Latent Dirichlet Allocation

CSC 696H - Advanced Topics in Probabilistic Graphical Models

Yang Hong
hong1@arizona.edu

University of Arizona

September 19, 2022

Outline

- Information Retrieval (IR)
  - tf-idf scheme
  - Unigram Model
  - Mixture of Unigram
- Latent Dirichlet Allocation (LDA)
- Latent Semantic Indexing (LSI) and probabilistic LSI (pLSI)
  - pLSI Model

Latent Dirichlet Allocation (LDA)

- Notation and terminology
- LDA **
- Graphical model
- Exchangeability
- Variational Inference **
- VI algorithm
- Parameter optimization **
- EM algorithm
- Applications and empirical results

Information Retrieval (IR)

- Fundamental concepts behind Internet search engine
- Basic idea: document scoring and ranking
- “How to do a presentation in 45 minutes like a pro?”
  - Video: How to Make a Good PowerPoint Presentation
  - How to Build a Perfect 45 Minute Talk
  - Which is the best way to prepare a 45 minute presentation in a few days, including PowerPoint slides?
- Large result set not a problem, just show first 10
  - First page of Google search results

Tokenization in IR

- Document – File, email, newspaper article, tweet, Facebook post, etc. A column in the term-document incidence matrix.
- Token == Word – A delimited string of characters as it appears in a document.
- Term – A "**normalized**" (case, morphology, spelling etc) and **unique** word. It is included in the index.
- Type – An equivalence class of tokens (e.g., "USA" and "U.S.A"). Not necessarily in the index.

Normalization (Text Preprocessing)

- Example: We want to match **U.S.A.** and **USA**
- Interaction between Normalization and Language Detection
    - PETER IS TALKING TO MIT. → MIT = mit
    - Prof. Pacheco was a postdoc at MIT. → MIT $\neq$ mit
- stop words = extremely common words which would appear to be of little value in helping select documents
    - Examples: a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with
- Input:
  $\boxed{\text{Friends, Romans, countrymen.}}$ $\boxed{\text{So let it be with Caesar}}$ . . .
- Output:
  $\boxed{\text{friend}}$ $\boxed{\text{roman}}$ $\boxed{\text{countryman}}$ $\boxed{\text{so}}$ . . .
    - Natural Language Processing (NLP)

tf: term frequency

- We wish to rank documents that are more relevant **higher than** documents that are less relevant.

- The term frequency $\mathrm{tf}_{t,d}$ of term t in document d is defined as the number of times that t occurs in d.
- We want to use tf when computing query-document match scores.
- But how?
- Raw term frequency is not what we want because:
- A document with $\mathrm{tf} = 10$ occurrences of the term is more relevant than a document with $\mathrm{tf} = 1$ occurrence of the term.
- But not 10 times more relevant.
- Relevance does not increase proportionally with term frequency.

idf: inverse document frequency

- df_t is the document frequency, the number of documents that t occurs in.
- df_t is an inverse measure of the informativeness of term t.
- We define the idf weight of term t as follows:

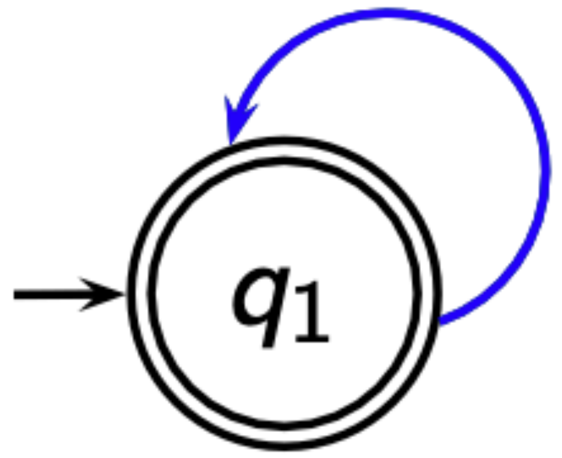
$$\mathrm{idf}_t = \log_{10} \frac{N}{\mathrm{df}_t}$$

  (N is the number of documents in the collection.)
- idf_t is a measure of the informativeness of the term.
- $[\log N/\mathrm{df}_t]$ instead of $[N/\mathrm{df}_t]$ to "dampen" the effect of idf
- Note that we use the log transformation for both term frequency and document frequency.

tf-idf scheme (weighting)

- The tf-idf weight of a term is the product of its tf weight and its idf weight.
- $$w_{t,d} = (1 + \log \mathrm{tf}_{t,d}) \cdot \log \frac{N}{\mathrm{df}_t}$$
- Best known weighting scheme in information retrieval
- Note: the “-” in tf-idf is a hyphen, not a minus sign!
- Alternative names: tf.idf, tf x idf

a Probabilistic Language Model



w	$P(w \mid q_1)$	w	$P(w \mid q_1)$
STOP	0.2	toad	0.01
the	0.2	said	0.03
a	0.1	likes	0.02
frog	0.01	that	0.04
		...	...

This is a one-state probabilistic finite-state automaton – a unigram language model – and the state emission distribution for its one state q_1. STOP is not a word, but a special symbol indicating that the automaton stops.

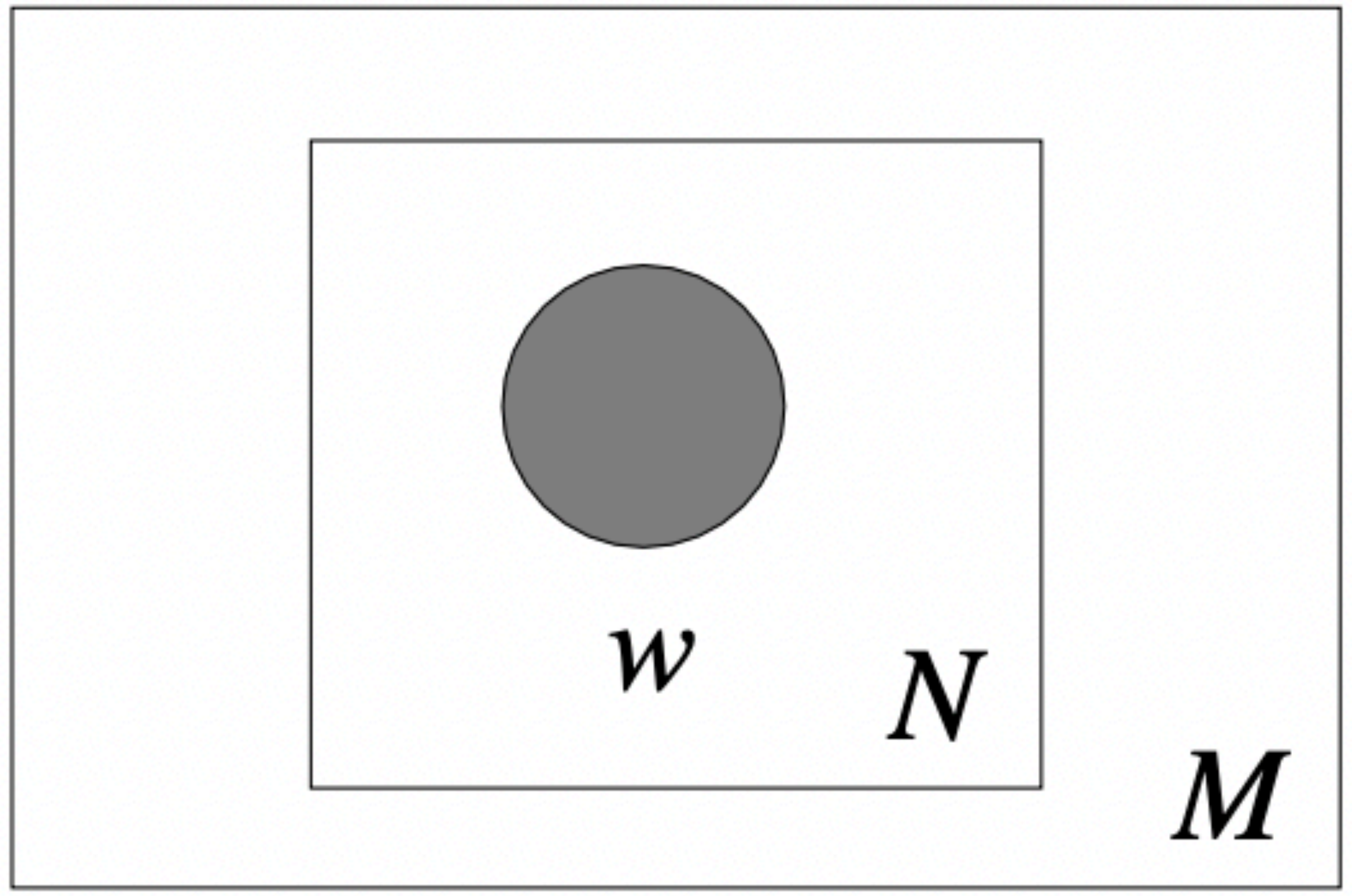
frog said that toad likes frog STOP

$P(\text{string}) = 0.01 \cdot 0.03 \cdot 0.04 \cdot 0.01 \cdot 0.02 \cdot 0.01 \cdot 0.2 = 0.000000000048$

Based on slides by Prof. Mihai Surdeanu in CSC 483/583 Text Retrieval and Web Search

Unigram model

- the words of every document are drawn **independently** from a multinomial distribution
- $$p(\mathbf{w}) = \prod_{n=1}^{N} p(w_n)$$
- $\mathbf{w}$:= a single document
- w_n := a single word
- N words



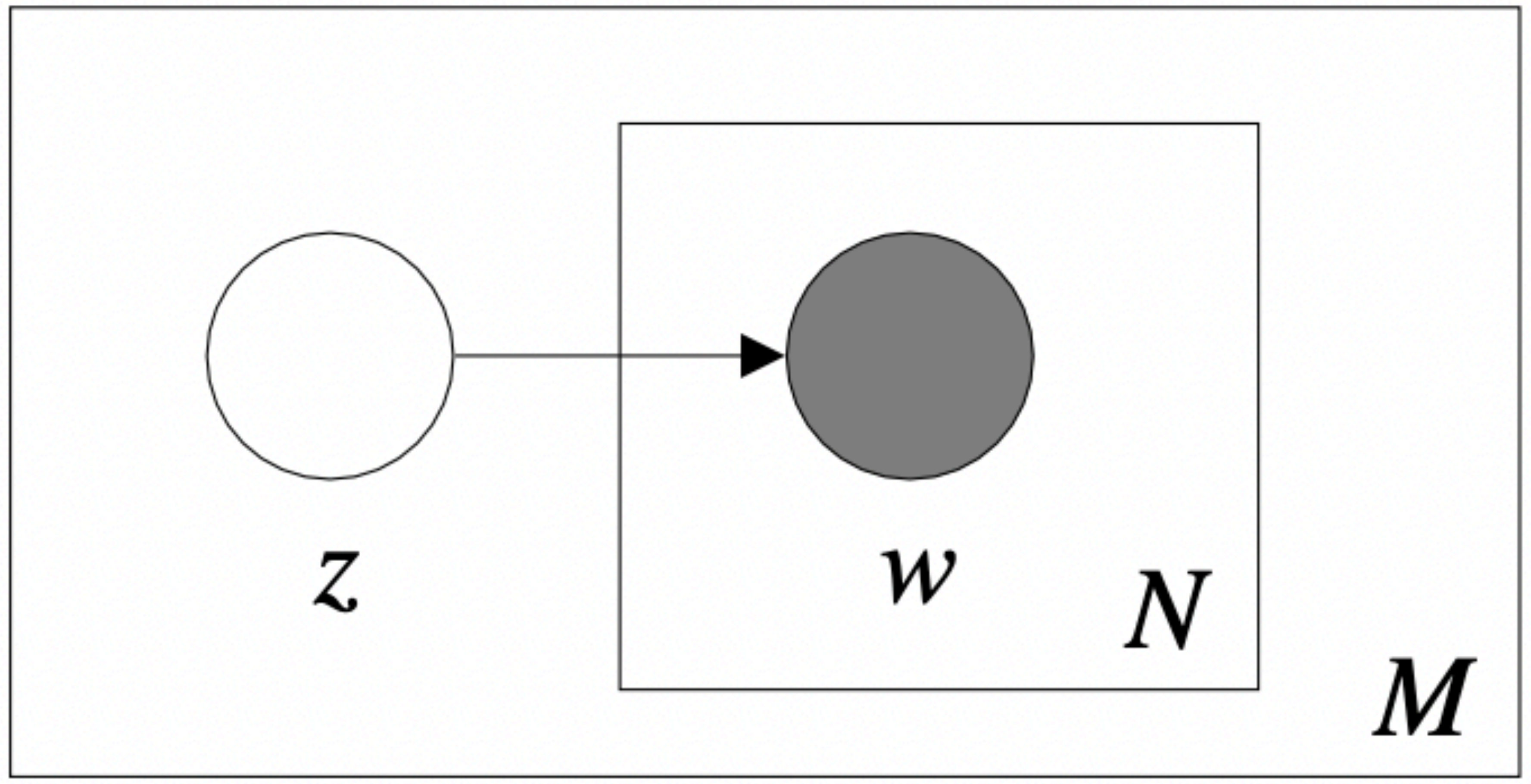
(a) unigram

Mixture of unigrams

- Introduce a discrete random topic variable z

- Choose a topic z, then generate N words independently from conditional multinomial distribution

- $$p(\mathbf{w}) = \sum_z p(z) \prod_{n=1}^{N} p(w_n \mid z)$$



(b) mixture of unigrams

Problems with Unigram and Mixture of unigrams

- Assuming 1 document is associated with 1 topic
  - Too limiting to effectively model a large collection of documents
- Offers little amount of reduction in description length
- Latent Semantic Indexing (LSI)
  - Requires linear algebra operations
    - dimensionality reduction
    - Singular value decomposition
  - probabilistic LSI (pLSI)
- “Bag of words” assumption

- LDA
  - 1 document exhibits multiple topics to different degrees
  - A dimensionality reduction technique in the spirit of LSI
    - But with proper underlying generative probabilistic semantics
  - This paper also assumes “bag of words”
    - Property of exchangeability
      - De Finetti’s Theorem
    - Can we do better?
      - Include a language model that describes the generation of sentences which would include the order dependence (the order of the words does matter)

Notation and terminology

- A word w := an item from a vocabulary indexed by {1, ..., V}
  - The basic unit of discrete data
  - How to construct the vocabulary for our task?
    - A topic is a distribution over the vocabulary
  - A unit-basis vector of shape V x 1 where v^{th} component is 1 and 0 elsewhere

v^{th} word

$\boxed{w^v}$

vocabulary

- A document $\mathbf{w}$:= $(w_1, w_2, w_3, \ldots, w_N)$
  - A sequence of N words where w_n is the n^{th} word in $\mathbf{w}$
  - What is w_n^v ?

n^{th} word

$\boxed{w_n}$

a sequence

- A text corpus D := $\left\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_M\right\}$
  - A collection of M documents
  - Notice $(\ldots)$ $\{\ldots\}$

m^{th} doc

$\boxed{\mathbf{w}_m}$

LDA in a nutshell

- Goal: want to find a model of a corpus that
  - Members of the corpus <= high probability (intra-doc)
  - “Similar” documents <= high probability (inter-docs)
- A generative probabilistic model of a corpus
  - Each $\mathbf{w}$: represented by random mixtures over latent topics z
  - Each z : characterized by a dist. over w(s), therefore a dist. over volcab.
- Three-level hierarchical Bayesian model

Generative process

- For each $\mathbf{w} \in D$:
  - Sample $N \sim \text{Poisson}(\xi)$
    - Not necessary: better distributions representing len$(\mathbf{w})$ as alternatives
    - Ancillary variable since $N \perp\!\!\!\perp \theta, \mathbf{z}$
  - Sample $\theta \sim \text{Dirichlet}(\alpha)$
    - Sample α by ancestral sampling
    - A probability vector of length k , a dist. over topics, a description of what a $\mathbf{w}$ is about
  - For each w_n :
    - Sample $z_n \sim \text{Multinomial}(\theta)$
      - Relationship between θ and z ?
    - Sample $w_n \sim p(w_n | z_n, \beta)$
      - a conditional multinomial probability assigning high probability to words relevant to z_n
- A generated document is literally a “bag of words” which is unreadable due to missing language structure but matches the statistics

Assumptions

- θ of dimensionality k => topic z of dimensionality k
  - z_n is a topic variable of length k for w_n
    - For simplicity, assume $z_n \sim \text{Categorical}(\theta)$
    - A special case of $\text{Multinomial}(\theta)$
- β : a k x V probability matrix

- θ lies in the $(k-1)$-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$ and has following pdf:

$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1}, \qquad (1)$$

  - α of length k and $\alpha_i \geq 0$ for $i \in \{1,\ldots,k\}$
- $\text{Dirichlet}(\alpha)$ is in the exponential family and forms a conjugate pair with $\text{Multinomial}(\theta)$
  - The property of conjugacy ensures that our posterior distribution takes a closed-form
  - Essential for variational inference (mean-field) and parameter estimation

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

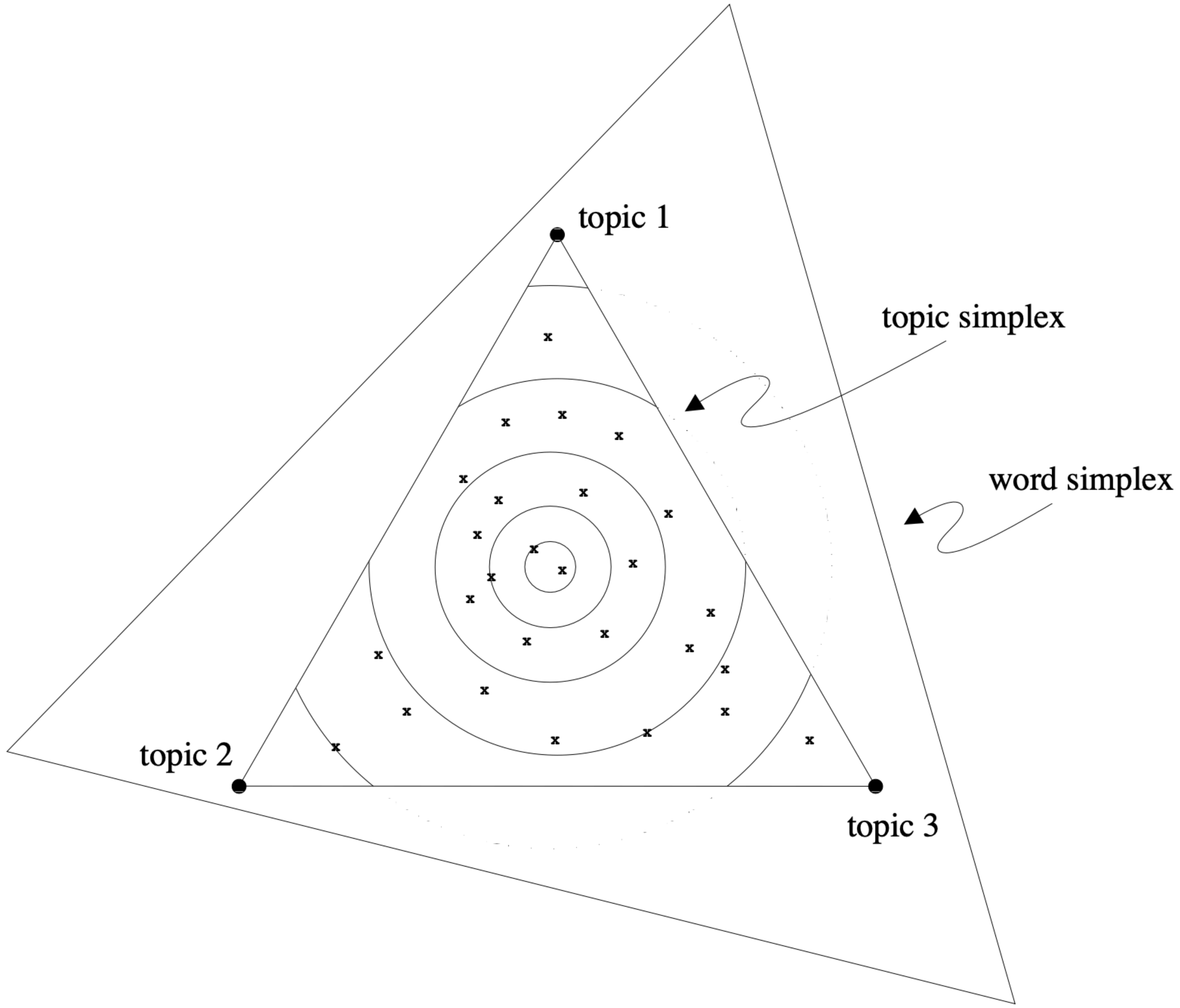
The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Figure 8: An example article from the AP corpus. Each color codes a different factor from which the word is putatively generated.

- Dirichlet constrains draws to lie in a probability simplex where $\sum_{n=1}^{k} \text{coordinates of } z_n = 1$ so it's a valid probability vector

- A continuous distribution on discrete probability distributions

- The generalization of $\text{Beta}()$



Other Conjugate Pairs

Likelihood	Model Parameters	Conjugate Prior
Normal	Mean	Normal
Normal	Mean / Variance	Normal-Inv-Gamma
Multivariate Normal	Mean / Variance	Normal-Inv-Wishart
Multinomial	Probability vector	Dirichlet
Gamma	Rate	Gamma
Poisson	Rate	Gamma
Exponential	Rate	Gamma

Wikipedia has a nice list of standard conjugate forms…

https://en.wikipedia.org/wiki/Conjugate_prior

- Joint distribution given the parameters α and β over a topic mixture $\theta, \mathbf{z}, \mathbf{w}$

  - $$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{2}$$

- Marginal distribution of a document $\mathbf{w}$

  - $$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta \tag{3}$$

- Probability of a text corpus

  - $$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d$$

  - What assumptions are facilitated here?

Three-levels to LDA representation

- α and β : corpus-level parameters, sampled once per generating a D
- θ_d : document-level variables, sampled once per $\mathbf{w}$
- z_{dn} and w_{dn} : word-level variables, sampled once for each $w \in \mathbf{w}$
  - z sampled repeatedly within the $\mathbf{w}$
- A classical Dirichlet-Multinomial clustering model is a two-level model
  - a Dirichlet sampled once per generating a D
  - a Multinomial clustering variable sampled once for each $\mathbf{w} \in D$
  - Restricts a $\mathbf{w}$ to being associated with a single z
- LDA enables a $\mathbf{w}$ to being associated with multiple $z(s)$

Graphical model

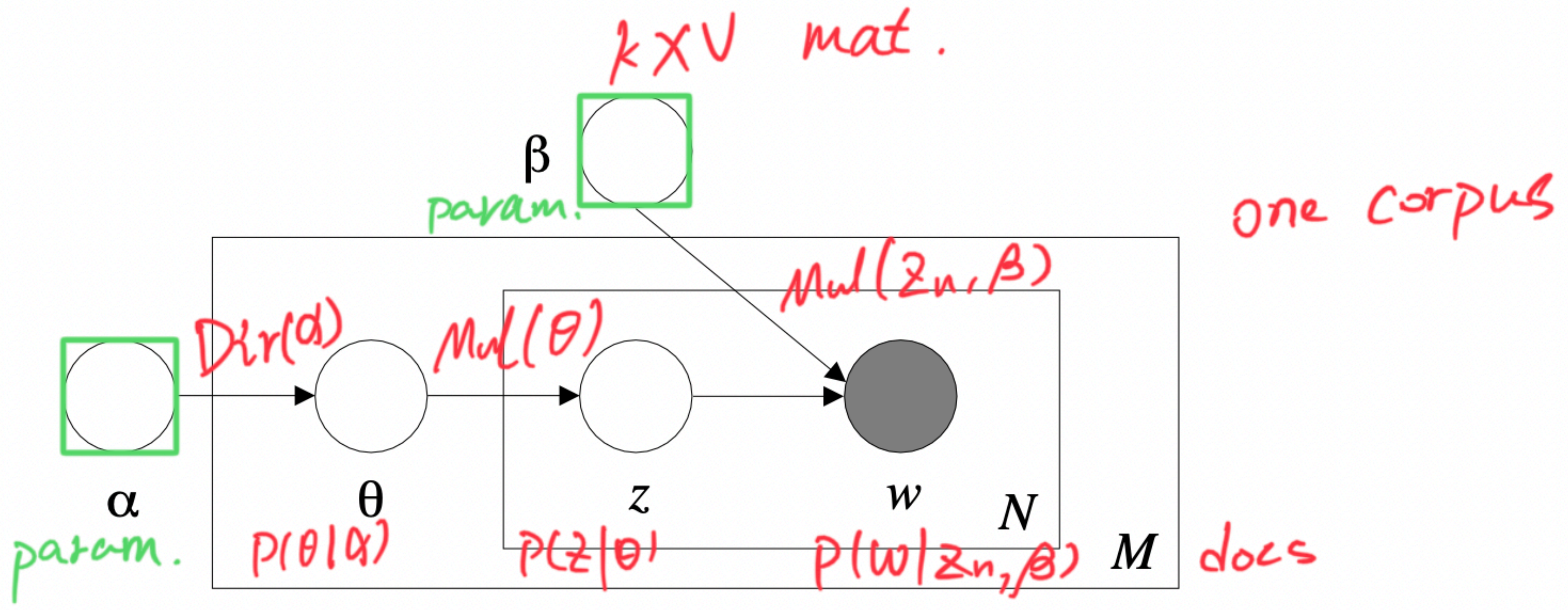


Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

- Latent random variables: do inference and compute posterior probabilities
- Parameters: do (maximum likelihood) estimation

Exchangeability

- Definition

A finite set of random variables $\{z_1, \ldots, z_N\}$ is said to be *exchangeable* if the joint distribution is invariant to permutation. If π is a permutation of the integers from 1 to N:

$$p(z_1, \ldots, z_N) = p(z_{\pi(1)}, \ldots, z_{\pi(N)}).$$

An infinite sequence of random variables is *infinitely exchangeable* if every finite subsequence is exchangeable.

Exchangeability

- De Finetti's representation theorem
  - Joint dist.(an **infinitely exchangeable** sequence of r.v.s) is as if
    - A r.param. $\sim \text{some dist}.()$
    - The r.v.s $\sim \text{i.i.d dist}.(\text{r.v.}\,|\,\text{param.})$
- Apply to LDA
  - $w_n \sim p(w_n \,|\, z_n, \beta)$ by fixed conditional distribution β
  - z are **infinitely exchangeable** within a $\mathbf{w}$
  - By de Finetti's theorem, the probability of a sequence of words and topics has the following form:
    - $p(\mathbf{w}, \mathbf{z}) = \int p(\theta) \left(\prod_{n=1}^{N} p(z_n \,|\, \theta) p(w_n \,|\, z_n) \right) d\theta$
  - We obtain LDA dist. on $\mathbf{w}$ in (3) by marginalizing out $\mathbf{z}$ variable and providing θ with a Dirichlet distribution

Intractability of the posterior distribution

The key inferential problem that we need to solve in order to use LDA is that of computing the posterior distribution of the hidden variables given a document:

$p(\theta, z, w . \alpha, \beta)$

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

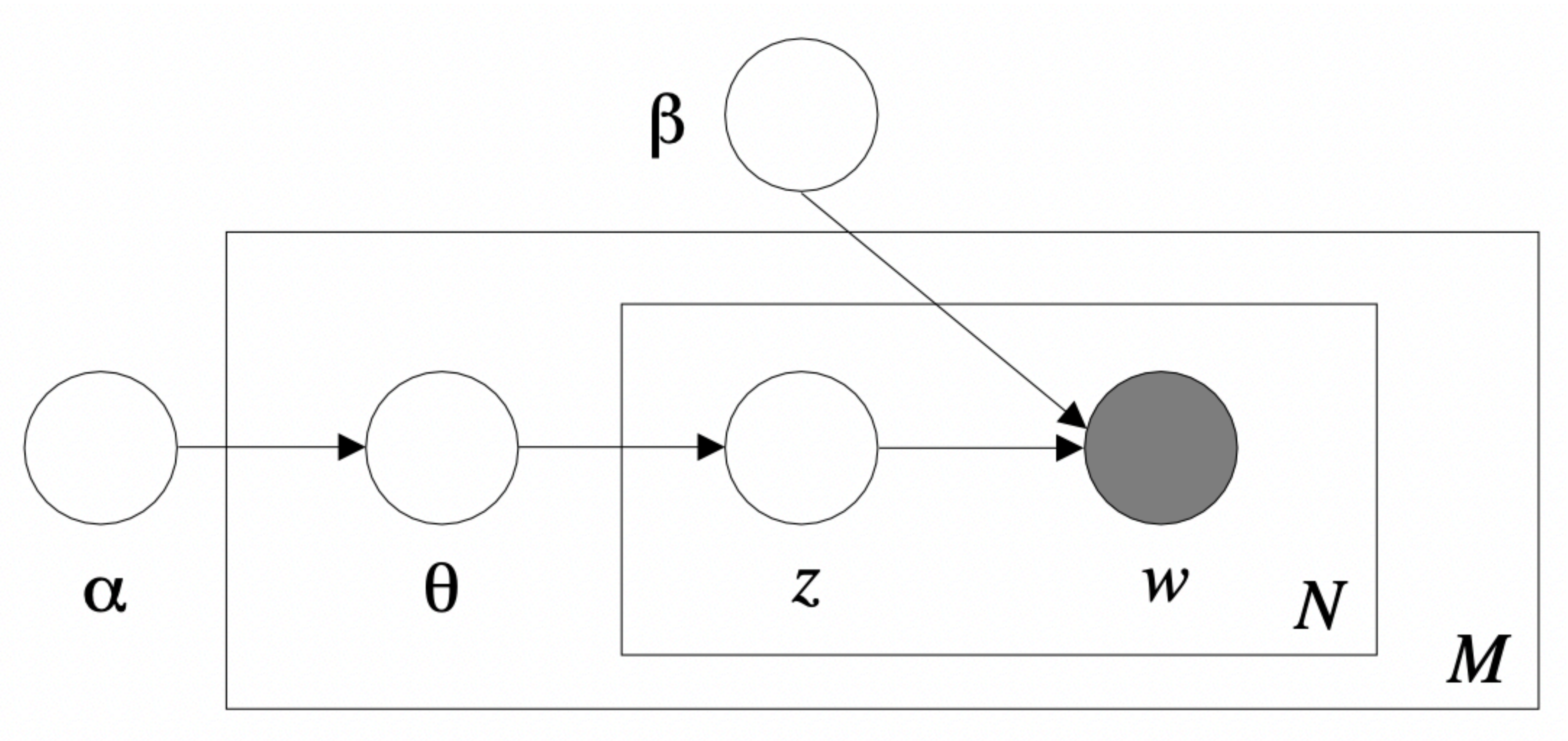
$$= \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)\,p(w_n|z_n,\beta)\right)d\theta$$

- Eq. (3) in terms of the model parameters

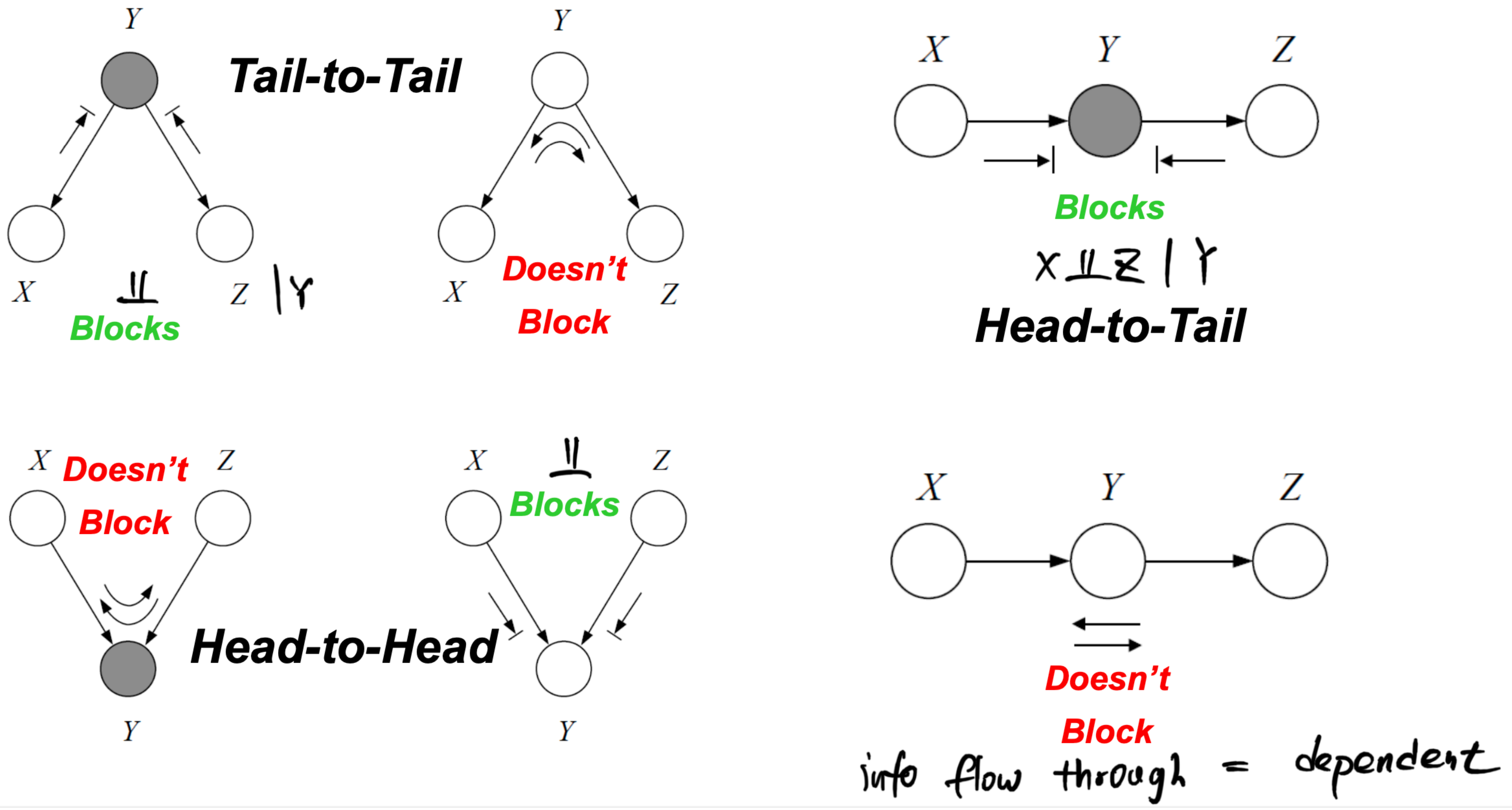
- $$p(\mathbf{w}\,|\,\alpha,\beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\right)\left(\prod_{n=1}^{N}\sum_{i=1}^{k}\prod_{j=1}^{V}\left(\theta_i \beta_{ij}\right)^{w_n^j}\right) d\theta$$

- Intractable to compute in general due to the coupling between θ and β in the summation over latent topics z_n

PGM for LDA



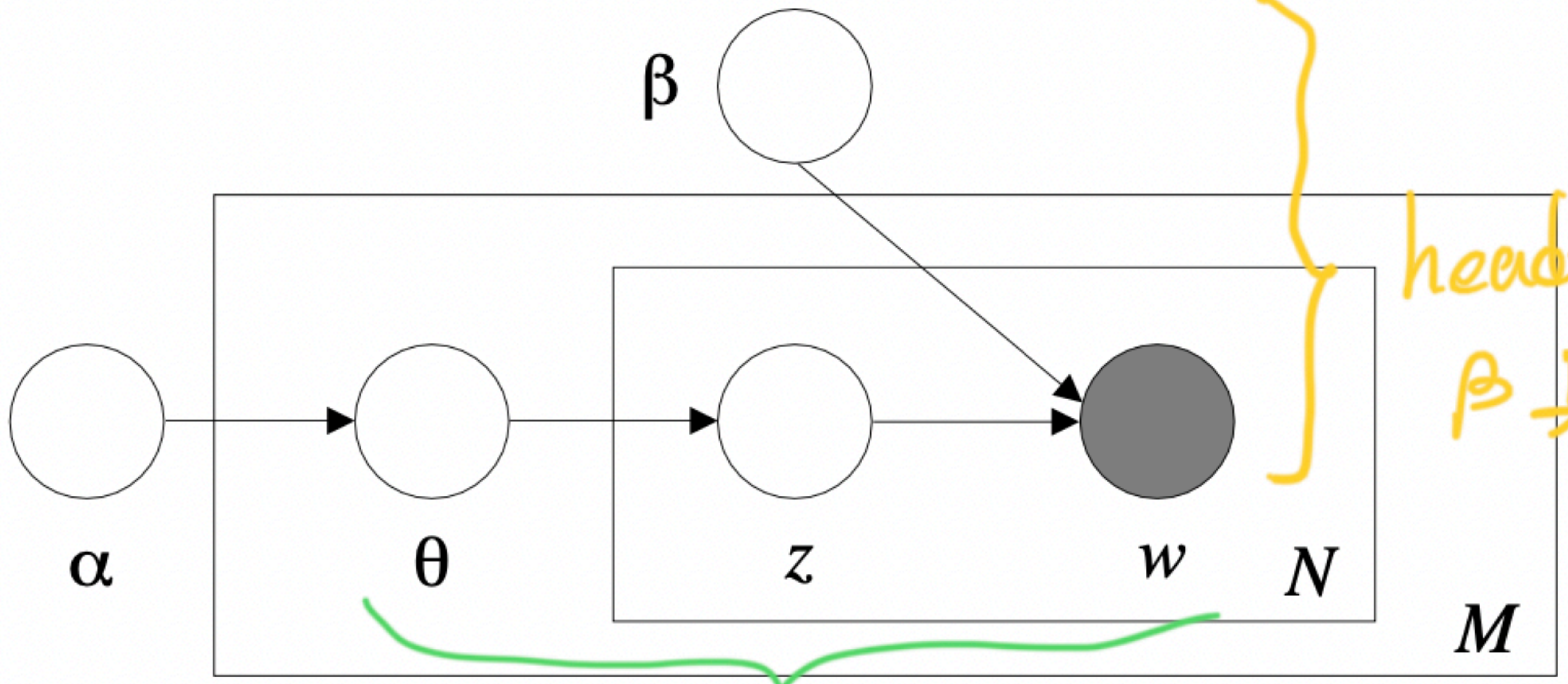
Bayes Ball Algorithm



Based on slides by Prof. Jason Pacheco in CSC 535 Probabilistic Graphical Models

Apply Bayes Ball Algorithm to PGM

θ, β dependency



head-to-head

$\beta \not\perp z$

head-to-tail : $\theta \not\perp w$

Figure 5: (Left) Graphical model representation of LDA. (Right

Variational inference

- A wide variety of approximate inference algorithms
  - Laplace approximation
  - Variational approximation
  - MCMC
- This paper
  - Convexity-based variational inference

Convexity-based variational inference

- Idea: utilize Jensen's inequality to obtain an adjustable lower bound on the log likelihood
  - Needs a tractable family of lower bounds
  - Needs a family of distributions
- By dropping the edges and $\mathbf{w}$, and providing with f.v.p. γ and ϕ, we obtain a family of dist. on latent θ and $\mathbf{z}$ characterized by following variational dist.:
  - $q(\theta, \mathbf{z} \mid \gamma, \phi) = q(\theta \mid \gamma) \prod_{n=1}^{N} q(z_n \mid \phi_n)$

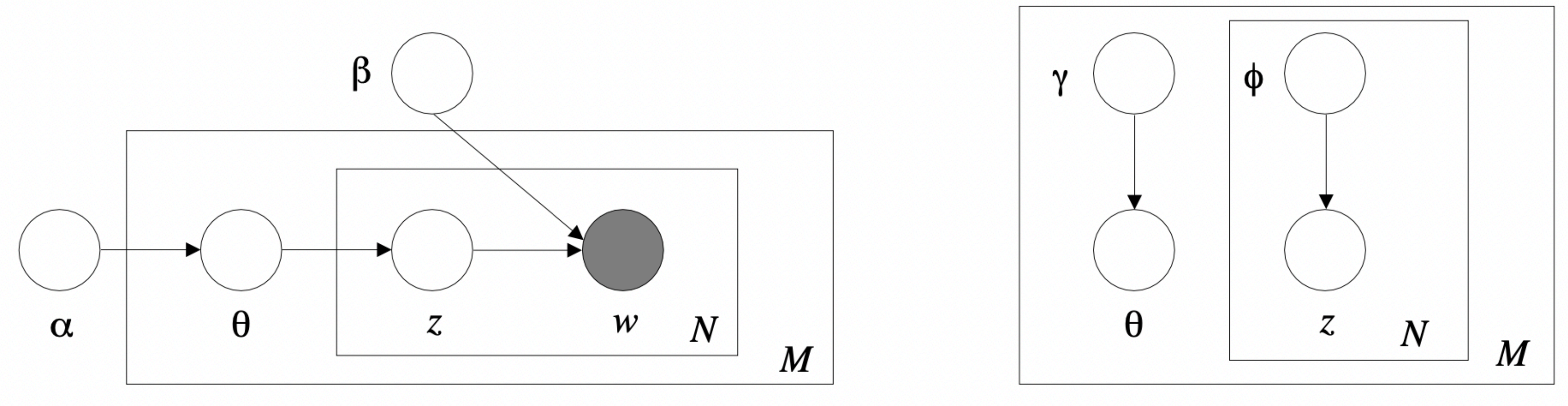


Figure 5: (Left) Graphical model representation of LDA. (Right) Graphical model representation of the variational distribution used to approximate the posterior in LDA.

  - where the Dir. param. γ and the Multi. params. $(\phi_1, \ldots, \phi_N)$ are the f.v.p.s
  - Mean-field assumption: picking a joint variational dist. based on the product of the marginals, so it doesn't capture any dependence => all r.v.s are marginally independent

Apply Jensen's inequality

- Bounding the log likelihood of a $\mathbf{w}$, omitting γ and ϕ for simplicity, we have:

$$
\begin{aligned}
\log p(\mathbf{w} \mid \alpha, \beta) &= \log \int \sum_{\mathbf{z}} p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) d\theta \\
&= \log \int \sum_{\mathbf{z}} \frac{p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) q(\theta, \mathbf{z})}{q(\theta, \mathbf{z})} d\theta \\
&\geq \int \sum_{\mathbf{z}} q(\theta, \mathbf{z}) \log p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) d\theta - \int \sum_{\mathbf{z}} q(\theta, \mathbf{z}) \log q(\theta, \mathbf{z}) d\theta \\
L(\gamma, \phi; \alpha, \beta) &= \mathrm{E}_q[\log p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta)] - \mathrm{E}_q[\log q(\theta, \mathbf{z})]. \qquad (12)
\end{aligned}
$$

- Jensen's inequality provides us with a lower bound on the log likelihood for $q(\theta, \mathbf{z} \mid \gamma, \phi)$
- It can be easily verified that
  - KL divergence(variational posterior || true posterior) = $\log p(\mathbf{w} \mid \alpha, \beta) - L(\gamma, \phi; \alpha, \beta)$
  - $\log p(\mathbf{w} \mid \alpha, \beta) = L(\gamma, \phi; \alpha, \beta) + D(q(\theta, \mathbf{z} \mid \gamma, \phi) \mid\mid p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta))$
  - Maximizing the lower bound L w.r.t. γ and $\phi \equiv$ minimizing the KL divergence

Obtaining variational parameter updates

- Turns lower bound maximization problem
  - => KL divergence minimization problem
  - => variational parameter optimization problem

$$(\gamma^*, \phi^*) = \arg\min_{(\gamma,\phi)} D(q(\theta, \mathbf{z} \mid \gamma, \phi) \parallel p(\theta, \mathbf{z} \mid \mathbf{w}, \alpha, \beta)). \tag{5}$$

  - γ^*, ϕ^* are found by minimizing the Kullback-Leibler (KL) divergence
  - Computing derivatives and setting them equal to 0, we obtain following update equations:

$$\phi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i) \mid \gamma]\} \tag{6}$$

$$\gamma_i = \alpha_i + \textstyle\sum_{n=1}^{N} \phi_{ni}. \tag{7}$$

  - Expectation in ϕ_{ni} update can be computed as follows (has a closed form):

$$E_q[\log(\theta_i) \mid \gamma] = \Psi(\gamma_i) - \Psi\left(\textstyle\sum_{j=1}^{k} \gamma_j\right), \tag{8}$$

    - where Ψ is the 1^{st} derivative of the $\log\Gamma$ function computable via Taylor approximation (digamma fn.)

Obtaining ϕ_{ni} update

- Expend by factorizations of p and q :

$$L(\gamma,\phi;\alpha,\beta) = \mathrm{E}_q[\log p(\theta\,|\,\alpha)] + \mathrm{E}_q[\log p(\mathbf{z}\,|\,\theta)] + \mathrm{E}_q[\log p(\mathbf{w}\,|\,\mathbf{z},\beta)] - \mathrm{E}_q[\log q(\theta)] - \mathrm{E}_q[\log q(\mathbf{z})]. \tag{14}$$

- Expend in terms of model and variational parameters:

$$\begin{aligned}
L(\gamma,\phi;\alpha,\beta) = {} & \log\Gamma\left(\textstyle\sum_{j=1}^k \alpha_j\right) - \sum_{i=1}^k \log\Gamma(\alpha_i) + \sum_{i=1}^k (\alpha_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\textstyle\sum_{j=1}^k \gamma_j\right)\right) \\
& + \sum_{n=1}^N \sum_{i=1}^k \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\textstyle\sum_{j=1}^k \gamma_j\right)\right) \\
& + \sum_{n=1}^N \sum_{i=1}^k \sum_{j=1}^V \phi_{ni} w_n^j \log\beta_{ij} \\
& - \log\Gamma\left(\textstyle\sum_{j=1}^k \gamma_j\right) + \sum_{i=1}^k \log\Gamma(\gamma_i) - \sum_{i=1}^k (\gamma_i - 1)\left(\Psi(\gamma_i) - \Psi\left(\textstyle\sum_{j=1}^k \gamma_j\right)\right) \\
& - \sum_{n=1}^N \sum_{i=1}^k \phi_{ni}\log\phi_{ni},
\end{aligned} \tag{15}$$

Obtaining ϕ_{ni} update

- Maximize (15) w.r.t. ϕ_{ni} , this is constrained since $\sum_{i=1}^{k} \phi_{ni} = 1$

- We form the Lagrangian by isolating the terms containing ϕ_{ni} :

$$L_{[\phi_{ni}]} = \phi_{ni}\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right)\right) + \phi_{ni}\log\beta_{iv} - \phi_{ni}\log\phi_{ni} + \lambda_n\left(\sum_{j=1}^{k}\phi_{ni} - 1\right),$$

  - where $\beta_{iv} = p(w_n^v = 1 \,|\, z^i = 1)$

- Take derivatives w.r.t. ϕ_{ni} :

$$\frac{\partial L}{\partial \phi_{ni}} = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right) + \log\beta_{iv} - \log\phi_{ni} - 1 + \lambda.$$

- Set it to 0 yields the maximum value of ϕ_{ni} :

$$\phi_{ni} \propto \beta_{iv}\exp\left(\Psi(\gamma_i) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right)\right). \qquad (16)$$

VI algorithm

initialize $\phi_{ni}^0 := 1/k$ for all i and n
initialize $\gamma_i := \alpha_i + N/k$ for all i
repeat
    for $n = 1$ **to** N N
        for $i = 1$ **to** k k $O((N+1)k)$
            $\phi_{ni}^{t+1} := \beta_{iw_n} \exp(\Psi(\gamma_i^t))$
        normalize ϕ_n^{t+1} to sum to 1.
    $\gamma^{t+1} := \alpha + \sum_{n=1}^N \phi_n^{t+1}$
until convergence

Figure 6: A variational inference algorithm for LDA.

- Empirically, the # of iterations required for a $\mathbf{w}$ depends on $|\mathbf{w}|$, thus roughly on the order of N^2k

Parameter optimization

- Given a $D = \{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_M\}$, want to find α and β s.t. the marginal log likelihood (theoretical) is maximized:

$$\ell(\alpha, \beta) = \sum_{d=1}^{M} \log p(\mathbf{w}_d \mid \alpha, \beta).$$

- Intractable to compute $p(\mathbf{w} \mid \alpha, \beta)$
- Approximate empirical estimates by variational EM procedure
  - Maximize a lower bound L w.r.t. γ and ϕ
  - For fixed values of γ and ϕ, maximize the lower bound w.r.t. α and β

Expectation Maximization

Find tightest lower bound of marginal likelihood,

$$\max_{\theta} \log p(\mathcal{Y} \mid \theta) \geq \max_{q,\theta} \mathbf{E}_q \left[\log \frac{p(z, \mathcal{Y} \mid \theta)}{q(z)} \right] \equiv \mathcal{L}(q, \theta)$$

Solve by coordinate ascent…

Initialize Parameters: $\theta^{(0)}$

At iteration t do:

E-Step: $q^{(t)} = \arg\max_q \mathcal{L}(q, \underset{\text{Fix } \theta}{\theta^{(t-1)}})$

M-Step: $\theta^{(t)} = \arg\max_\theta \mathcal{L}(\underset{\text{Fix q}}{q^{(t)}}, \theta)$

Until convergence

Based on slides by Prof. Jason Pacheco in CSC 535 Probabilistic Graphical Models

EM algorithm

- While true:
  - (E-step) for each $\mathbf{w} \in D$:
    - Find the optimizing values of $\left\{\gamma_d^*, \phi_d^* : d \in D\right\}$
    - done in VI algorithm
  - (M-step) With fixed γ^* and ϕ^*, maximize the resulting lower bound on the log likelihood w.r.t. α and β
    - Update for the conditional multinomial parameter β can be written out as:

$$\beta_{ij} \propto \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}^* w_{dn}^j.$$

- Repeat until the lower bound on the log likelihood converges

Applications and empirical results

- For a D of M documents, the perplexity is defined as following:

$$perplexity(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(\mathbf{w}_d)}{\sum_{d=1}^{M} N_d}\right\}$$

  - A lower perplexity score indicates better generalization performance

- The latent variable models perform better than the simple unigram model

- LDA consistently performs better than the other models

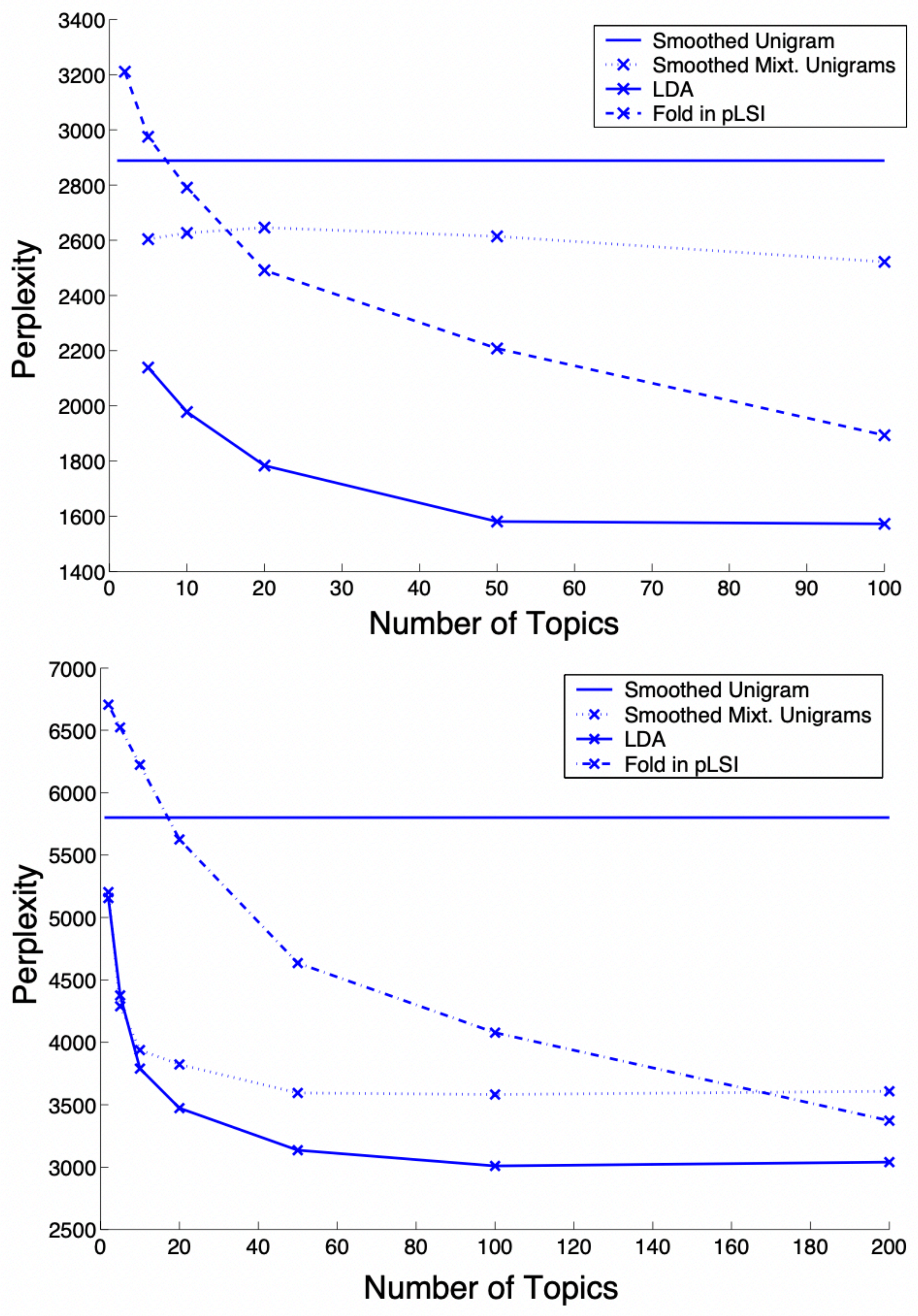


Figure 9: Perplexity results on the nematode (Top) and AP (Bottom) corpora for LDA, the unigram model, mixture of unigrams, and pLSI.

Applications and empirical results

Num. topics (k)	Perplexity (Mult. Mixt.)	Perplexity (pLSI)
2	22,266	7,052
5	2.20×10^{8}	17,588
10	1.93×10^{17}	63,800
20	1.20×10^{22}	2.52×10^{5}
50	4.19×10^{106}	5.04×10^{6}
100	2.39×10^{150}	1.72×10^{7}
200	3.51×10^{264}	1.31×10^{7}

Table 1: Overfitting in the mixture of unigrams and pLSI models for the AP corpus. Similar behavior is observed in the nematode corpus (not reported).

Applications and empirical results

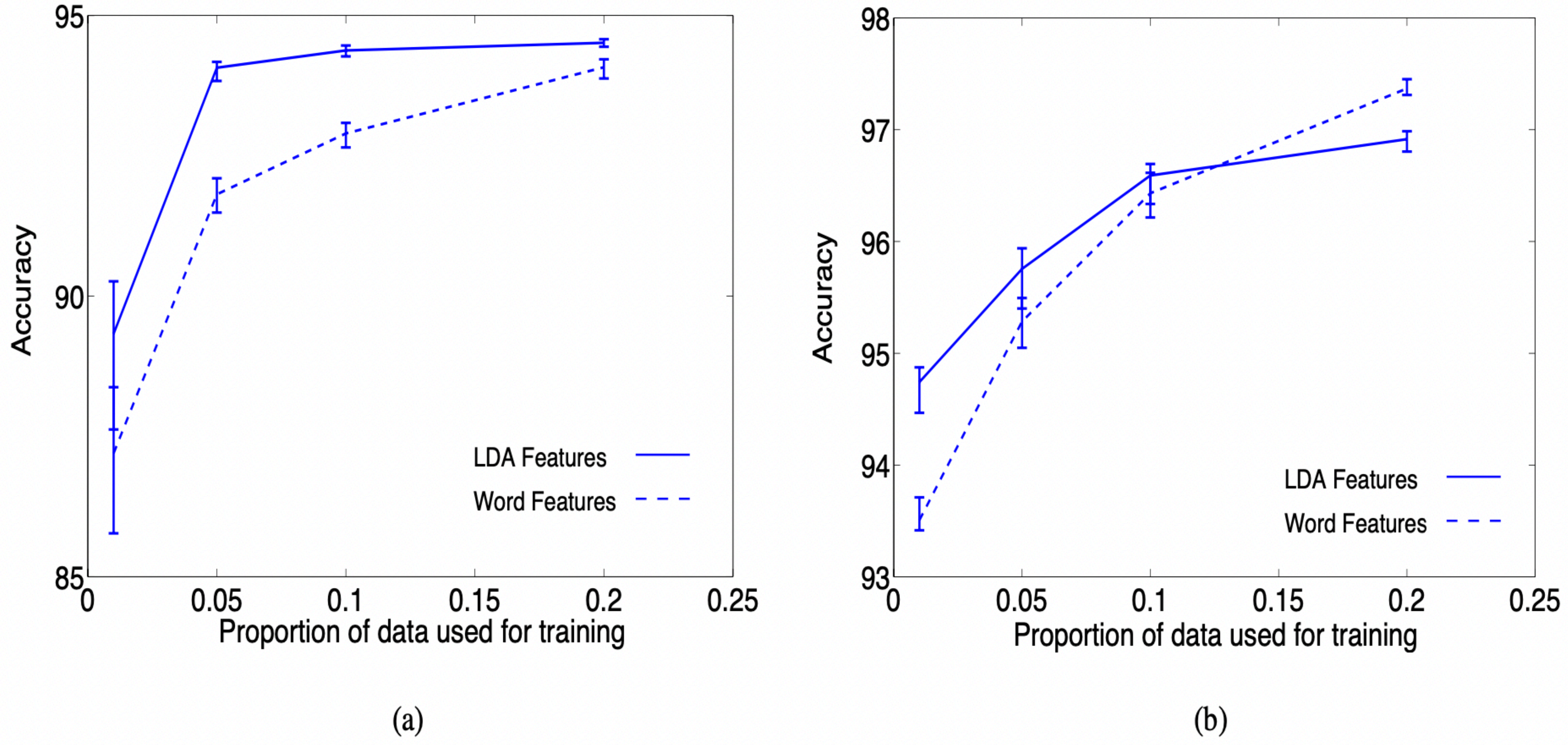


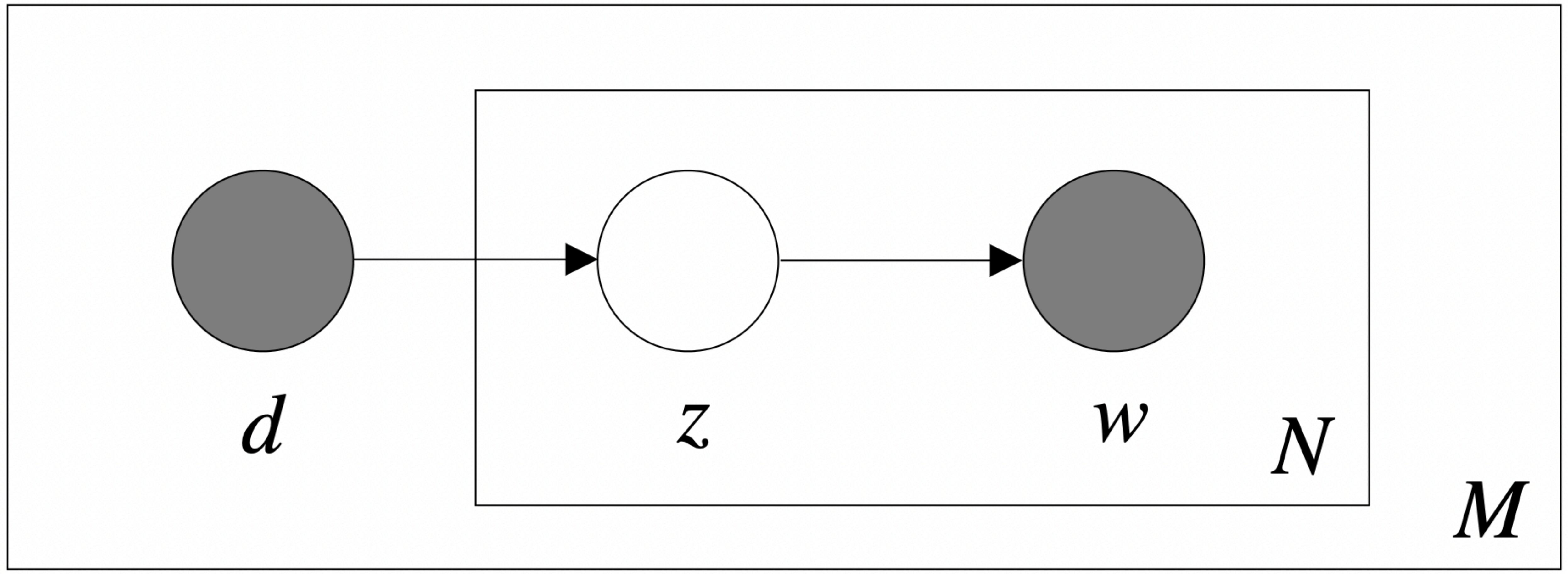
Figure 10: Classification results on two binary classification problems from the Reuters-21578 dataset for different proportions of training data. Graph (a) is EARN vs. NOT EARN. Graph (b) is GRAIN vs. NOT GRAIN.

- A little drop in classification performance using LDA-based features
- However, in almost all cases, the performance is improved with the LDA features, suggesting topic-based representation may be useful as a fast filtering algorithm for feature selection in text classification

Latent Semantic Indexing (LSI)

- Use a singular value decomposition of the *X* matrix to identify a linear subspace in the space of *tf-idf* features that captures most of the variance in the collection
- Strengths
    - Significant compression in large collections
    - Derived features are linear combinations of the original *tf-idf* features, can capture some aspects of basic linguistic notions such as synonymy and polysemy
- A generative probabilistic model to study the ability of LSI
    - pLSI

Probabilistic Latent Semantic Indexing (pLSI)



(c) pLSI/aspect model

$$p(d, w_n) = p(d) \sum_z p(w_n \mid z) p(z \mid d).$$

Probabilistic Latent Semantic Indexing (pLSI)

- Attempts to relax the simplifying assumption made in the mixture of unigrams model that each document is generated from only one topic

- It does capture the possibility that a document may contain multiple topics

- However

    - d is a dummy index into the list of documents in the *training set*

    - *T*he model learns the topic mixtures $p(z \mid d)$ only for those documents on which it is trained

- For above reasons,

    - pLSI is not a well-defined generative model of documents

    - No natural way to assign probability to a previously unseen document

Thank you!

Questions?