Padhraic Smyth:

Statistical Approaches to Joint Modeling of Text and Network Data

What if the corpus has network structure?



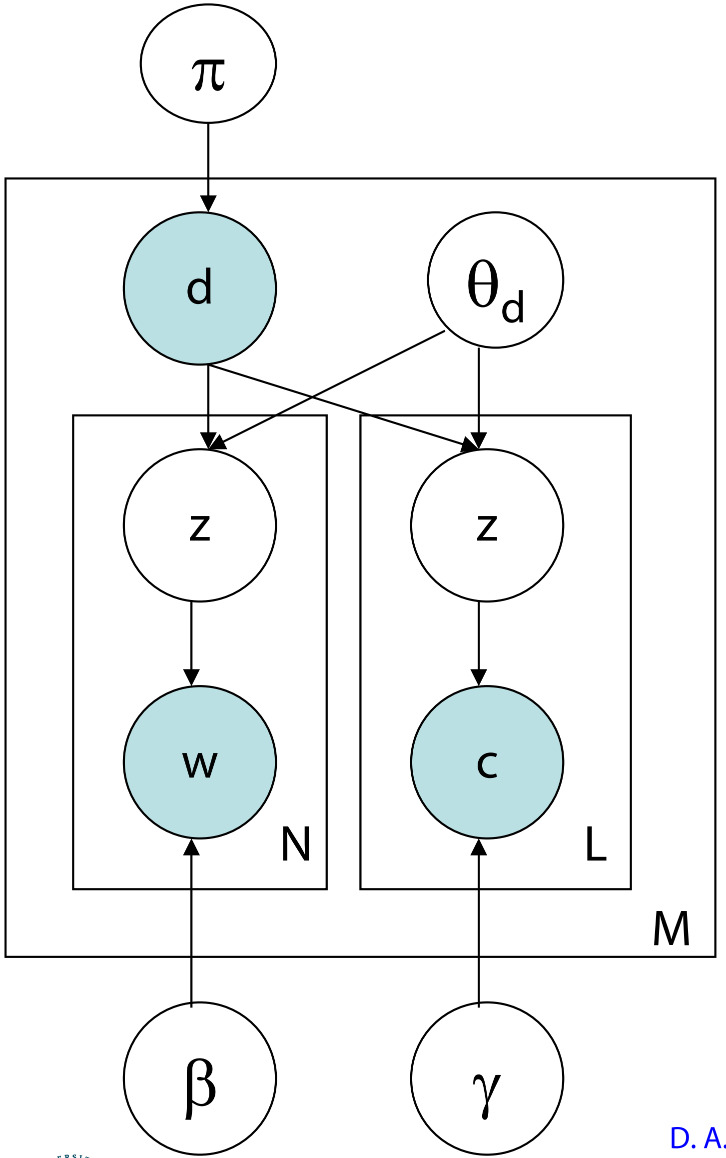
CORA citation network. Figure from [Chang, Blei, AISTATS 2009]

J. Chang, and D. Blei. Relational Topic Models for Document Networks. AISTATS, volume 5 of JMLR Proceedings, page 81-88. JMLR.org, **2009**.

Outline

- Topic Models for Community Analysis
  - Citation Modeling
    - with pLSI
    - with LDA
  - Modeling influence of citations
  - Relational Topic Models

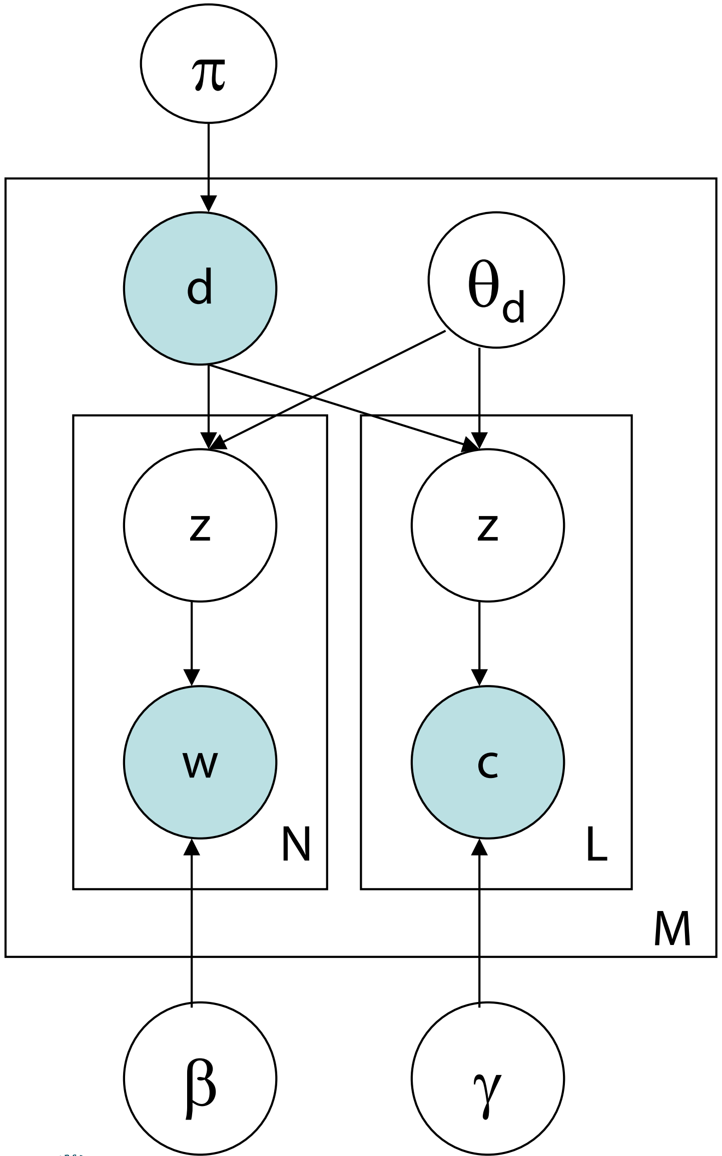
Hyperlink Modeling Using pLSI



- Select document $d \sim \mathrm{Mult}(\pi)$
  - For each position $n = 1,\ldots, N_d$
    - Generate $z_n \sim \mathrm{Mult}(\cdot\,|\theta_d)$
    - Generate $w_n \sim \mathrm{Mult}(\cdot\,|\beta_{z_n})$
  - For each citation $j = 1,\ldots, L_d$
    - Generate $z_j \sim \mathrm{Mult}(\cdot\,|\theta_d)$
    - Generate $c_j \sim \mathrm{Mult}(\cdot\,|\gamma_{z_j})$

D. A. Cohn, Th. Hofmann, The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity, In: Proc. NIPS, pp. 430-436, **2000**

Hyperlink Modeling Using pLSI



- pLSI likelihood

$$\prod_{d=1}^{M} P(w_1, \dots, w_{N_d}, d | \theta, \beta, \pi)$$

$$= \prod_{d=1}^{M} \pi_d \left(\prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \beta_{kw_n} \right)$$

- New likelihood

$$\prod_{d=1}^{M} P(w_1, \dots, w_{N_d}, c_1, \dots, c_{L_d}, d | \theta, \beta, \gamma, \pi)$$

$$= \prod_{d=1}^{M} \pi_d \left(\prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \beta_{kw_n} \right) \left(\prod_{j=1}^{L_d} \sum_{k=1}^{K} \theta_{dk} \gamma_{kc_j} \right)$$

- Learning using EM

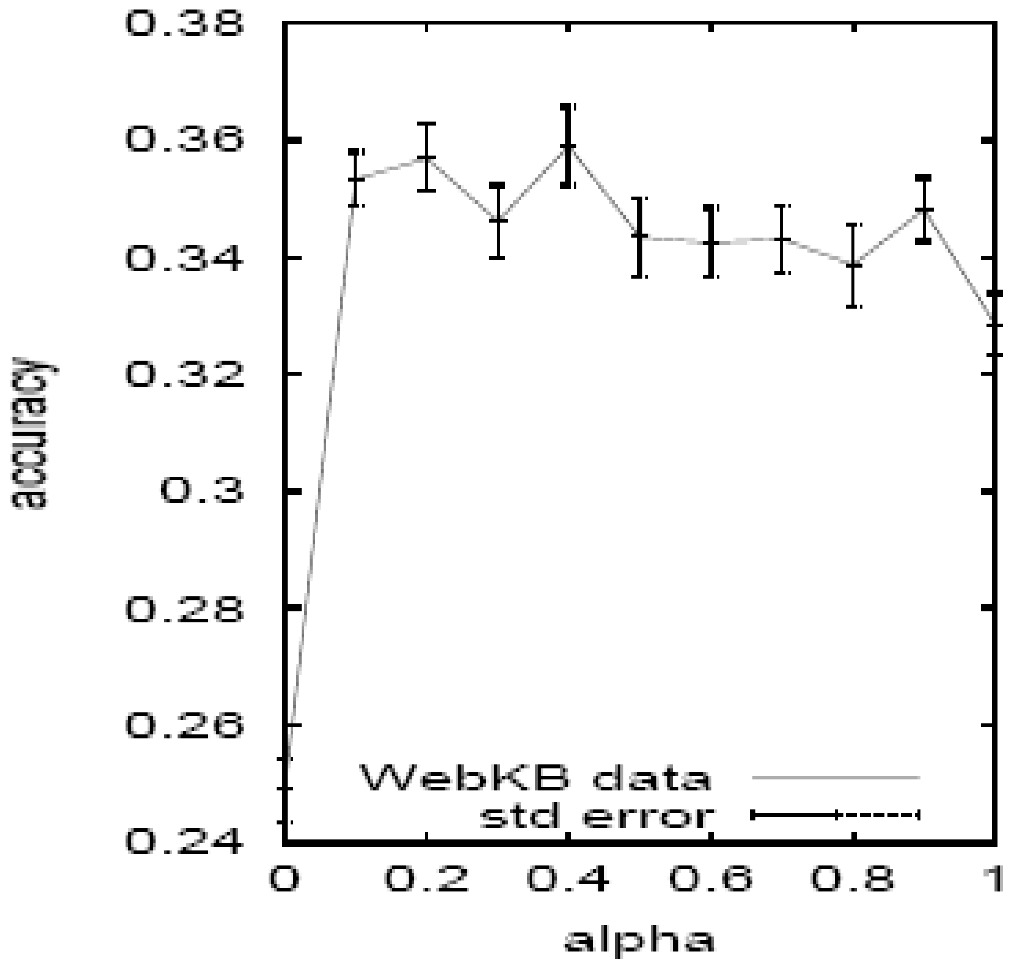
Hyperlink Modeling Using pLSI

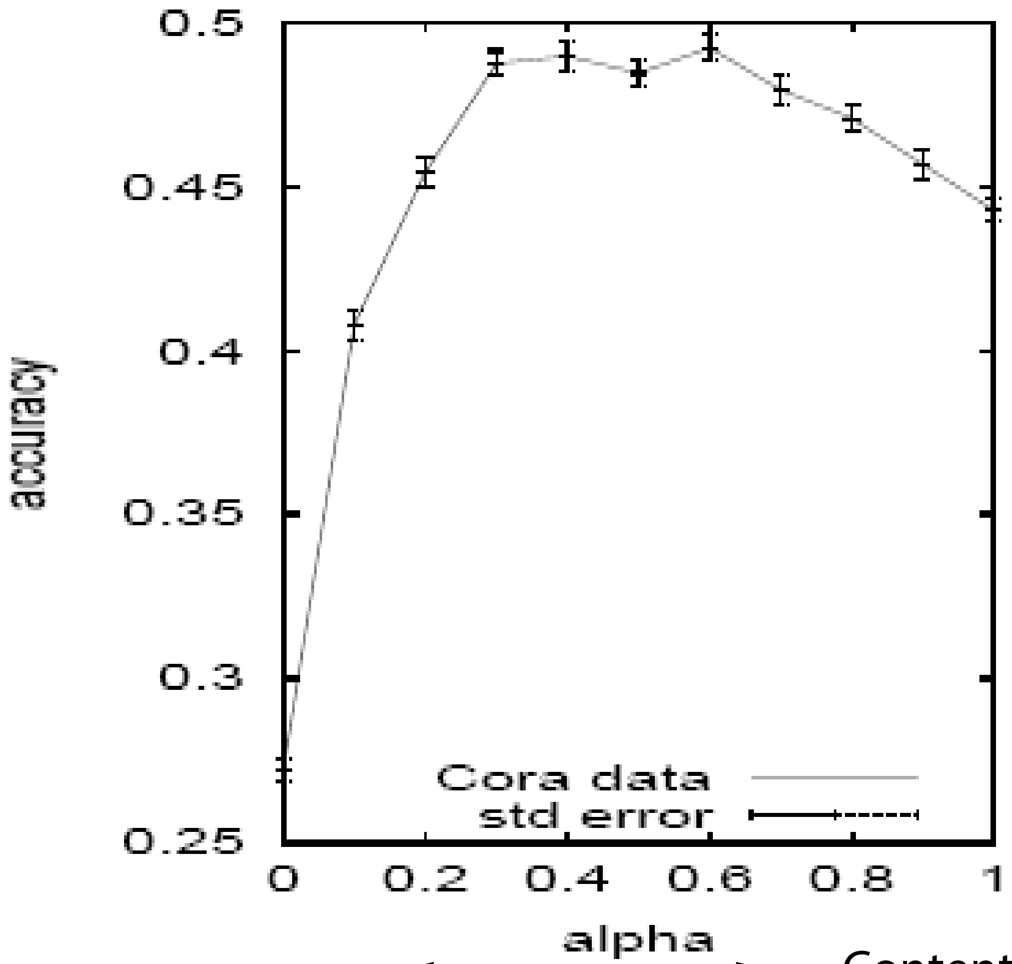
- Heuristic
  - $0 < \alpha < 1$ determines the relative importance of content and hyperlinks

$$\prod_{d=1}^{M} P\left(w_1, \dots, w_{N_d}, c_1, \dots, c_{L_d}, d \middle| \theta, \beta, \gamma, \pi\right)$$

$$= \prod_{d=1}^{M} \pi_d \left(\prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \beta_{k w_n}\right)^{\alpha} \left(\prod_{j=1}^{L_d} \sum_{k=1}^{K} \theta_{dk} \gamma_{k c_j}\right)^{1-\alpha}$$

Hyperlink Modeling Using pLSI

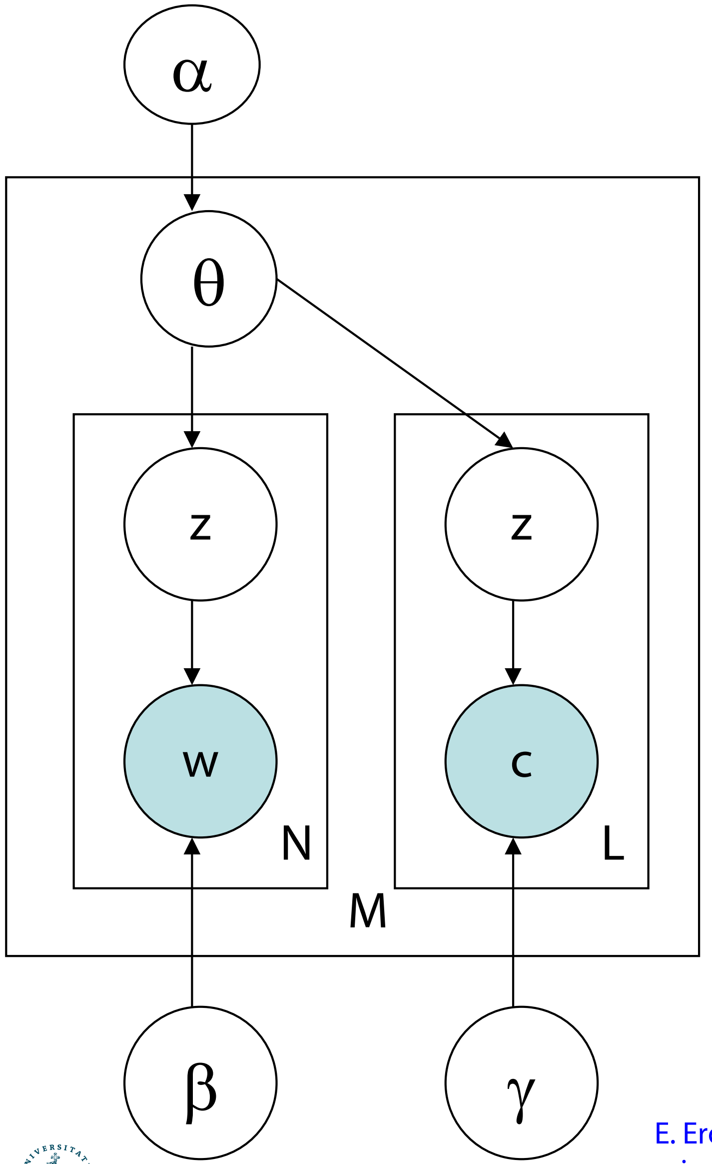




Hyperlink ⟷ Content

Hyperlink ⟷ Content

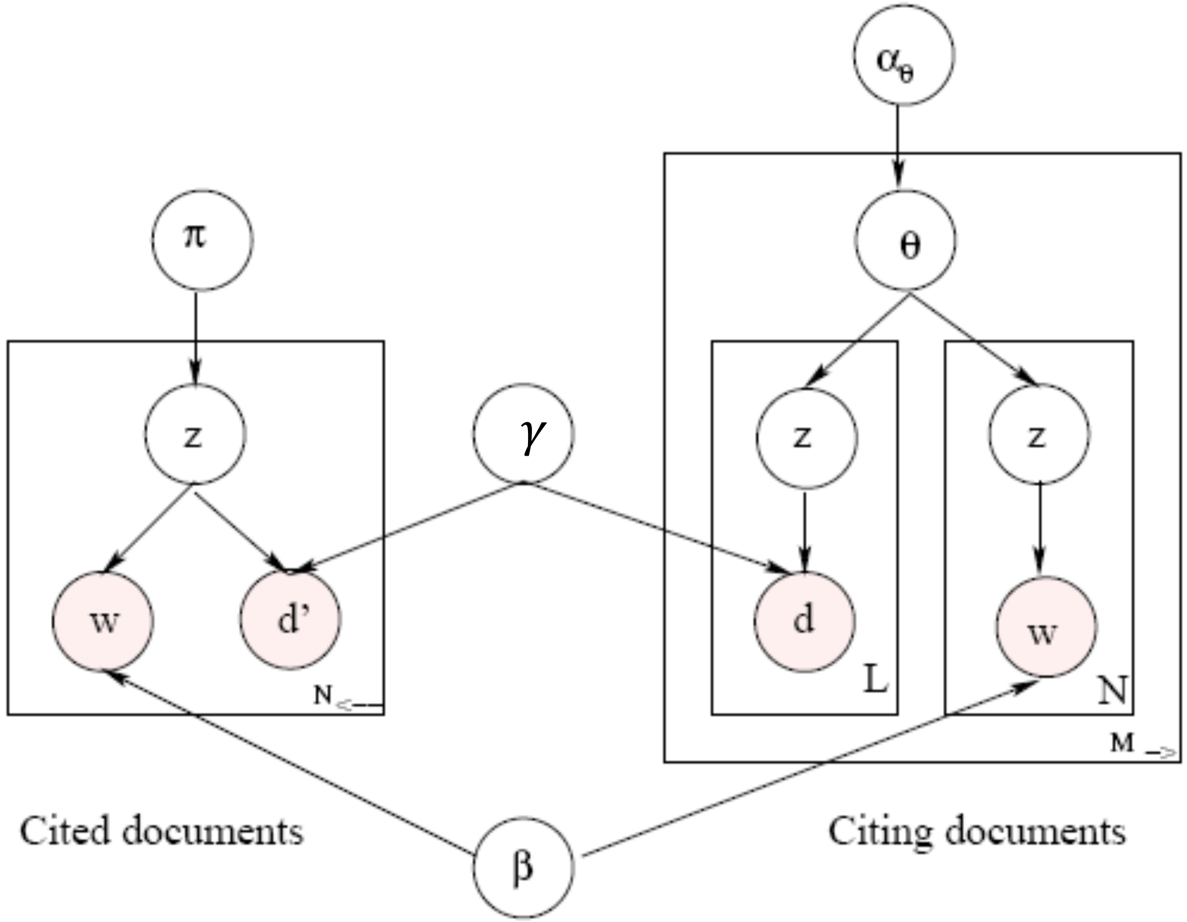
Hyperlink modeling using LDA



- For each document d,
  - Generate $\theta_d \sim \text{Dirichlet}(\alpha)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot\,|\theta_d)$
    - Generate a word $w_i \sim \text{Mult}\,(\cdot\,|\beta_{z_n})$
  - For each citation $j = 1, \ldots, L_c$
    - Generate $z_i \sim \text{Mult}(\theta_d)$
    - Generate $c_i \sim \text{Mult}\,(\cdot\,|\gamma_{z_j})$
- Learning using variational EM, Gibbs sampling

E. Erosheva, S Fienberg, J. Lafferty, Mixed-membership models of scientific publications. Proc National Academy Science U S A. 2004 Apr 6;101 Suppl 1:5220-7. Epub **2004** Mar 12.

Link-pLSI-LDA: Topic Influence in Blogs

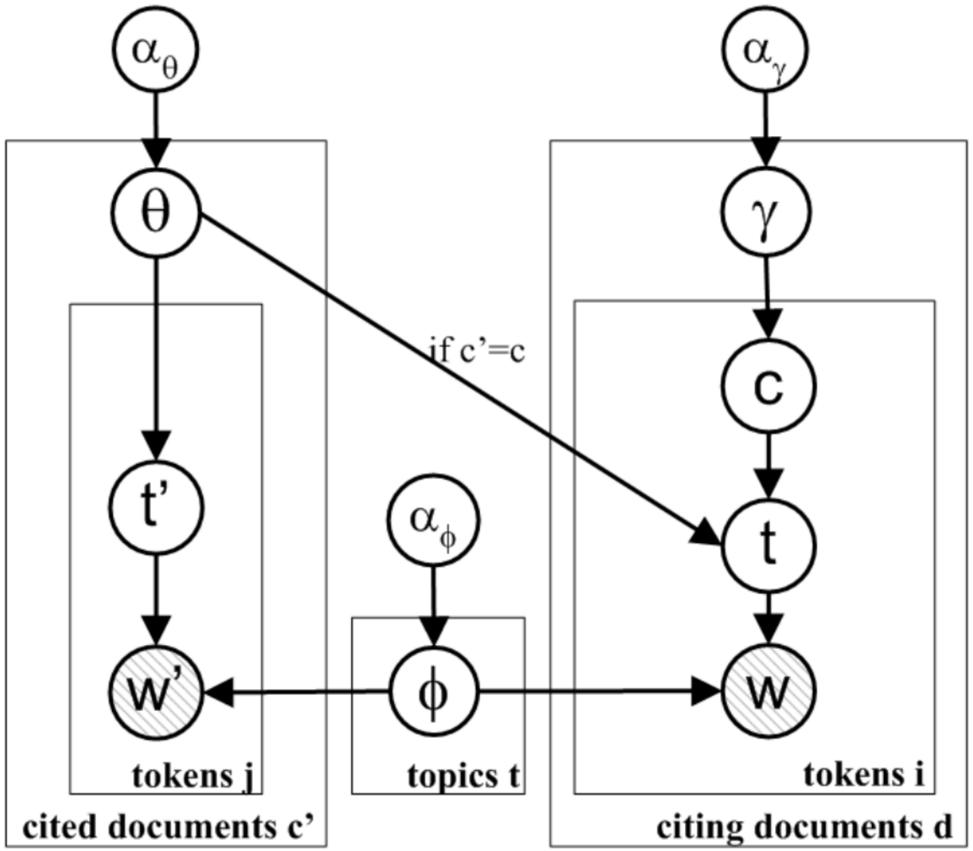


R. Nallapati, A. Ahmed, E. Xing, W.W. Cohen, Joint Latent Topic Models for Text and Citations, In: Proc. KDD, **2008**.

Topic 21 "CIA LEAK" 0.067	Topic 7 "IRAQ WAR" 0.062	Topic 16 "SUPREME COURT NOMINATIONS" 0.06	Topic 20 "SEARCH ENGINE MARKET" 0.04
TOP TOPICAL TERMS			
rove	will	robert	will
his	war	court	search
who	attack	bush	new
time	iraq	his	market
cooper	terrorist	supreme	post
karl	who	john	product
cia	world	nominate	brand
bush	terror	judge	permalink
know	muslim	will	time
report	america	conservative	yahoo
story	one	right	you
source	people	president	year
house	think	justice	comment
leak	bomb	nominee	company
plame	against	senate	business
TOP BLOG POSTS ON TOPIC			
billmon.org Whiskey Bar	willisms.com Iraq what might	themoderatevoice.com The Moderate Voice	edgeperspectives. typepad.com John Hagel
qando.net Free Markets & People	instapunk.com InstaPun***K	blogsforbush.com Blogs for Bush	.comparisonengines.com Comparison of Engines
captainsquartersblog .com, Captain's Quarters	jihadwatch.org Jihad Watch	michellemalkin.com Michelle Malkin	blogs.forrester.com Charlene Li's Blog
coldfury.com The Light Of Reason	thesharpener.net The Sharpener	captainsquartersblog.com Captain's Quarters	longtail.typepad.com The Long Tail
thismodernworld.com Tom Tomorrow	thedonovan.com Jonah's Military	wizbangblog.com Wizbang	.searchenginejournal.com Search Engine Journal

Modeling Citation Influences - Copycat Model

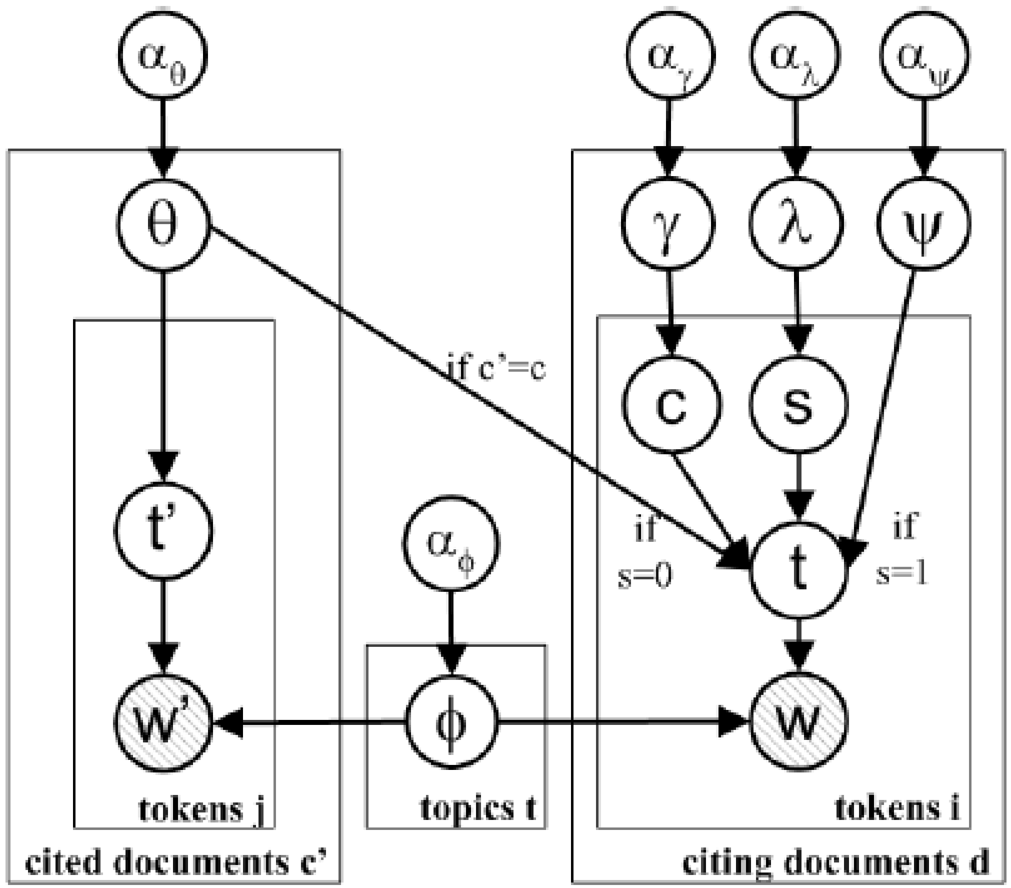
- Each topic in a citing document is drawn from one of the topic mixtures of cited publications



L. Dietz, St. Bickel, and T. Scheffer, Unsupervised Prediction of Citation Influences, In: Proc. ICML **2007**.

Modeling Citation Influences

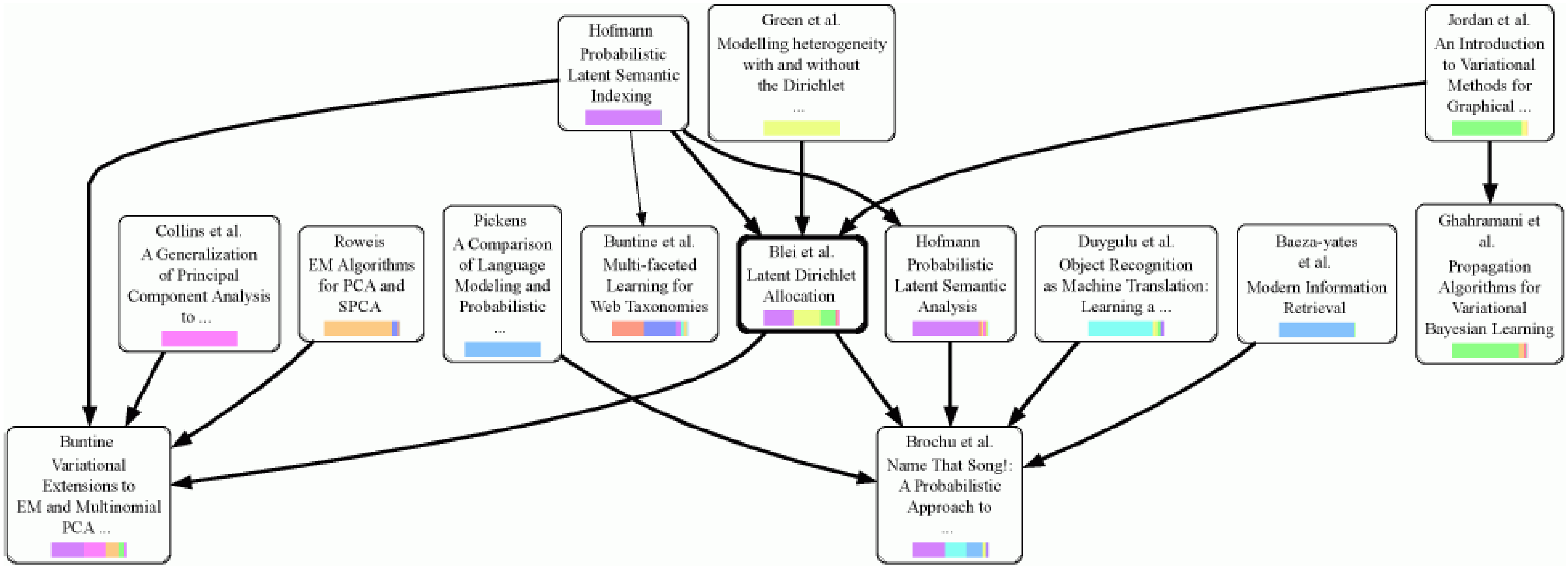
- Citation influence model: Combination of LDA and Copycat model



L. Dietz, St. Bickel, and T. Scheffer, Unsupervised Prediction of Citation Influences, In: Proc. ICML **2007**.

Modeling Citation Influences

- Citation influence graph for LDA paper



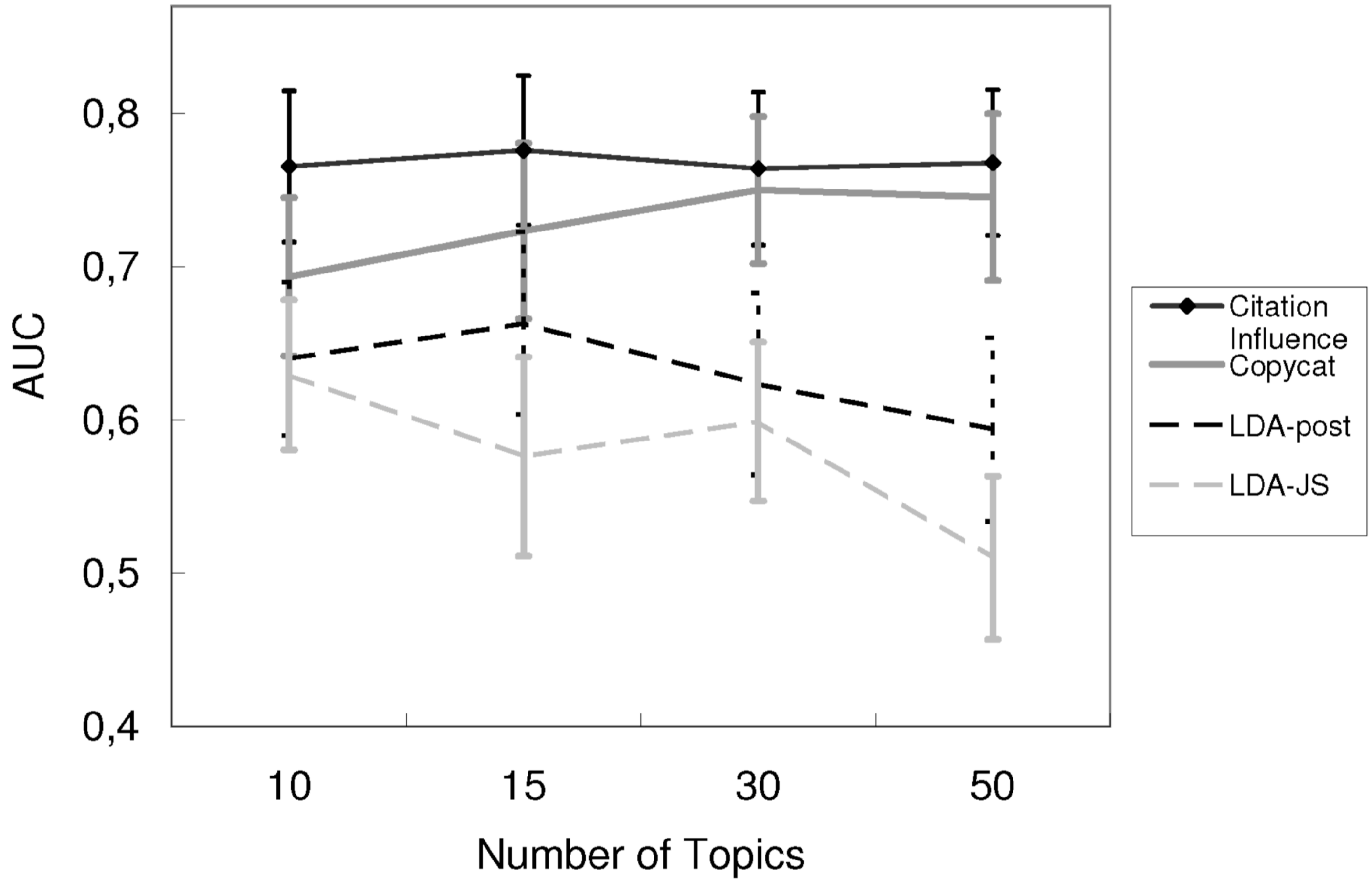
Modeling Citation Influences

- Words in LDA paper assigned to citations

Cited Title	Associated Words	γ
Probabilistic Latent Semantic Indexing	text(0.04), latent(0.04), modeling(0.02), model(0.02), indexing(0.01), semantic(0.01), document(0.01), collections(0.01)	0.49
Modelling heterogeneity with and without the Dirichlet process	dirichlet(0.02), mixture(0.02), allocation(0.01), context(0.01), variable(0.0135), bayes(0.01), continuous(0.01), improves(0.01), model(0.01), proportions(0.01)	0.25
Introduction to Variational Methods for Graphical Methods	variational(0.01), inference(0.01), algorithms(0.01), including(0.01), each(0.01), we(0.01), via(0.01)	0.22

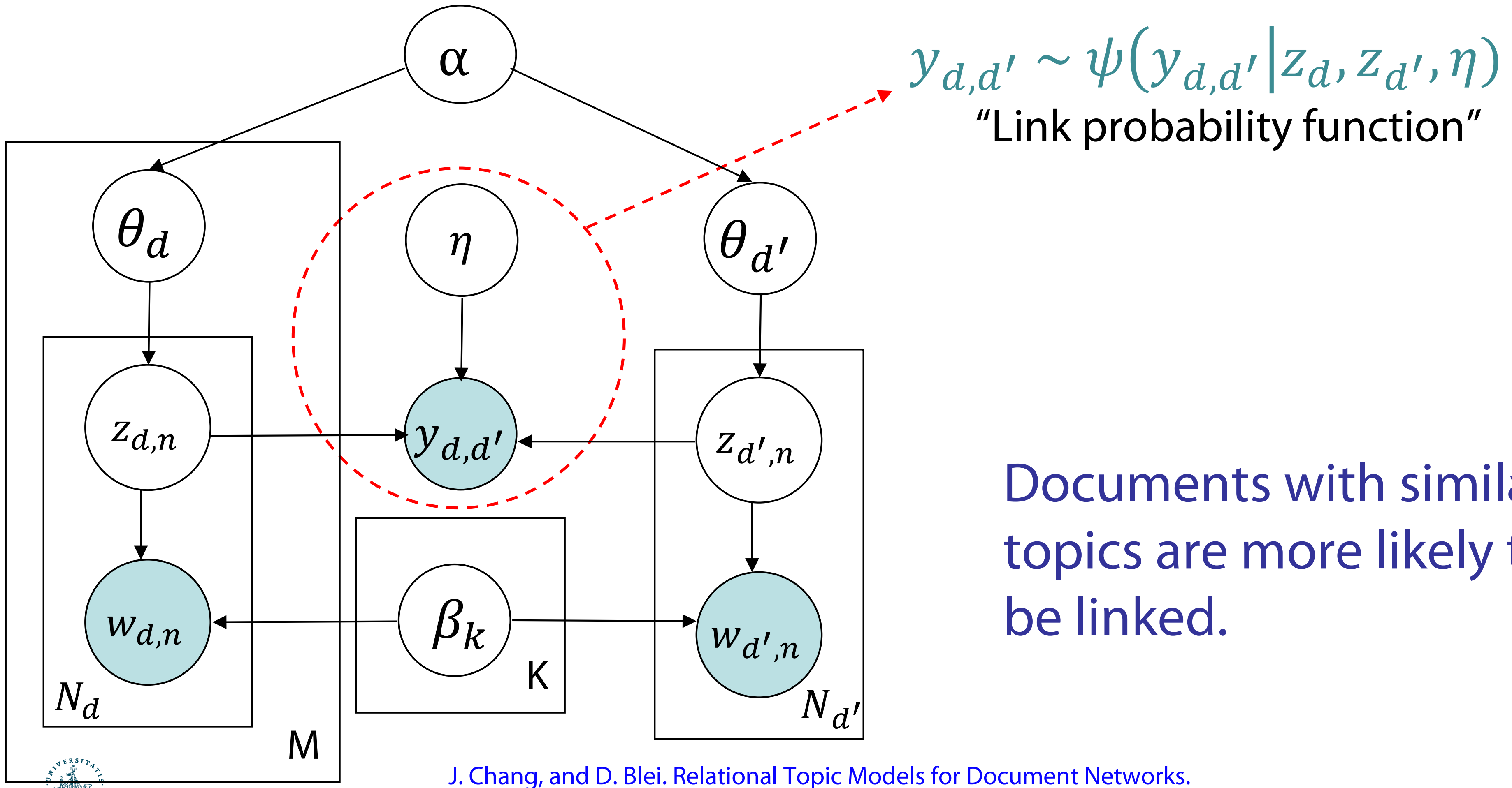
Modeling Citation Influences

- Predictive Performance



Relational Topic Model (RTM) [ChangBlei 2009]

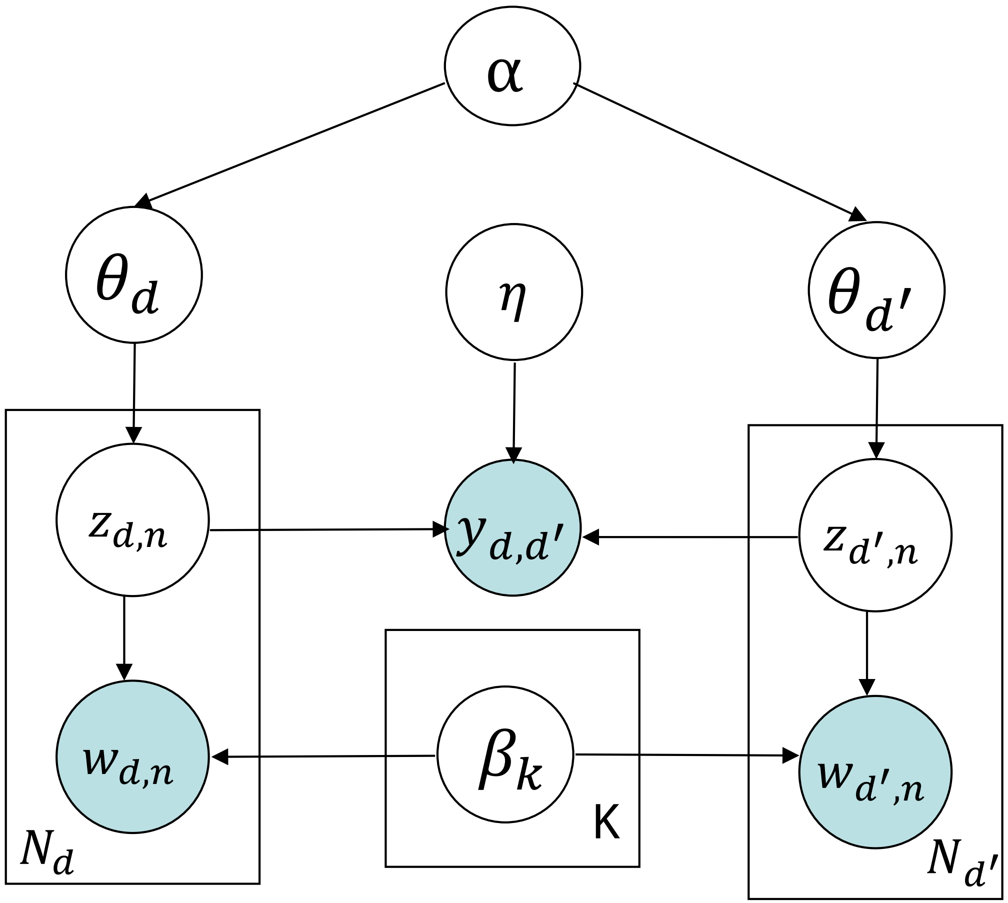
- Same setup as LDA, except now we have observed network information across documents



Documents with similar topics are more likely to be linked.

J. Chang, and D. Blei. Relational Topic Models for Document Networks. AISTATS, volume 5 of JMLR Proceedings, page 81-88. JMLR.org, **2009**.

Relational Topic Model (RTM) [ChangBlei 2009]



- For each document d
  - Draw topic proportions $\theta_d|\alpha \sim Dir(\alpha)$
  - For each word $w_{d,n}$
    - Draw assignment $z_{d,n}|\theta_d \sim Mult(\theta_d)$
    - Draw word $w_{d,n}|z_{d,n}, \beta_{1:K} \sim Mult(\beta_{z_{d,n}})$
  - For each pair of documents d, d'
    - Draw binary link indicator $y|z_d, z_{d'} \sim \psi(\cdot\, |z_d, z_{d'})$

Collapsed Gibbs Sampling for RTM

- Conditional distribution of each z:

$$P\left(z_{d,n} = k \middle| z^{\neg d,n}, \cdot\right) \propto \left(N_{dk}^{\neg d,n} + \alpha\right) \frac{N_{kw}^{\neg d,n} + \beta}{N_k^{\neg d,n} + W\beta}$$ ← LDA term

$$\prod_{d' \neq d : y_{d,d'} = 1} \psi_e\left(y_{d,d'} = 1 \middle| \mathbf{z}_d, \mathbf{z}_{d'}, \eta\right)$$ ← "Edge" term

$$\prod_{d' \neq d : y_{d,d'} = 0} \psi_e\left(y_{d,d'} = 0 \middle| \mathbf{z}_d, \mathbf{z}_{d'}, \eta\right)$$ ← "Non-edge" term

- Using the exponential link probability function, it is computationally efficient to calculate the "edge" term.
- It is **very costly** to compute the "non-edge" term exactly.

Approximating the Non-edges

1. Assume non-edges are “missing” and ignore the term entirely (Chang/Blei)

2. Make the following fast approximation:

$$\prod_i \big(1 - \exp(c_i)\big) \approx (1 - \exp(\bar{c}_i))^N, \text{where } \bar{c}_i = \frac{1}{N}\sum_i c_i$$

3. Subsample non-edges and exactly calculate the term over subset.

4. Subsample non-edges but instead of recalculating statistics for every $z_{d,n}$ *token*, calculate statistics *once per document* and <u>*cache*</u> them over each Gibbs sweep.

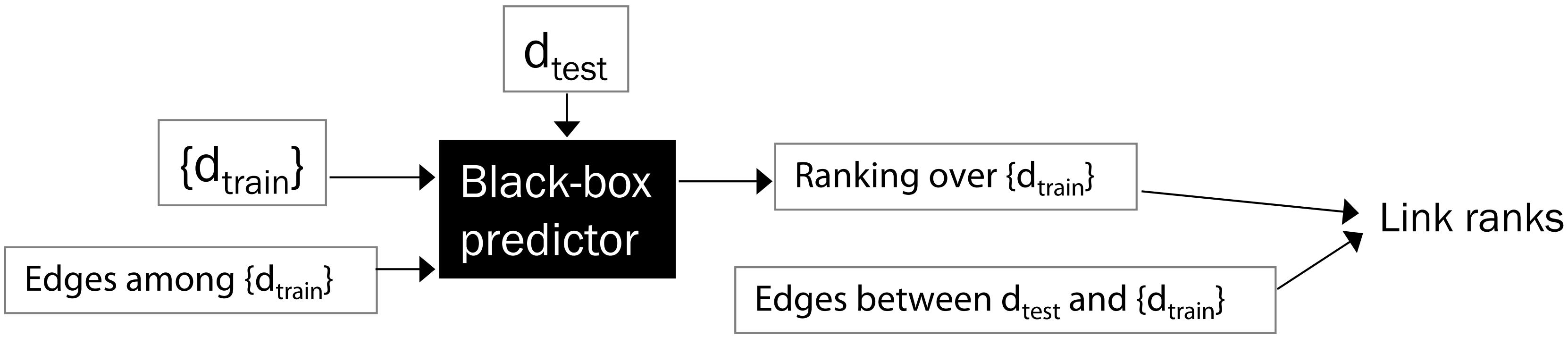
Document networks

	# Docs	# Links	Ave. Doc-Length	Vocab-Size	Link Semantics
CORA	4,000	17,000	1,200	60,000	Paper citation (undirected)
Netflix Movies	10,000	43,000	640	38,000	Common actor/director
Enron (Undirected)	1,000	16,000	7,000	55,000	Communication between person i and person j
Enron (Directed)	2,000	21,000	3,500	55,000	Email from person i to person j

UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR INFORMATIONSSYSTEME

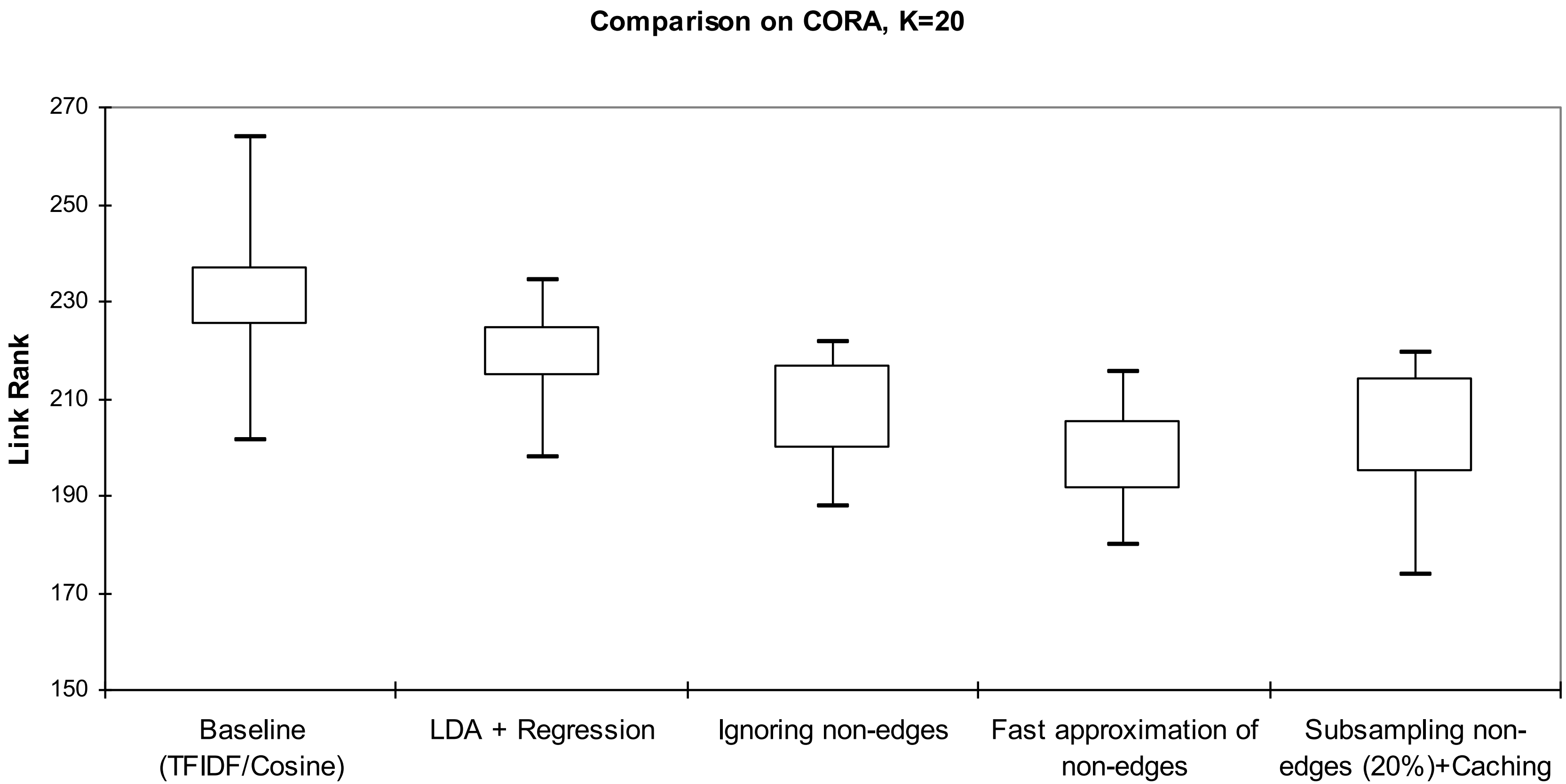
Link Rank

- "Link rank" on held-out data as evaluation metric
  - Lower is better



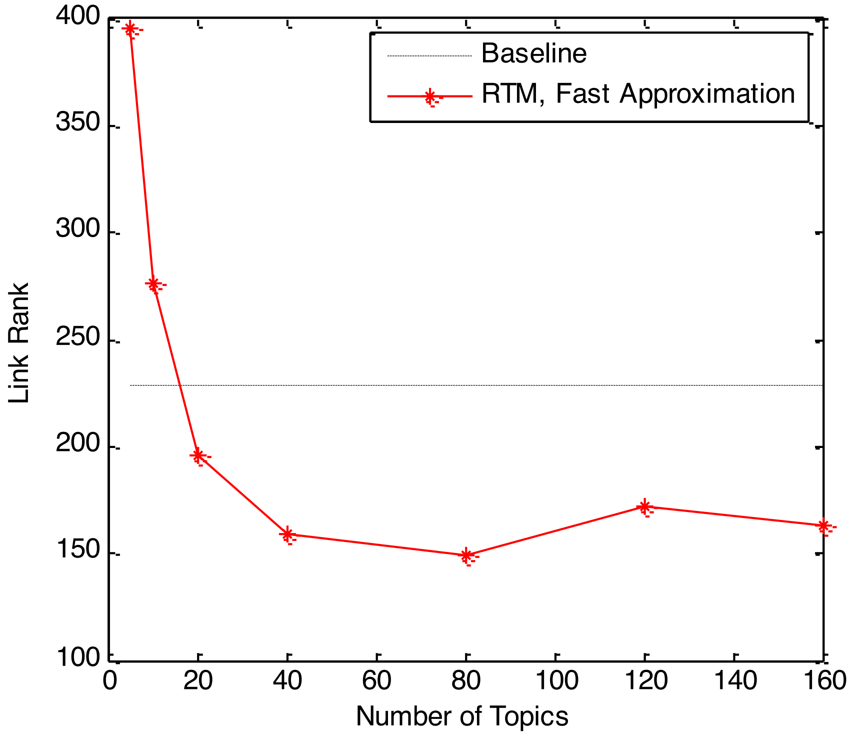
- How to compute link rank (simplified):
  1. Train model with $\{d_{train}\}$.
  2. Given the model, calculate probability that d_{test} would link to each d_{train}. Rank $\{d_{train}\}$ according to these probabilities.
  3. For each observed link between d_{test} and $\{d_{train}\}$, find the "rank", and average all these ranks to obtain the "link rank"

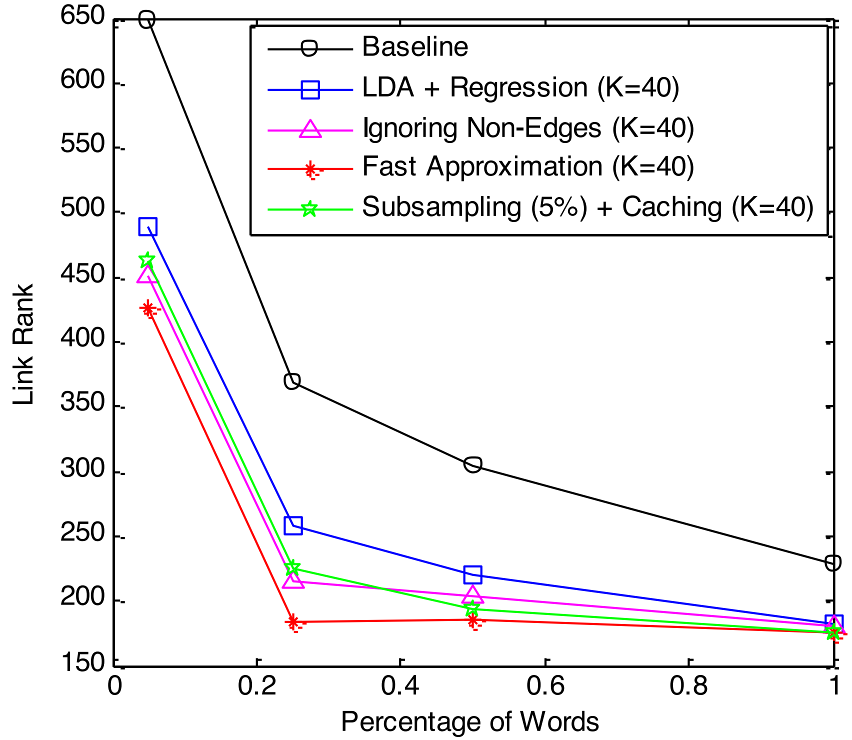
Results on CORA data



8-fold cross-validation. Random guessing gives link rank = 2000.

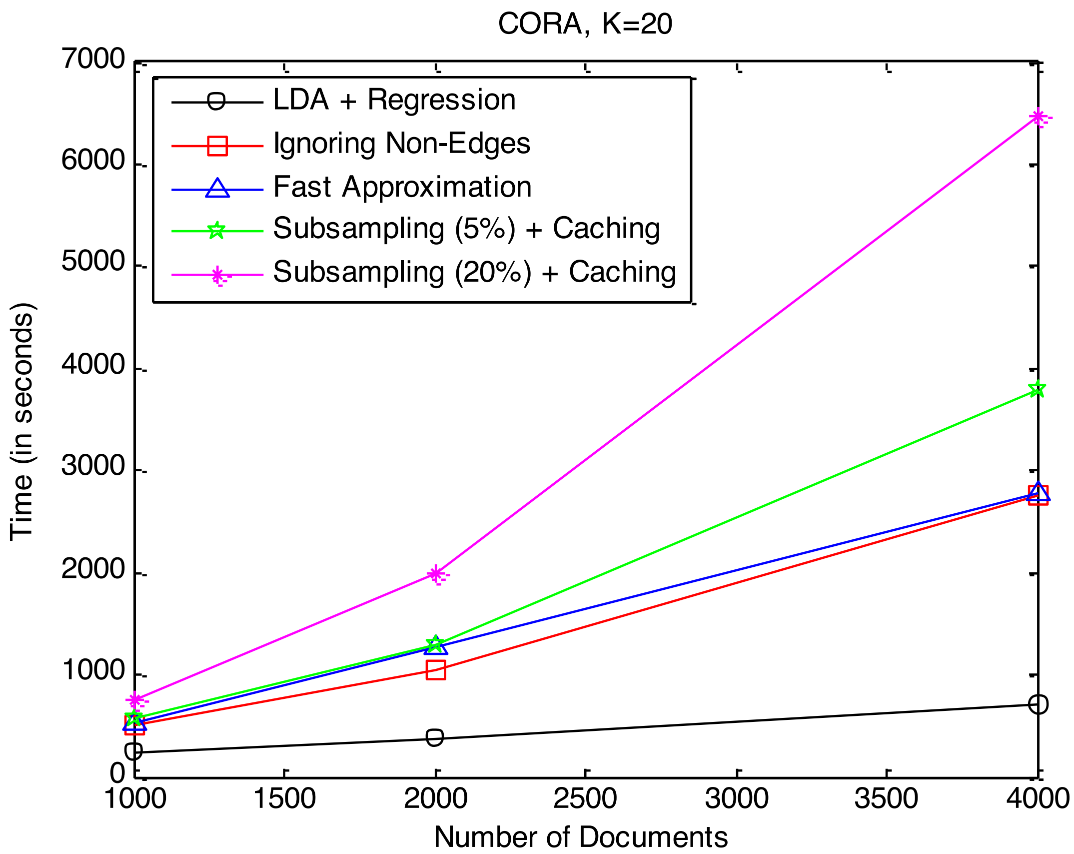
Results on CORA data





- Model does better with more topics
- Model does better with more words in each document

Timing Results on CORA



"Subsampling (20%) without caching" not shown since it takes 62,000 seconds for D=1000 and 3,720,150 seconds for D=4000

Conclusion

- Relational topic modeling provides a useful start for combining text and network data in a single statistical framework

- RTM can improve over simpler approaches for link prediction

- Opportunities for future work:
  - Faster algorithms for larger data sets
  - Better understanding of non-edge modeling
  - Extended models