Document and Topic Models: pLSA and LDA

Andrew Levandoski and Jonathan Lobo

CS 3750 Advanced Topics in Machine Learning

2 October 2018

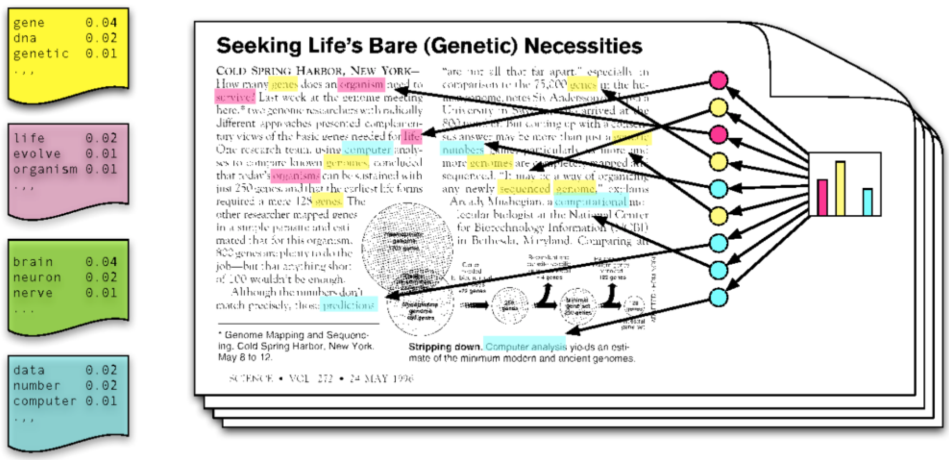
Outline

- Topic Models
- pLSA
  - LSA
  - Model
  - Fitting via EM
  - pHITS: link analysis
- LDA
  - Dirichlet distribution
  - Generative process
  - Model
  - Geometric Interpretation
  - Inference

2

Topic Models: Visual Representation

Topics | Documents | Topic proportions and assignments



3

Topic Models: Importance

- For a given corpus, we learn two things:
  1. **Topic:** from full vocabulary set, we learn important subsets
  2. **Topic proportion:** we learn what each document is about
- This can be viewed as a form of dimensionality reduction
  - From large vocabulary set, extract basis vectors (topics)
  - Represent document in topic space (topic proportions)
  - Dimensionality is reduced from $\{w_i\} \in \mathbb{Z}_V^N$ to $\theta \in \mathbb{R}^K$
- Topic proportion is useful for several applications including document classification, discovery of semantic structures, sentiment analysis, object localization in images, etc.

4

Topic Models: Terminology

- **Document Model**
  - **Word:** element in a vocabulary set
  - **Document:** collection of words
  - **Corpus:** collection of documents
- **Topic Model**
  - **Topic:** collection of words (subset of vocabulary)
  - Document is represented by (latent) mixture of topics
    - $p(w|d) = p(w|z)p(z|d)$ (z : topic)
- **Note:** document is a *collection* of words (not a *sequence*)
  - *'Bag of words'* assumption
  - In probability, we call this the *exchangeability* assumption
    - $p(w_1, \dots, w_N) = p(w_{\sigma(1)}, \dots, w_{\sigma(N)})$ (σ: permutation)

5

Topic Models: Terminology (cont'd)

- Represent each document as a vector space
- A word is an item from a vocabulary indexed by $\{1, \dots, V\}$. We represent words using unit-basis vectors. The v^{th} word is represented by a V vector w such that $w^v = 1$ and $w^u = 0$ for $v \neq u$.
- A document is a sequence of n words denoted by $\mathrm{w} = (w_1, w_2, \dots w_n)$ where w_n is the nth word in the sequence.
- A corpus is a collection of M documents denoted by

$$D = \{w_1, w_2, \dots w_m\}.$$

6

Probabilistic Latent Semantic Analysis (pLSA)

Motivation

- Learning from text and natural language
- Learning meaning and usage of words without prior linguistic knowledge
- Modeling semantics
  - Account for polysems and similar words
  - Difference between what is said and what is meant

Vector Space Model

- Want to represent documents and terms as vectors in a lower-dimensional space
- N × M word-document co-occurrence matrix $\boldsymbol{N}$

$$D = \{d_1, \ldots, d_N\}$$

$$\mathrm{W} = \{w_1, \ldots, w_M\}$$

$$\boldsymbol{N} = \left(n(d_i, w_j)\right)_{ij}$$

  - limitations: high dimensionality, noisy, sparse
  - solution: map to lower-dimensional latent semantic space using SVD

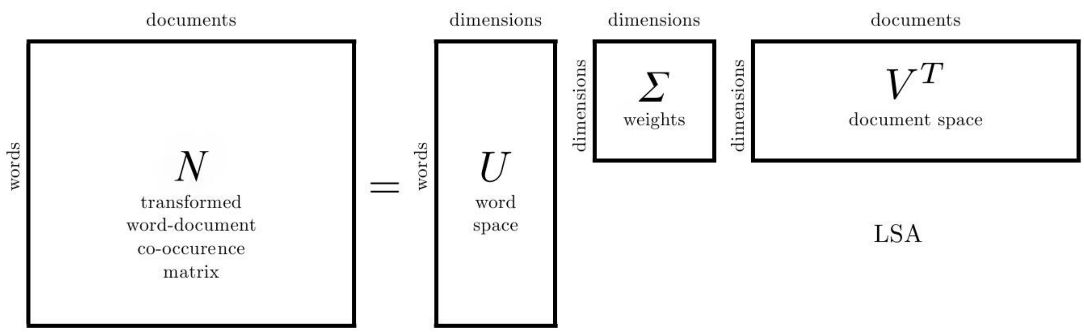
9

Latent Semantic Analysis (LSA)

- Goal
  - Map high dimensional vector space representation to lower dimensional representation in latent semantic space
  - Reveal semantic relations between documents (count vectors)
- SVD
  - $N = U\Sigma V^T$
  - U: orthogonal matrix with left singular vectors (eigenvectors of NN^T)
  - V: orthogonal matrix with right singular vectors (eigenvectors of N^TN)
  - Σ: diagonal matrix with singular values of N
- Select *k* largest singular values from Σ to get approximation $\widetilde{N}$ with minimal error
  - Can compute similarity values between document vectors and term vectors

10

LSA



LSA Strengths

- Outperforms naïve vector space model
- Unsupervised, simple
- Noise removal and robustness due to dimensionality reduction
- Can capture synonymy
- Language independent
- Can easily perform queries, clustering, and comparisons

LSA Limitations

- No probabilistic model of term occurrences
- Results are difficult to interpret
- Assumes that words and documents form a joint Gaussian model
- Arbitrary selection of the number of dimensions k
- Cannot account for polysemy
- No generative model

13

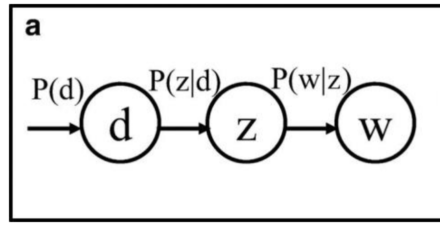
Probabilistic Latent Semantic Analysis (pLSA)

- Difference between topics and words?
  - Words are observable
  - Topics are not, they are latent
- Aspect Model
  - Associates an unobserved latent class variable $z \epsilon \mathbb{Z} = \{z_1, \ldots, z_K\}$ with each observation
  - Defines a joint probability model over documents and words
  - Assumes *w* is independent of *d* conditioned on *z*
  - Cardinality of *z* should be much less than than *d* and *w*

14

pLSA Model Formulation

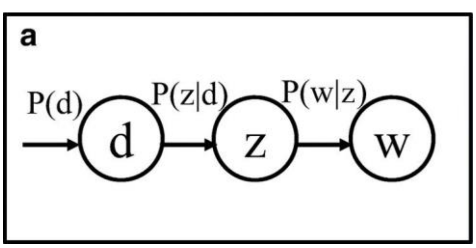
- Basic Generative Model
  - Select document *d* with probability *P(d)*
  - Select a latent class *z* with probability *P(z|d)*
  - Generate a word *w* with probability *P(w|z)*
- Joint Probability Model

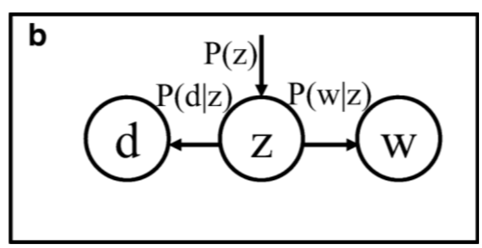


$$P(d,w) = P(d)P(w|d) \qquad P(w|d) = \sum_{z \in \mathbb{Z}} P(w|z)P(z|d)$$

15

pLSA Graphical Model Representation





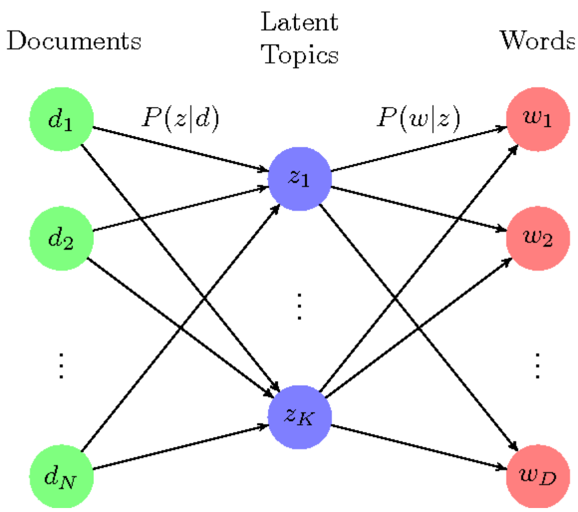
$$P(d,w) = P(d)P(w|d)$$

$$P(w|d) = \sum_{z \in \mathbb{Z}} P(w|z)P(z|d)$$

$$P(d,w) = \sum_{z \in \mathbb{Z}} P(z)P(d|z)P(w|z)$$

16

pLSA Joint Probability Model



$$P(d,w) = P(d)P(w|d)$$

$$P(w|d) = \sum_{z \in \mathbb{Z}} P(w|z)P(z|d)$$

Maximize:

$$\mathcal{L} = \sum_{d \epsilon D} \sum_{w \epsilon W} n(d,w) \log P(d,w)$$

Corresponds to a minimization of KL divergence (cross-entropy) between the empirical distribution of words and the model distribution P(w|d)

17

Probabilistic Latent Semantic Space

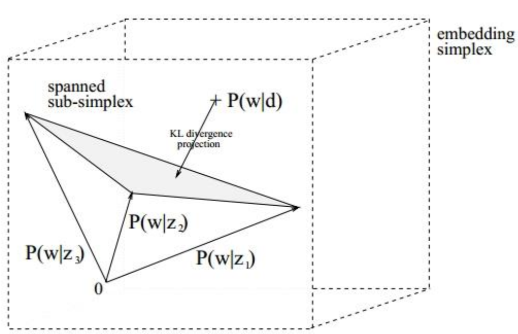


Figure 1: Sketch of the probability sub-simplex spanned by the aspect model.

- P(w|d) for all documents is approximated by a multinomial combination of all factors P(w|z)
- Weights P(z|d) uniquely define a point in the latent semantic space, represent how topics are mixed in a document

18

Probabilistic Latent Semantic Space

- Topic represented by probability distribution over words

$z_i = (w_1, \ldots, w_m) \qquad z_1 = (0.3, 0.1, 0.2, 0.3, 0.1)$

- Document represented by probability distribution over topics

$d_j = (z_1, \ldots, z_n) \qquad d_1 = (0.5, 0.3, 0.2)$

19

Model Fitting via Expectation Maximization

- E-step

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')}$$

Compute posterior probabilities for latent variables z using current parameters

- M-step

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{d,w'} n(d,w')P(z|d,w')}$$

$$P(d|z) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{d',w} n(d',w)P(z|d',w)}$$

$$P(z) = \frac{1}{R}\sum_{d,w} n(d,w)P(z|d,w), \qquad R \equiv \sum_{d,w} n(d,w)$$

Update parameters using given posterior probabilities

20

pLSA Strengths

- Models word-document co-occurrences as a mixture of conditionally independent multinomial distributions
- A mixture model, not a clustering model
- Results have a clear probabilistic interpretation
- Allows for model combination
- Problem of polysemy is better addressed

21

pLSA Strengths

- Problem of polysemy is better addressed

tie					spring				
trousers	season	scoreline	wires	operatic	beginning	dampers	flower	creek	humid
blouse	teams	goalless	cables	soprano	until	brakes	flowers	brook	winters
waistcoat	winning	equaliser	wiring	mezzo	months	suspension	flowering	river	summers
skirt	league	clinching	electrical	contralto	earlier	absorbers	fragrant	fork	ppen
sleeved	finished	scoreless	wire	baritone	year	wheels	lilies	piney	warm
pants	championship	replay	cable	coloratura	last	damper	flowered	elk	temperatures

22

pLSA Limitations

- Potentially higher computational complexity
- EM algorithm gives local maximum
- Prone to overfitting
  - Solution: Tempered EM
- Not a well defined generative model for new documents
  - Solution: Latent Dirichlet Allocation

23

pLSA Model Fitting Revisited

- Tempered EM
  - Goals: maximize performance on unseen data, accelerate fitting process
  - Define control parameter β that is continuously modified
- Modified E-step

$$P_\beta(z|d,w) = \frac{P(z)[P(d|z)P(w|z)]^\beta}{\sum_{z'} P(z')[P(d|z')P(w|z')]^\beta}$$

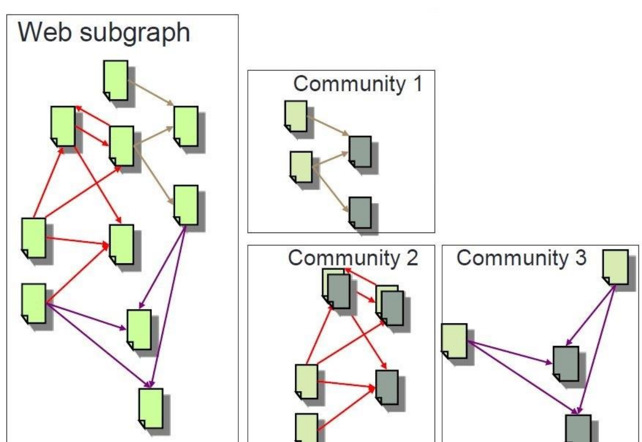
24

Tempered EM Steps

1) Split data into training and validation sets
2) Set β to 1
3) Perform EM on training set until performance on validation set decreases
4) Decrease β by setting it to ηβ, where η <1, and go back to step 3
5) Stop when decreasing β gives no improvement

25

Example: Identifying Authoritative Documents



26

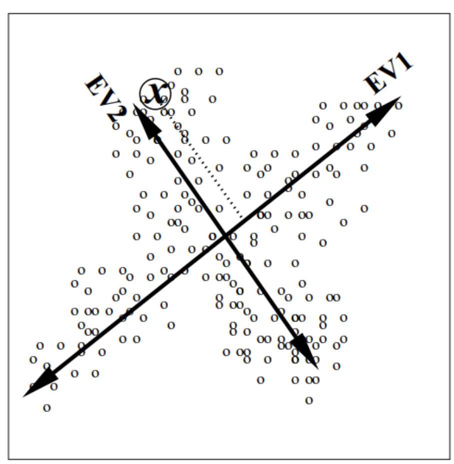
HITS

- Hubs and Authorities
  - Each webpage has an authority score *x* and a hub score *y*
  - Authority – value of content on the page to a community
    - likelihood of being cited
  - Hub – value of links to other pages
    - likelihood of citing authorities
  - A good hub points to many good authorities
  - A good authority is pointed to by many good hubs
- Principal components correspond to different communities
  - Identify the principal eigenvector of co-citation matrix

27

HITS Drawbacks

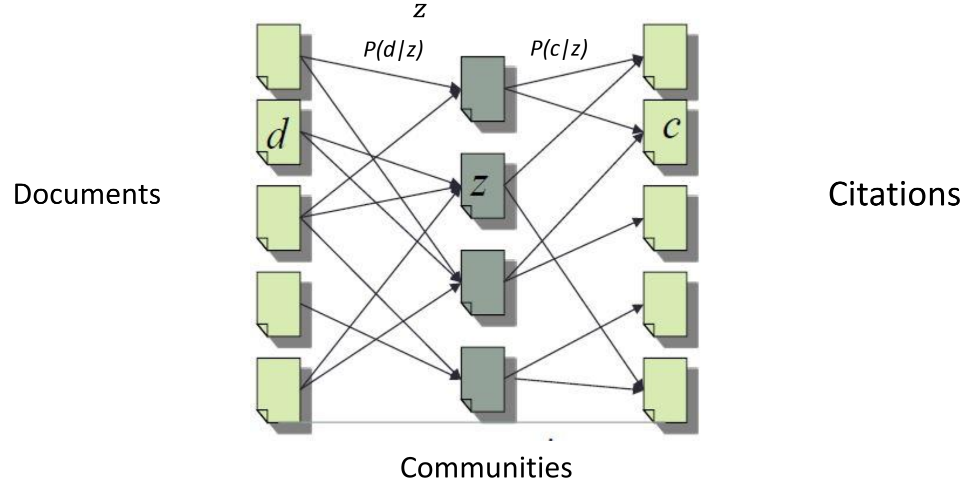
- Uses only the largest eigenvectors, not necessary the only relevant communities
- Authoritative documents in smaller communities may be given no credit
- Solution: Probabilistic HITS



28

pHITS

$$P(d,c) = \sum_{z} P(z)P(c|z)P(d|z)$$



29

Interpreting pHITS Results

- Explain *d* and *c* in terms of the latent variable "community"
- Authority score: P(c|z)
  - Probability of a document being cited from within community *z*
- Hub Score: P(d|z)
  - Probability that a document *d* contains a reference to community *z*.
- Community Membership: P(z|c).
  - Classify documents

30

Joint Model of pLSA and pHITS

- Joint probabilistic model of document content (pLSA) and connectivity (pHITS)
  - Able to answer questions on both structure and content
  - Model can use evidence about link structure to make predictions about document content, and vice versa
  - Reference flow – connection between one topic and another
- Maximize log-likelihood function

$$\mathcal{L} = \sum_j \left[\alpha \sum_i \frac{N_{ij}}{\sum_{i'} N_{i'j}} log \sum_k P(w_i|z_k)P(z_k|d_j) + (1-\alpha) \sum_l \frac{A_{lj}}{\sum_{l'} A_{l'j}} log \sum_k P(c_l|z_k)P(z_k|d_j) \right]$$

31

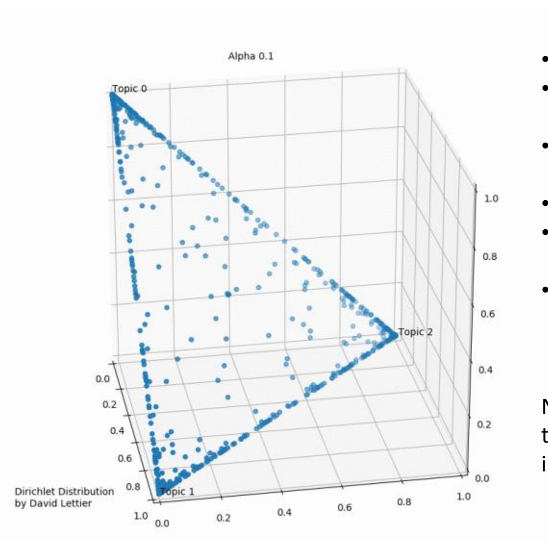
pLSA: Main Deficiencies

- Incomplete in that it provides no probabilistic model at the document level i.e. no proper priors are defined.
- Each document is represented as a list of numbers (the mixing proportions for topics), and there is no generative probabilistic model for these numbers, thus:
  1. The number of parameters in the model grows linearly with the size of the corpus, leading to overfitting
  2. It is unclear how to assign probability to a document outside of the training set
- Latent Dirichlet allocation (LDA) captures the exchangeability of both words *and* documents using a Dirichlet distribution, allowing a coherent generative process for test data

32

Latent Dirichlet Allocation (LDA)

LDA: Dirichlet Distribution



- A 'distribution of distributions'
- Multivariate distribution whose components all take values on (0,1) and which sum to one.
- Parameterized by the vector α, which has the same number of elements (k) as our multinomial parameter θ.
- Generalization of the beta distribution into multiple dimensions
- The alpha hyperparameter controls the mixture of topics for a given document
- The beta hyperparameter controls the distribution of words per topic

Note: Ideally we want our composites to be made up of only a few topics and our parts to belong to only some of the topics. With this in mind, alpha and beta are typically set below one.

LDA: Dirichlet Distribution (cont'd)

- A k-dimensional Dirichlet random variable θ can take values in the (k-1)-simplex (a k-vector θ lies in the (k-1)-simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$) and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1},$$

  where the parameter α is a k-vector with components $\alpha_i > 0$ and where $\Gamma(x)$ is the Gamma function.
- The Dirichlet is a convenient distribution on the simplex:
  - In the exponential family
  - Has finite dimensional sufficient statistics
  - Conjugate to the multinomial distribution

35

LDA: Generative Process

LDA assumes the following generative process for each document w in a corpus D:

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the N words w_N:
   a. Choose a topic $z_n \sim Multinomial(\theta)$.
   b. Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Example: Assume a group of articles that can be broken down by three topics described by the following words:

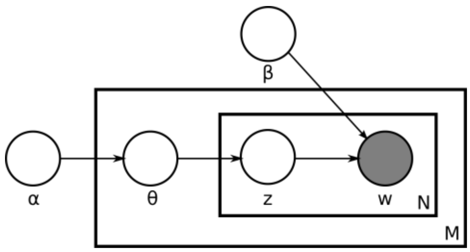
- Animals: dog, cat, chicken, nature, zoo
- Cooking: oven, food, restaurant, plates, taste
- Politics: Republican, Democrat, Congress, ineffective, divisive

To generate a new document that is 80% about animals and 20% about cooking:

- Choose the length of the article (say, 1000 words)
- Choose a topic based on the specified mixture (~800 words will coming from topic 'animals')
- Choose a word based on the word distribution for each topic

36

LDA: Model (Plate Notation)



α is the parameter of the Dirichlet prior on the per-document topic distribution,

β is the parameter of the Dirichlet prior on the per-topic word distribution,

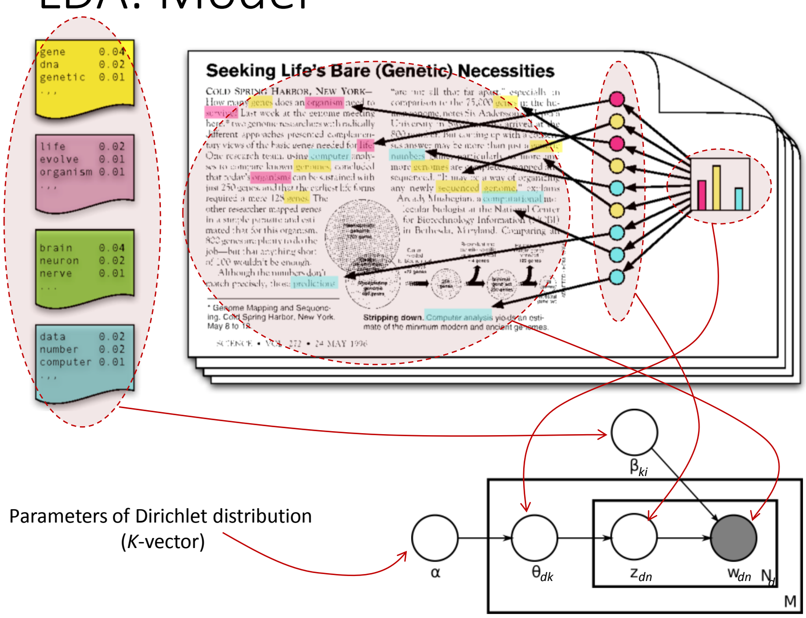
θ_M is the topic distribution for document *M,*

z_{MN} is the topic for the *N-th* word in document *M,* and

w_{MN} is the word.

37

LDA: Model



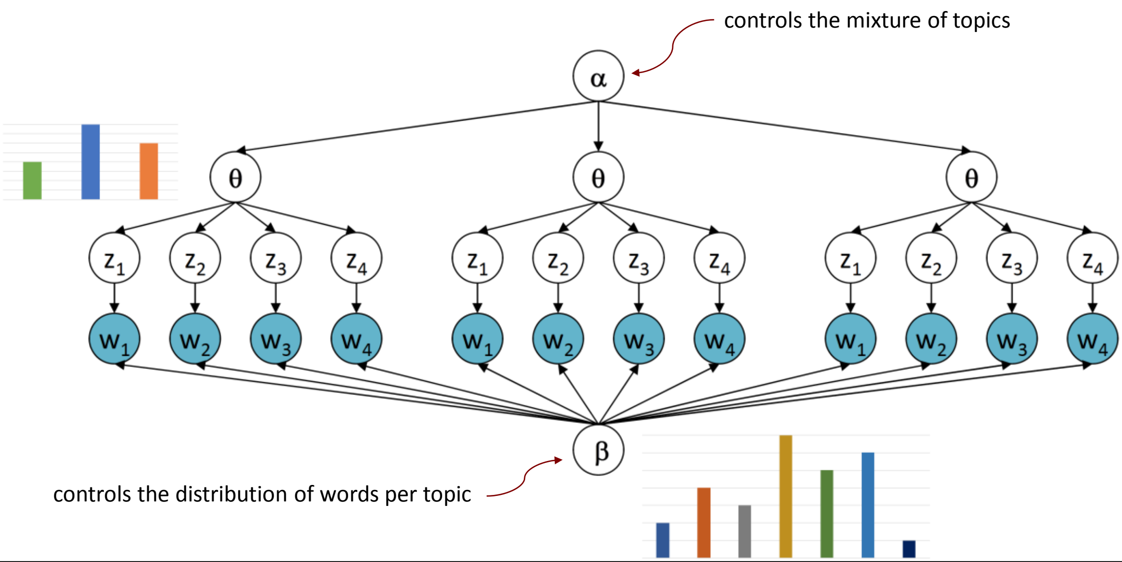
	1 … topic … *K*
1 ⋮ doc ⋮ *M*	θ_{dk}

	1 … *n*th word … N_d
1 ⋮ doc ⋮ *M*	$z_{dn} = \{1, \ldots, K\}$

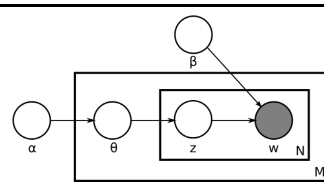
	1 … word idx … *V*
1 ⋮ topic ⋮ *M*	$\beta_{ki} = p(w\|z)$

38

LDA: Model (cont'd)



LDA: Model (cont'd)



Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta),$$

where $p(z_n|\theta)$ is θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z, we obtain the marginal distribution of a document:

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

Finally, taking the products of the marginal probabilities of single documents, we obtain the probability of a corpus:

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

LDA: Exchangeability

- A finite set of random variables $\{x_1, \dots, x_N\}$ 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:

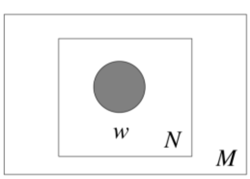
$$p(x_1, \dots, x_N) = p(x_{\pi_1}, \dots, x_{\pi_N})$$

- An infinite sequence of random numbers is infinitely exchangeable if every finite sequence is exchangeable
- We assume that words are generated by topics and that those topics are infinitely exchangeable within a document
- By De Finetti's Theorem:

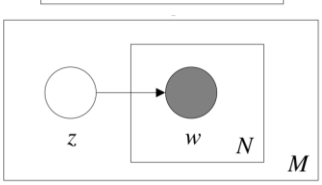
$$p(w, z) = \int p(\theta) \left(\prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n) \right) d\theta$$

41

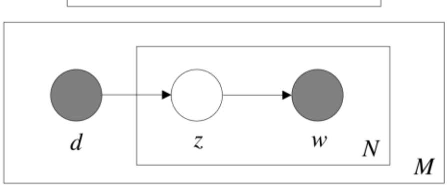
LDA vs. other latent variable models



Unigram model: $p(w) = \prod_{n=1}^{N} p(w_n)$



Mixture of unigrams: $p(w) = \sum_z p(z) \prod_{n=1}^{N} p(w_n|z)$

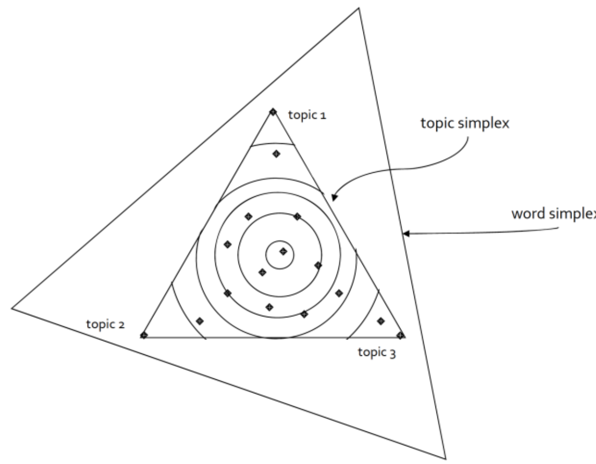


pLSI: $p(d, w_n) = p(d) \sum_z p(w_n|z) p(z|d)$

42

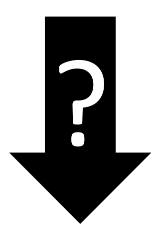
LDA: Geometric Interpretation

- Topic simplex for three topics embedded in the word simplex for three words
- Corners of the word simplex correspond to the three distributions where each word has probability one
- Corners of the topic simplex correspond to three different distributions over words
- Mixture of unigrams places each document at one of the corners of the topic simplex
- pLSI induces an empirical distribution on the topic simplex denoted by diamonds
- LDA places a smooth distribution on the topic simplex denoted by contour lines



LDA: Goal of Inference

LDA inputs: Set of words per document for each document in a corpus



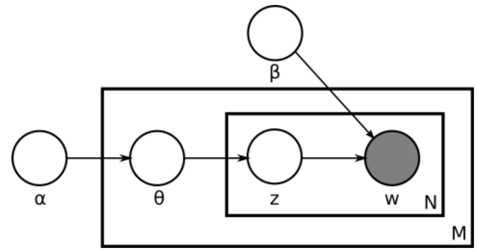
LDA outputs: Corpus-wide topic vocabulary distributions

Topic assignments per word

Topic proportions per document

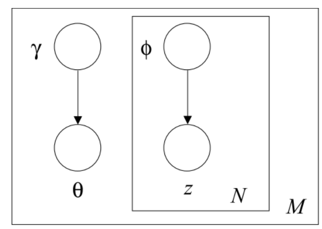
LDA: Inference

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



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

This formula is intractable to compute in general (the integral cannot be solved in closed form), so to normalize the distribution we marginalize over the hidden variables:



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

LDA: Variational Inference

- Basic idea: make use of Jensen's inequality to obtain an adjustable lower bound on the log likelihood
- Consider a family of lower bounds indexed by a set of variational parameters chosen by an optimization procedure that attempts to find the tightest possible lower bound
- Problematic coupling between θ and β arises due to edges between θ, z and w. By dropping these edges and the w nodes, we obtain a family of distributions on the latent variables characterized by the following variational distribution:

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

where γ and $(\phi_1, \ldots, \phi_n)$ and the free variational parameters.

LDA: Variational Inference (cont'd)

- With this specified family of probability distributions, we set up the following optimization problem to determine θ and ϕ:

$$(\gamma^*, \phi^*) = argmin_{(\gamma,\phi)} D(q(\theta, z|\gamma, \phi) \parallel p(\theta, z|w, \alpha, \beta))$$

- The optimizing values of these parameters are found by minimizing the KL divergence between the variational distribution and the true posterior $p(\theta, z|w, \alpha, \beta)$
- By computing the derivatives of the KL divergence and setting them equal to zero, we obtain the following pair of update equations:

$$\phi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i)\,|\gamma]\}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni}$$

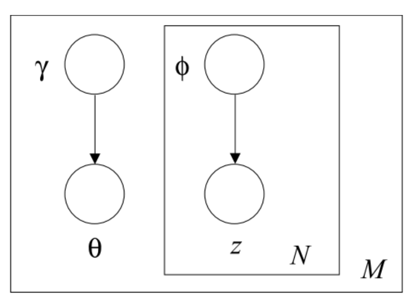
- The expectation in the multinomial update can be computed as follows:

$$E_q[\log(\theta_i)\,|\gamma] = \Psi(\gamma_i) - \Psi(\prod_{j=1}^{k} \gamma_j)$$

where Ψ is the first derivative of the $\log\Gamma$ function.

LDA: Variational Inference (cont'd)

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
    for $i = 1$ **to** 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



LDA: Parameter Estimation

- Given a corpus of documents $D = \{w_1, w_2 \ldots, w_M\}$, we wish to find α and β that maximize the marginal log likelihood of the data:

$$\ell(\alpha, \beta) = \sum_{d=1}^{M} logp(w_d | \alpha, \beta)$$

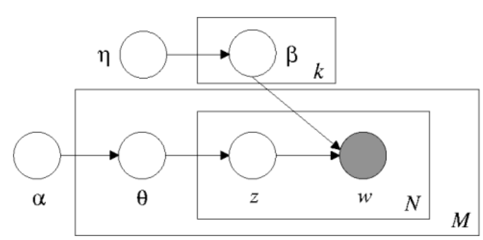
- Variational EM yields the following iterative algorithm:
    1. (E-step) For each document, find the optimizing values of the variational parameters $\{\gamma_d^*, \phi_d^*: d \in D\}$
    2. (M-step) Maximize the resulting lower bound on the log likelihood with respect to the model parameters α and β

These two steps are repeated until the lower bound on the log likelihood converges.

49

LDA: Smoothing

- Introduces Dirichlet smoothing on β to avoid the zero frequency word problem
- Fully Bayesian approach:



$$q(\beta_{1:k}, z_{1:M}, \theta_{1:M} | \lambda, \phi, \gamma) = \sum_{i=1}^{k} Dir(\beta_i | \lambda_i) \prod_{d=1}^{M} q_d(\theta_d, z_d | \phi_d, \gamma_d)$$

where $q_d(\theta, z \,|\phi, \gamma)$ is the variational distribution defined for LDA. We require an additional update for the new variational parameter λ:

$$\lambda_{ij} = \eta + \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni}^* w_{dn}^j$$

50

Topic Model Applications

- Information Retrieval
- Visualization
- Computer Vision
  - Document = image, word = “visual word”
- Bioinformatics
  - Genomic features, gene sequencing, diseases
- Modeling networks
  - cities, social networks

51

pLSA / LDA Libraries

- gensim (Python)
- MALLET (Java)
- topicmodels (R)
- Stanford Topic Modeling Toolbox

References

David M. Blei, Andrew Y. Ng, Michael I. Jordan. Latent Dirichlet Allocation. JMLR, 2003.

David Cohn and Huan Chang. Learning to probabilistically identify Authoritative documents. ICML, 2000.

David Cohn and Thomas Hoffman. The Missing Link - A Probabilistic Model of Document Content and Hypertext Connectivity. NIPS, 2000.

Thomas Hoffman. Probabilistic Latent Semantic Analysis. UAI-99, 1999.

Thomas Hofmann. Probabilistic Latent Semantic Indexing. SIGIR-99, 1999.