CS 589 Fall 2020

Latent semantic indexing

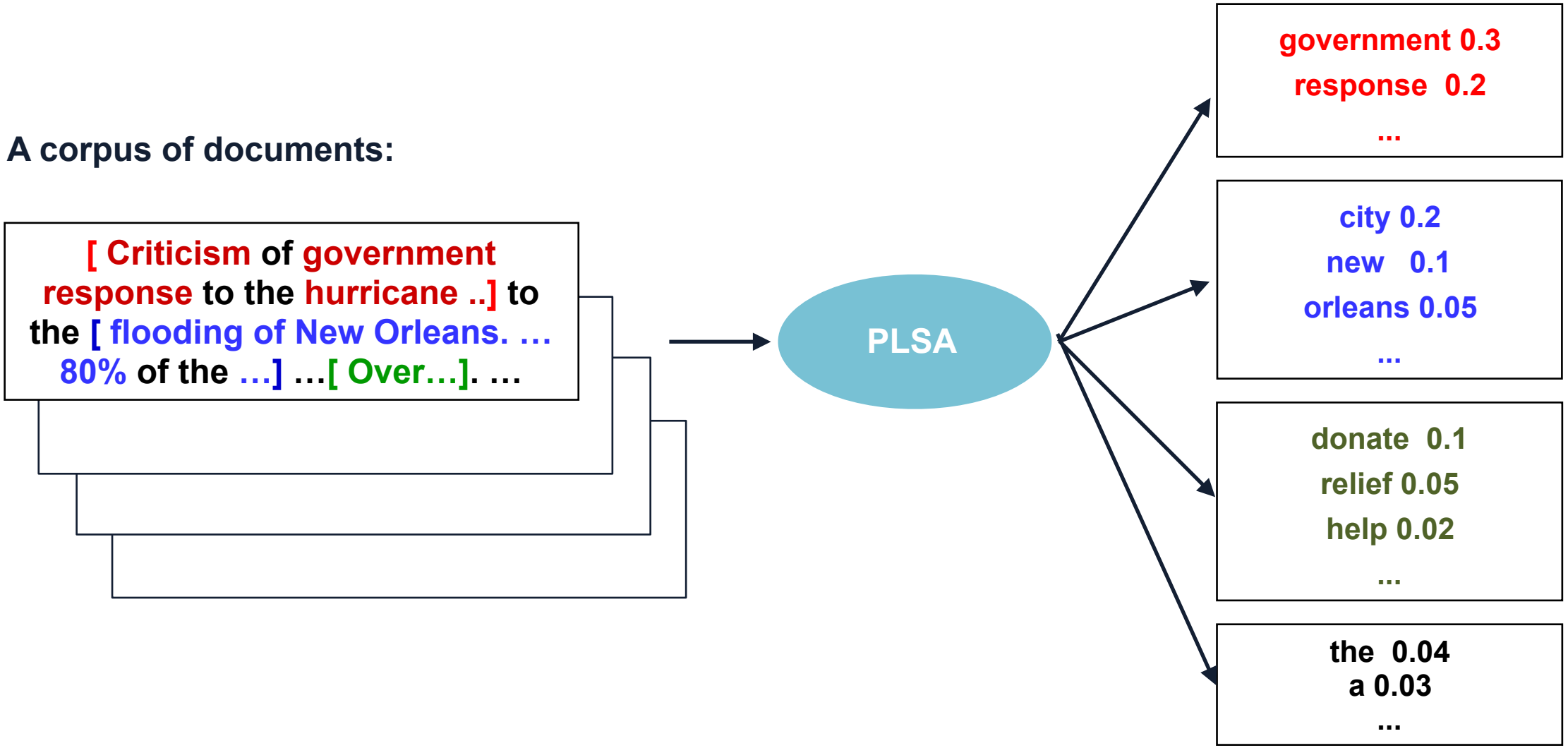
Latent Dirichlet allocation

Instructor: Susan Liu
TA: Huihui Liu

Stevens Institute of Technology

Most slides come from UIUC CS510 LDA lecture by Chase Geigle, ChengXiang Zhai

Review of the last lecture



Today's lecture

- Latent semantic indexing

- Continue on topic model
  - Bayesian inference of topic model
  - Variational inference for LDA
  - Gibbs sampling, Markov chain Monte-Carlo

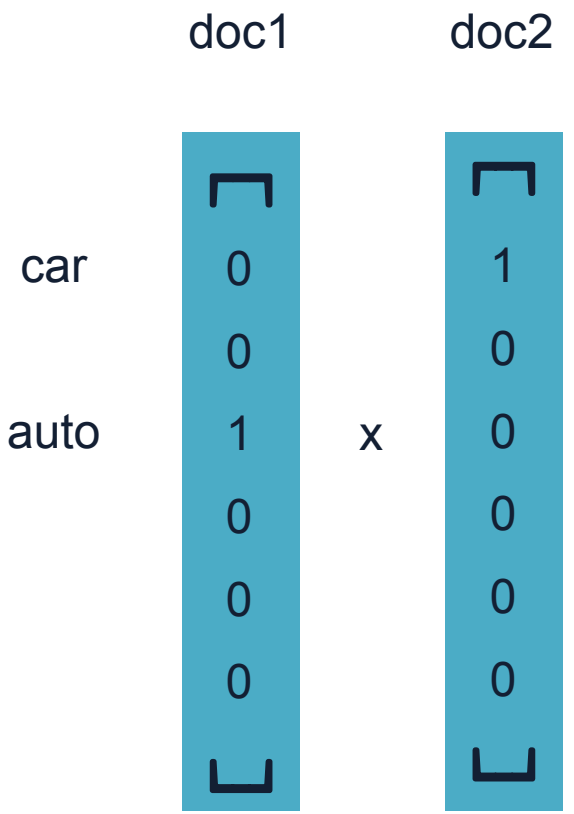
Vocabulary gap problem with vector space model

- Vector space model:

$$score(q, d) = \frac{q \cdot d}{\|q\| \cdot \|d\|}$$

- Challenges matching synonyms
  - e.g., auto vs. car

- Challenges matching polysemy
  - e.g., apple (fruit vs. company)



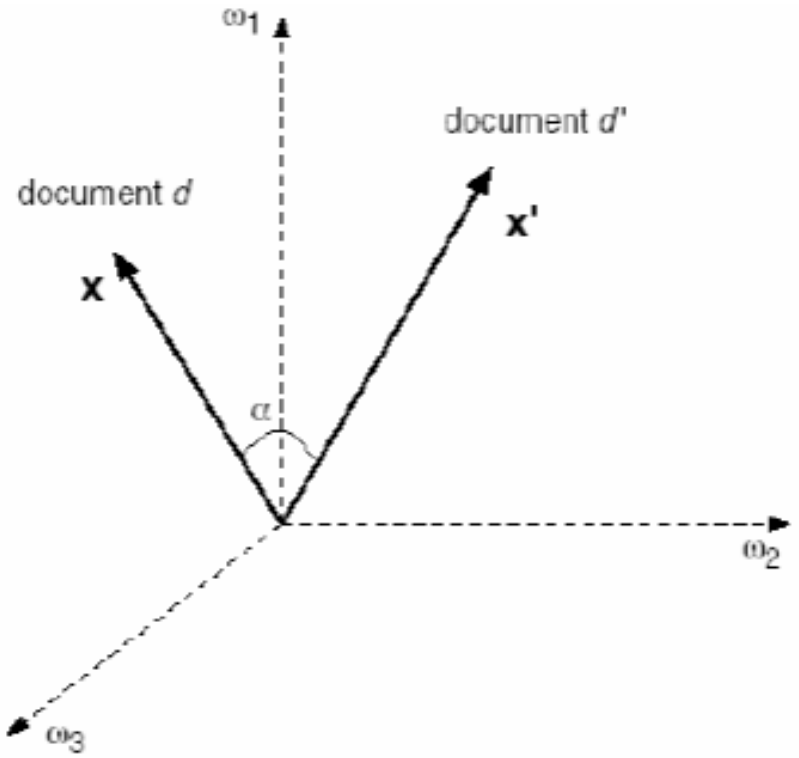
Low-dimensional, dense vector representation [Deerwester et al. 1998

Latent semantic indexing

- Uses statistically derived conceptual indices instead of individual words for retrieval •
- Assumes that there is some underlying or latent structure in word usage that is obscured by variability in word choice •
- Key idea: instead of representing documents and queries as vectors in a low dimensional space of terms

Low dimensional vector representation of words

- Axes are concepts, also called principle components (PCA)



Singular value decomposition

- For an min matrix A of rank r, there exists a factorization (SVD) as follows:

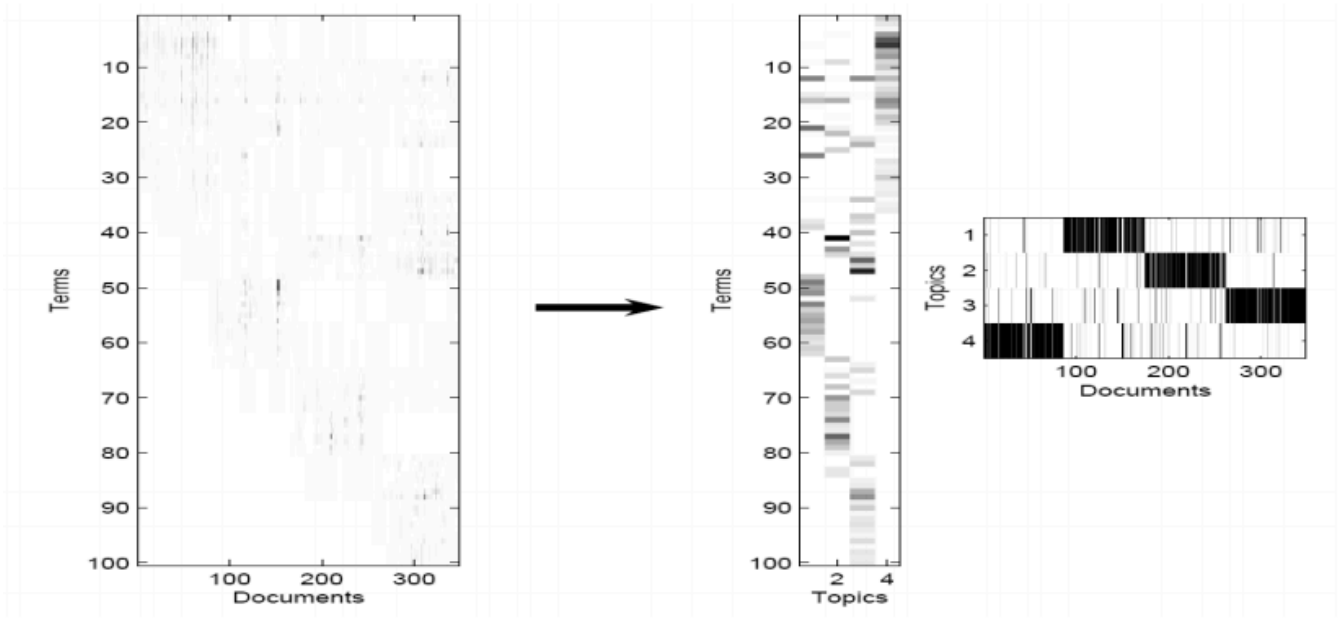
$$A = U\Sigma V^T$$

- U: orthogonal eigen vectors of AA^T
- V: orthogonal eigen vectors of A^TA
- Sigma: eigen values

$$\Sigma = \mathrm{diag}\left(\sigma_1 \ldots \sigma_r\right)$$

Dimension reduction

- Map documents and queries to a low dimensional space
- Retrieval in this space may be superior to retrieval in the original space



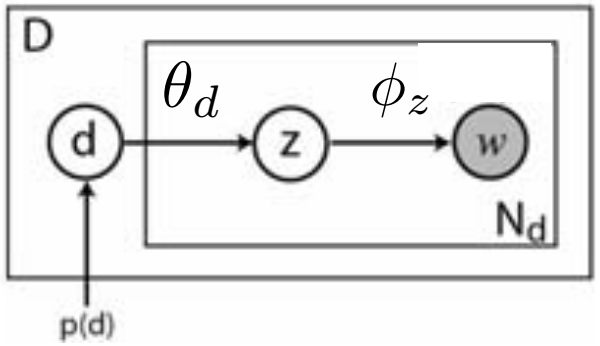
What LSI can do

- LSI effectively does
  - – Dimensionality reduction
  - – Noise reduction
  - – Exploitation of redundant data
  - – Correlation analysis and Query expansion (with related words)
- Some of the individual effects can be achieved with simpler techniques (e.g. thesaurus construction). LSI does them together
- LSI handles **synonymy** well, not so much **polysemy** (vs word embedding)
- Challenge: SVD is complex to compute (O(n3)) – Needs to be updated as new documents are found/updated

Review of PLSA

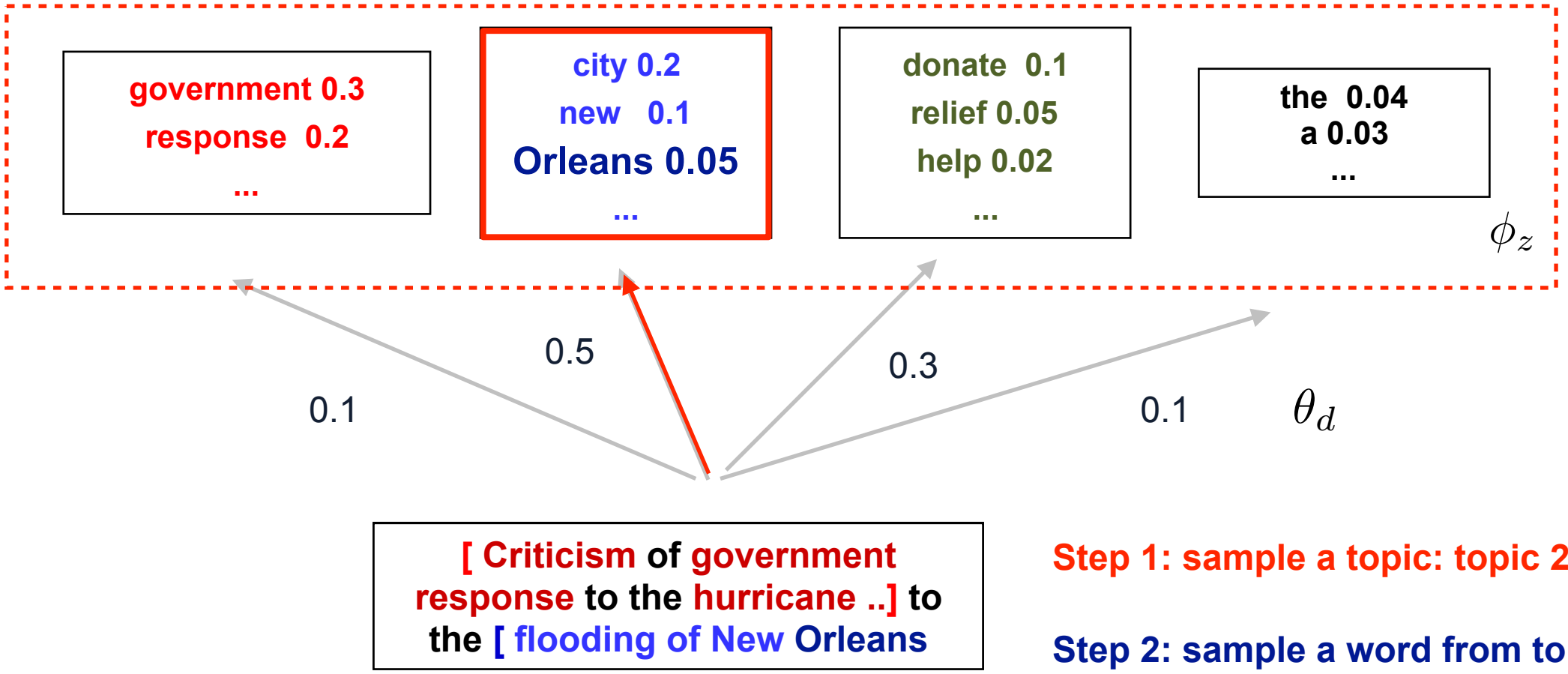
- Documents are generated by sampling words from k latent topics

- For each document d:
  - For each token position i
  - Choose a topic z ~ Multinomial(θ_d)
  - Choose a term w ~ Multinomial(ϕ_z)



Probabilistic Latent Semantic Analysis. Thomas Hoffman. 2001.

Review of PLSA



Step 1: sample a topic: topic 2

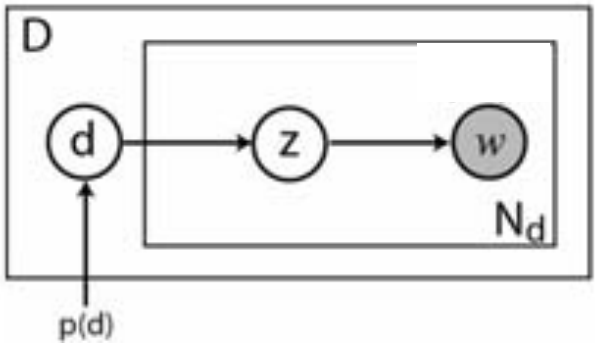
Step 2: sample a word from topic 2

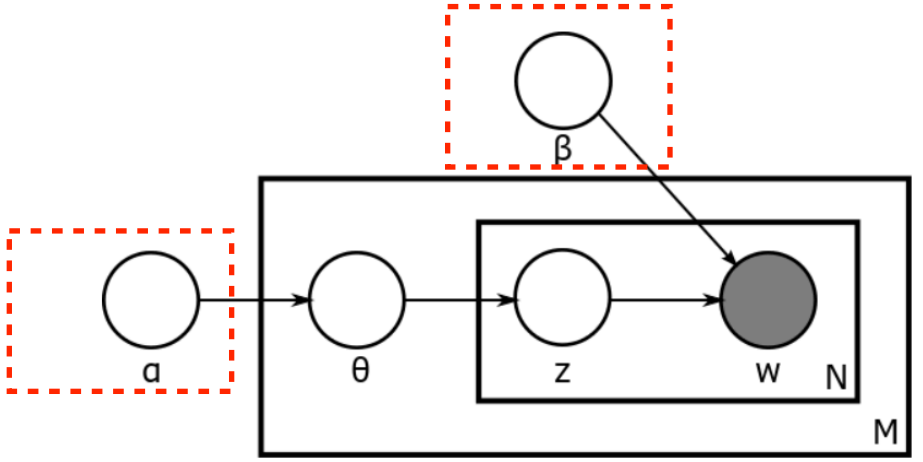
Probabilistic Latent Semantic Analysis. Thomas Hoffman. 2001.

Latent Dirichlet allocation [Blei et al. 2003]

- A generative statistical topic model

- Adding a prior hyper parameter alpha to PLSA model
  - The document-topic probability is sampled from the same prior distribution





Bayes' rules

Chain rule: joint distribution

$$P(A,B) = P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

Bayes' rule:

posterior likelihood prior

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \left[\frac{P(B|A)}{\sum_{X\in\{A,\overline{A}\}} P(B|X)P(X)}\right] P(A)$$

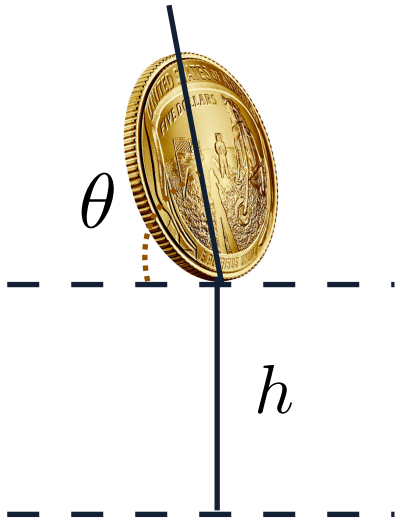
$$P(A|B) \propto P(B|A)P(A)$$

$$\sum_{A} P(A|B) = 1$$

skipping estimating $P(B)$

trick for estimating the posterior

Frequentist vs. Bayesian inference



- Frequentist treats parameters as **fixed**

sequence = 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0

$$\mu = \frac{\#ups}{\#ups + \#downs}$$

$$x_i \sim \mu$$

- Bayesians treats parameters as **random**
  - Consider the force that causes the coin to be biased
  - The forces are explained a prior distribution:

$$x_i \sim \mu$$

$$\mu \sim Beta(\alpha, \beta)$$

Maximum a Posterior (MAP) estimation

- Maximum likelihood estimation:

$$\theta_{MLE} = arg\max_{\theta} p(x_1, \cdots, x_n|\theta)$$

- Choose the parameter with the maximum posterior probability

$$\begin{aligned}\theta_{MAP} &= arg\max_{\theta} p(\theta|x_1, \cdots, x_n) \\ &= arg\max_{\theta} p(x_1, \cdots, x_n|\theta)p(\theta)\end{aligned}$$

How many parameters in PLSA vs. LDA? PLSA: d*k + k*V, LDA: d*k+k*V + d + V

PLSA -> LDA

- PLSA:

$$p_d\left(w \mid \{\theta_d\}, \{\phi_z\}\right) = \sum_{z=1}^{k} \theta_{d,z}\phi_{z,w}$$

$$\log p\left(D \mid \{\phi_z\}, \{\theta_d\}\right) = \sum_{d \in D}\sum_{w \in V} c(w,d) \log \left[\sum_{z=1}^{k} \theta_{d,z}\theta_{z,w}\right]$$

- LDA:

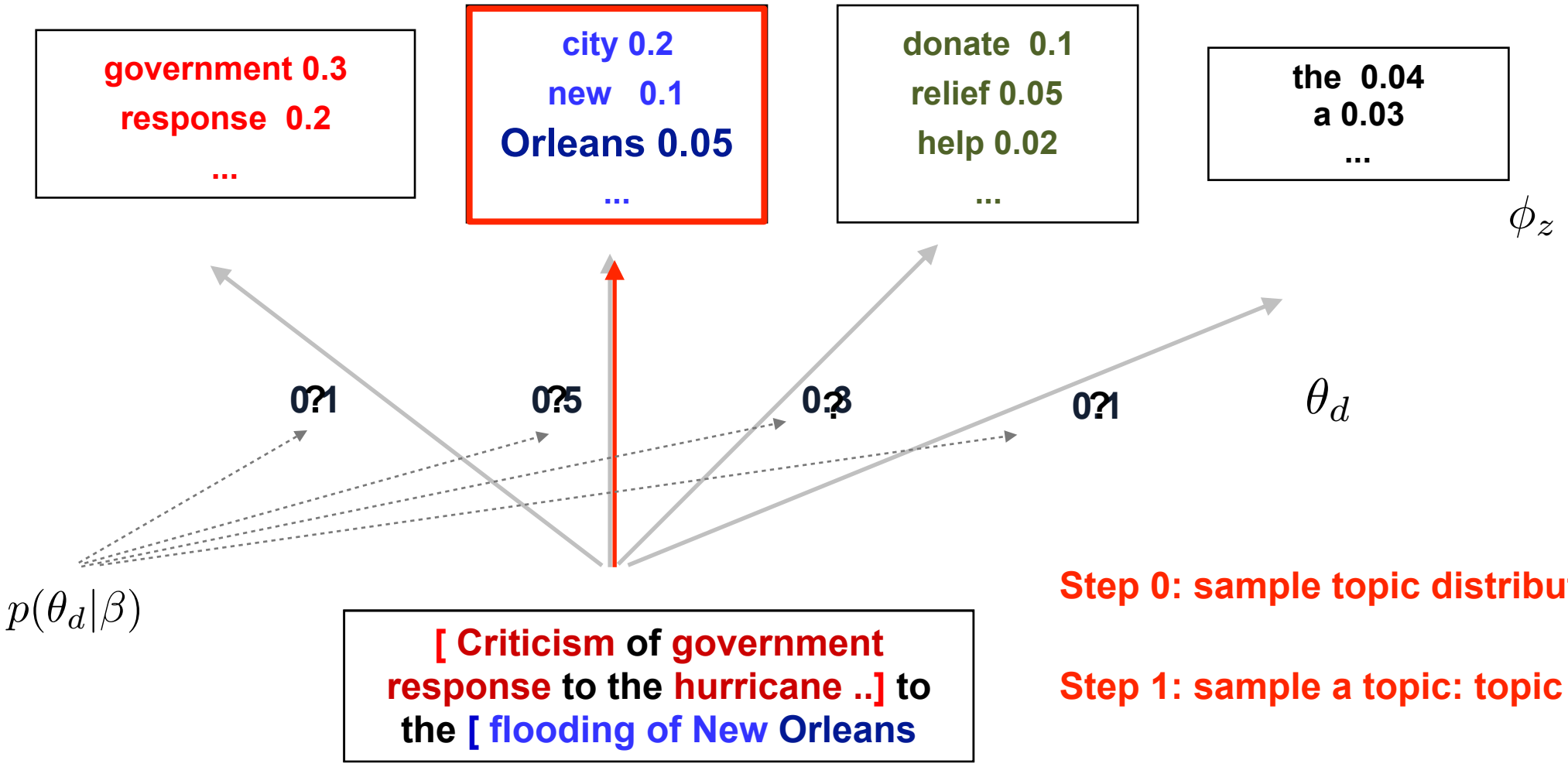
$$p_d\left(w \mid \{\theta_d\}, \{\phi_z\}\right) = \sum_{z=1}^{k} \theta_{d,z}\phi_{z,w}$$

$$\log p\left(d \mid \alpha, \{\phi_z\}\right) = \int \sum_{w \in V} c(w,d) \log \left[\sum_{j=1}^{k} \theta_{d,z}\phi_{z,w}\right] p\left(\theta_d \mid \alpha\right) d\theta_d$$

marginalized probability:

$$\log p(D \mid \alpha, \beta) = \int \sum_{d \in D} \log p\left(d \mid \alpha, \{\phi_z\}\right) \prod_{z=1}^{k} p\left(\phi_z \mid \beta\right) d\phi_1 \ldots d\phi_k$$

PLSA -> LDA



Step 0: sample topic distribution

Step 1: sample a topic: topic 2

Step 2: sample a word from topic 2

Solving Maximum a Posteriori inference for LDA

- Maximum likelihood estimation:

$$\mathcal{L} = \log p(\mathcal{W} \mid \mathbf{R}, \Phi, \Theta) = \sum_{d}^{D} \sum_{d_i}^{N_d} \sum_{z}^{T} R_{(w_{d_i} z)} \left(\log \phi_{(z, w_{di})} + \log \theta_{(d,z)} \right)$$
$$+ \left(\sum_{d=1}^{D} \lambda_d \left(1 - \sum_{z=1}^{T} \theta_{(d,z)} \right) + \sum_{z=1}^{T} \sigma_k \left(1 - \sum_{w=1}^{V} \phi_{z,w} \right) \right]$$

M step:

$$\theta_{d,z} \propto \sum_{v=1}^{V} R_{d,v,z} count(v, d)$$

$$\phi_{z,v} \propto \sum_{d=1}^{D} [z_{d,v} == z] count(v, d)$$

E step:

$$R_{(w_{d_i,z})} \propto \phi_{z,v} \theta_{d,z}$$

Solving Maximum a Posteriori inference for LDA

- Exact inference is intractable:

$$p(Z, \Phi, \Theta | D, \alpha, \beta) = \frac{p(D, Z, \Phi, \Theta | \alpha, \beta)}{p(D | \alpha, \beta)}$$

$$\log p(D \mid \alpha, \beta) = \int \sum_{d \in D} \log p\left(d \mid \alpha, \{\phi_z\}\right) \prod_{z=1}^{k} p\left(\phi_z \mid \beta\right) d\phi_1 \ldots d\phi_k$$

- Equation (1) is computationally intractable due to the coupling of beta and phi in the denominator

- Question: **why do we need to compute the denominator?**

Variational inference

- Key idea: use a **surrogate distribution** to approximate the posterior distribution of latent variables
  - Surrogate distribution is simpler to estimate than the true posterior

- Goal: finding the “best” surrogate distribution from a certain parametric family by **minimizing** the KL-divergence between the surrogate function (Q) to the true posterior (P)

- Typical surrogate distributions
  - Mean-field approximation [Blei et al. 03]
  - Expectation propagation [Minka et al. 02]
  - Collapsed variational Bayes [Teh et al. 07]

Evidence lower bound (ELBO)

- Given that $p(Z|D) = \frac{p(D,Z)}{p(D)}$, we want to minimize the KL divergence between Q(Z) and p(Z|D):

$$
\begin{aligned}
D_{\mathrm{KL}}(q\|p) &= \sum_Z q(Z)\left[\log\frac{q(Z)}{p(Z,D)} + \log p(D)\right] \\
&= \sum_Z q(Z)[\log q(Z) - \log p(Z,D)] + \sum_Z q(Z)[\log p(D)] \\
&= \sum_Z q(Z)[\log q(Z) - \log p(Z,D)] + \log p(D)
\end{aligned}
$$

$$
\Rightarrow \log p(D) = D_{\mathrm{KL}}(q\|p) - \mathbb{E}_{\mathrm{q}}[\log q(Z) - \log p(Z,D)] = D_{\mathrm{KL}}(q\|p) + \mathcal{L}(q)
$$

Evidence lower bound (ELBO)

- Given that $p(Z|D) = \frac{p(D,Z)}{p(D)}$, we want to minimize the K
  Q(Z) and p(Z|D):

https://stats.stackexchange.com/questions/205506/why-do-we-use-the-mean-field-approximation-for-variational-bayes
https://www.cs.colorado.edu/~jbg/

$$
\begin{aligned}
D_{\mathrm{KL}}(q\|p) &= \int_Z q(Z)\left[\log\frac{q(Z)}{p(Z,D)} + \log p(D)\right] \\
&= \int_Z q(Z)[\log q(Z) - \log p(Z,D)] + \int_Z q(Z)[\log p(D)] \\
&= \int_Z q(Z)[\log q(Z) - \log p(Z,D)] + \log p(D)
\end{aligned}
$$

$$
\Rightarrow \log p(D) = D_{\mathrm{KL}}(q\|p) - \mathbb{E}_{\mathrm{q}}[\log q(Z) - \log p(Z,D)] = D_{\mathrm{KL}}(q\|p) + \mathcal{L}(q)
$$

Evidence lower bound (ELBO)

- p(D) does not rely on the latent variable Z:

$$\Rightarrow \log p(D) = D_{\mathrm{KL}}(q||p) - \mathbb{E}_{\mathrm{q}}[\log q(Z) - \log p(Z, D)] = D_{\mathrm{KL}}$$

https://stats.stackexchange.com/questions/205506/why-do-we-use-the-mean-field-approximation-for-variational-bayes
https://www.cs.colorado.edu/~jbg/

- Therefore minimizing KL divergence = maximizing L(q), which we call evidence lower bound, as KL divergence is non-negative

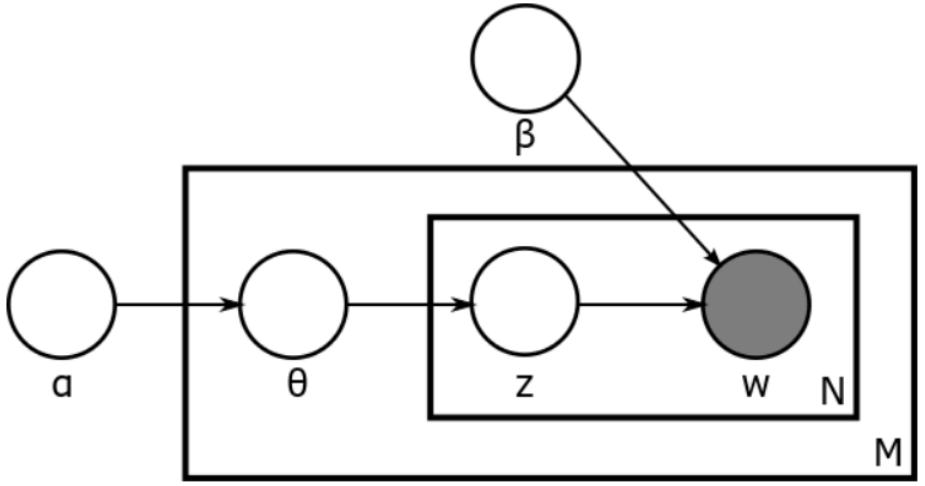
- The lower bound is more tractable to optimize than the evidence:

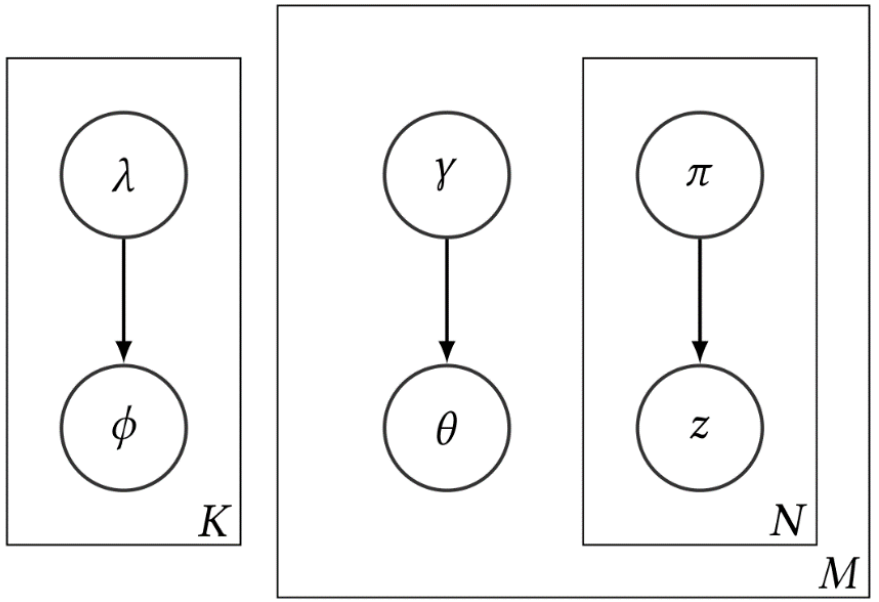
$$L(q) = \int_Z q(Z)[\log q(Z) - \log p(Z, D)]$$

LDA variational inference

- Assumption: q belongs to a family of distribution which can be factorized

$$q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q\left(z_n \mid \phi_n\right)$$





LDA variational inference

- ELBO of LDA under the factorization assumption:

$$L(\gamma, \phi; \alpha, \beta) = \mathrm{E}_q[\log p(\theta \mid \alpha)] + \mathrm{E}_q[\log p(\mathbf{z} \mid \theta)] + \mathrm{E}_q[\log p(\mathbf{w} \mid \mathbf{z}, \beta)] - \mathrm{E}_q[\log q(\theta)] - \mathrm{E}_q[\log q(\mathbf{z})]$$

$$\begin{aligned} L(\gamma, \phi; \alpha, \beta) = {} & \log \Gamma\left(\sum_{j=1}^{k} \alpha_j\right) - \sum_{i=1}^{k} \log \Gamma(\alpha_i) + \sum_{i=1}^{k} (\alpha_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right) \\ & + \sum_{n=1}^{N} \sum_{i=1}^{k} \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right) \\ & + \sum_{n=1}^{N} \sum_{i=1}^{k} \sum_{j=1}^{V} \phi_{ni} w_n^j \log \beta_{ij} \\ & - \log \Gamma\left(\sum_{j=1}^{k} \gamma_j\right) + \sum_{i=1}^{k} \log \Gamma(\gamma_i) - \sum_{i=1}^{k} (\gamma_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)\right) \\ & - \sum_{n=1}^{N} \sum_{i=1}^{k} \phi_{ni} \log \phi_{ni} \end{aligned}$$

no θ , why?

$$\mathrm{E}_q[\log(\theta_i) \mid \gamma] = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k} \gamma_j\right)$$

LDA variational inference

- LDA variational inference algorithm: optimizing γ, φ, β

1. Randomly initialize variational parameters (can't be uniform)
2. For each iteration:
    1. For each document, update γ and φ
    2. For corpus, update β
    3. Compute L for diagnostics
3. Return **expectation of variational parameters** for solution to latent variables

$$\gamma_i = \alpha_i + \sum_n \phi_{ni}$$

$$\phi_{ni} \propto \beta_{iv} \exp\left(\psi\left(\gamma_i\right) - \Psi\left(\sum_j \gamma_j\right)\right)$$

$$\beta_{ij} \propto \sum_d \sum_n \phi_{dni} w_{dn}^j$$

Pros and Cons of Variational inference

- Pros
  - Deterministic algorithm - easy to tell when converged
  - Parallelizable

- Cons:
  - Speed: make many calls to transcendental functions (no close form solution)
  - Quality is questionable due to the factorization assumption
  - Memory usage: requires O(MNK) to store the per-token variational distributions

Collapsed variational inference

- If your priors are conjugate, they can be integrated out!
- Let $\sigma_{j,k} = \sum_{z=1} \mathbf{1}\left(z_{j,z} = k\right)$ be the number of times topic z is observed in document j
- Let $\delta_{i,r} = \sum_{j=1} \sum_{z=1} \mathbf{1}\left(w_{j,t} = r\right)$ be the number of times word r is assigned to topic z

$$P(\boldsymbol{W}, \boldsymbol{Z} \mid \alpha, \beta) = P(\boldsymbol{Z} \mid \alpha) P(\boldsymbol{W} \mid \boldsymbol{Z}, \beta)_M$$

$$P(\boldsymbol{Z} \mid \alpha) = \int P(\Theta \mid \alpha) P(\boldsymbol{Z} \mid \Theta) d\Theta = \prod_{i=1}^{M} \frac{1}{B(\alpha)} \prod_{j=1}^{K} \theta_{j,i}^{\sigma_{j,i}+\alpha_i-1} d\theta_j = \prod_{j=1}^{M} \frac{B(\alpha+\sigma_j)}{B(\alpha)}$$

$$P(\boldsymbol{W}, \boldsymbol{Z} \mid \alpha, \beta) = \prod^{M} \frac{B\left(\alpha + \sigma_j\right)}{B(\alpha)} \prod^{K} \frac{B\left(\beta + \delta_i\right)}{B(\beta)}$$

B is beta function

Collapsed variational inference

$$P(\boldsymbol{W}, \boldsymbol{Z} \mid \alpha, \beta) = \prod_{j=1}^{M} \frac{B\left(\alpha + \sigma_j\right)}{B(\alpha)} \prod_{i=1}^{K} \frac{B\left(\beta + \delta_i\right)}{B(\beta)}$$

- How do we get back phi and theta? MAP estimates from σ and δ

$$\hat{\theta}_{j,k} = \frac{\sigma_{j,k} + \alpha_k}{\sum_{i=1}^{K} \sigma_{j,i} + \alpha_i} \text{ and } \hat{\phi_{k,v}} = \frac{\delta_{k,v} + \beta_v}{\sum_{r=1}^{V} \delta_{k,r} + \beta_r}$$

- How do we get σ and δ ?
  - Gibbs sampling: sample values of Z, count from those examples

Collapsed Gibbs sampling

- **Key idea:** construct a well-behaved Markov chain such that
  1. the states of the chain represent an assignment of $\boldsymbol{Z}$
  2. state transitions occur between states that differ in only one $z_{j,t}$
  3. transition probabilities are based on the **full conditional**:
  4. $$P\Big(z_{j,t} = k \mid \boldsymbol{Z}_{\neg j,t}, \boldsymbol{W}, \alpha, \beta\Big) = \frac{P(z_{j,t} = k, \boldsymbol{Z}_{\neg j,t}, \boldsymbol{W} \mid \alpha,\ \beta)}{P(\boldsymbol{Z}_{\neg j,t}, \boldsymbol{W} \mid \alpha,\ \beta)} \propto P\Big(z_{j,t} = k, Z_{\neg j,t}, \boldsymbol{W} \mid \alpha, \beta\Big)$$
  5. where $\boldsymbol{Z}_{\neg j,t}$ is the set of assignments for $\boldsymbol{Z}$ without position $\big(j, t\big)$

Collapsed Gibbs sampling

- For each position (j, t), sample a new value of $z_{j,t}$ based on the current values of the rest of the z. Use the newly sampled value for computing the probability for the next sample.

$$P\left(z_{j,t} = k \mid Z_{\neg j,t}, W, \alpha, \beta\right) \propto \frac{\sigma_{j,k}^{\neg j,t} + \alpha_k}{\sum_{i=1}^{K} \sigma_{j,i}^{-j,t} + \alpha_i} \times \frac{\delta_{k,w_{j,t}}^{j,t} + \beta_{w_{j,t}}}{\sum_{r=1}^{V} \delta_{k,r}^{\neg j,t} + \beta_r}$$

Collapsed Gibbs sampling

			iteration 1	iteration 2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs sampling

			iteration 1	iteration 2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	?
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Slide source: Tom Griffiths' presentation at https://cocosci.berkeley.edu/tom/talks/compling.ppt

Collapsed Gibbs sampling

			iteration 1	iteration 2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	?
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

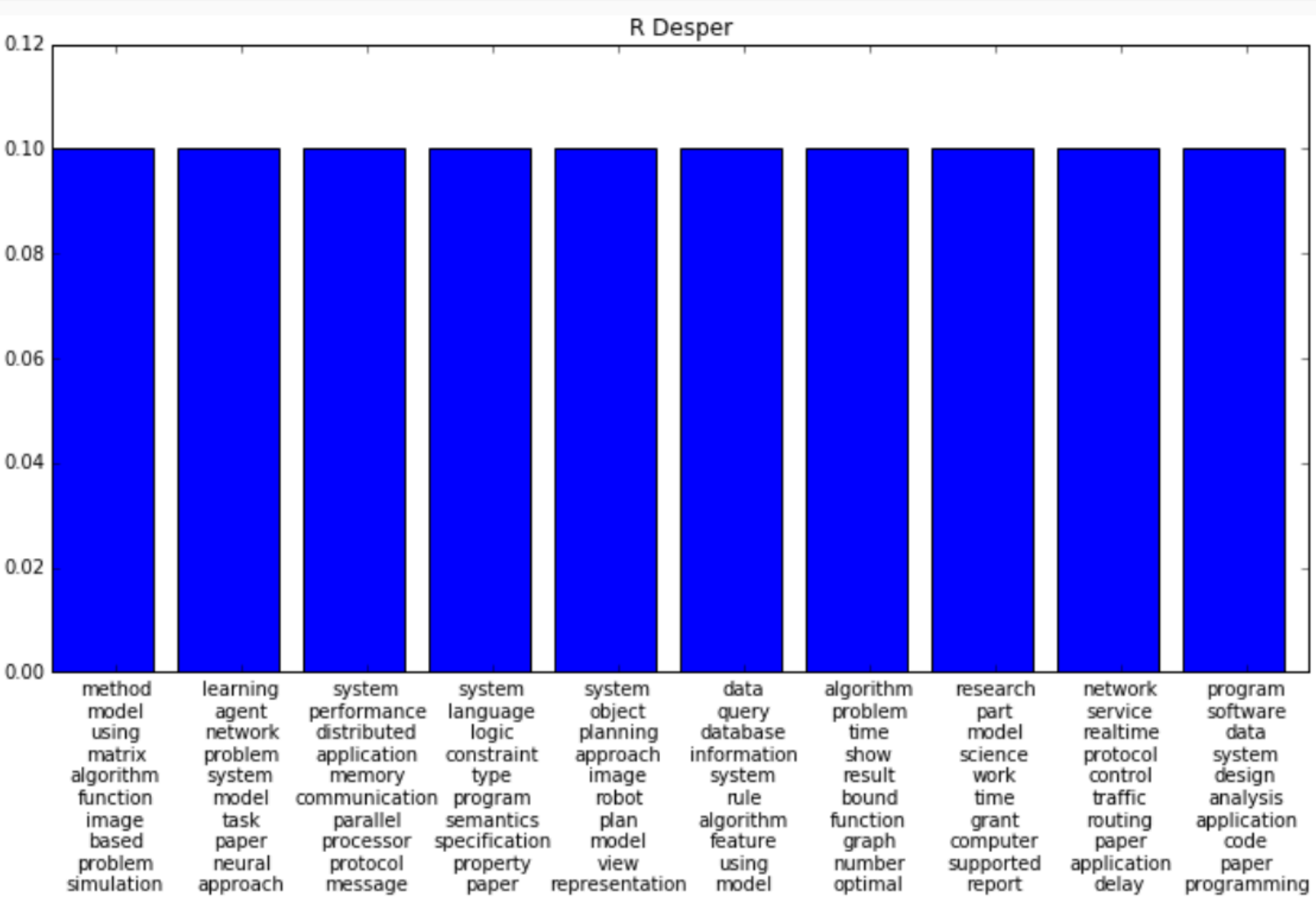
$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Pros/Cons of Gibbs sampling

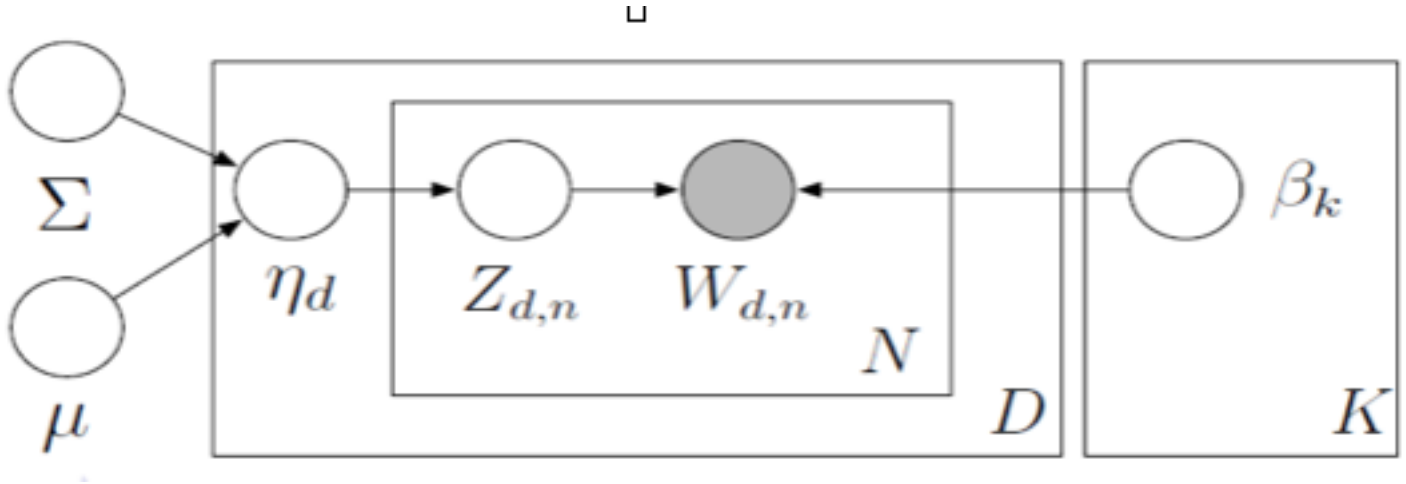
- Pros:
  - Ease of implementation
  - Fast iterations (no transcendental functions), fast convergence (at least relative to a full Gibbs sampler)
  - Low memory usage (only require $O(MN)$ storage for the current values of $z_{j,t}$)

- Cons:
  - No obvious parallelization strategy (each iteration depends on previous)
  - Can be difficult to assess convergence

LDA resu

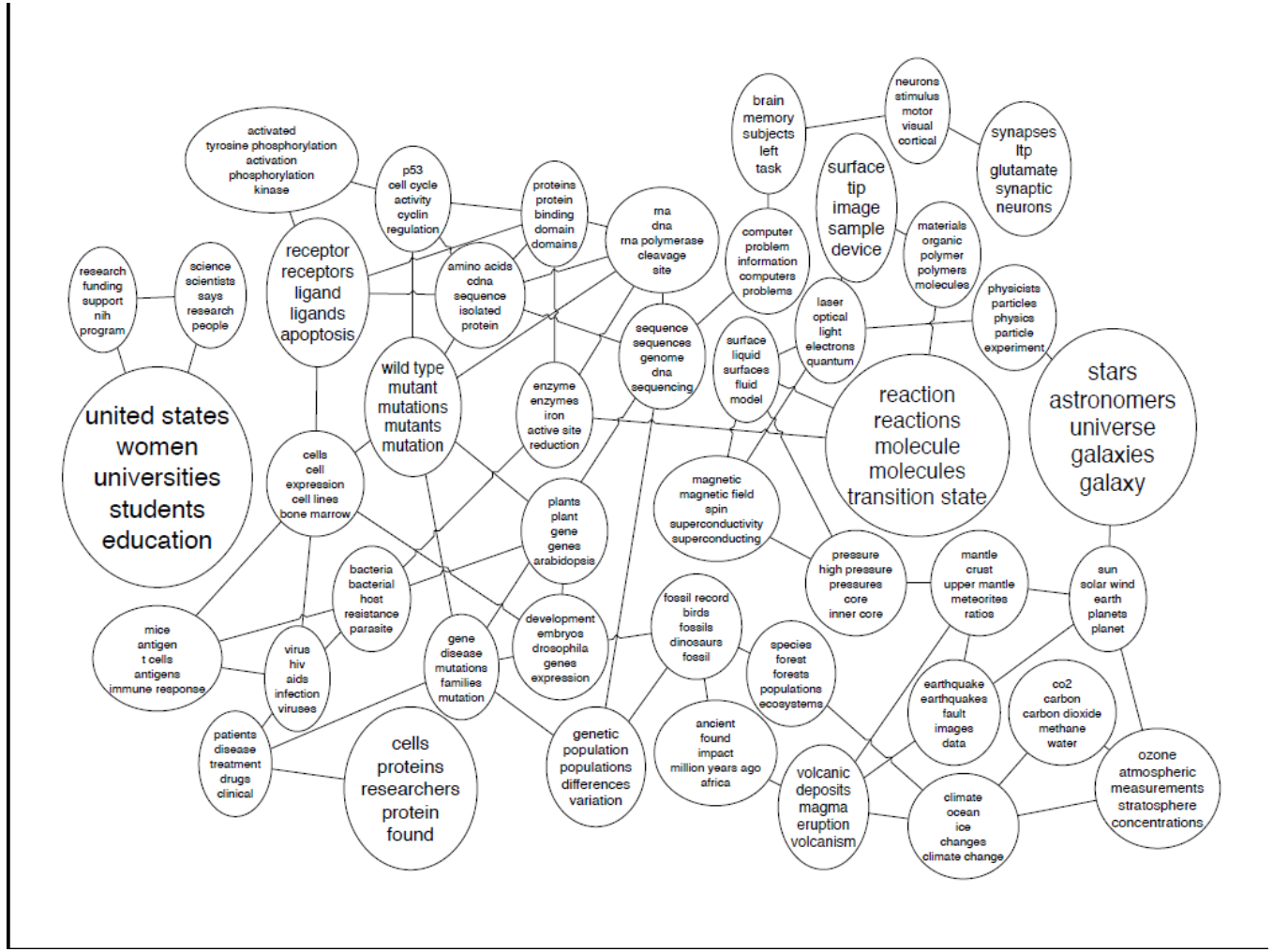

R Desper
0.12
0.10
0.08
0.06
0.04
0.02
0.00
method model using matrix algorithm function image based problem simulation
learning agent network problem system model task paper neural approach
system performance distributed application memory communication parallel processor protocol message
system language logic constraint type program semantics specification property paper
system object planning approach image robot plan model view representation
data query database information system rule algorithm feature using model
algorithm problem time show result bound function graph number optimal
research part model science work time grant computer supported report
network service realtime protocol control traffic routing paper application delay
program software data system design analysis application code paper programming

Correlated topic models (CTM) [Blei et al. 2005]

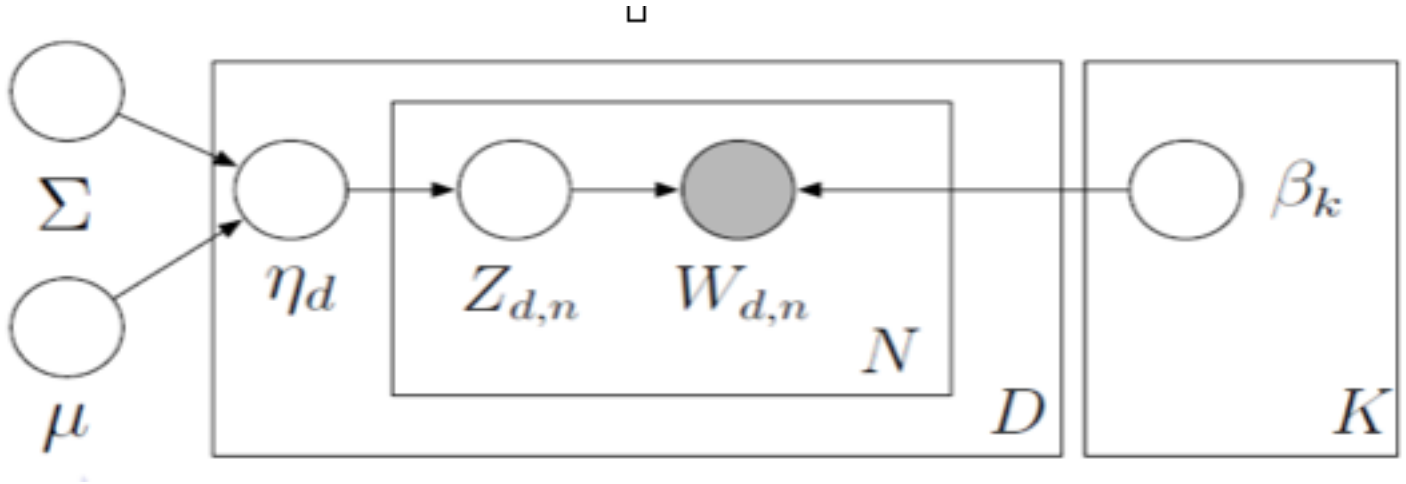


- Draw topics from a logistic normal, where topic occurrence can exhibit correlations

Correlated topic models (CTM) [Blei et al. 2005]

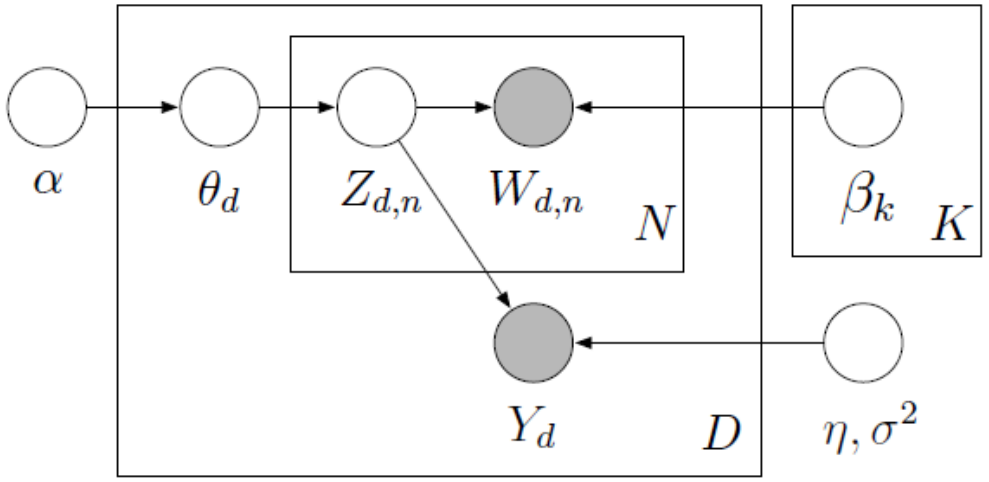


Correlated topic models (CTM) [Blei et al. 2005]



- Draw topics from a logistic normal, where topic occurrence can exhibit correlations

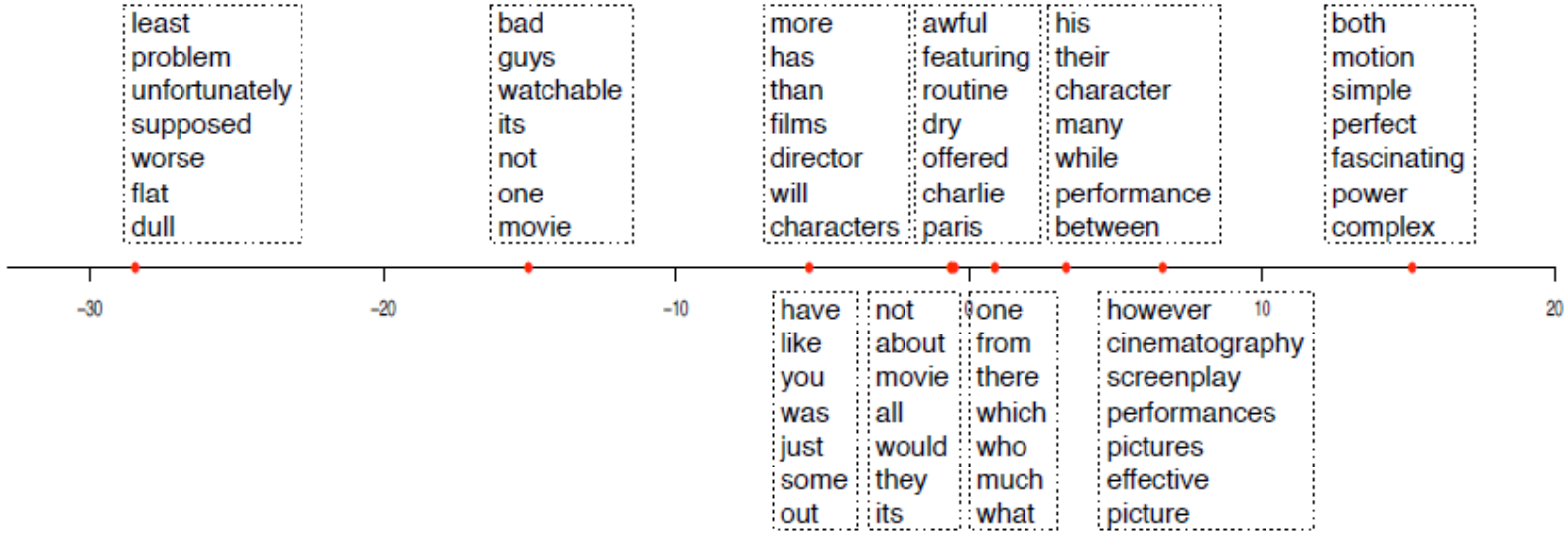
Supervised LDA [Blei & McAuliffe 07]



1. Draw topic proportions $\theta \mid \alpha \sim \mathrm{Dir}(\alpha)$.
2. For each word
   - Draw topic assignment $z_n \mid \theta \sim \mathrm{Mult}(\theta)$.
   - Draw word $w_n \mid z_n, \beta_{1:K} \sim \mathrm{Mult}(\beta_{z_n})$.
3. Draw response variable $y \mid z_{1:N}, \eta, \sigma^2 \sim \mathrm{N}(\eta^\top \bar{z}, \sigma^2)$, where

$$\bar{z} = (1/N) \sum_{n=1}^{N} z_n.$$

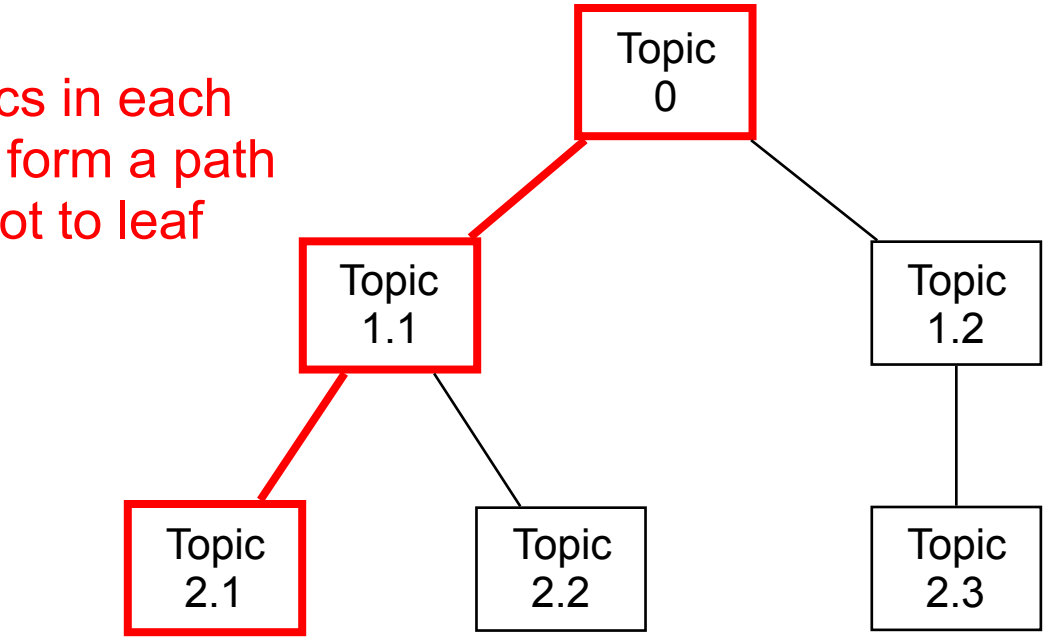
Supervised LDA [Blei & McAuliffe 07]



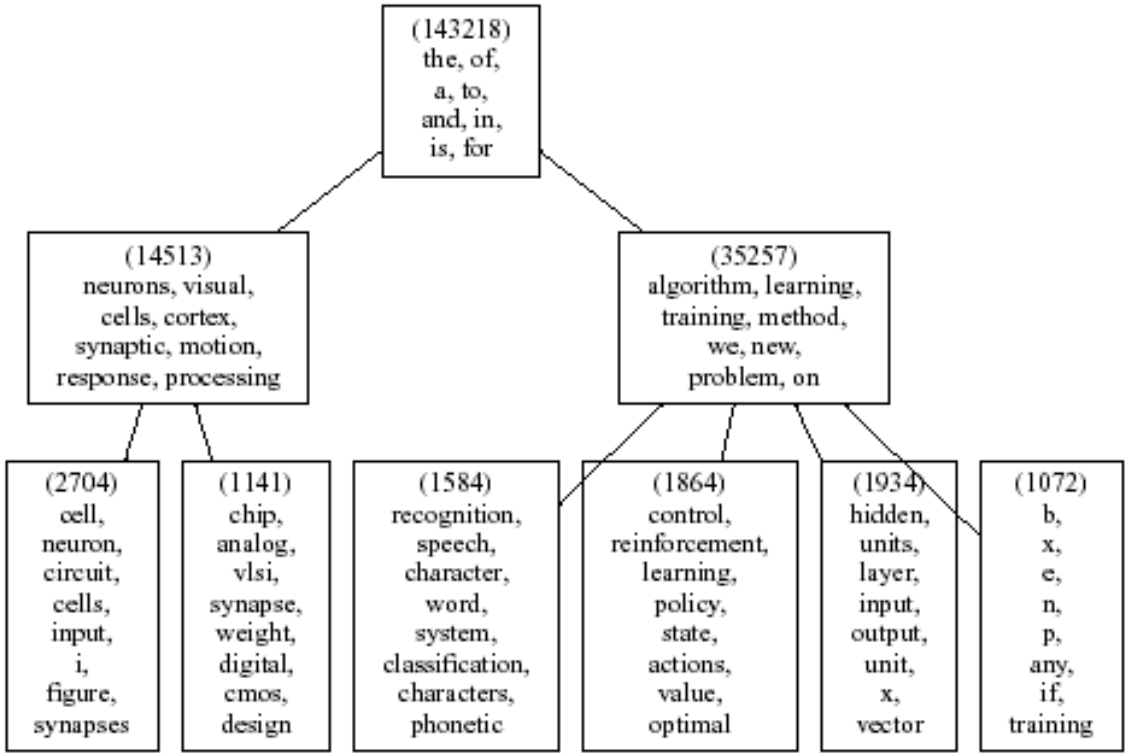
- 10-topic sLDA model on movie reviews (Pang and Lee, 2005).
- Response: number of stars associated with each review

Learning topic hierarchies

The topics in each document form a path from root to leaf



Learning topic hierarchies



Today's lecture

- Latent semantic indexing

- Continue on topic model
  - Bayesian inference of topic model
  - Variational inference for LDA
  - Gibbs sampling, Markov chain Monte-Carlo