Bayesian Inference for Mixture Language Models

Chase Geigle, ChengXiang Zhai

Department of Computer Science
University of Illinois, Urbana-Champaign

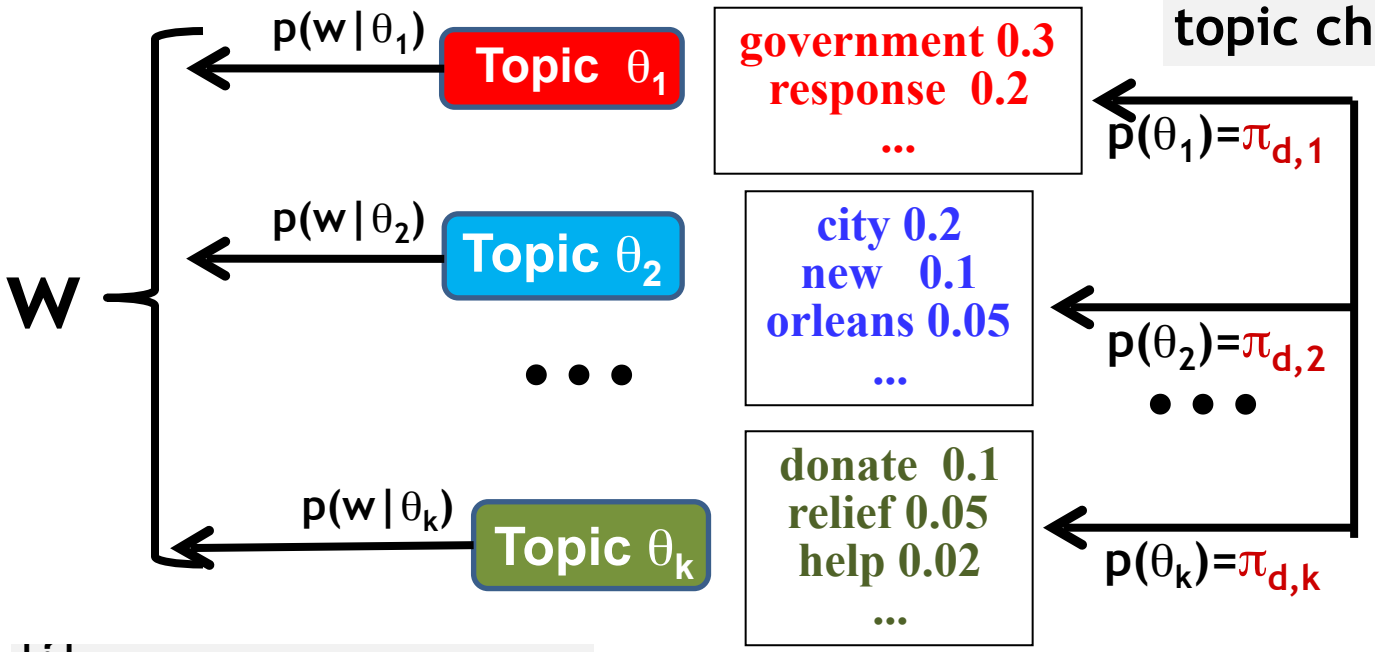
Deficiency of PLSA

- Not a generative model
  - Can't compute probability of a new document (why do we care about this?)
  - Heuristic workaround is possible, though
- Many parameters ➔ high complexity of models
  - Many local maxima
  - Prone to overfitting
- Overfitting is not necessarily a problem for text mining (only interested in fitting the "training" documents)

Latent Dirichlet Allocation (LDA)

- Make PLSA a generative model by imposing a Dirichlet prior on the model parameters ➔
  - LDA = Bayesian version of PLSA
  - Parameters are regularized
- Can achieve the same goal as PLSA for text mining purposes
  - Topic coverage and topic word distributions can be inferred using Bayesian inference

PLSA → LDA



Both word distributions and topic choices are free in PLSA

$\vec{\pi}_d = (\pi_{d,1},\ldots,\pi_{d,k})$

$p(\vec{\pi}_d) = \text{Dirichlet}(\vec{\alpha})$

$\vec{\alpha} = (\alpha_1,\ldots,\alpha_k), \alpha_i > 0$

$\vec{\theta}_i = (p(w_1 \mid \theta_i),\ldots,p(w_M \mid \theta_i))$

$p(\vec{\theta}_i) = \text{Dirichlet}(\vec{\beta})$

$\vec{\beta} = (\beta_1,\ldots,\beta_M), \beta_i > 0$

LDA imposes a prior on bot

Likelihood Functions for PLSA vs. LDA

PLSA

$$p_d(w|\{\theta_j\},\{\pi_{d,j}\}) = \sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)$$

Core assumption in all topic models

$$\log p(d|\{\theta_j\},\{\pi_{d,j}\}) = \sum_{w\in V} c(w,d)\log[\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)]$$

$$\log p(C|\{\theta_j\},\{\pi_{d,j}\}) = \sum_{d\in C}\log p(d|\{\theta_j\},\{\pi_{d,j}\})$$

LDA

$$p_d(w|\{\theta_j\},\{\pi_{d,j}\}) = \sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)$$

PLSA component

$$\log p(d|?,\{\theta_j\}) = \int\Big[\sum_{w\in V} c(w,d)\log[\sum_{j=1}^{k}\pi_{d,j}p(w|\theta_j)]\Big]p(\pi_d|?,?)d\pi_d$$

$$\log p(C|?,?) = \int\sum_{d\in C}\log p(d|?,\{\theta_j\})\prod_{j=1}^{k}p(\theta_j|?)d\theta_1\ldots d\theta_k$$

Added by LDA

Parameter Estimation and Inferences in LDA

- Parameters can be estimated using ML estimator

$$(\alpha^{?}, \beta^{?}) = \arg\max_{\alpha^{?},\beta^{?}} \log p(C \mid \alpha^{?}, \beta^{?})$$

How many parameters in LDA vs. PLSA?

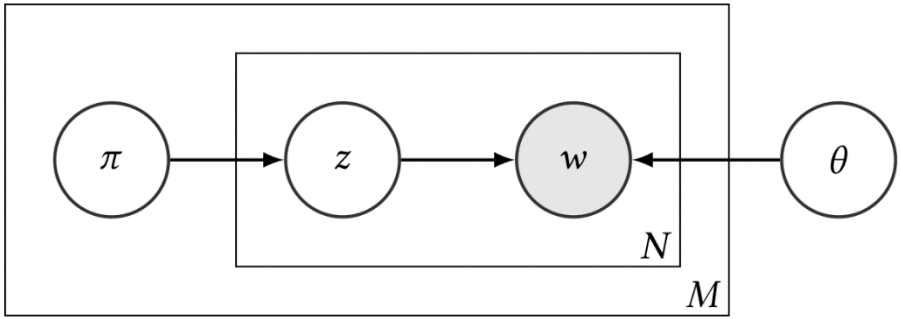
- However, $\{\theta_j\}$ **and** $\{\pi_{d,j}\}$ must now be computed using posterior inference

$$p(\{\theta_j\}, \{\pi_{d,j}\} \mid C, \alpha^{?}, \beta^{?}) = ?$$

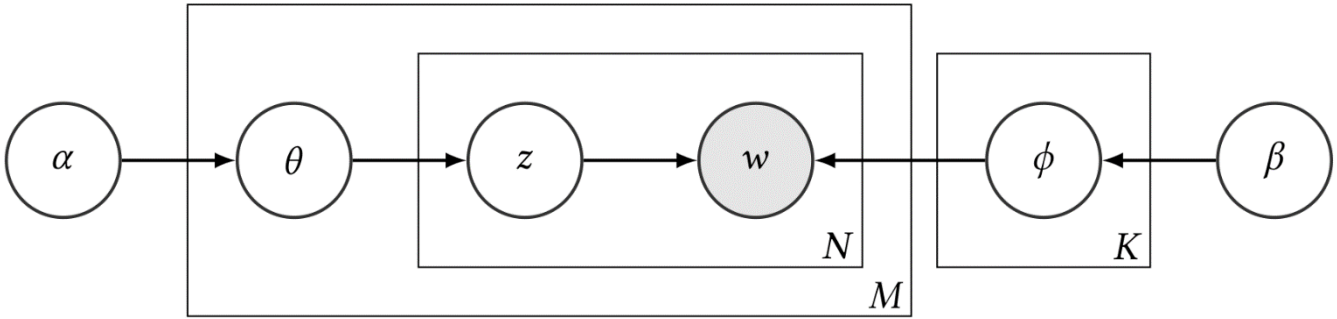
  - Computationally intractable
  - Must resort to approximate inference
  - Many different inference methods are available

Plate Notation

- PLSA:



- LDA:



Why is exact inference intractable?

$$P(\mathbf{Z}, \Theta, \Phi \mid \mathbf{W}, \alpha, \beta) = \frac{P(\mathbf{W}, \mathbf{Z}, \Theta, \Phi \mid \alpha, \beta)}{P(\mathbf{W} \mid \alpha, \beta)}$$

where

$$P(\mathbf{W} \mid \alpha, \beta) = \int_{\Phi} \int_{\Theta} \sum_{\mathbf{Z}} P(\mathbf{W}, \mathbf{Z}, \Theta, \Phi \mid \alpha, \beta) d\Theta d\Phi$$

$$= \int_{\Phi} p(\Phi \mid \beta) \int_{\Theta} p(\Theta \mid \alpha) \sum_{\mathbf{Z}} p(\mathbf{Z} \mid \Theta) p(\mathbf{W} \mid \mathbf{Z}, \Phi) d\Theta d\Phi$$

Denominator integral is intractable due to **coupling** between θ and ϕ in the summation over **Z**

Approximate Inference for LDA

- Many different ways; each has its pros & cons
- Deterministic approximation
  - variational Bayes [Blei et al. 03a]
  - collapsed variational Bayes [Teh et al. 07]
  - expectation propagation [Minka & Lafferty 02]
- Markov chain Monte Carlo
  - full Gibbs sampler [Pritchard et al. 00]
  - collapsed Gibbs sampler [Griffiths & Steyvers 04]

Very efficient and quite popular, but can only work with conjugate priors

Variational Inference

- Key idea: Use a **surrogate distribution** to approximate the posterior
  - Surrogate distribution is **simpler** than the **true posterior** (typically by making strong independence assumptions to break coupling)

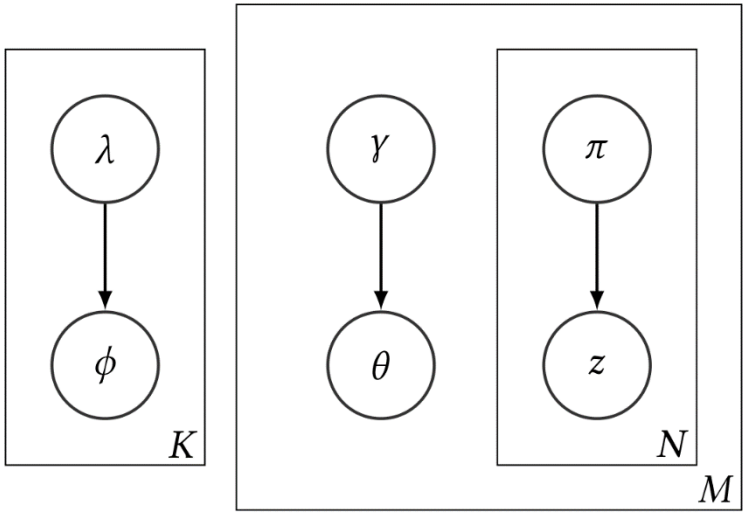
- **Goal:** Find the "best" surrogate distribution from a certain parametric family by minimizing the KL-divergence from the surrogate (Q) to the true posterior (P)
  - Transforms inference into a deterministic optimization!

  No free lunch: approximation quality depends on variational family chosen

A Mean Field Approximation for LDA

$$Q(\Phi, \mathbf{Z}, \Theta \mid \lambda, \pi, \gamma) = \prod_{i=1}^{K} Dir(\phi_i \mid \lambda_i) \prod_{j=1}^{M} q_j(\theta_j \mid \gamma_j) \prod_{t=1}^{N_j} q_j(z_{j,t} \mid \pi_{j,t})$$

- Uses a **fully factorized** surrogate distribution
- λ, γ, π are the **variational parameters**
- These are adjusted to minimize



Variational Inference for LDA

- Solve a series of constrained minimizations for each variational parameter (since they don't depend on each other)

- **Result:** a simple coordinate ascent algorithm (update each variable in turn and iterate until convergence)

$$\pi_{j,t,i} \propto \phi_{i,w_{j,t}} \exp\left(\Psi(\gamma_{j,i}) - \Psi\left(\sum_{k=1}^{K} \gamma_{j,k}\right)\right)$$

$$\gamma_{j,i} = \alpha_i + \sum_{t=1}^{N_j} \pi_{j,t,i}.$$

$$\lambda_{i,v} = \beta_v + \sum_{j=1}^{M} \sum_{t=1}^{N_j} \pi_{j,t,v} \mathbb{1}(w_{j,t} = v)$$

Variational Inference: (Non-exhaustive) Pros/Cons

- Pros:
  1. Deterministic algorithm---easy to tell when you've converged
  2. Embarrassingly parallel over documents

- Cons:
  1. Speed: makes many calls to transcendental functions
  2. Quality: fully factorized variational distribution is a questionable approximator
  3. Memory usage: requires to store the per-token variational distributions

Collapsed Inference Algorithms

- **If your priors are conjugate**, they can be integrated out!
- **Result:** a simpler distribution (less variables) from which to build inference methods

- Let be the number of times topic is observed in document
- Let be the number of times word type is assigned topic

-

so then

where is the multivariate Beta function.

Collapsed Inference Algorithms (cont'd.)

- Distribution over only **W** and **Z**
- How do we get back and ? MAP estimate from **Z**:
  and

- But how do we get the count vectors and ?
  1. Gibbs sampling: sample values of **Z**, count from those samples
  2. Variational inference: use a surrogate distribution to model the probability of **Z** and use its expectation ("expected counts")

Collapsed Gibbs Sampling for LDA

- **Key idea:** construct a well-behaved Markov chain such that
  1. the states of the chain represent an assignment of
  2. state transitions occur between states that differ in only one
  3. transition probabilities are based on the **full conditional**:

     where is the set of assignments for without position

- **Guaranteed** to produce **true samples** from the posterior!

  IF you run it for long enough…

Collapsed Gibbs Sampling for LDA (cont'd.)

Algorithm: for each position , sample a new value of based on the current values of the rest of the . Use the newly sampled value for computing the probability for the next sample.

Key equation:

number of times topic occurs in document

number of times word type assigned to topic

count excluding current position

Gibbs sampling in LDA: Illustration

			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	

What's the most likely topic for w_i in d_i?

How likely would d_i choose topic j?

How likely would topic j generate word w_i ?

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

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

Gibbs sampling in LDA: Illustration

			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	

Count of all words assigned with topic j

words in d_i assigned with topic j

Count of cases where w_i is assigned with topic j

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

words in d_i assigned with any topic

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

Gibbs sampling in LDA: Illustration

			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

Gibbs sampling in LDA: Illustration

			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	

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

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

Gibbs sampling in LDA: Illustration

			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	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

Gibbs sampling in LDA: Illustration

			iteration			
			1	2	…	1000
i	w_i	d_i	z_i	z_i		z_i
1	MATHEMATICS	1	2	2		2
2	KNOWLEDGE	1	2	1		2
3	RESEARCH	1	1	1		2
4	WORK	1	2	2		1
5	MATHEMATICS	1	1	2		2
6	RESEARCH	1	2	2		2
7	WORK	1	2	2		2
8	SCIENTIFIC	1	1	1	…	1
9	MATHEMATICS	1	2	2		2
10	WORK	1	1	2		2
11	SCIENTIFIC	2	1	1		2
12	KNOWLEDGE	2	1	2		2
.	.	.	.	.		.
.	.	.	.	.		.
.	.	.	.	.		.
50	JOY	5	2	1		1

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

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

Collapsed Gibbs Sampling: (Non-exhaustive) Pros/Cons

- Pros:
  - Ease of implementation
  - Fast iterations (no transcendental functions), fast convergence (at least relative to a full Gibbs sampler)
  - Low memory usage (only require storage for the current values of)
- Cons:
  - No obvious parallelization strategy (each iteration depends on previous)
  - Can be difficult to assess convergence
    - "Variational inference is that thing you implement while waiting for your Gibbs sampler to converge." – David Blei

Other Inference Variations

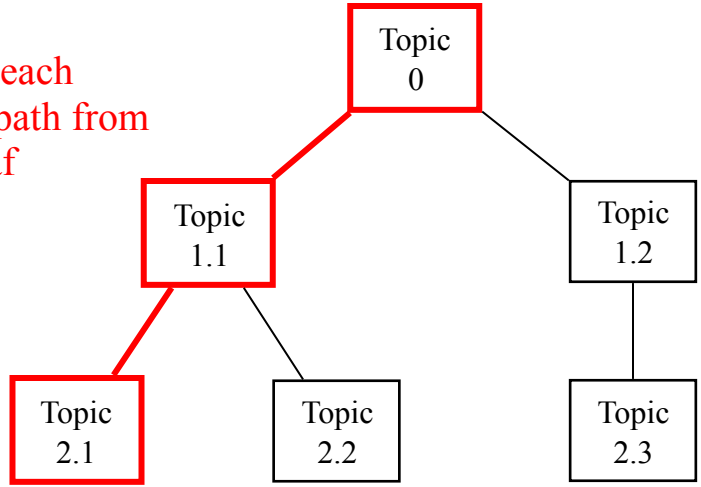
- Parallelization: AD-LDA [Newman et al. 09]
  - **Idea:** Distribute documents across cores (or machines)
    - locally maintain “differences” instead of updating globally shared counts
    - Resolve these “differences” once all documents have been sampled
    - Can be used for both CGS and CVB/CVB0
- Online learning: Stochastic Variational Inference [Hoffman et al. 13]
  - **Idea:** Learn LDA models on streaming data (learned Wikipedia topic model with low resource usage
  - Stochastic Collapsed VI also possible [Foulds et al. 13]

Extensions of LDA

- Capturing topic structures
- Capturing context
- With supervision

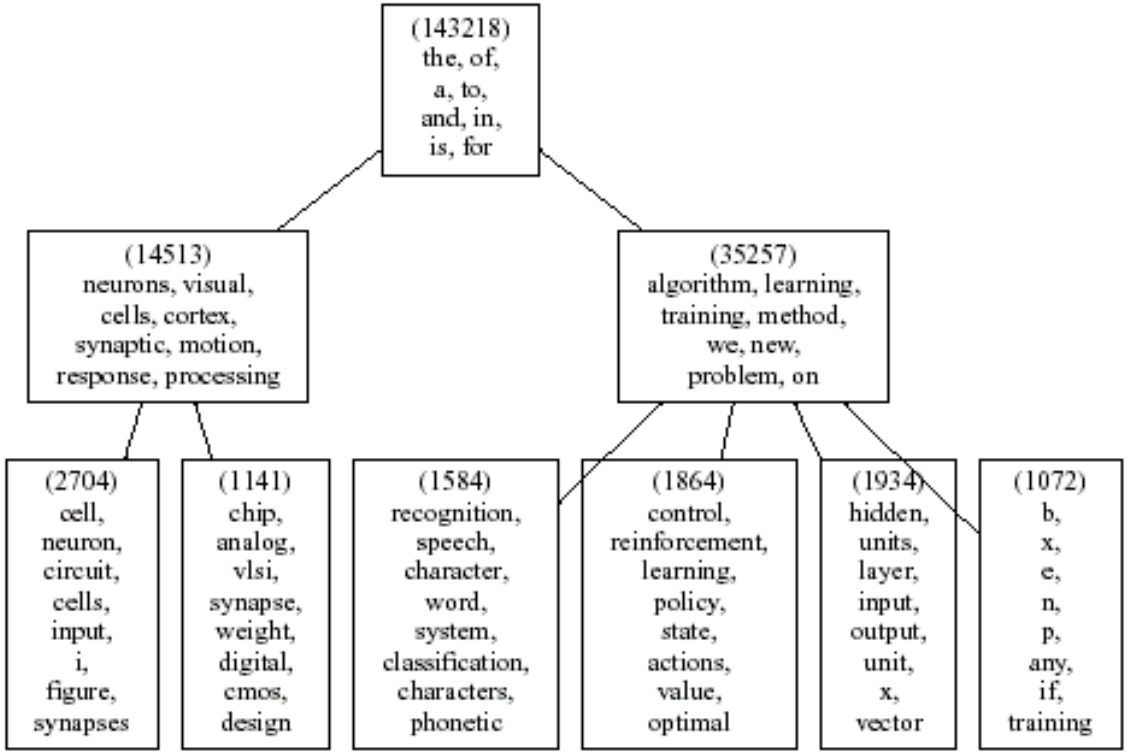
Learning topic hierarchies

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

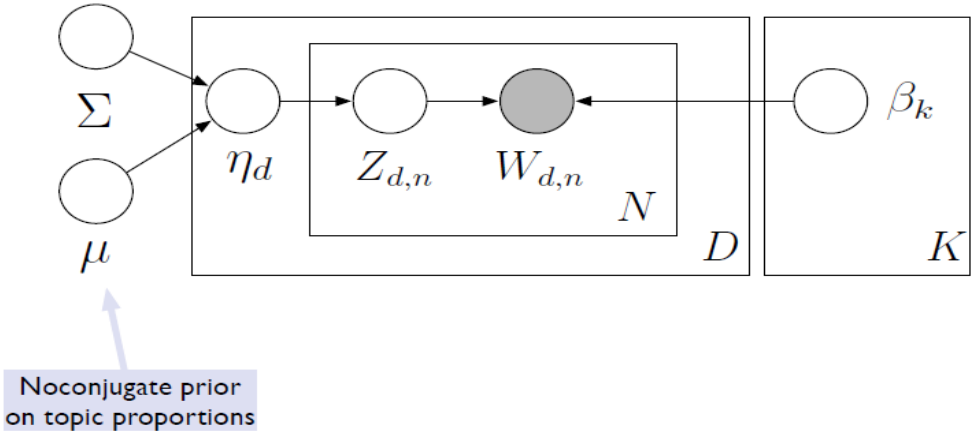


- Fixed hierarchies: [Hofmann 99c]
- Learning hierarchies:[Blei et al. 03b]

Twelve Years of NIPS [Blei et al. 03b]

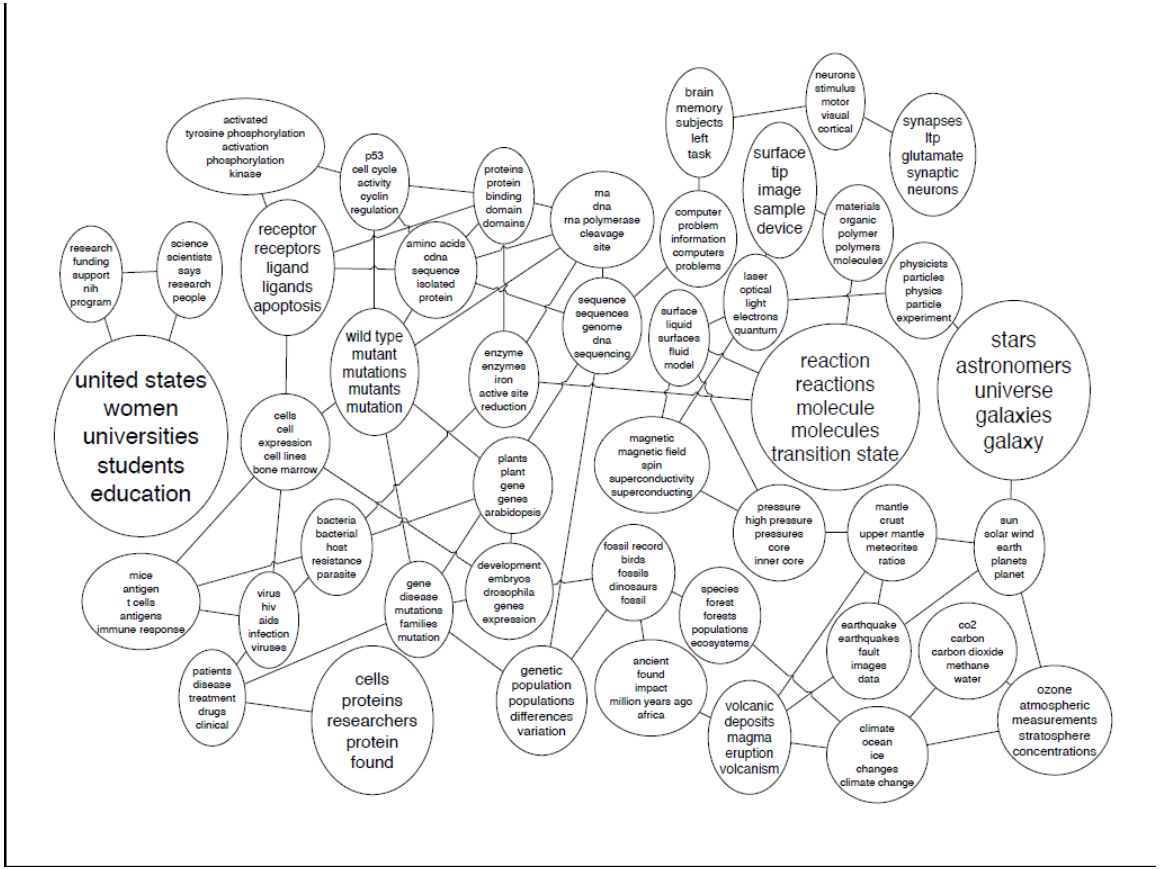


Capturing Topic Structures: Correlated Topic Model (CTM) [Blei & Lafferty 05]



- Draw topic proportions from a logistic normal, where topic occurrences can exhibit correlation.
- Use for:
  - Providing a "map" of topics and how they are related
  - Better prediction via correlated topics

Sample Result of CTM



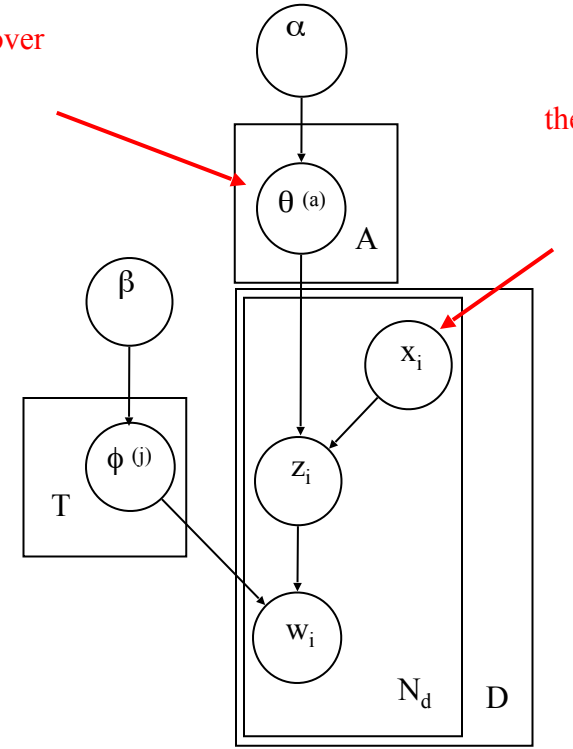
The Author-Topic model

[Rosen-Zvi et al. 04]

each author has a distribution over topics

$\theta^{(a)} \sim \text{Dirichlet}(\alpha)$

$\phi^{(j)} \sim \text{Dirichlet}(\beta)$



the author of each word is chosen uniformly at random

$x_i \sim \text{Uniform}(A^{(d)})$

$z_i \sim \text{Discrete}(\theta^{(x_i)})$

$w_i \sim \text{Discrete}(\phi^{(z_i)})$

Four example topics from NIPS

TOPIC 19	
WORD	**PROB.**
LIKELIHOOD	0.0539
MIXTURE	0.0509
EM	0.0470
DENSITY	0.0398
GAUSSIAN	0.0349
ESTIMATION	0.0314
LOG	0.0263
MAXIMUM	0.0254
PARAMETERS	0.0209
ESTIMATE	0.0204
AUTHOR	**PROB.**
Tresp_V	0.0333
Singer_Y	0.0281
Jebara_T	0.0207
Ghahramani_Z	0.0196
Ueda_N	0.0170
Jordan_M	0.0150
Roweis_S	0.0123
Schuster_M	0.0104
Xu_L	0.0098
Saul_L	0.0094

TOPIC 24	
WORD	**PROB.**
RECOGNITION	0.0400
CHARACTER	0.0336
CHARACTERS	0.0250
TANGENT	0.0241
HANDWRITTEN	0.0169
DIGITS	0.0159
IMAGE	0.0157
DISTANCE	0.0153
DIGIT	0.0149
HAND	0.0126
AUTHOR	**PROB.**
Simard_P	0.0694
Martin_G	0.0394
LeCun_Y	0.0359
Denker_J	0.0278
Henderson_D	0.0256
Revow_M	0.0229
Platt_J	0.0226
Keeler_J	0.0192
Rashid_M	0.0182
Sackinger_E	0.0132

TOPIC 29	
WORD	**PROB.**
REINFORCEMENT	0.0411
POLICY	0.0371
ACTION	0.0332
OPTIMAL	0.0208
ACTIONS	0.0208
FUNCTION	0.0178
REWARD	0.0165
SUTTON	0.0164
AGENT	0.0136
DECISION	0.0118
AUTHOR	**PROB.**
Singh_S	0.1412
Barto_A	0.0471
Sutton_R	0.0430
Dayan_P	0.0324
Parr_R	0.0314
Dietterich_T	0.0231
Tsitsiklis_J	0.0194
Randlov_J	0.0167
Bradtke_S	0.0161
Schwartz_A	0.0142

TOPIC 87	
WORD	**PROB.**
KERNEL	0.0683
SUPPORT	0.0377
VECTOR	0.0257
KERNELS	0.0217
SET	0.0205
SVM	0.0204
SPACE	0.0188
MACHINES	0.0168
REGRESSION	0.0155
MARGIN	0.0151
AUTHOR	**PROB.**
Smola_A	0.1033
Scholkopf_B	0.0730
Burges_C	0.0489
Vapnik_V	0.0431
Chapelle_O	0.0210
Cristianini_N	0.0185
Ratsch_G	0.0172
Laskov_P	0.0169
Tipping_M	0.0153
Sollich_P	0.0141

Dirichlet-multinomial Regression (DMR) [Mimno & McCallum 08]

Allows arbitrary features to be used to influence choice of topics

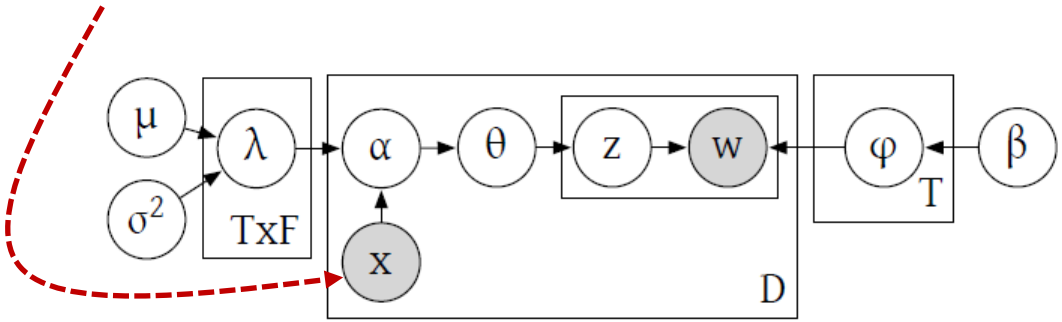
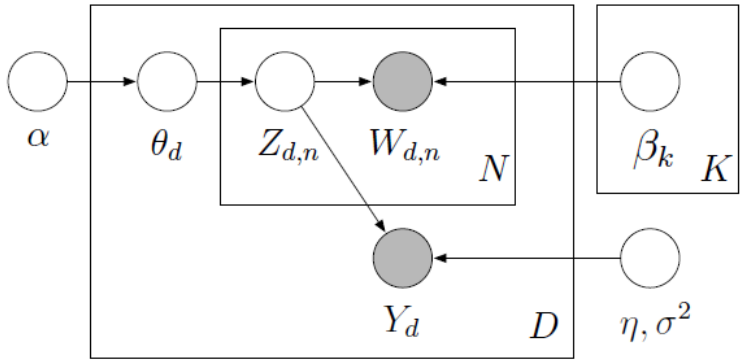


Figure 3: The Dirichlet-multinomial Regression (DMR) topic model. Unlike all previous models, the prior distribution over topics, α, is a function of observed document features, and is therefore specific to each distinct combination of metadata feature values.

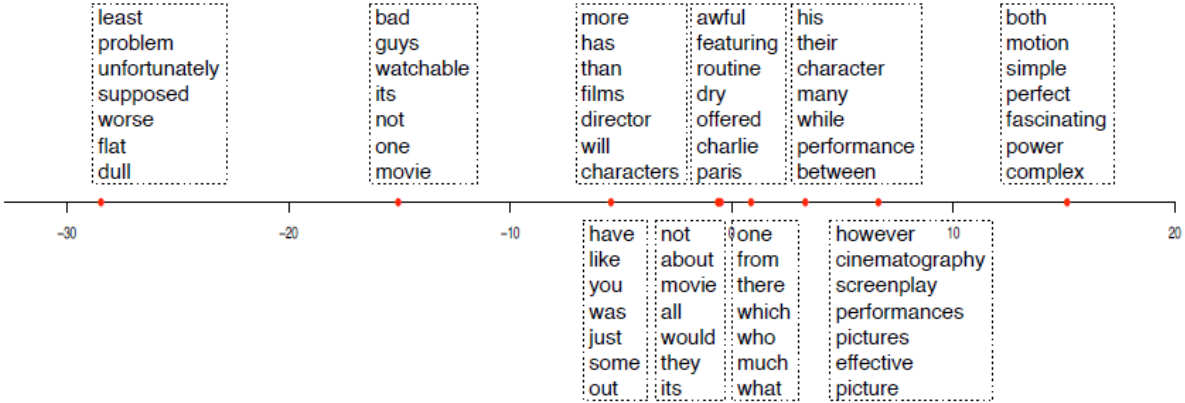
Supervised LDA [Blei & McAuliffe 07]



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

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

Sample Results of Supervised LDA



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

Challenges and Directions for Future Research

- **Challenge 1: How can we quantitatively evaluate the benefit of topic models for text mining?**
  - Currently, most quantitative evaluation is based on perplexity which doesn't reflect the actual utility of a topic model for text mining
  - Need to separately evaluate the quality of both topic word distributions and topic coverage (do more in line with "intrusion test"[Chang et al. 09])
  - Need to consider multiple aspects of a topic (e.g., coherent?, meaningful?) and define appropriate measures
  - Need to compare topic models with alternative approaches to solving the same text mining problem (e.g., traditional IR methods, non-negative matrix factorization)
  - Need to create standard test collections

Challenges and Directions for Future Research (cont.)

- **Challenge 2: How can we help users interpret a topic?**
  - Most of the time, a topic is manually labeled in a research paper; this is insufficient for real applications
  - Automatic labeling can help, but the utility still needs to evaluated
  - Need to generate a summary for a topic to enable a user to navigate into text documents to better understand a topic
  - Need to facilitate post-processing of discovered topics (e.g., ranking, comparison)

Challenges and Directions for Future Research (cont.)

- **Challenge 3: How can we address the problem of multiple local maxima?**
  - All topic models have the problem of multiple local maxima, causing problems with reproducing results
  - Need to compute the variance of a discovered topic
  - Need to define and report the confidence interval for a topic
- **Challenge 4: How can we develop efficient estimation and inference algorithms for sophisticated models?**
  - How can we leverage a user's knowledge to speed up inferences for topic models?
  - Need to develop parallel estimation/inference algorithms

Challenges and Directions for Future Research (cont.)

- **Challenge 5: How can we incorporate linguistic knowledge into topic models?**
  - Most current topic models are purely statistical
  - Some progress has been made to incorporate linguistic knowledge (e.g., [Griffiths et al. 04, Wallach 08])
  - More needs to be done
- **Challenge 6: How can we incorporate more sophisticated supervision into topic modeling to maximize its support for user tasks?**
  - Current models are mostly pre-specified with little flexibility for an analyst to "steer" the analysis process; need to develop a **general analysis framework** to enable an analyst to use multiple topic models together to perform complex text mining tasks
  - Need to combine topic modeling (as a generative model for feature learning) with **deep learning** to optimize task support