Intelligent Agents
Topic Analysis: pLSI and LDA

Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Summary and Agenda

- IR Agents
  - Task/goal: Information retrieval
  - Agents visits document repositories and returns doc recommendations
  - Means:
    - Vector space (bag-of-words)
      - Dimension reduction (LSI)
    - Probability based retrieval (binary)
      - Language models

Non-standard Databases and Data Mining

- Today: Language models with dimension reduction
  - Probabilistic Latent Semantic Indexing (pLSI)
  - Latent Dirichlet Allocation (LDA): Topic Models
- Soon: What agents can take with them
  - What agents can leave at the repository (win-win)

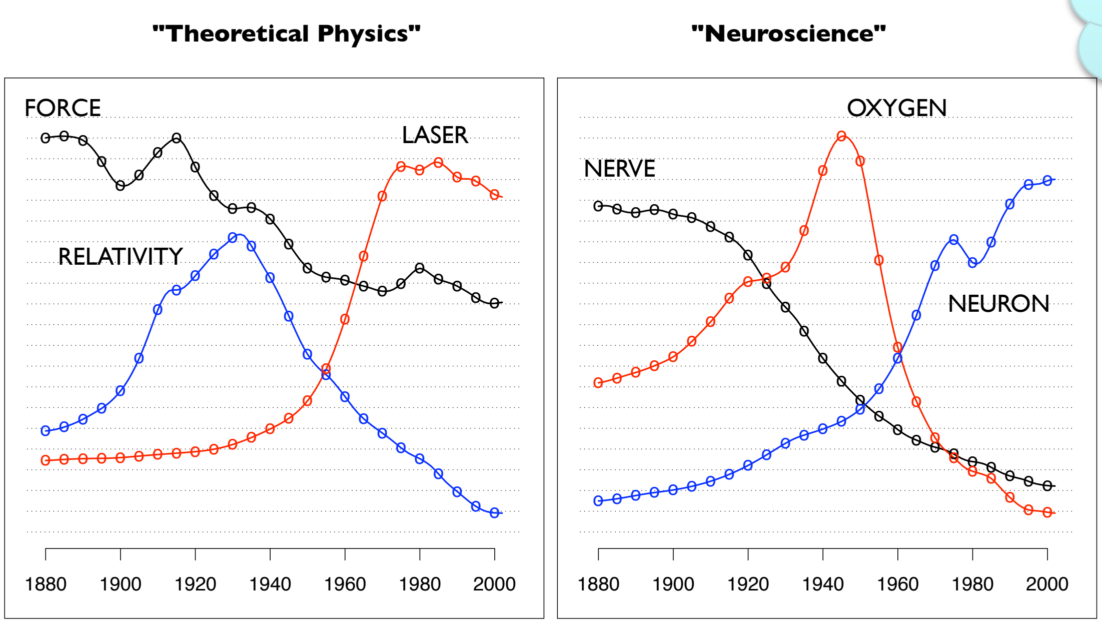
Acknowledgments

Ramesh M. Nallapati
presentation on
Generative Topic Models for Community Analysis
&
Sina Miran
presentation on
Probabilistic Latent Semantic Indexing (PLSI)
&
David M. Blei
presentation on
Probabilistic Topic Models

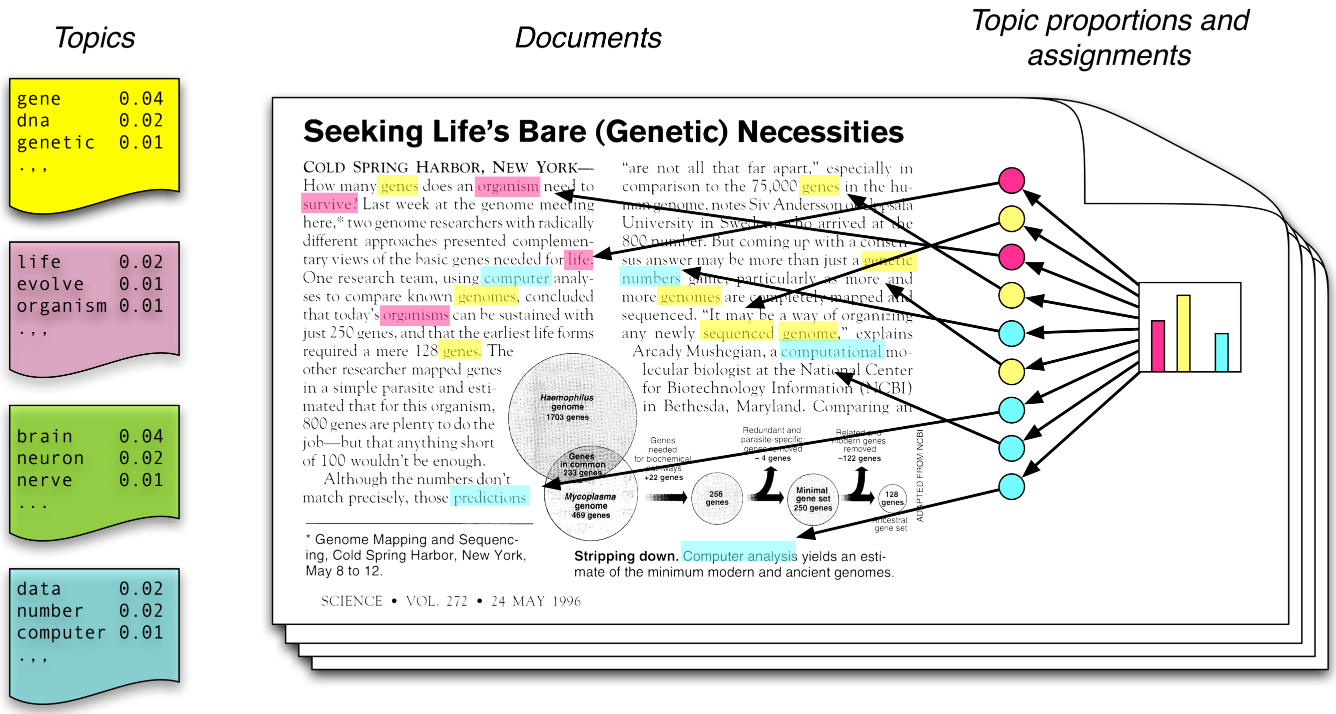
Topic Models

- 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

Just for illustration purposes

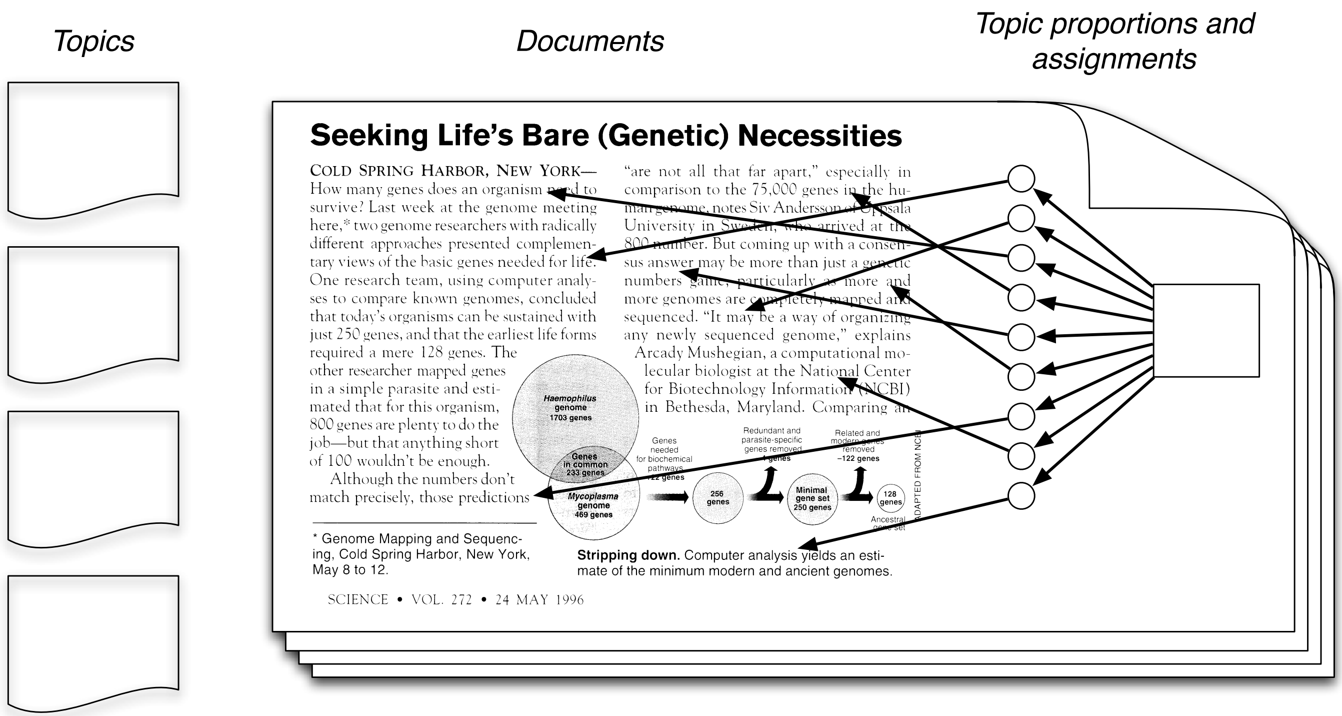


Topic Modeling Scenario



- 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

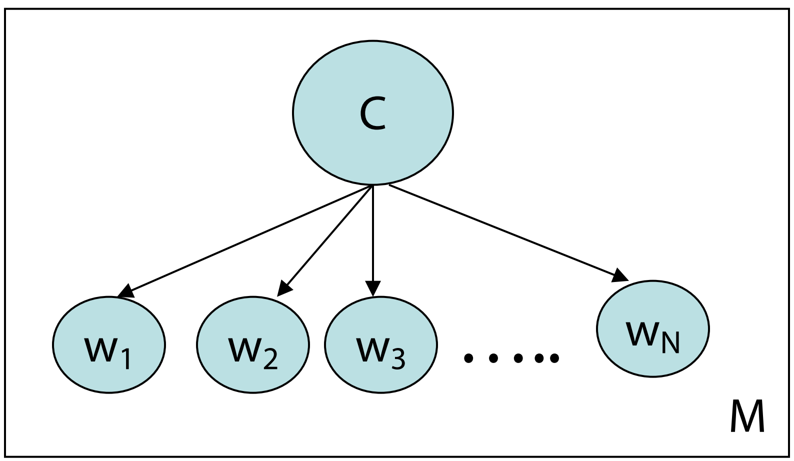
Topic Modeling Scenario

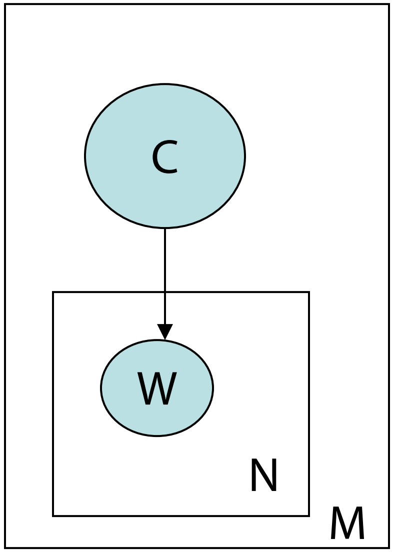


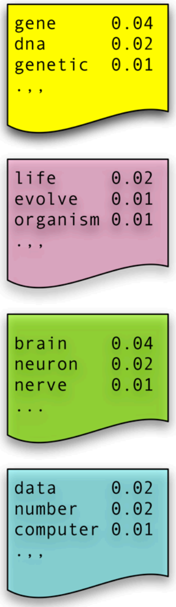
- In reality, we only observe the documents
- The other structures are hidden variables
- Topic modeling algorithms infer these variables from data

Plate Notation

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







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

Generative vs. Descriptive Models

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

Forward Sampling No Evidence

Input: Bayesian network

$X = \{X_1, \ldots, X_N\}$, N- #nodes, T - # samples

Output: T samples

Process nodes in topological order – first process the ancestors of a node, then the node itself:

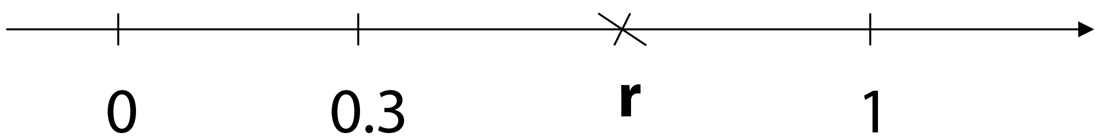
1. For t = 0 to T
2. For i = 0 to N
3. $X_i \leftarrow$ sample x_i^t from $P(x_i \mid pa_i)$

M. Henrion, "Propagating uncertainty in Bayesian networks by probabilistic logic sampling", Uncertainty in AI, pp. = 149-163, **1988**

Sampling A Value

What does it mean to sample x_i^t from $P(X_i \mid pa_i)$?

- Assume $D(X_i)=\{0,1\}$
- Assume $P(X_i \mid pa_i) = (0.3, 0.7)$

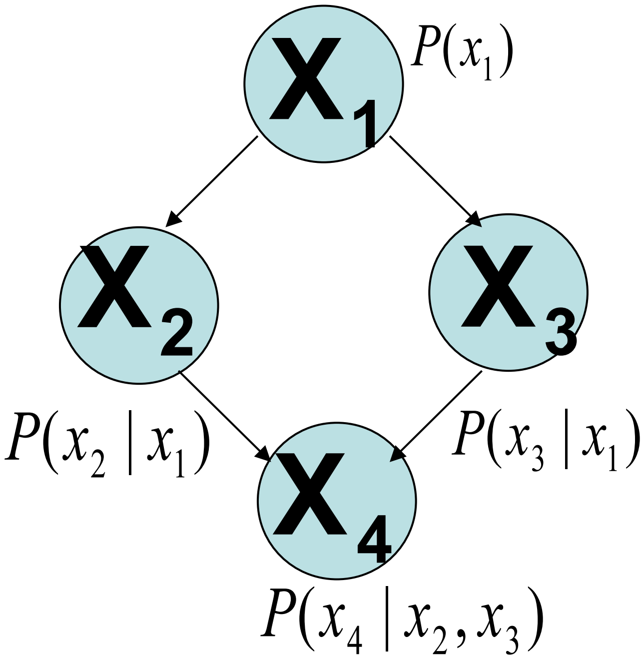


- Draw a random number **r** from [0,1]

  If **r** falls in [0,0.3], set $X_i = 0$

  If **r** falls in [0.3,1], set $X_i=1$

Forward Sampling (Example)



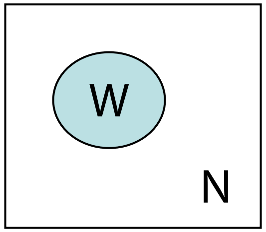
$\text{Evidence} : X_3 = 0$

// generate sample k

1. Sample x_1 from $P(x_1)$
2. Sample x_2 from $P(x_2 \mid x_1)$
3. Sample x_3 from $P(x_3 \mid x_1)$
4. If $x_3 \neq 0$, reject sample and start from 1, otherwise
5. sample x_4 from $P(x_4 \mid x_{2,} x_3)$

Rejection sampling
(rather inefficient)

Earlier Topic Models: Topics Known



- Unigram
  - No context information

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

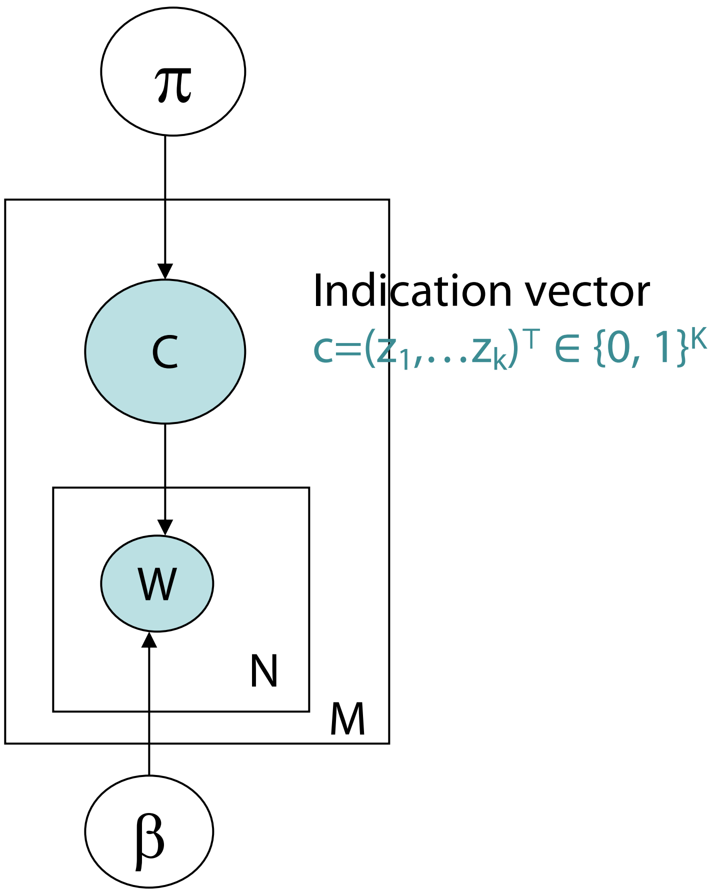
```
fifth, an, of, futures, the, an, incorporated, a,
a, the, inflation, most, dollars, quarter, in, is,
mass

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

that, or, limited, the
```

Automatically generated sentences from a unigram model

Multinomial Naïve Bayes



- How to specify Domain(C)?
  - Domain(C) = {1, 2, …, k} or
  - Domain(C) = $\{0, 1\}^k$
- How to specify $P(c_d)$?
  - Define a table

	P(C)
1	p_1
…	…
K	p_K

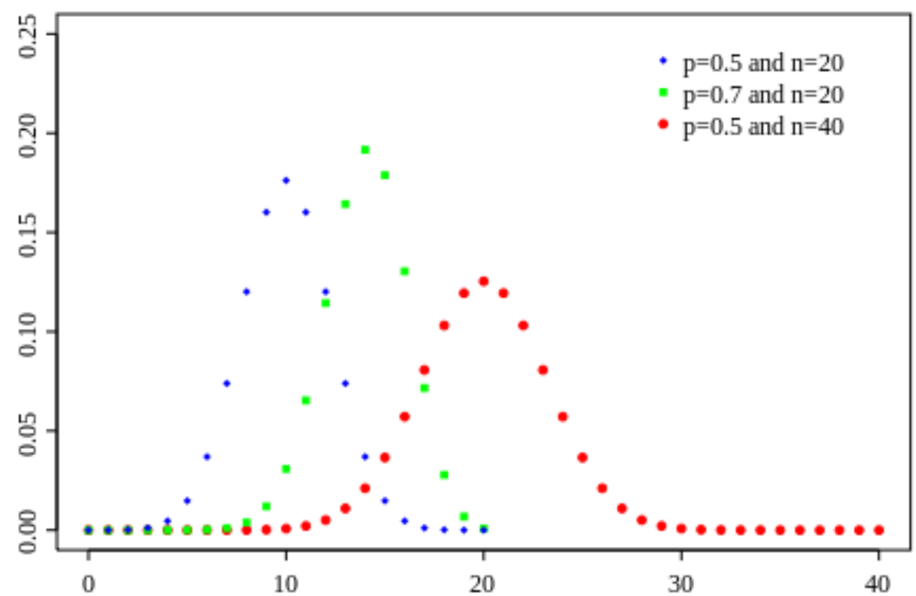
  - or use parameterized distribution $\pi = (p_1, \ldots, p_K)$
    - P(C=c|π)= $\prod_{k=1}^{K} \pi_k{}^{z_k}$

Recap: Binomial Distribution

- Describes the number of successes in a series of independent trials with two possible outcomes “success” or “no success”
- n = #trials
  p = #successful trials / n
- Description of frequency of having exactly k successful trials as a function

$$B_{p,n}(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

$B_{p,n}(k)$



k

- Es gilt: $$\sum_{i=0} B_{p,n}(i) = 1$$
- If n=1: Bernoulli distribution

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

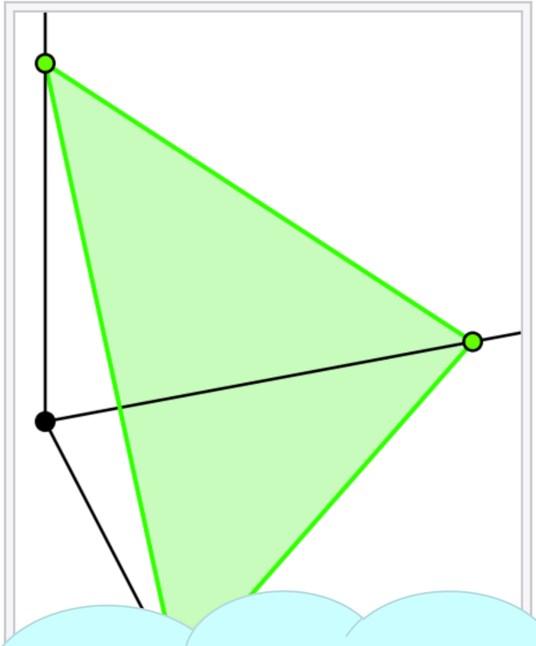
Multinomial Distribution Mult(n | π)

- Generalization of binomial distribution
  - K possible outcomes instead of 2
  - Probability mass function
    - n = number of trials
    - x_j ∈ {0, 1} a count for how often class j occurs $\sum_{i=1}^{k} x_i = n$
    - p_j = probability of class j occurring

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

    - Here, the input to $\Gamma(\cdot)$ is a positive integer, so $\Gamma(n) = (n-1)!$
  - If n=1: called categorial distribution ("multinoulli")
    - Often written $Mult(\,.\,; p_1, \dots, p_K)$ or $Mult(\,.\,|\, p_1, \dots, p_K)$
    - Generates a one-hot vector

Sampling



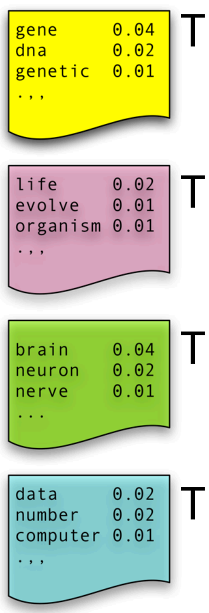
- A variable value a can be sampled from a discrete distribution $\pi = (p_1, \ldots, p_K)$
- Notation: a ~ Mult(. | π)

One-hot vector to be generated with position probability of indicator controlled by π

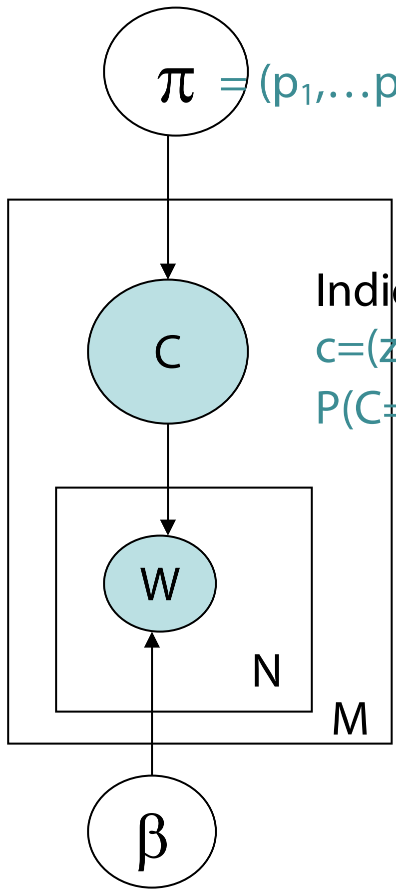
- Generate random number x from (0, 1]
- Find $l \in \{1, 2, \ldots, k\}$ such that $\sum_{i=1}^{l-1} p_i < x \leq \sum_{i=1}^{l} p_i$
- Return $(z_1, \ldots, z_K)$ such that $z_l = 1$ and $z_i = 0$ für $i \neq l$

Multinomial with Matrices

- Let β be a $K \times V$ matrix (V vocabulary size), each row denotes a word distribution of a topic
- Select row k before applying multinomial:
  - Notation: $\mathrm{Mult}(\,.\,|\,\beta_k)$ or $\mathrm{Mult}(\,.\,|\,\beta, k)$ or $\mathrm{Mult}(\,.\,|\,k, \beta)$



Mixture of Unigrams: Known Topics



$\pi = (p_1,\ldots p_k)^T$

Indication vector
$c=(z_1,\ldots z_k)^T \in \{0, 1\}^K$
$P(C=c|\pi)= \prod_{k=1}^{K} \pi_k{}^{zk}$

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

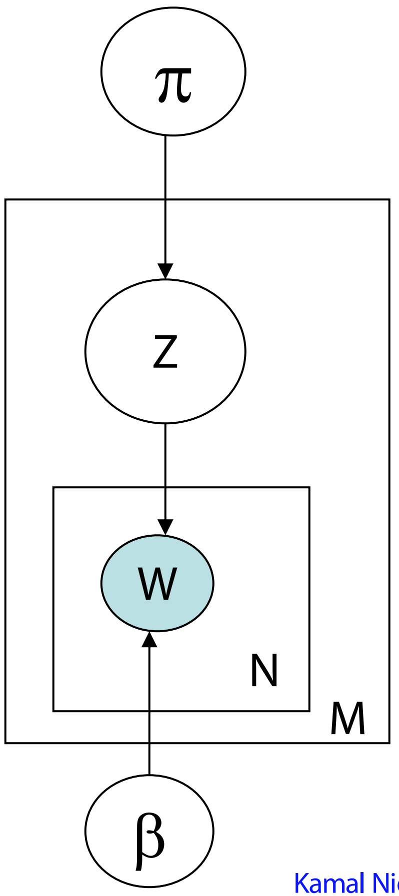
$$\prod_{d=1}^{M} P\left(w_1, \ldots, w_{N_d}, c_d \mid \beta, \pi\right)$$

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

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

multinomial

Mixture of Unigrams: Unknown Topics



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

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

  - Sum over topics (K = number of topics)

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

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

Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun & Tom Mitchell, Learning to Classify Text from Labeled and Unlabeled Documents, Proc. AAAI 98, Pages 792–799, **1998**.

Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun & Tom Mitchell Text Classification from Labeled and Unlabeled Documents using EM Journal of Machine Learning volume 39, pages 103–134, **2000**.

Mixture of Unigrams: Learning

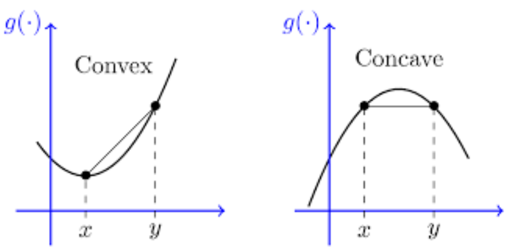
- Learn parameters π and β

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

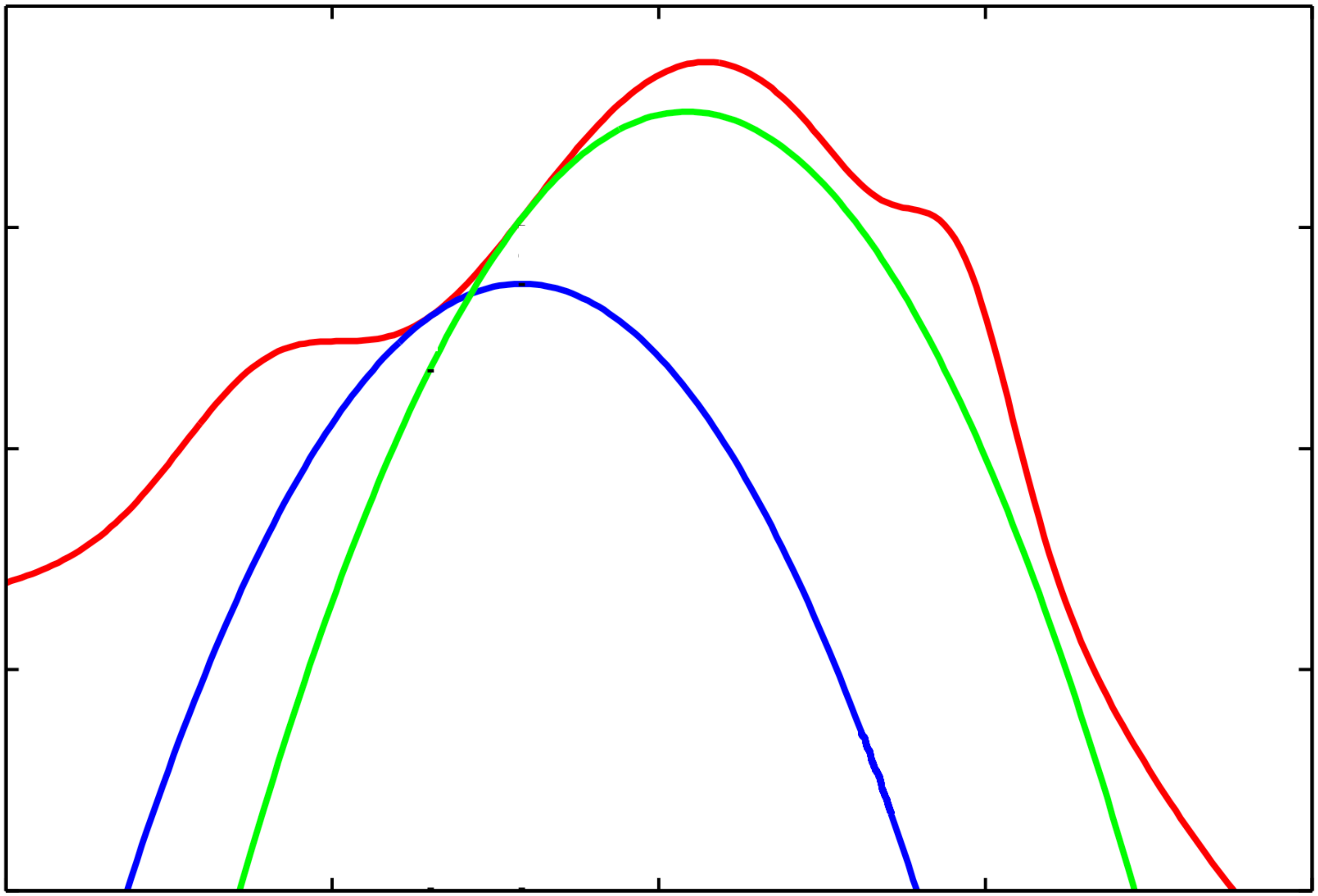
- Use likelihood

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

- 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

$\pi_{z_k} := P(z_k|\pi)$
$\beta_{z_k,w_i} := P(w_i|\beta, z_k)$

- 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(**a**·**b**) ≥ **a** · log **b**

a, **b** distribution vectors

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

H(γ)

Entropy of γ_{di}
Sometimes called I(.)

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

Mixture of Unigrams: Learning

- Optimization problem for each document

$$argmax_{\beta\pi} \sum_{k=1}^{K} \left(\gamma_k \log(\pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k,w_i}) \right) + H(\gamma)$$

Convex?
Concave?

- We have introduced a new latent variable γ to approximate the original functional to be optimized
- Each document is assumed to be associated with a latent variable $\gamma \in [0,1]^K$, $\Sigma_k \gamma_k = 1$ independent of other random variables
- Can be seen as a class in the new space $\gamma_k, \pi_{z_k}, \beta_{z_k,w_i}$

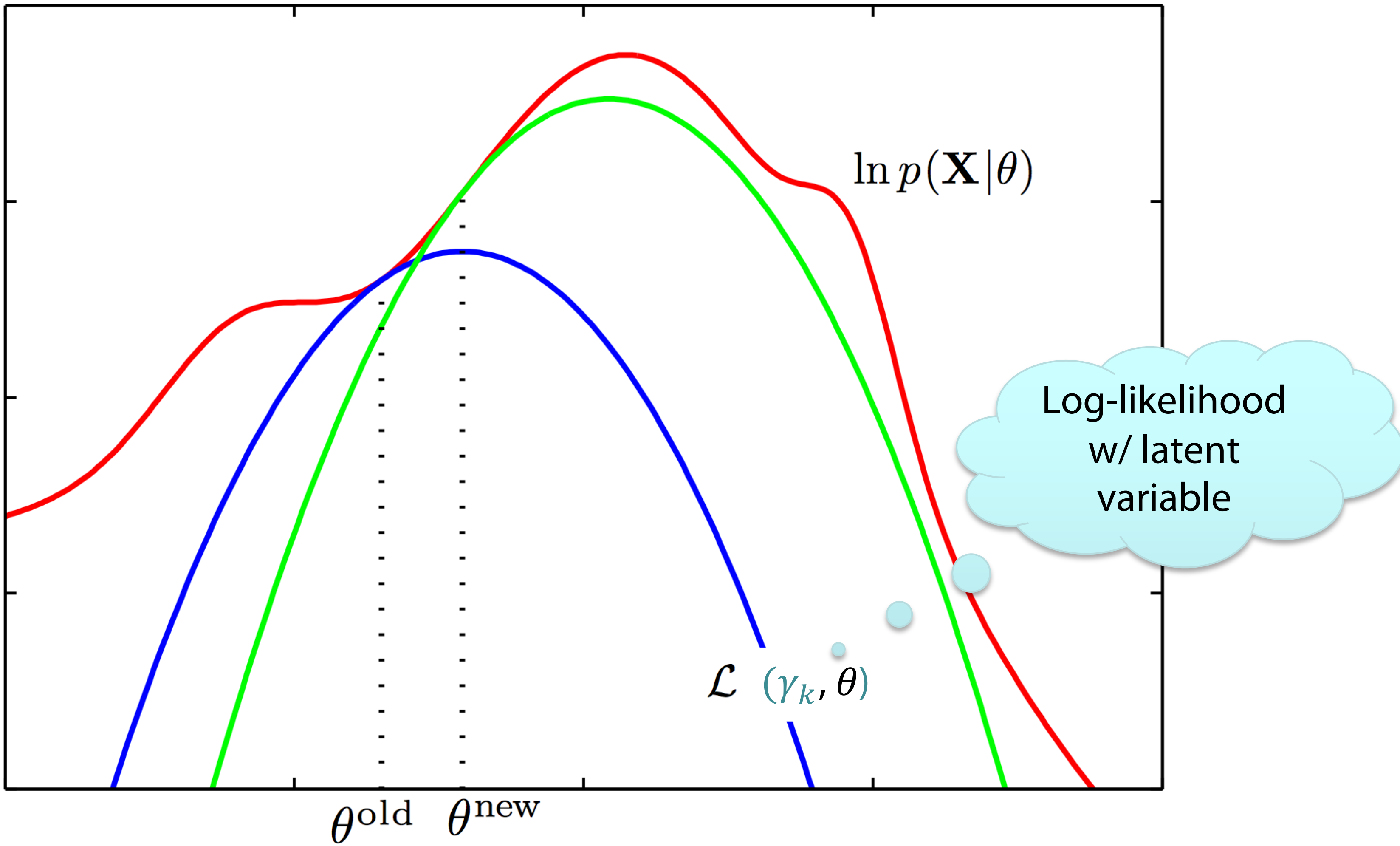
Mixture of Unigrams: Learning

- New optimization problem:

$$argmax_{\beta\pi} \sum_{k=1}^{K} \left(\gamma_k \log(\pi_{z_k} \prod_{i=1}^{N_d} \beta_{z_k,w_i}) \right) + H(\gamma)$$

- Solution: Expectation Maximization
  - Iterative algorithm to find local optimum
  - Guess values of $\gamma_k, \pi_{z_k}, \beta_{z_k,w_i}$
  - Compute $\gamma_k = P(\gamma_k | \pi_{z_k}, \beta_{z_k,w_i})$ according to model
  - Use Maximum-likelihood estimation of to optimize $\pi_{z_k}, \beta_{z_k,w_i}$
  - Until no further improvement
- Guaranteed to maximize a lower bound on the log-likelihood of the observed data
- Use $\pi_{z_k}, \beta_{z_k,w_i}$ to estimate $P(z_k|\pi), P(w_i|\beta, z_k)$, respectively

Graphical Idea of the EM Algorithm



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

Mixture of Unigrams: Learning

- EM solution
  - E step (compute $\gamma_k = P(\gamma_k | \pi_{z_k}, \beta_{z_k, w_i})$)

$$\gamma_k^{(t+1)} = \frac{\gamma_k^{(t)} \pi_{z_k}^{(t)} \prod_{i=1}^{N_d} \beta_{z_k, w_i}^{(t)}}{\sum_{j=1}^{K} \gamma_{z_{dj}}^{(t)} \pi_{z_j}^{(t)} \prod_{i=1}^{N_d} \beta_{z_j, w_i}^{(t)}}$$

Independence assumption

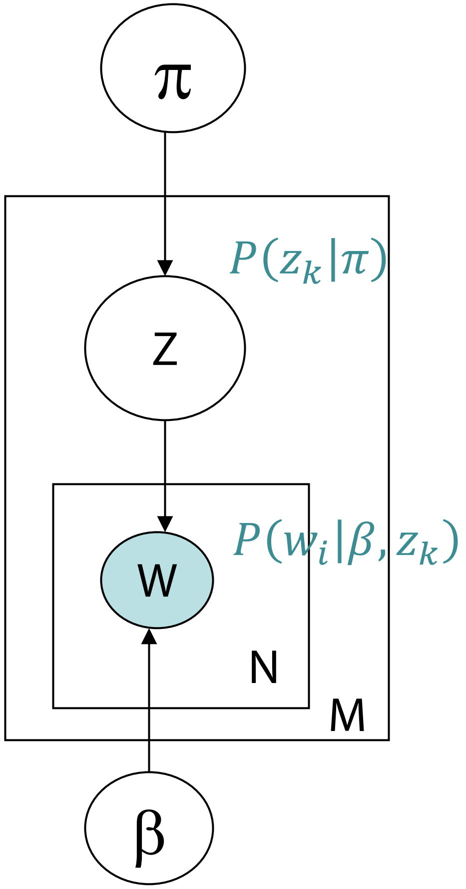
  - M step (maximum likelihood optimization: use frequencies)

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

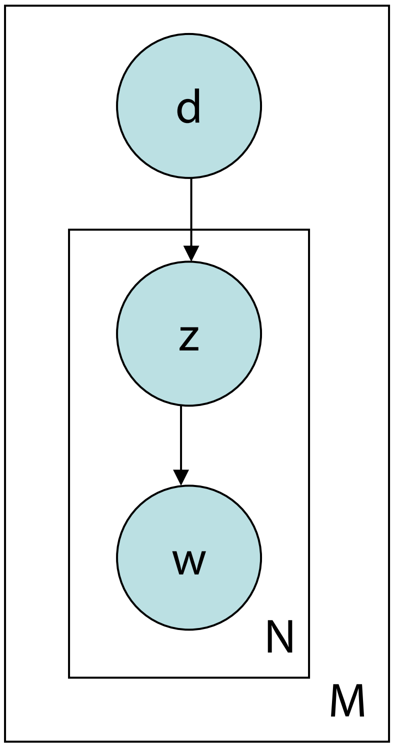
$n(d, w_i)$ number of times word w_i occurs in document d

Back to Topic Modeling Scenario

- Documents are associated with a single topic
- Words do not depend on context
  - Bag-of-words model



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**

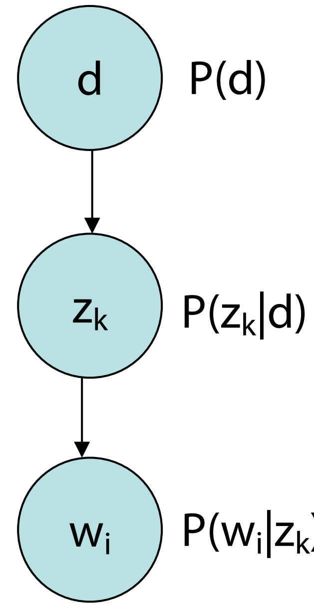
pLSI

- Joint probability for all documents, words

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

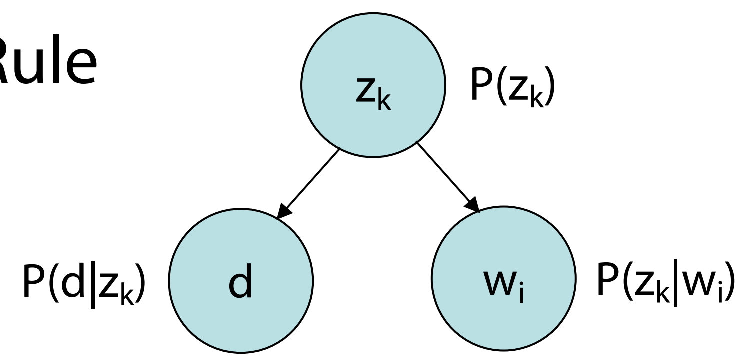
- Distribution for document d, word w_i

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



- 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

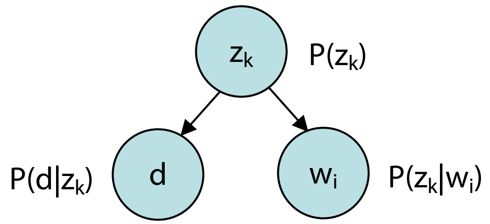
$$L = \sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)\log P(d,w_i) = \sum_{d=1}^{M}\sum_{i=1}^{N_d} n(d,w_i)\log\sum_{k=1}^{K} P(d|z_k)P(z_k)P(w_i|z_k)$$

- Parameters to learn (M step)

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

- (E step)

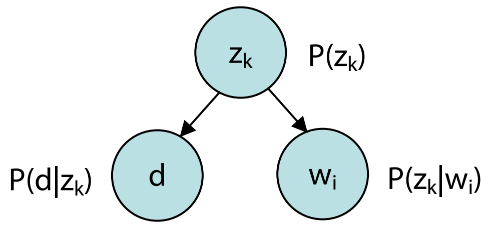
$$P(z_k|d,w_i)$$



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

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

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

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

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

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

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

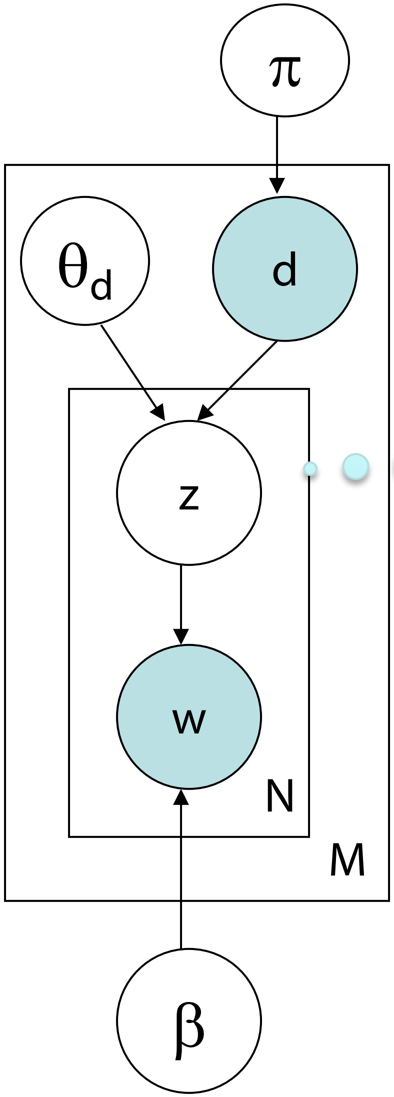
- Difference:
  - LSI: minimize Frobenius (L-2) norm
  - pLSI: log-likelihood of training data

Intelligent Agents
Topic Analysis: pLSI and LDA

Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

pLSI with Multinomials

$$\pi_d := P(d|\pi)$$
$$\beta_{k,w_i} := P(w_i|\beta, z_k)$$
$$\theta_{d,k} := P(z_k|\theta_d)$$



Multiple topics

- 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 $w_i \sim \text{Mult}(\cdot \mid z_i, \beta_k)$

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

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

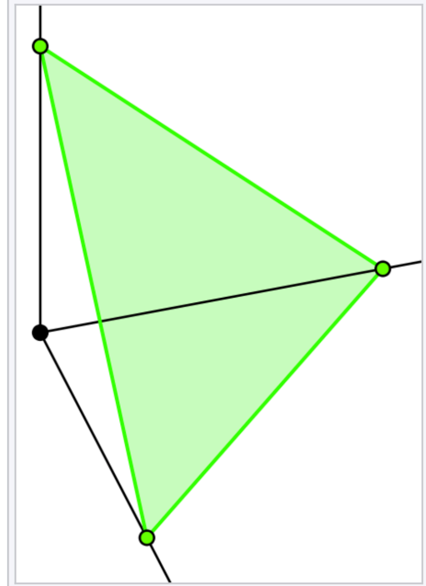
$$= \prod_{d=1}^{M} \pi_d \prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{d,k} \beta_{k,w_i}$$

Maximum-likelihood learning?

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

Prior Distribution for Topic Mixture

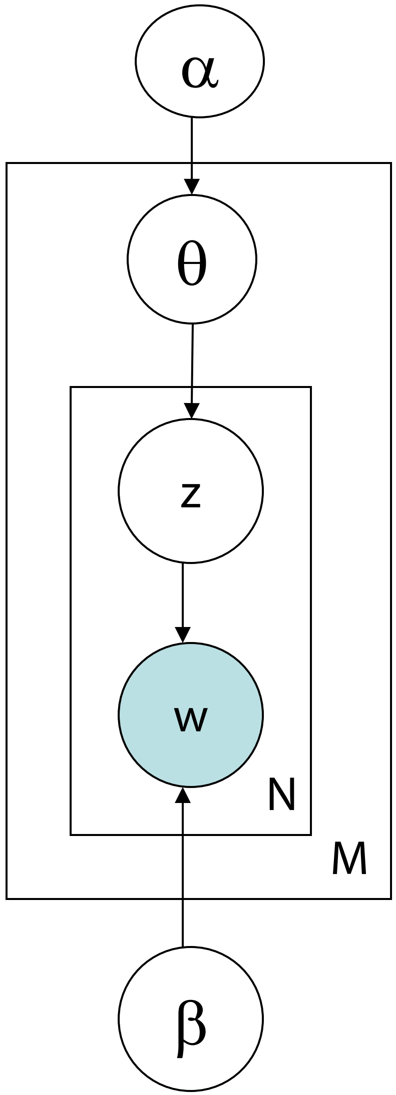
- Goal: topic mixture proportions for each document could drawn from some distribution.
  - Distribution on multinomials (k-tuples of non-negative numbers that sum to one)
- The space of all of these multinomials can be interpreted geometrically as a (k-1)-*simplex*
  - K-1 independent values
  - Simplex = 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



The possible probabilities for the categorical distribution with $k = 3$ are the 2-simplex $p_1 + p_2 + p_3 = 1$, embedded in 3-space.

[Wikipedia]

LDA Model – Parameters



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

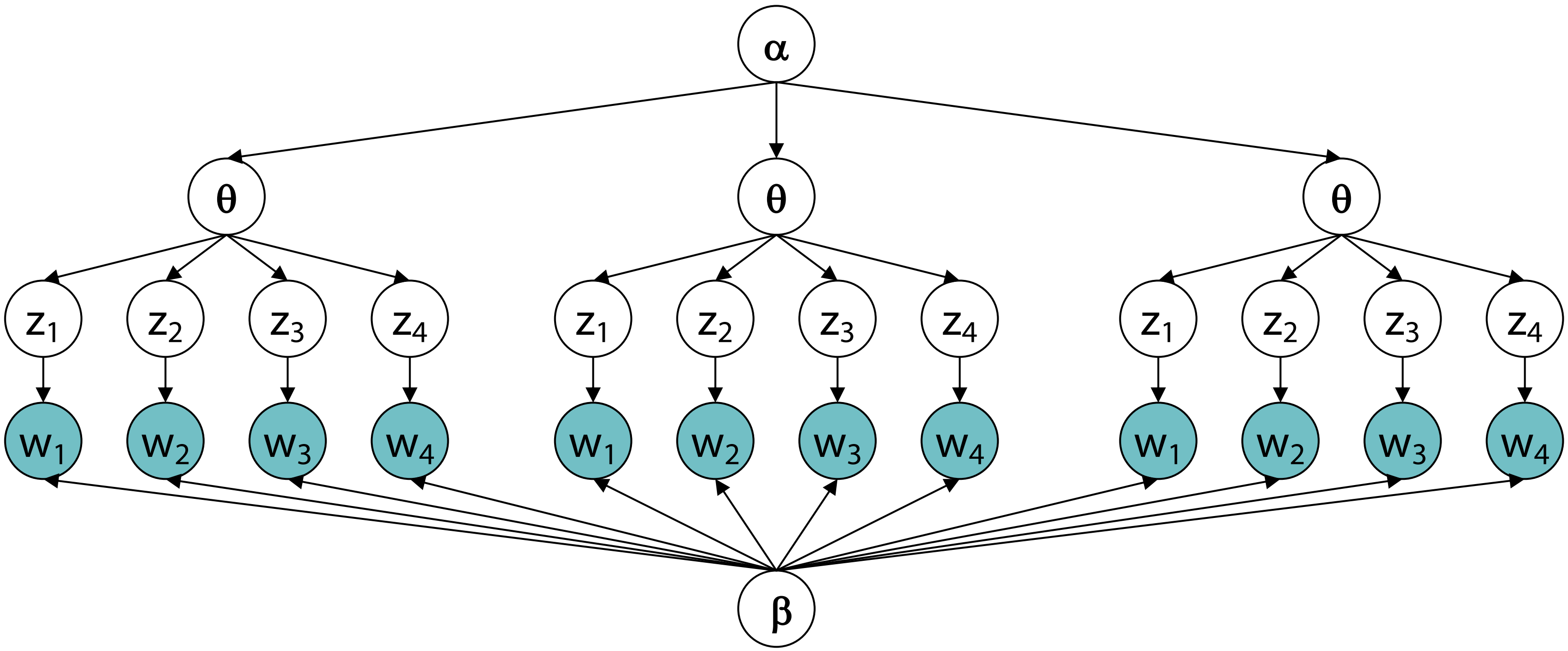
← Per-document topic distribution
(k-dimensional vector of
probabilities summing up to 1)

← 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)

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

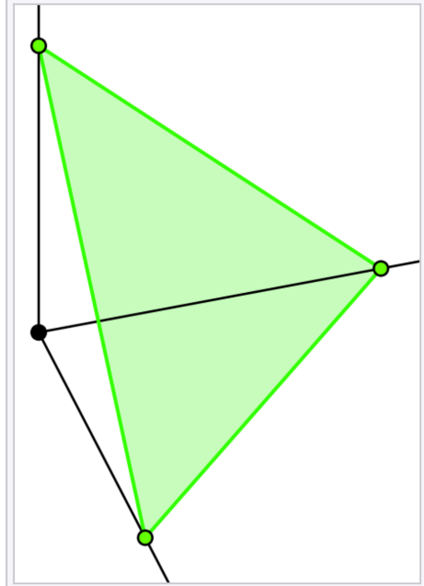
Latent Dirichlet Allocation

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

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

Dirichlet Distributions

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

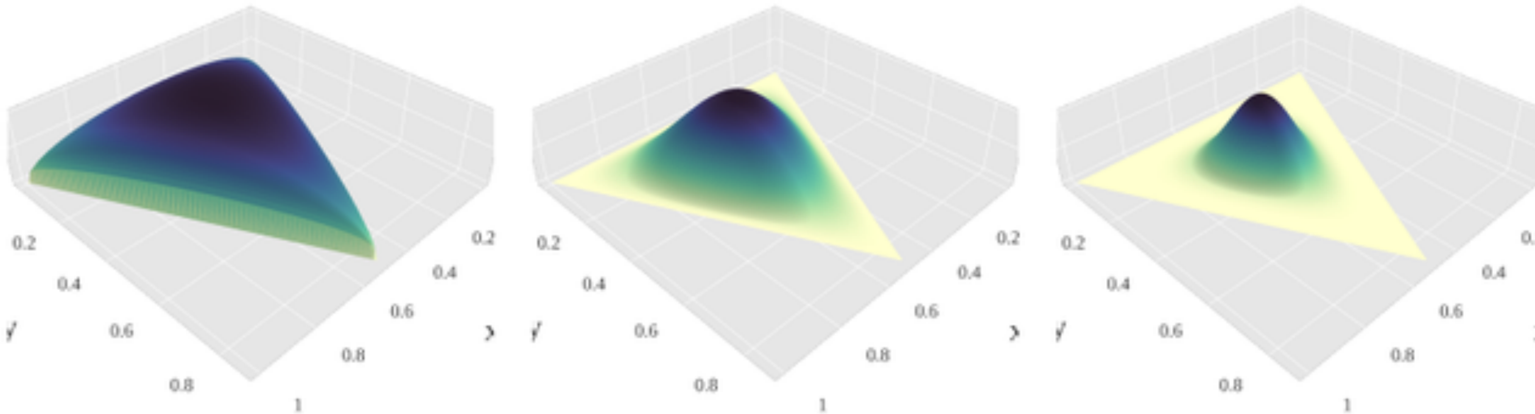


The possible probabilities for the categorical distribution with $k = 3$ are the 2-simplex $p_1 + p_2 + p_3 = 1$, embedded in 3-space.

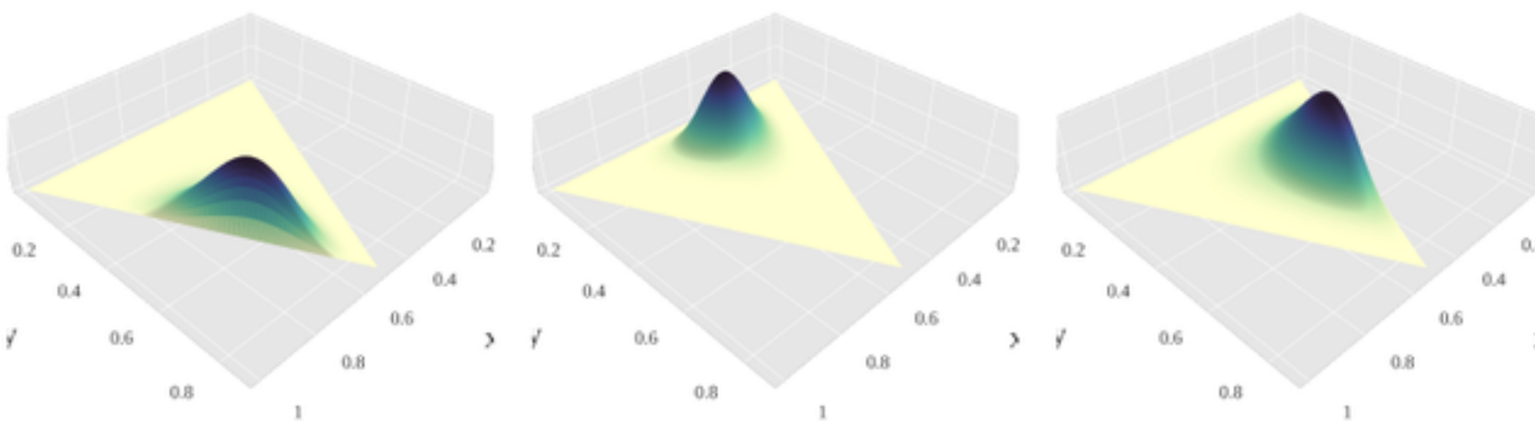
- Defined over a (k-1)-simplex
  - Takes K non-negative arguments which sum to one.
  - Consequently it is a natural distribution to use over multinomial distributions.
- The Dirichlet parameter α_i can be thought of as a prior count of the i^{th} class

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

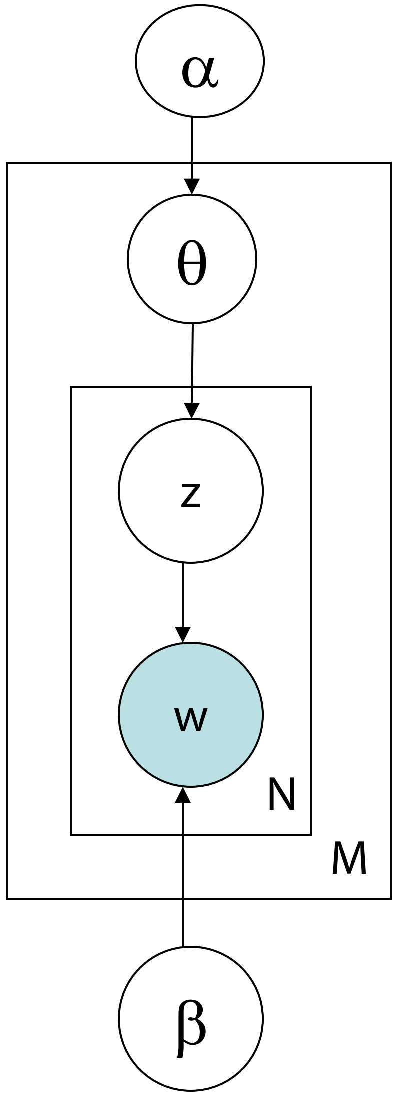
Dirichlet Distribution over a 2-Simplex



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



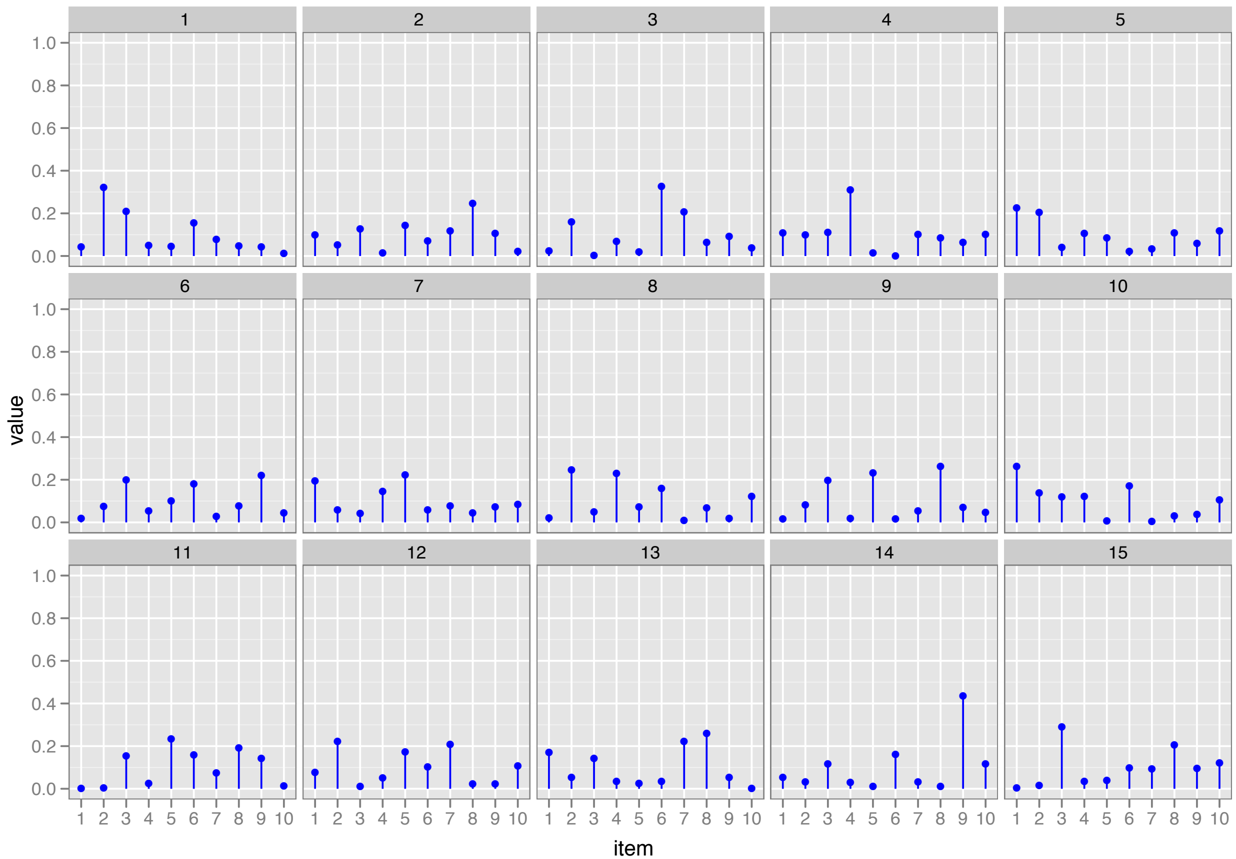
- For each document d,
  - Generate $\theta_d \sim \text{Dirichlet}(\alpha)$
  - For each position $i = 1, \ldots, N_d$
    - Generate a topic $z_i \sim \text{Mult}(\cdot \mid \theta_d)$
    - Generate a word $w_i \sim \text{Mult}(\cdot \mid z_i, \beta)$

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

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

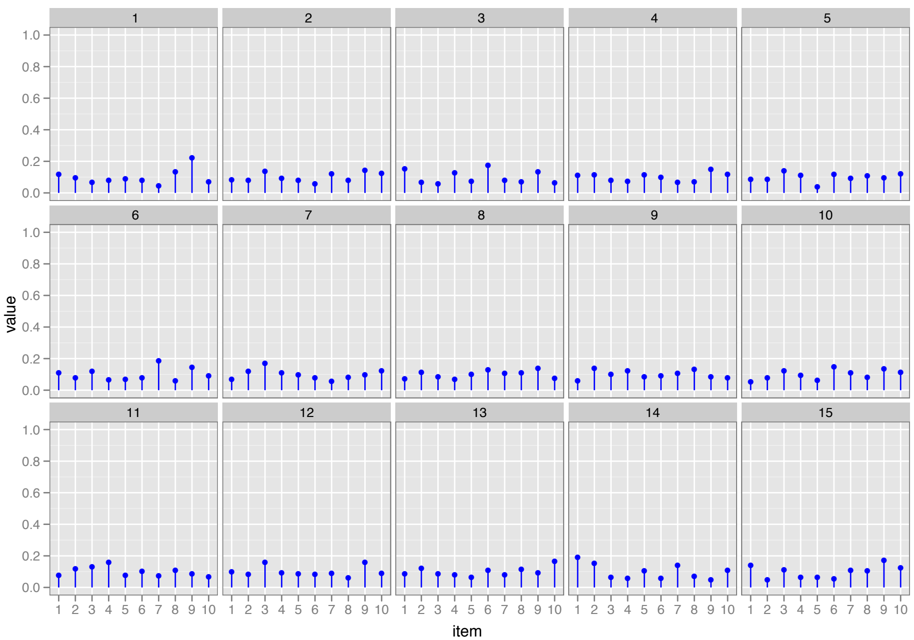
Corpus-level Parameter α (uniform: $\alpha_{i} = \alpha_{j}$)

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

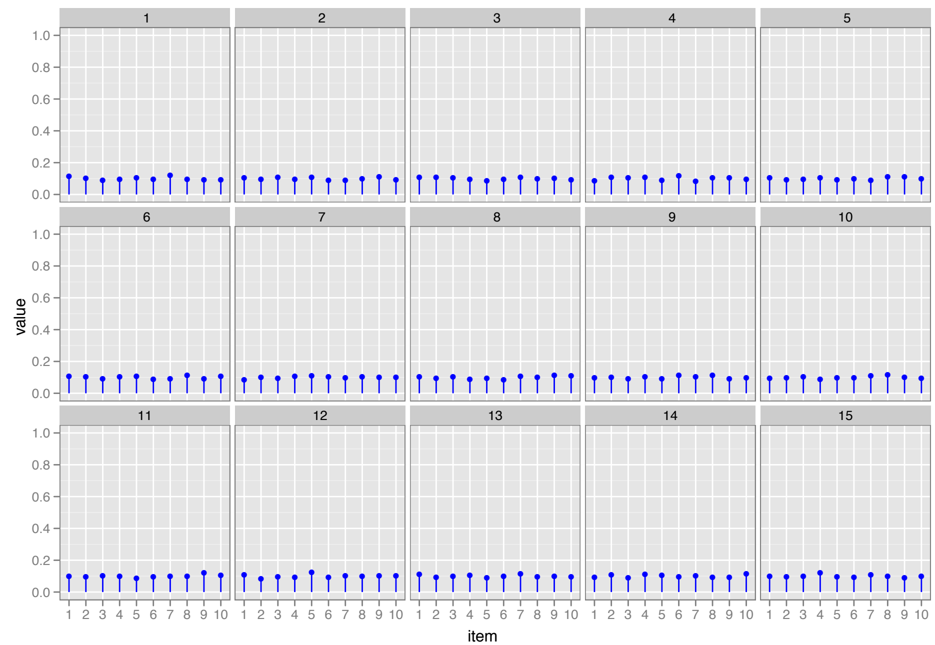


Corpus-level Parameter α

- $\alpha = 10$

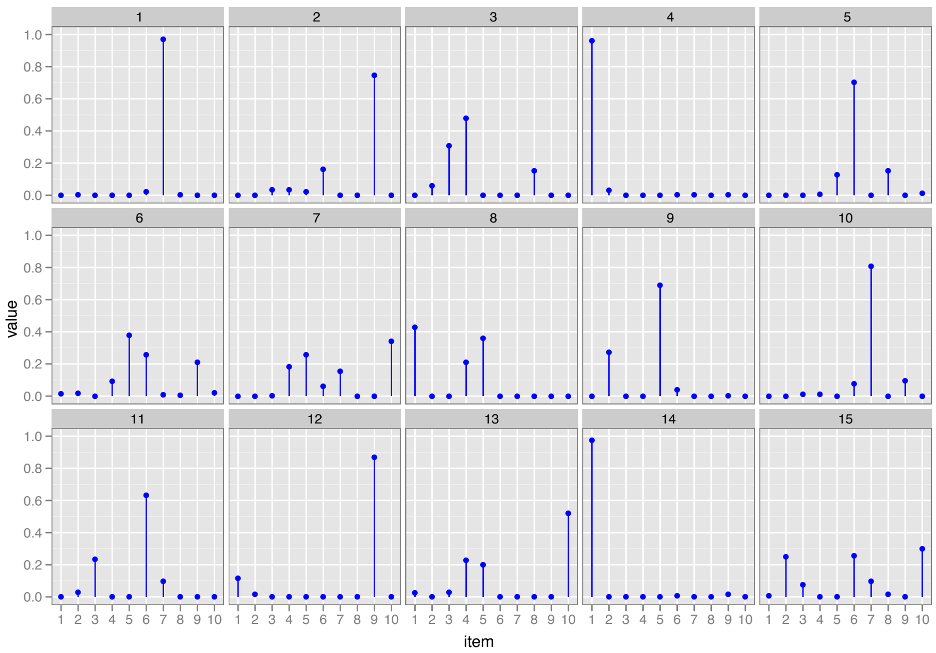


- $\alpha = 100$

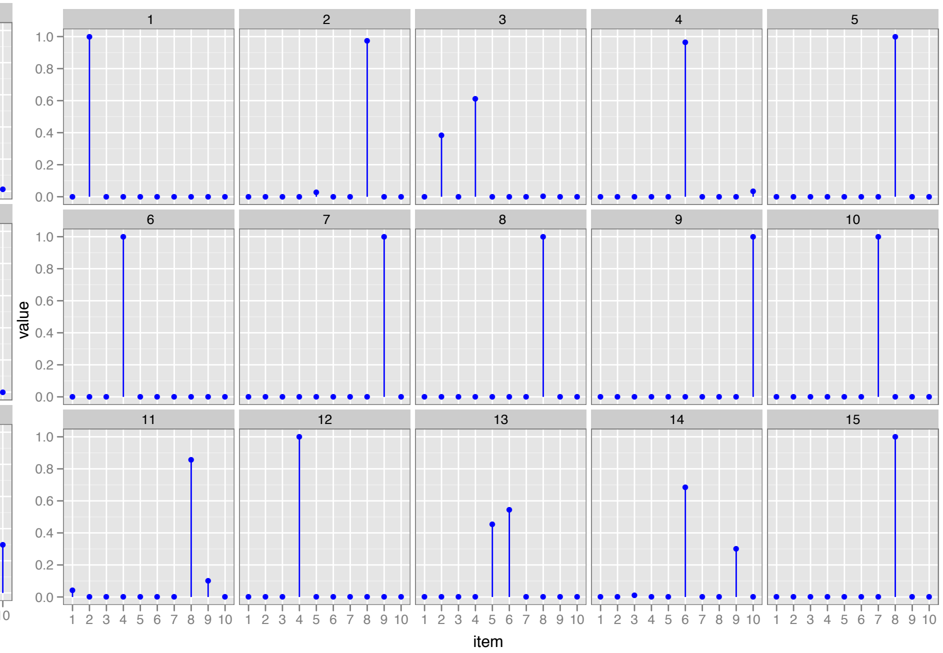


Corpus-level Parameter α

- $\alpha = 0.1$



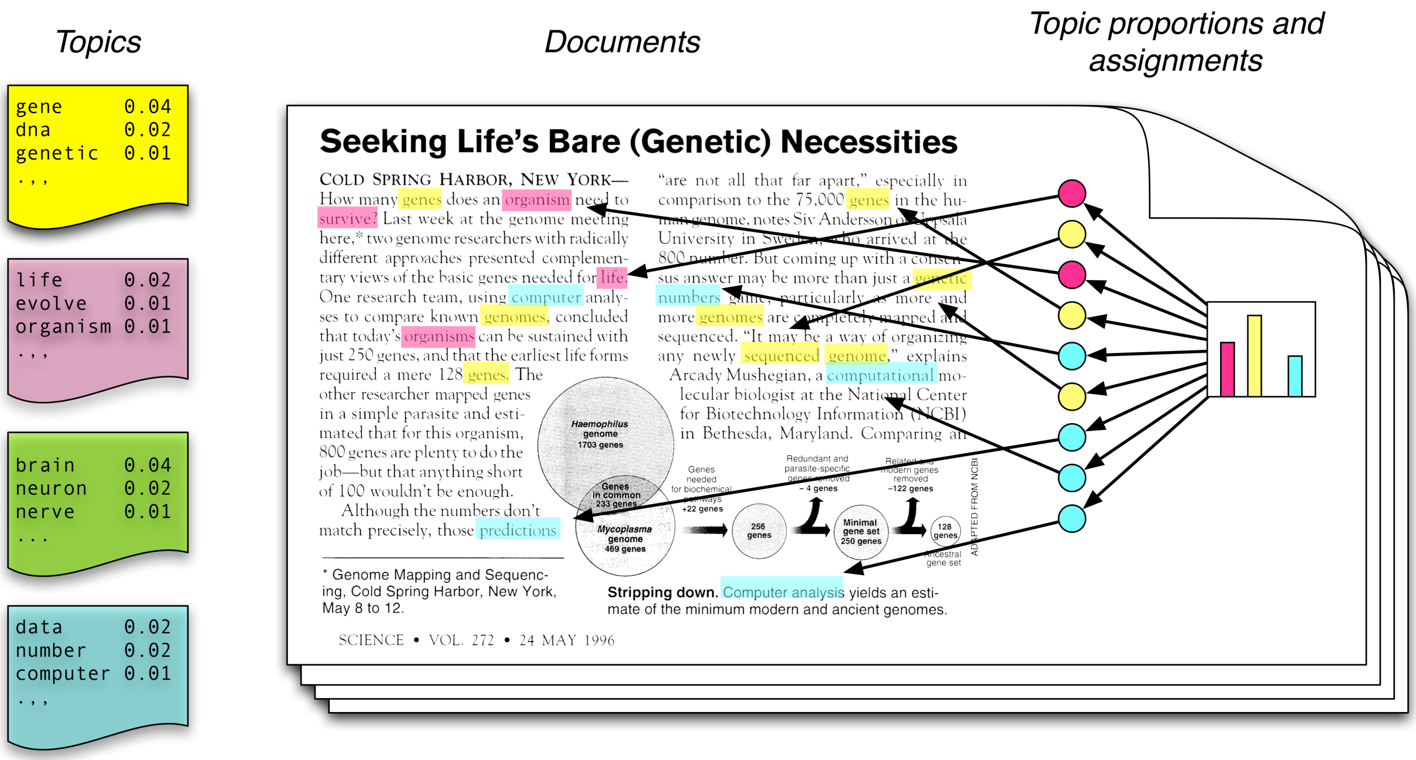
- $\alpha = 0.01$

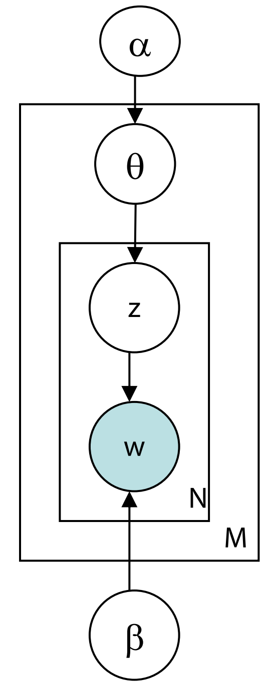


Back to Topic Modeling Scenario

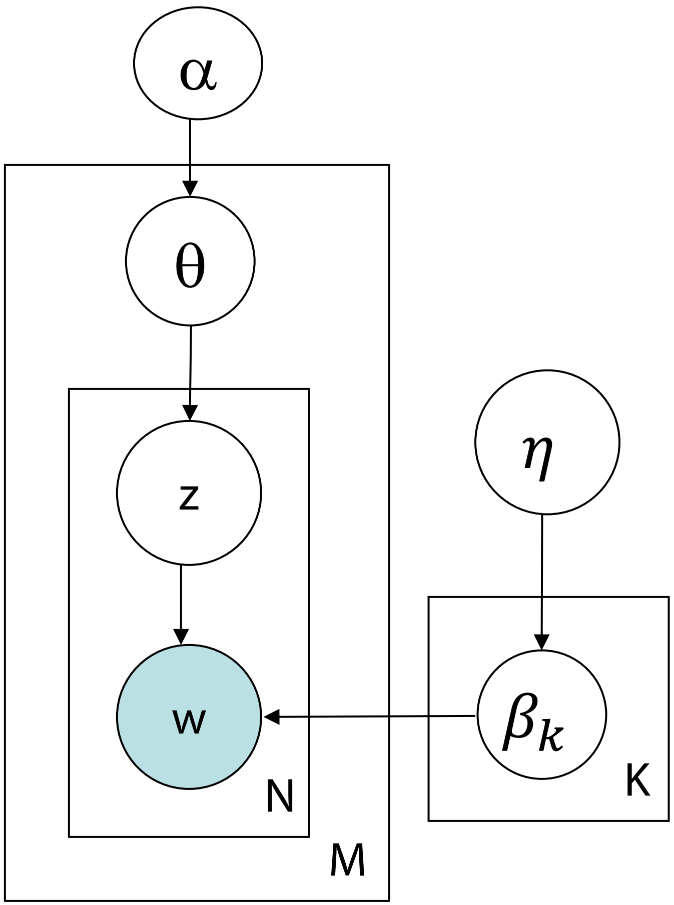
What are the words' topics and word distribs of topics?

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





Topic-specific Words: “Smoothed” LDA Model

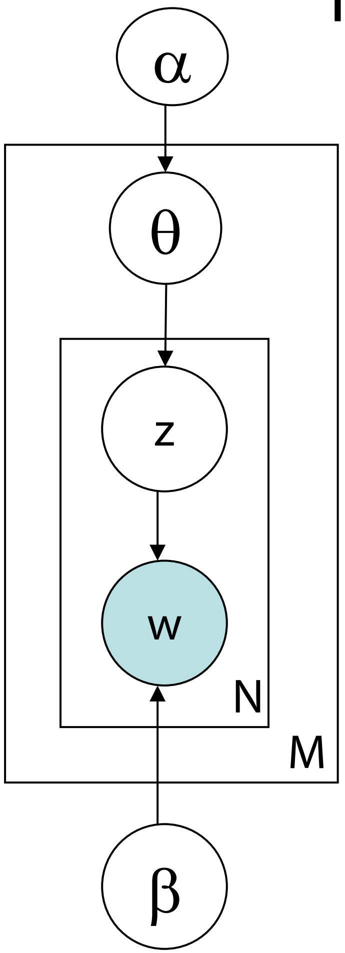


- Give a different word distribution to each topic
  - β is $K \times V$ matrix (V vocabulary size), each row denotes word distribution of a topic
- For each document d
  - Choose $\theta_d \sim \text{Dirichlet}(\alpha)$
  - Choose $\beta_k \sim \text{Dirichlet}(\eta)$
  - For each position $i = 1, \ldots,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

But why does LDA actually 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 non-negligible 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

Query Answering Problem (non-smoothed version)



To which topics does a given document belong?

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

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

$$P(\boldsymbol{w} | \alpha, \beta) = \int \textstyle\sum_{k=1}^{K} P(\boldsymbol{w}, \theta, \boldsymbol{z} \, | \alpha, \beta) \; d\theta =$$

$$\int \textstyle\sum_{k=1}^{K} P(\theta | \alpha) \prod_{i=1}^{N} P(z_i | \theta) P(w_i | z_i, \beta) \, d\theta =$$

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

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

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

LDA Learning

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

We have a
lecture on
Approximation
Algorithms for
Probabilistic
Models !

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

Back to Agents

- Agents not only *use* models
- Agents *build* models that are appropriate to fulfil the agents' goals …
  - … or maximize the utilities derived from preference structures and goals
- Agents *derive approximation algorithms* for query answering on the models they find appropriate

Agents also "learn" QA strategies

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-7 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

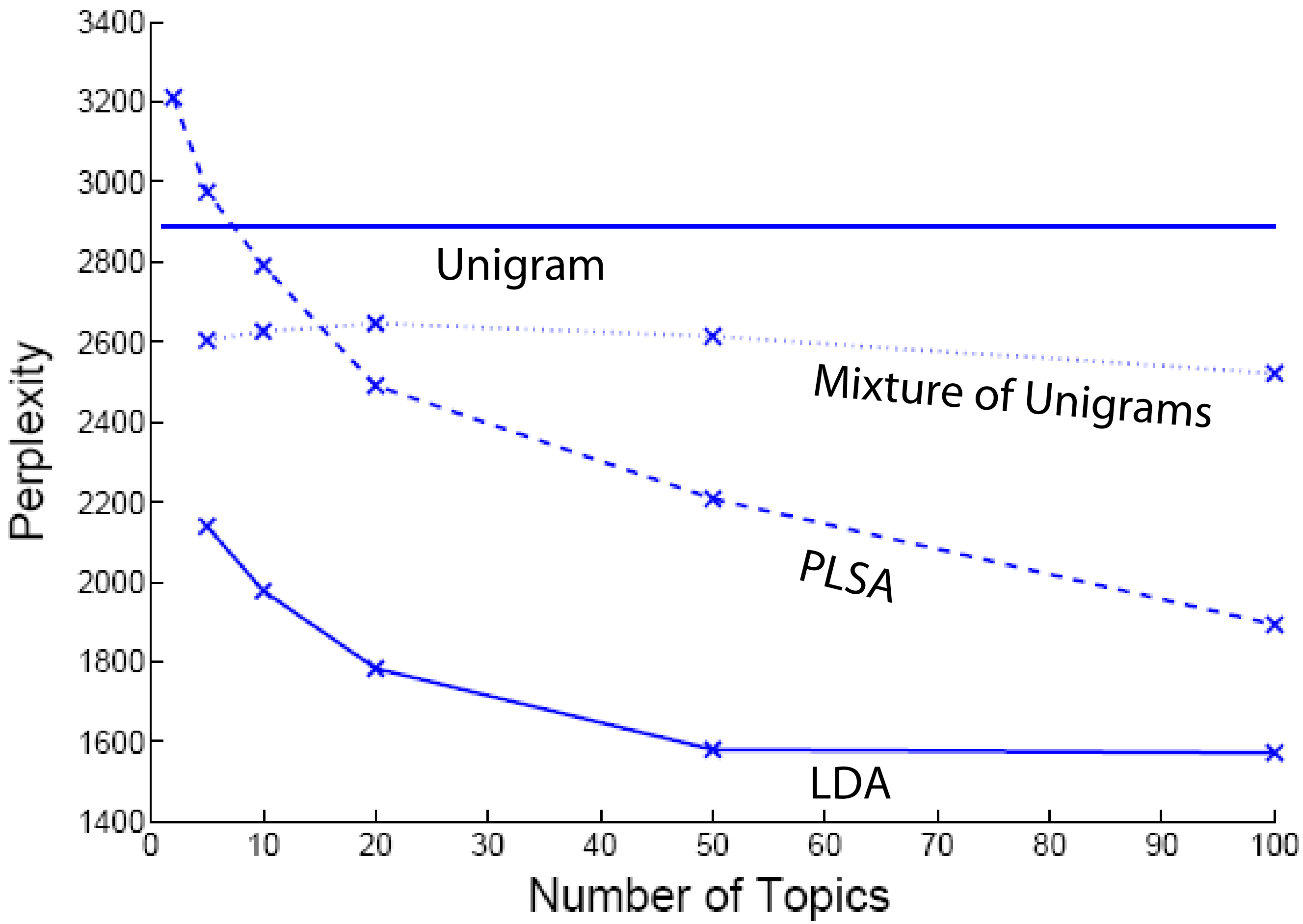
- Perplexity of a probability model
- Describe how well a probability distribution or probability model predicts a sample
  - q: Model of an unknown probability distribution p based on a training sample drawn from p
  - Evaluate q by asking how well it predicts a separate test sample $x_1, \ldots, x_N$ 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 $q(x_i)$ $\rightarrow$ lower perplexity ("less surprised by sample")

[Wikipedia]

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 a 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

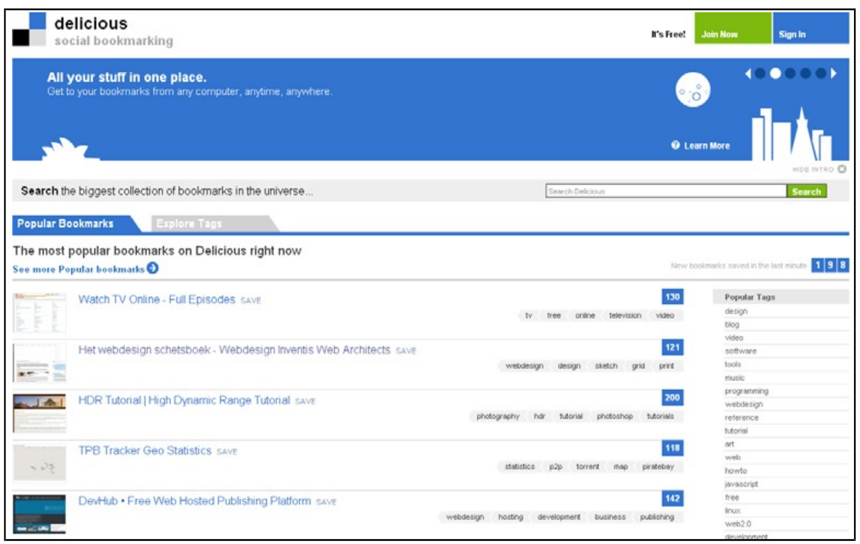
Acknowledgements

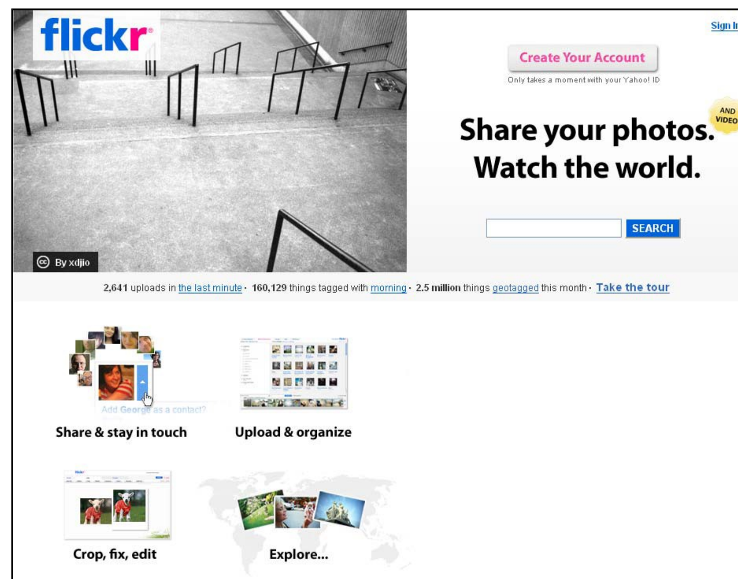
Topic modelling

Tomoharu Iwata

Social annotation services

- Delicious, Flickr, CiteULike, youtube, Last.fm, Technorati, Hatena
- Users can attach annotations freely to objects, and share the annotations.





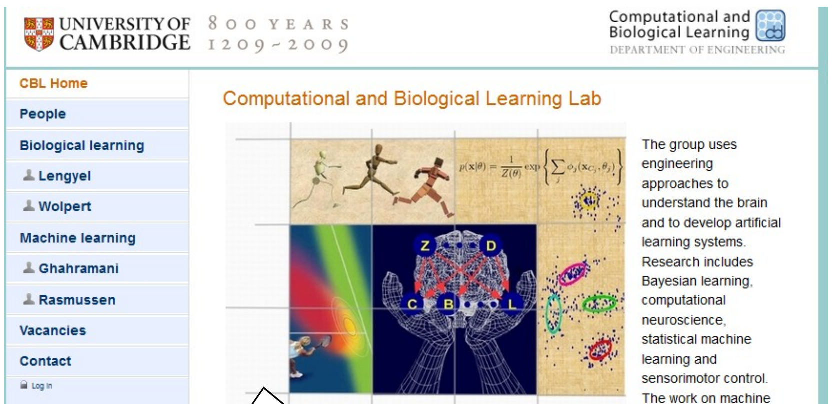
Derive content-unrelated annotations

- manufacturer of camera that took the photo
  - ‘nikon’, ‘canon’
- when they were taken
  - ‘2008’, ‘november’
- remind the annotator
  - ‘toread’
- qualities
  - ‘great’, ‘*****’
- ownership

Proposed model

- generative model for contents (words) and annotations with relevance based on topic models
- infer relevance to the content for each annotation

content



group engineering brain develop theory learning human research systems modelling

(bag-of-words)

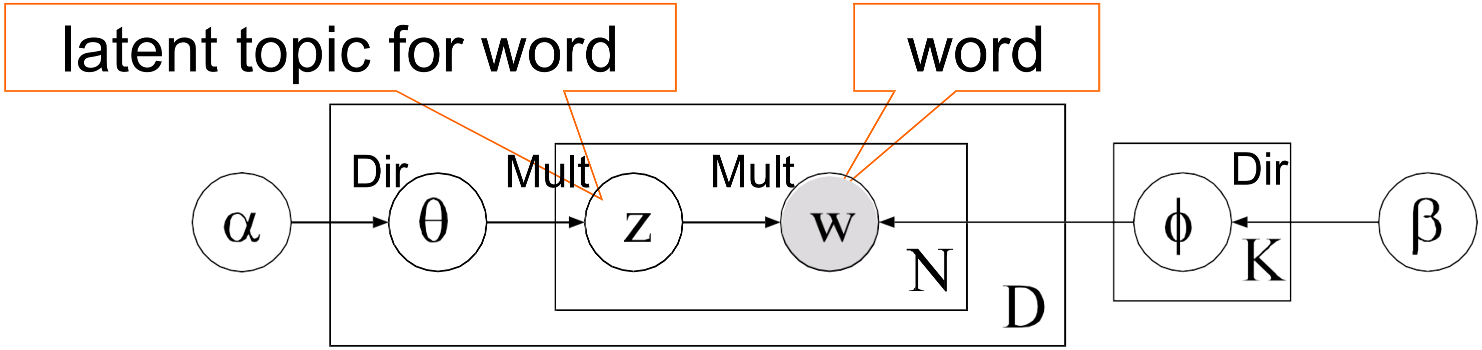
annotation

machine-learning toread bayes ***** neuroscience

Content-related: machine-learning bayes, neuroscience

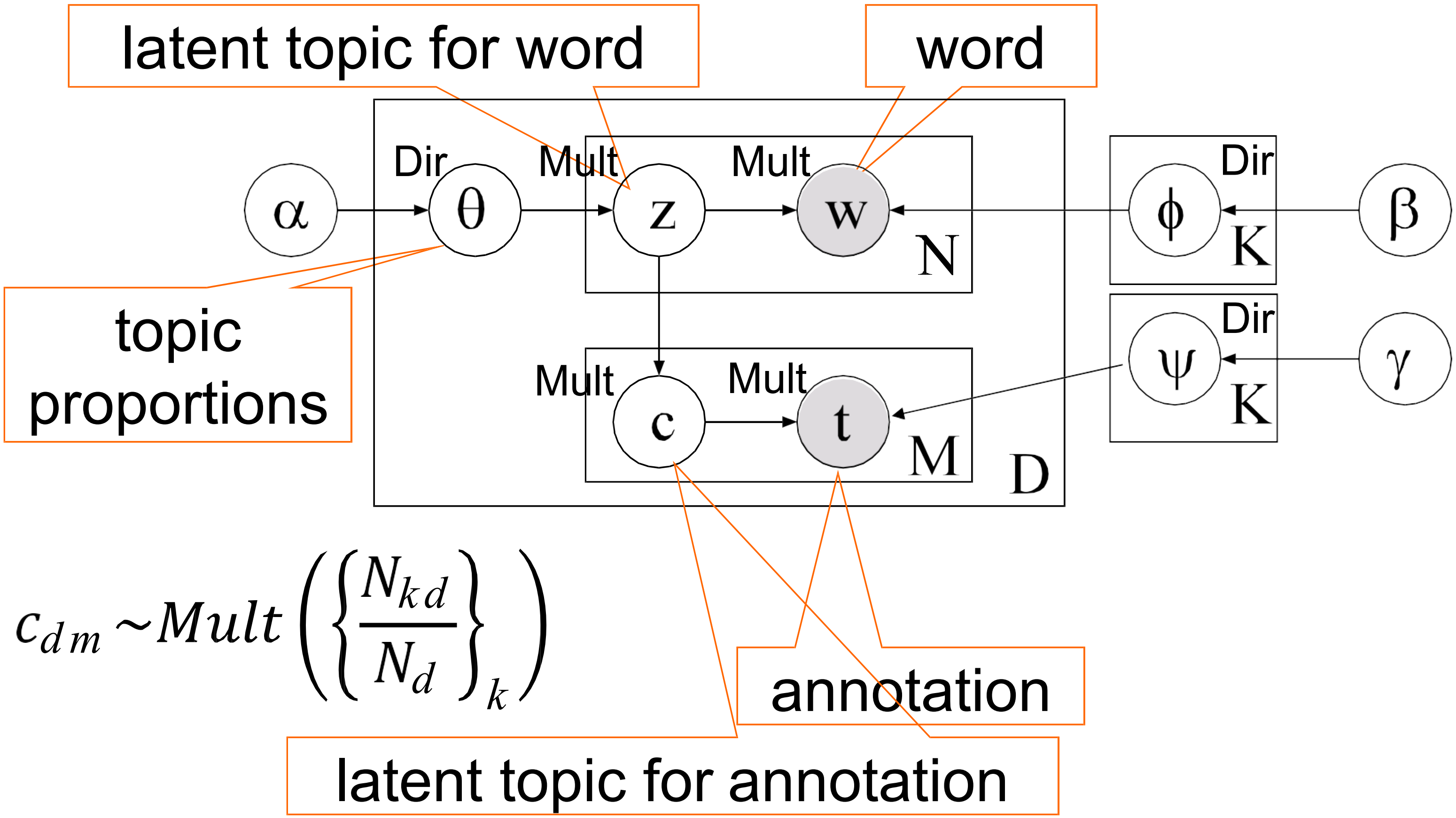
Content-unrelated: toread *****

Latent Dirichlet allocation



[Blei et. al. Latent Dirichlet Allocation, JMLR2003]

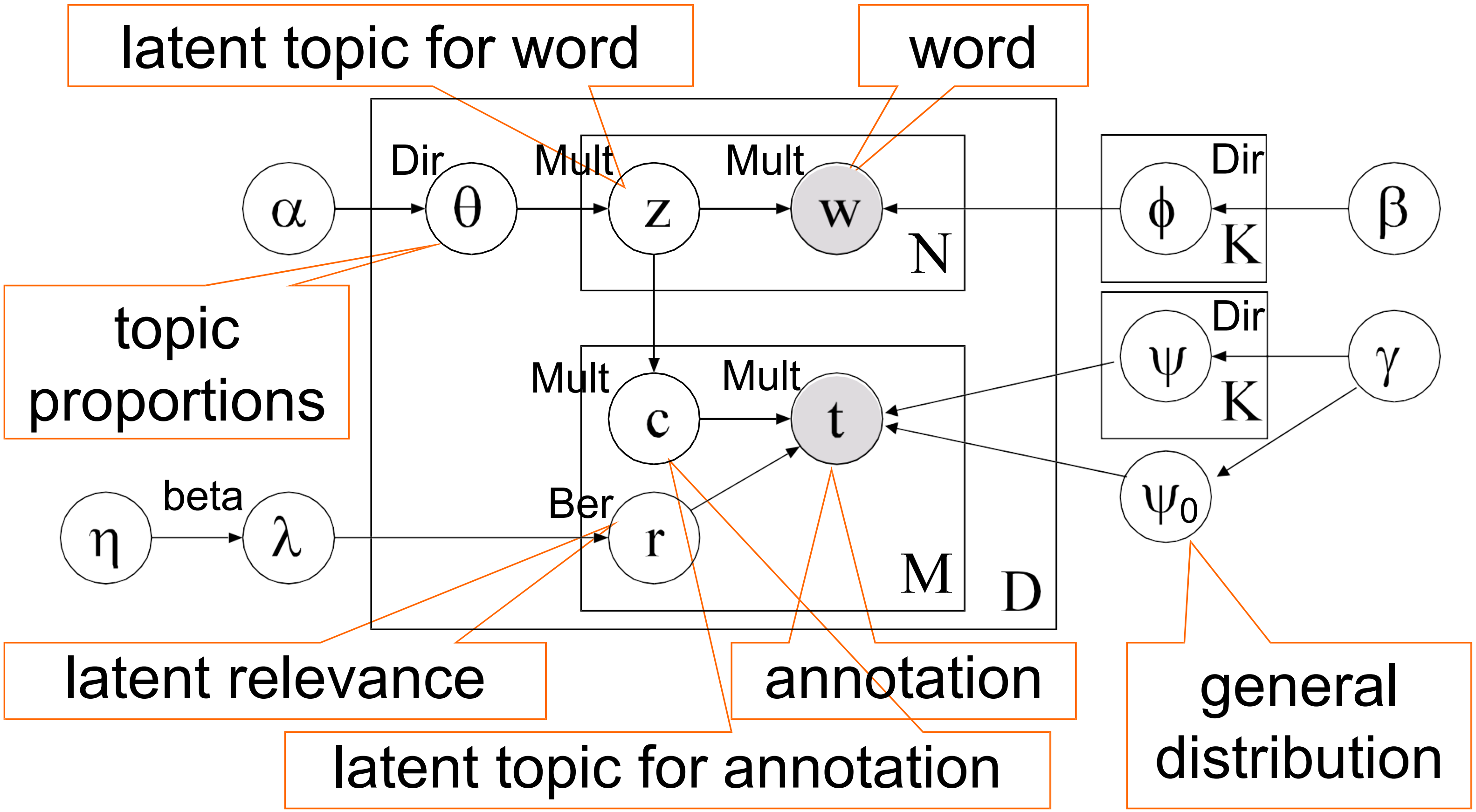
Correspondence LDA



$$c_{dm} \sim Mult\left(\left\{\frac{N_{kd}}{N_d}\right\}_k\right)$$

David M. Blei and Michael I. Jordan. Modeling annotated data. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03). Association for Computing Machinery, New York, NY, USA, 127–134. **2003**.

Proposed model (Inference with Gibbs Sampling)

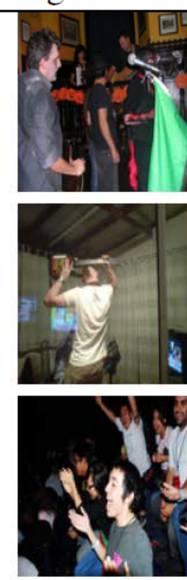
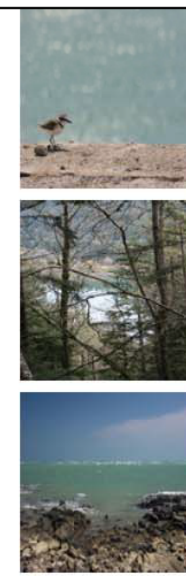


- N: #words, M: #annotations, D: #documents, K: #topics
- each annotation is associated with a latent variable r, r=1 if content-related, r=0 otherwise

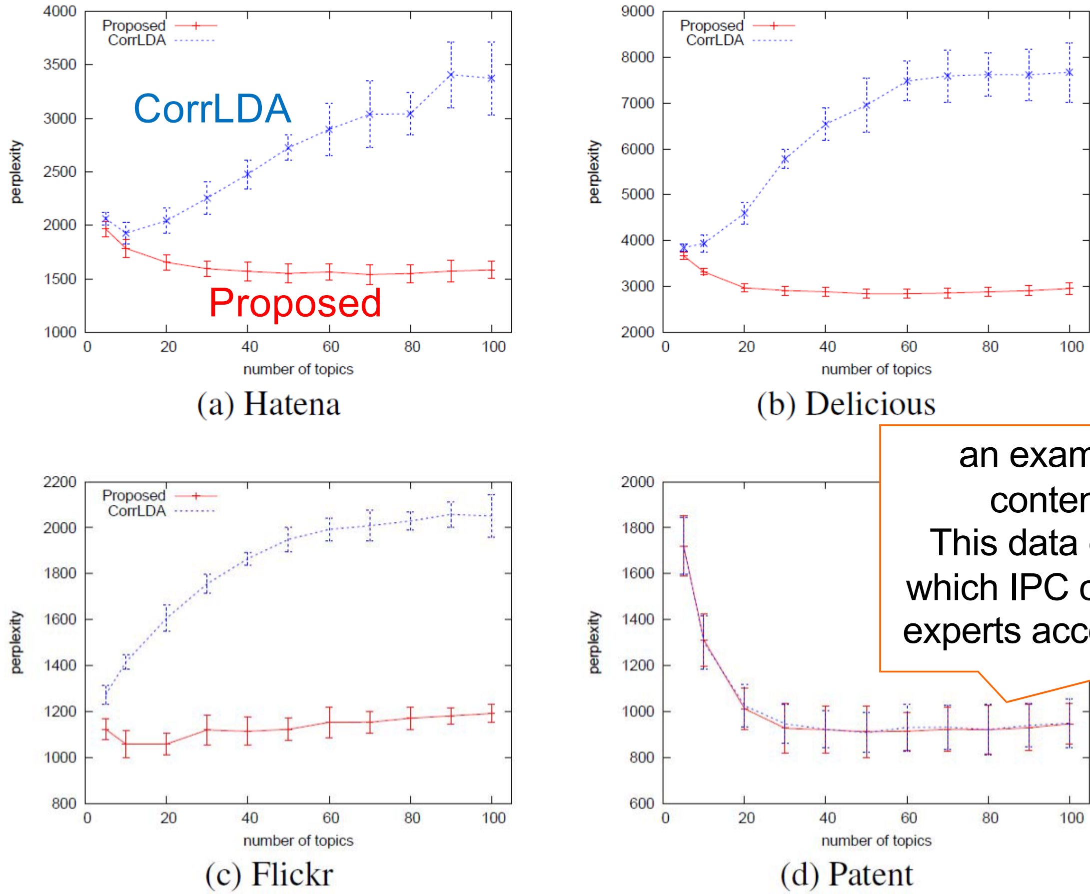
Topics in Delicious

	unrelated	Topic1	Topic2	Topic3	Topic4	Topic5
annotation	reference	money	video	opensource	food	windows
	web	finance	music	software	recipes	linux
	imported	economics	videos	programming	recipe	sysadmin
	design	business	fun	development	cooking	Windows
	internet	economy	entertainment	linux	Food	security
	online	Finance	funny	tools	Recipes	computer
	cool	financial	movies	rails	baking	microsoft
	toread	investing	media	ruby	health	network
	tools	bailout	Video	webdev	vegetarian	Linux
	blog	finances	film	rubyonrails	diy	ubuntu
content word		money	music	project	recipe	windows
		financial	video	code	food	system
		credit	link	server	recipes	microsoft
		market	tv	ruby	make	linux
		economic	movie	rails	wine	software
		october	itunes	source	made	file
		economy	film	file	add	server
		banks	amazon	version	love	user
		government	play	files	eat	files
		bank	interview	development	good	ubuntu

Topics in Flickr

	unrelated	Topic1	Topic2	Topic3	Topic4	Topic5
annotation	2008	dance	sea	autumn	rock	beach
	nikon	bar	sunset	trees	house	travel
	canon	dc	sky	tree	party	vacation
	white	digital	clouds	mountain	park	camping
	yellow	concert	mountains	fall	inn	landscape
	red	bands	ocean	garden	coach	texas
	photo	music	panorama	bortescristian	creature	lake
	italy	washingtondc	south	geotagged	halloween	cameraphone
	california	dancing	ireland	mud	mallory	md
	color	work	oregon	natura	night	sun
probable image						

Perplexity



x-axis: #topics
y-axis: perplexity

an example of data without content-unrelated tags.
This data consist of patents, to which IPC code were attached by experts according to their content.

The proposed method performed better than Corr-LDA in the case of noisy social annotation data.

Acknowledgements

Generative Topic Models for Community Analysis

Ramesh Nallapati

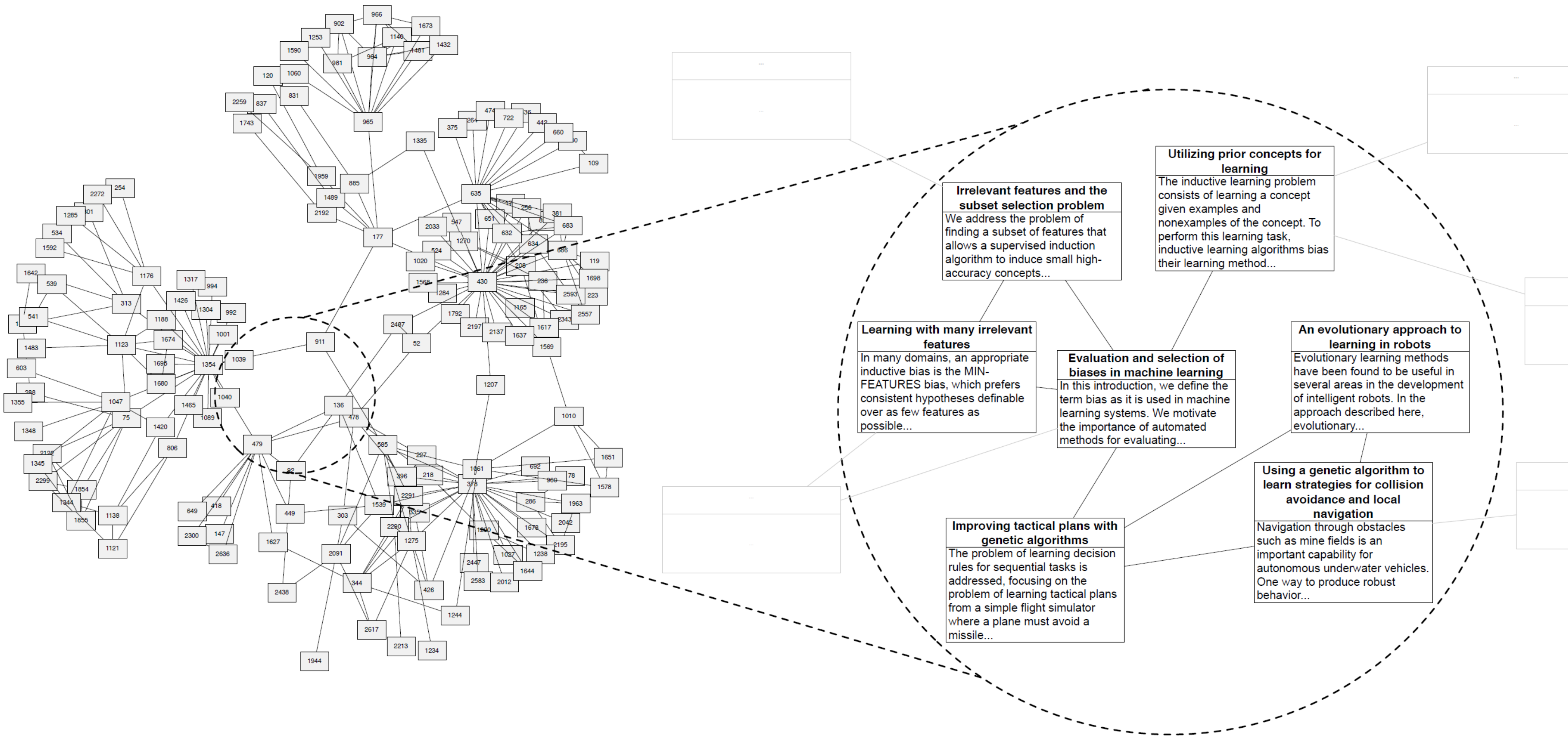
http://www.cs.cmu.edu/~wcohen/10-802/lda-sep-18.ppt

&

Arthur Asuncion, Qiang Liu, Padhraic Smyth:

Statistical Approaches to Joint Modeling of Text and Network Data

What if the corpus has network structure?



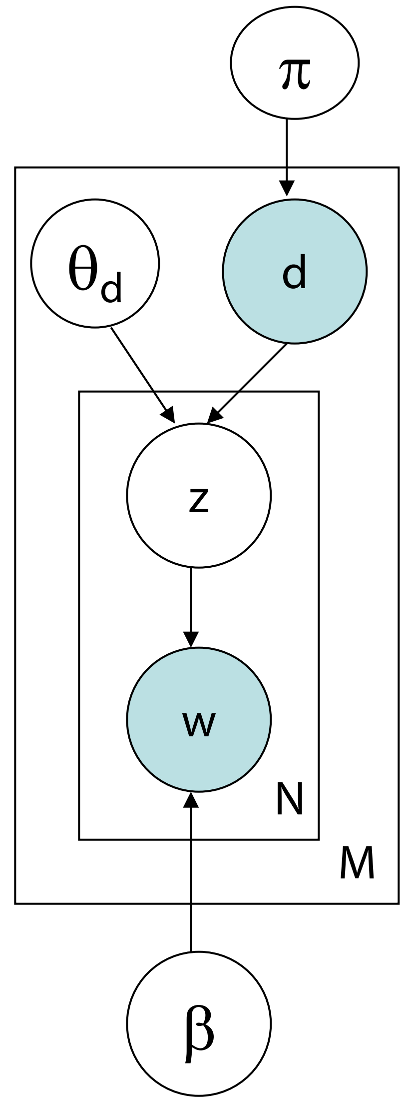
CORA citation network. Figure from [Chang, Blei, AISTATS 2009]

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

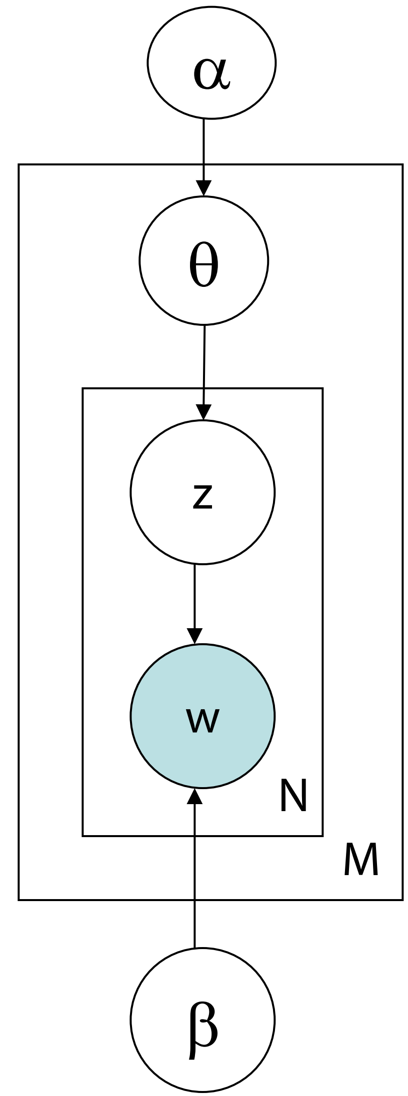
Outline

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

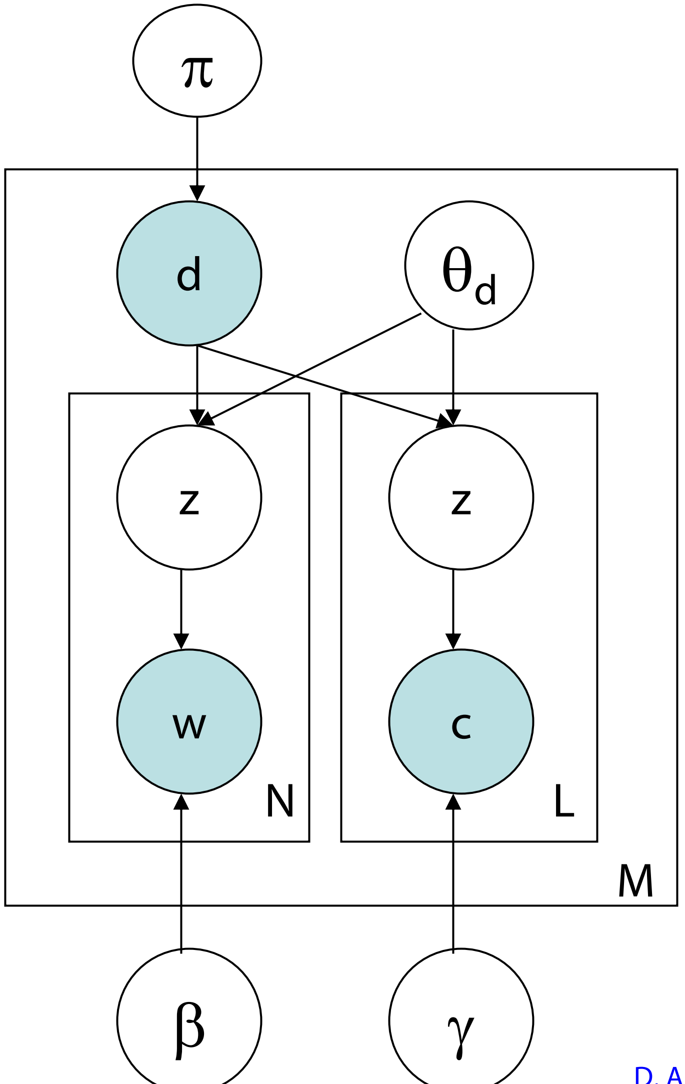
pLSI



LDA



Hyperlink Modeling Using pLSI

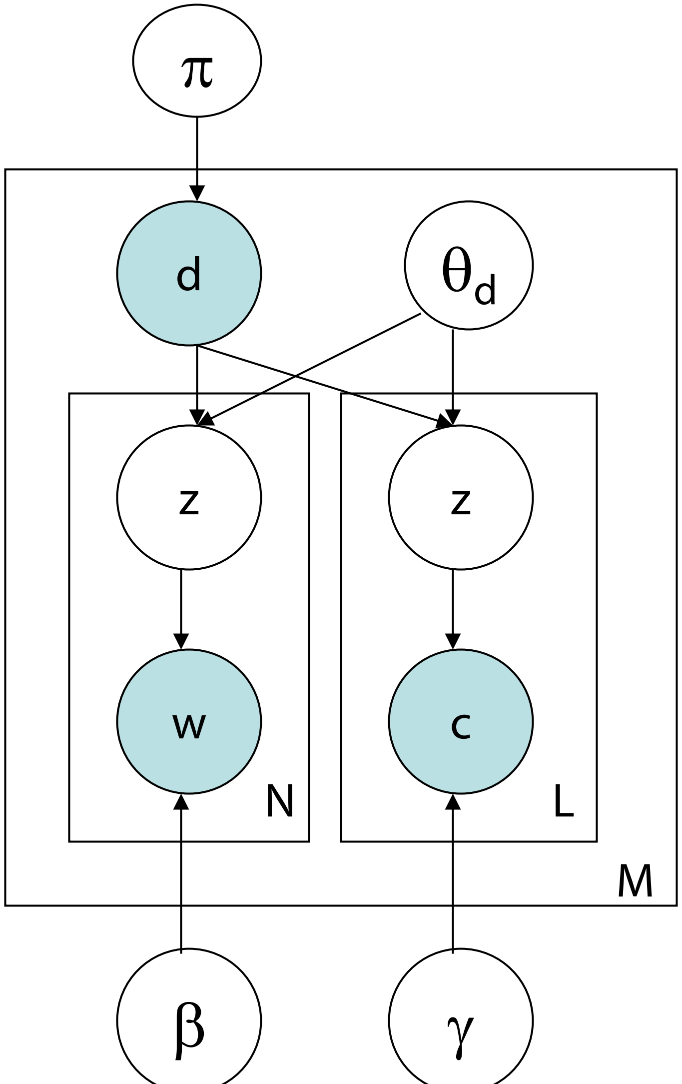


- Select document $d \sim \text{Mult}(. \,|\pi)$
  - For each position $n = 1,\ldots, N_d$
    - Generate $z_n \sim \text{Mult}(\cdot\,|\theta_d)$
    - Generate $w_n \sim \text{Mult}(\cdot\,|\beta_{z_n})$
  - For each citation $j = 1,\ldots, L_d$
    - Generate $z_j \sim \text{Mult}(\cdot\,|\theta_d)$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

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

$$\pi_d := P(d|\pi)$$
$$\beta_{kw_i} := P(w_i|\beta, z_k)$$
$$\gamma_{kc_i} := P(c_i|\gamma, z_k)$$
$$\theta_{dk} := P(z_k|\theta_d)$$



- pLSI likelihood

$$\prod_{d=1}^{M} P(w_1, \ldots, w_{N_d}, d|\theta, \beta, \pi)$$
$$= \prod_{d=1}^{M} \pi_d \left(\prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \beta_{kw_n} \right)$$

- New likelihood

$$\prod_{d=1}^{M} P(w_1, \ldots, w_{N_d}, c_1, \ldots, c_{L_d}, d|\theta, \beta, \gamma, \pi)$$
$$= \prod_{d=1}^{M} \pi_d \left(\prod_{i=1}^{N_d} \sum_{k=1}^{K} \theta_{dk} \beta_{kw_n} \right) \left(\prod_{j=1}^{L_d} \sum_{k=1}^{K} \theta_{dk} \gamma_{kc_j} \right)$$

- 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 PLSA

- Experiments: Text Classification
- Datasets:
  - Web KB
    - 6000 CS dept web pages with hyperlinks
    - 6 Classes: faculty, course, student, staff, etc.
  - Cora
    - 2000 Machine learning abstracts with citations
    - 7 classes: sub-areas of machine learning
- Methodology:
  - Learn the model on complete data and obtain θ_d for each document
  - Test documents classified into the label of the nearest neighbor in training set
  - Distance measured as cosine similarity in the θ space
  - Measure the performance as a function of α

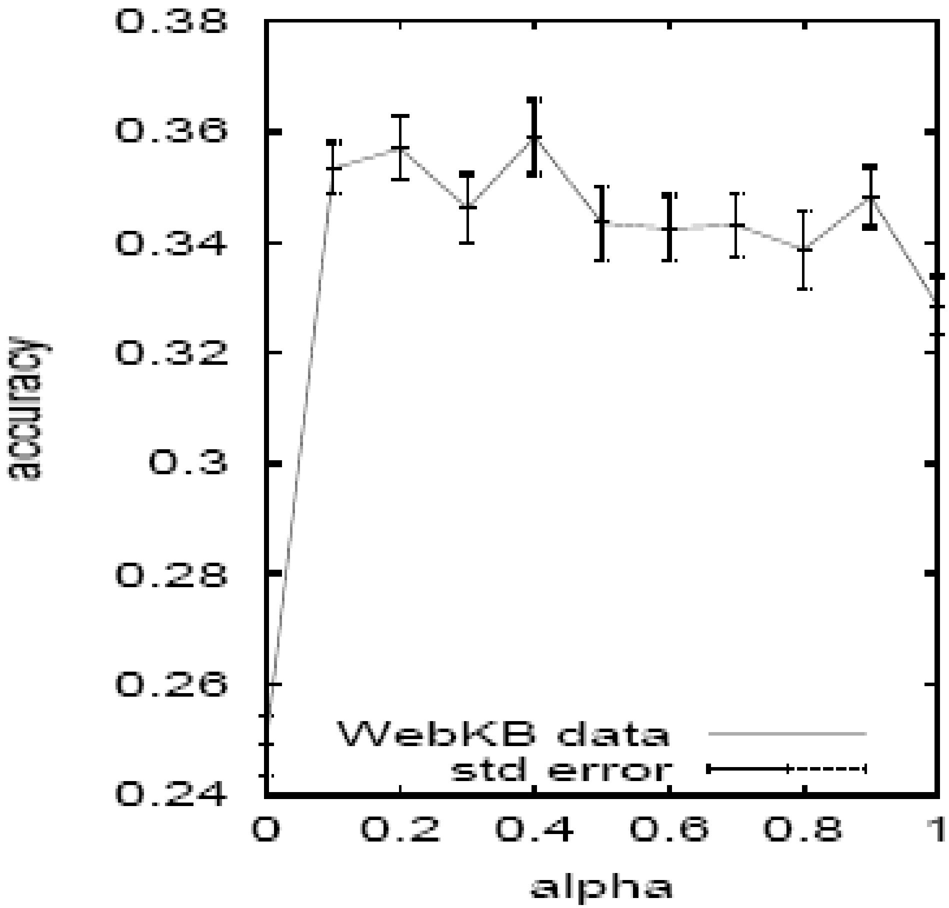
Overview on Evaluation Measures

		True condition			
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$
	Predicted condition negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR–}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR–) $= \frac{\text{FNR}}{\text{TNR}}$	

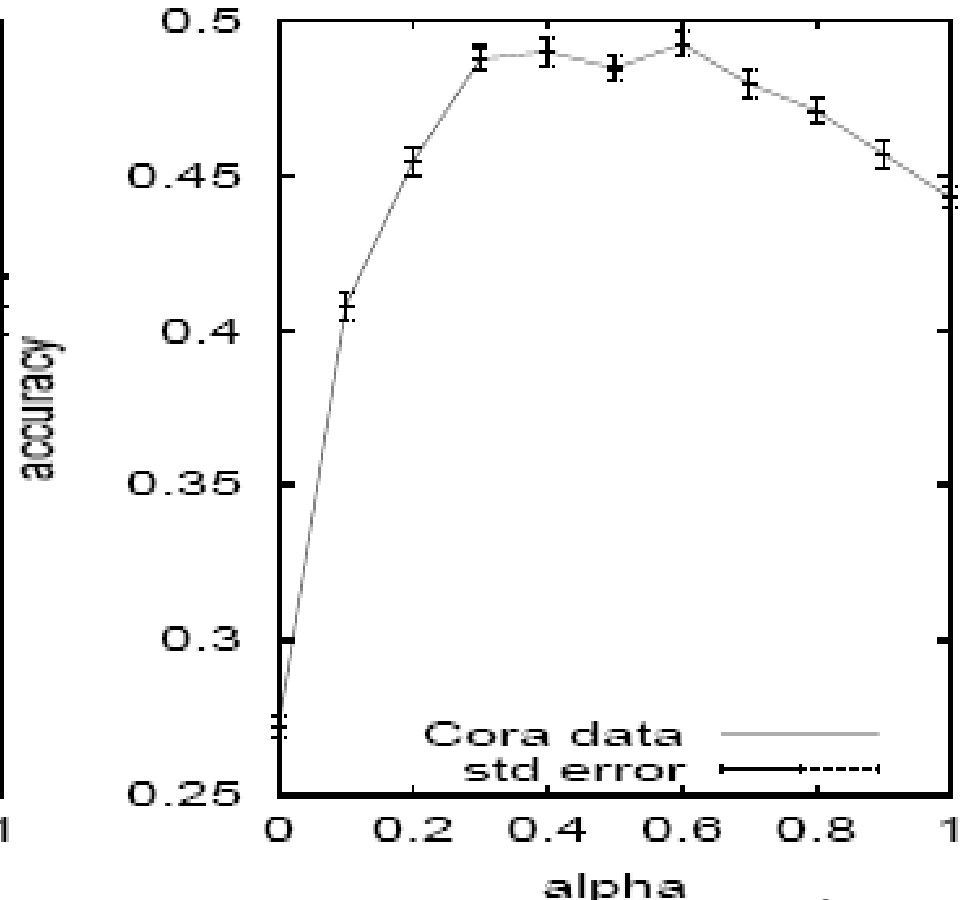
[Wikipedia]

https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)#Mean_average_precision

Hyperlink Modeling Using pLSI

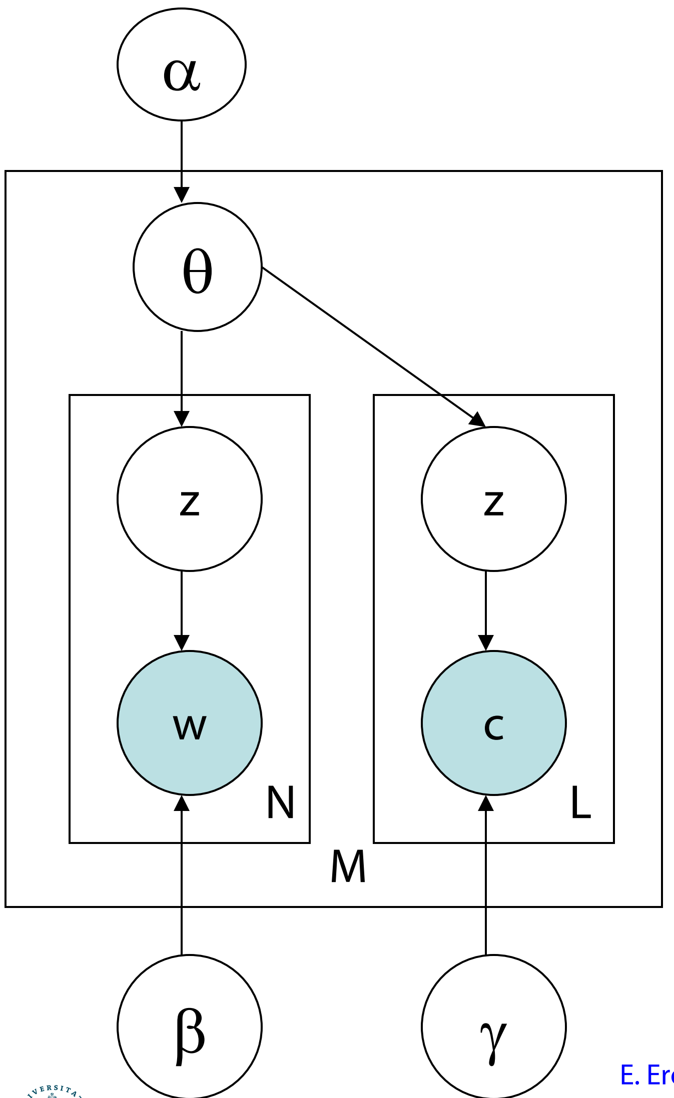


Hyperlink ⟷ Content



Hyperlink ⟷ Content

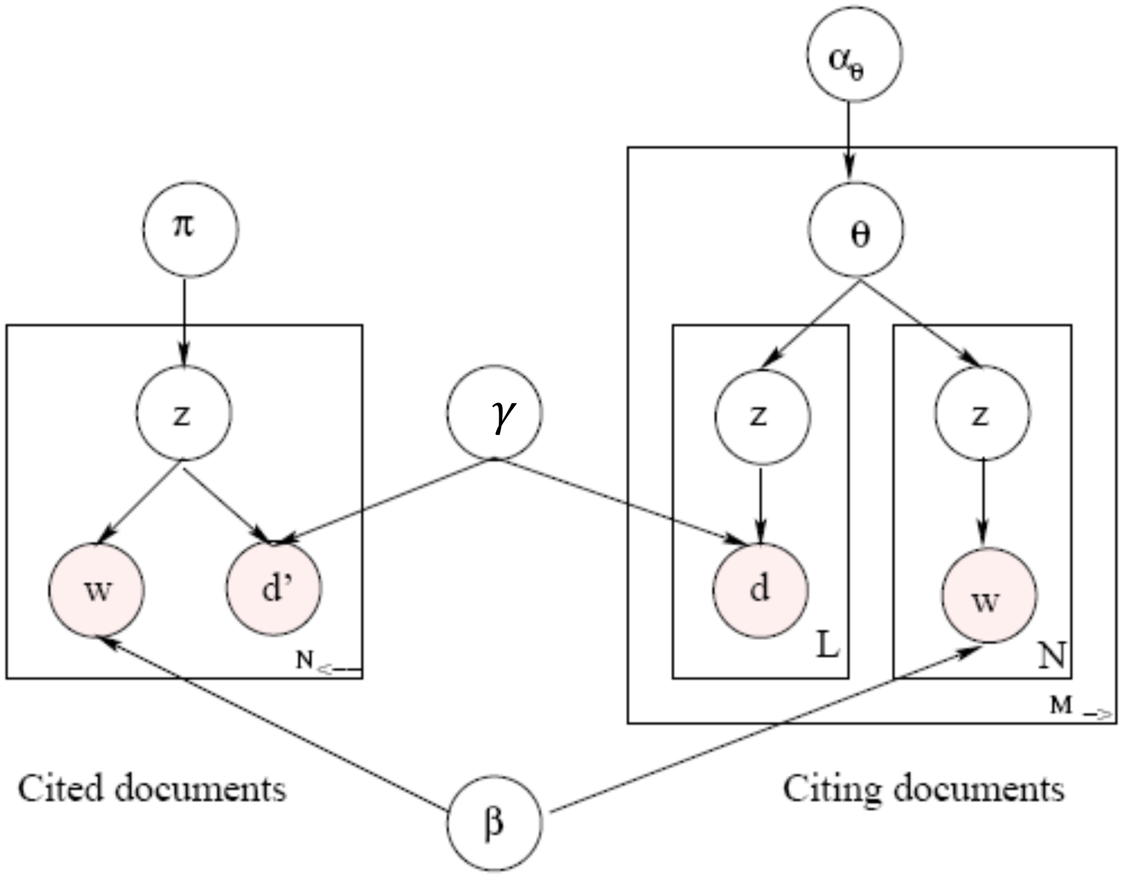
Hyperlink modeling using LDA



- 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.

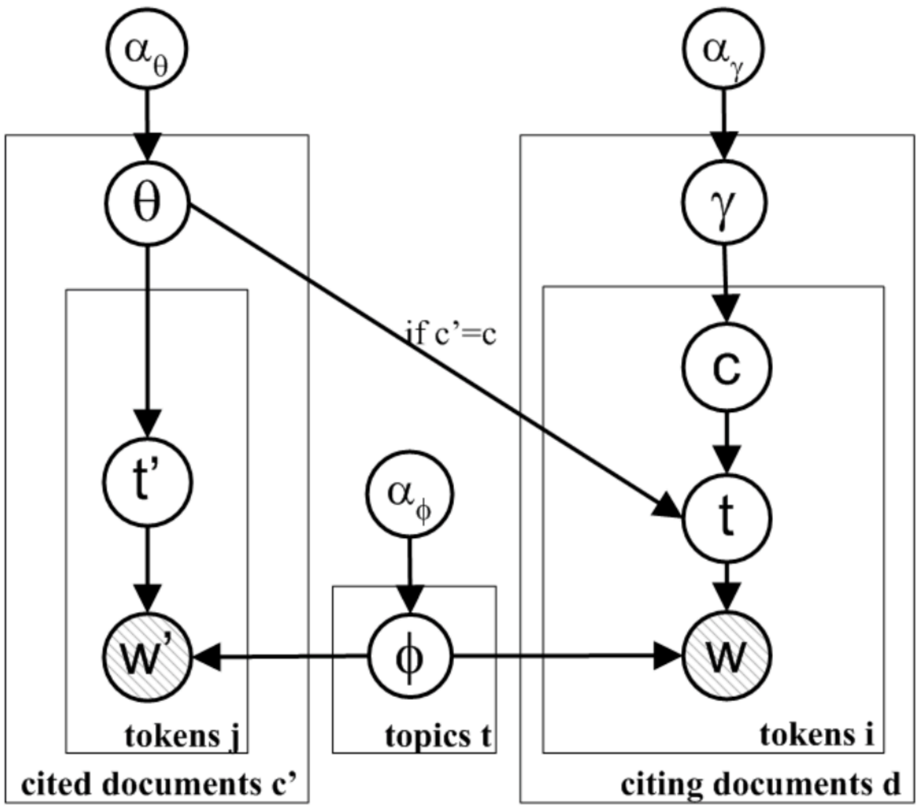
Link-pLSI-LDA: Topic Influence in Blogs



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

Modeling Citation Influences - Copycat Model

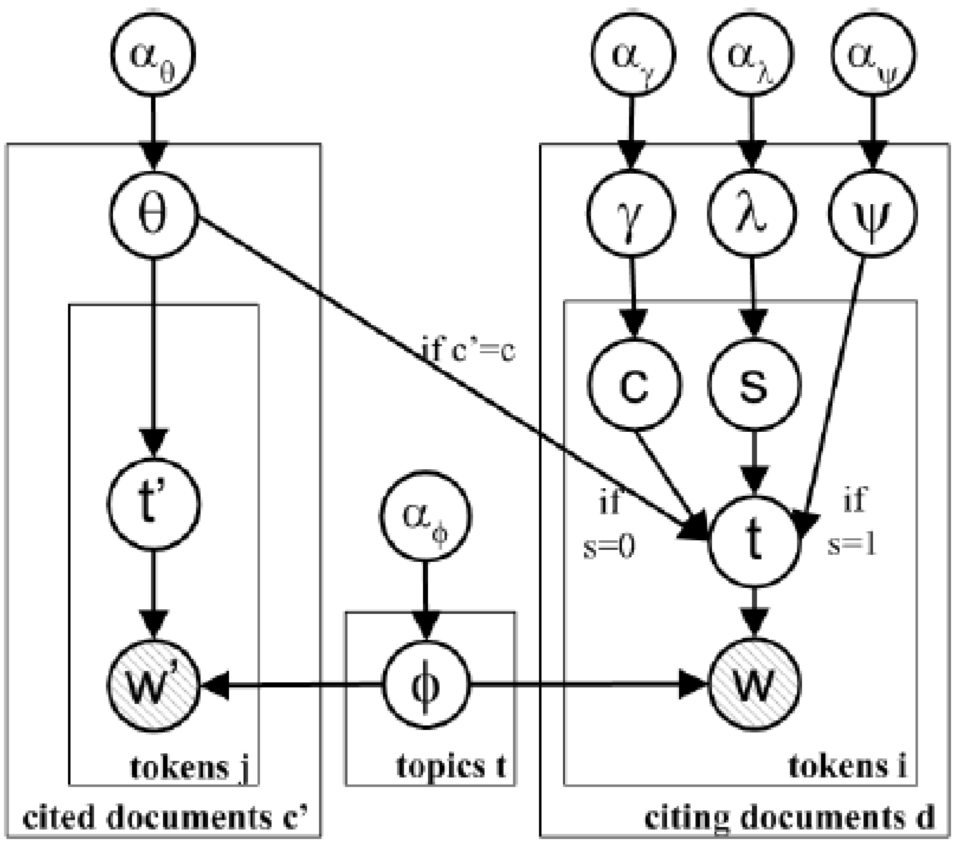
- Each topic in a citing document is drawn from one of the topic mixtures of cited publications



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

Modeling Citation Influences

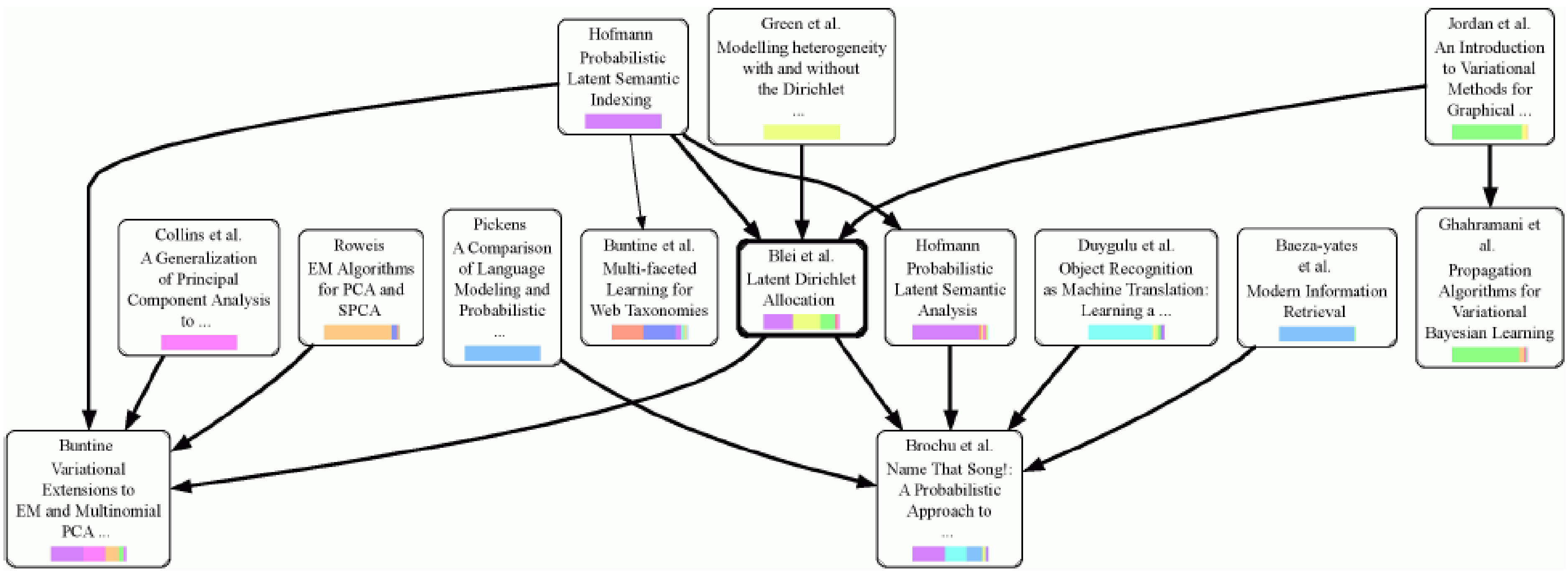
- Citation influence model: Combination of LDA and Copycat model



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



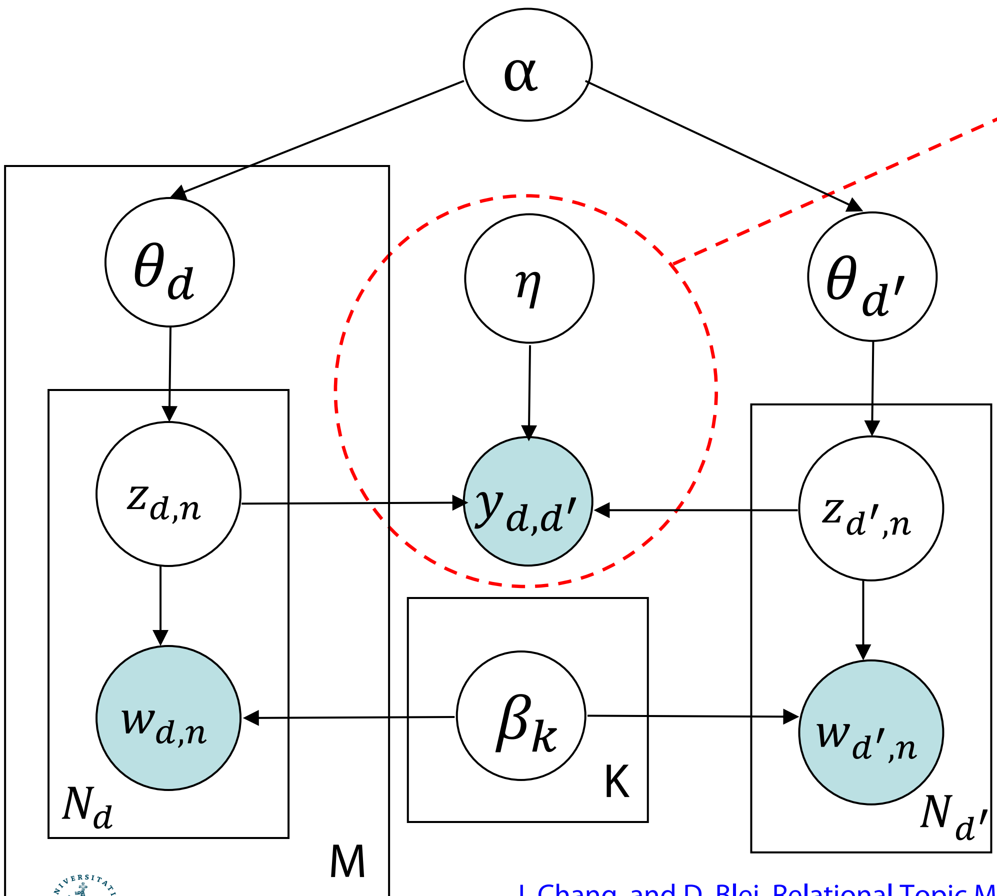
Modeling Citation Influences

- 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

Relational Topic Model (RTM) [ChangBlei 2009]

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



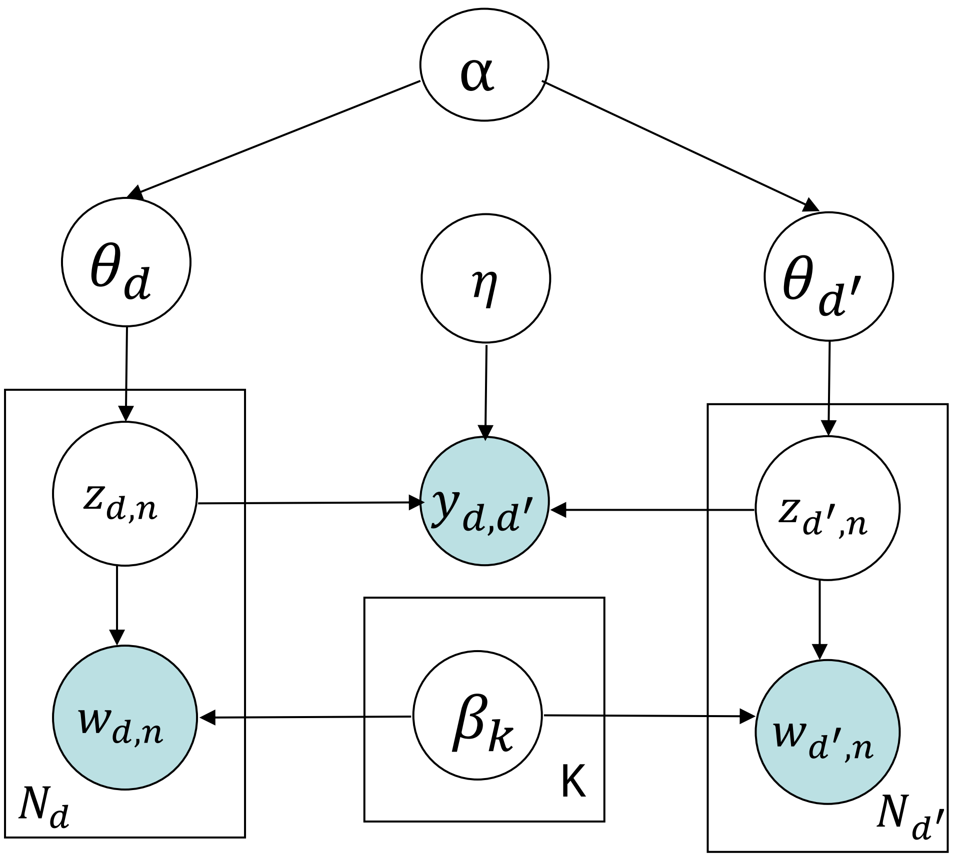
$y_{d,d'} \sim \psi(y_{d,d'} | z_d, z_{d'}, \eta)$

"Link probability function"

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 $y | z_d, z_{d'} \sim \psi(\cdot \, | z_d, z_{d'}, \eta)$

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

Conclusion

- Topic Modeling
- 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