1 Introduction

- LDA is one of the early versions of a 'topic model' which was first presented by David Blei, Andrew Ng, and Michael I. Jordan in 2003.[p1]
- In essence, LDA is a generative model that allows observations about data to be explained by unobserved latent variables that explain why some parts of the data are similar, or potentially belong to groups of similar 'topics'
- Originally presented in a graphical model and using Variational Inference

1.1 Topic Modeling - Motivation

- Goal: Would like to take an object, say a book, and be able to describe it as a collection of weights on various topics. For instance, a book about the great depression into a vector of weights on: Economics 60% American History 30% and 10% Human suffering
- When this is accomplished distance measures can be used to compare object in the 'topic' space and recommendations can be derived on a 'nearest neighbor' basis
- This is to be done in an unsupervised way
2 LDA Algorithm

2.1 LDA Model

Two main parts to LDA generative model:

1. A set of 'topics' or distributions on 'words'
2. For each 'document' a distribution on topics

Generative Process for LDA:

1. For each document, generate a distribution on topics
   \[ \theta_d \sim \text{Dirichlet}(\alpha), d = 1, ..., D \] (1)

2. For the n'th word in the d’th document
   (a) Pick a topic for the word \( z_{dn} \sim \text{Discrete}(\theta_d) \)
   (b) Generate the word from the selected topic, \( w_{dn} \sim P(w_{dn}|z_{dn}, \beta) \) where \( \beta \) is a \( K \times |W| \) matrix with elements \( \beta_{ij} = p(w_j = 1|z_i = 1) \)

2.2 LDA Model - Observations

- Bayesian Model - requires an inference algorithm for learning a posterior distribution on parameters given data
- Only data is known - many unknown - distributions on topics, works, topics, indicators, etc.
- 'Bag of Words' model - no reference to order of words/topics and generative results are useless from a linguistic perspective

2.3 Graphical Representation of LDA Model
Which helps get the following factorization for the joint:

\[ P(\theta, Z, W|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta) \]  

(2)

### 2.4 Smoothed LDA Model

An additional Dirichlet distribution can be placed on the distribution of words given a topic. This can help control sparsity of the solution and help concentrate the distribution of words on relevant points.

**Generative Process for Smoothed LDA:**

1. Generate each topic, which is a distribution on words

\[ \varphi_k \sim Dirichlet(\beta), k = 1, ..., K \]  

(3)

2. For each document, generate a distribution on topics

\[ \theta_d \sim Dirichlet(\alpha), d = 1, ..., D \]  

(4)

3. For the \( n \)'th word in the \( d \)'th document
   
   (a) Pick a topic for the word \( z_{dn} \sim Discrete(\theta_d) \)
   
   (b) Generate the word from the selected topic, \( w_{dn} \sim Discrete(\varphi_{z_{dn}}) \)

### 2.5 Graphical Representation of Smoothed LDA

#### 2.6 Dirichlet Distribution

**Definition 1** (Dirichlet Probability Density).

\[ p(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \theta_1^{\alpha_1-1}...\theta_K^{\alpha_K-1} \]  

(5)
where the parameter $\alpha$ is a $k$-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function

2.7 Dirichlet Process (over a Gaussian)

constant $\alpha$ going higher as figures go down (1, 10, 100, 1000

2.8 Dirichlet Distribution - Observations

- 'distribution over distributions'
- Natural choice of prior for discrete variable
- Is a probability distribution on discrete probability distributions - although it's a continuous distribution
- depending on choice of $\alpha$ distribution can favor sparse discrete probabilities over $k$ - high $\alpha$ gives more 'topics per document and vice versa
- for $k = 2$ Dirichlet converges to Beta distribution

$$E(\theta_i) = \frac{\alpha_i}{\sum \alpha_i}$$

3 Inference

3.1 Framing the Problem

1. Known - $W$ - The 'words'
2. Unknown - $\beta$ - distribution over works per topic ( $K \times |W|$ matrix
3. Latent variable - $Z$ - topic per word
4. Latent variable - $\theta$ - distribution of topics per document
The two latent variable scale with the size of the input data where \( \beta \) is a fixed size input. Inference of latent variables requires calculation of the posterior:

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

(6)

However, the denominator is intractable. Two main approaches are typically used for inference:

1. Variational Inference - via EM - faster and in the original paper
2. Gibbs Sampling via MCMC (or collapsed Gibbs sampling) - exact

### 3.2 Variational Inference

1. Simplifying assumption is made on the posterior. In this case the 'mean-field assumption', where each latent variable is controlled by its own variational variable to define a class of possible approximate posteriors \( q \):

\[
q(\theta, z|\gamma) = q(\theta|\gamma_\theta) \prod_{n=1}^{N} q(z_n|\gamma_n)
\]

2. Derive an algorithm for computing \( \gamma \) in a way that minimize the KL divergence, or equivalently maximizing the ELBO, of \( q \) with respect to true posterior \( p \) as a function of \( \gamma \)

3. Implement EM algorithm until ELBO converges:
   
   (a) E-step: compute \( \gamma \) per document as discussed in (2)
   
   (b) M-step: with choice of \( \gamma \) maximize the lower bound on log likelihood of \( p(w|\alpha, \beta) \) - This corresponds to maximum likelihood under the approximate posterior from E-step.

   - 'ELBO': evidence lower bound

\[
\log p(w|\alpha, \beta) = L(\gamma; \alpha, \beta) + KL(q, p)
\]

\[
L(\gamma; \alpha, \beta) = \mathbb{E}_q[\log P(\theta, z, w|\alpha, \beta)] - \mathbb{E}_q[\log q(\theta, z)]
\]

   - Solutions to both the E and M steps are iterative and can be derived by taking derivatives of ELBO with respect to the relevant variables - 'coordinate ascent'

### 3.3 MCMC

**Gibbs Sampling:**

- In this setting can treat \( \beta \) as a latent variable too.
- \( p(\beta, \theta, z|w) \sim \{\text{GibbsSample}\} \)
- start with initial guesses for \( \beta, \theta, z \)
- iterate through all dimensions of each variables
– for example: \( z_{i}^{t+1} \sim p(z_i | \beta, \theta, z) \)

• do this over many steps until posterior is sampled

**Collapsed Gibbs Sampling** [p4]

• By analytically marginaling out variables we are able to reduce the need to samples distributions for all latent variables but \( z \sim p(z|w) \)

• \( \theta \) and \( \beta \) are recovered by using expectations overs \( p(z|q) \)

• This method is much faster than normal Gibbs Sampling and in fact is also faster than VI

• Method also work seems to work on mini-batches of data

### 4 Applications

#### 4.1 LDA and Matrix Factorization

For a document, the probability of a given word, given a distribution over topics for the document and a distribution of words for each topic can be represented as a product of two non-negative matrices

\[
P(x_{dn} = i | \beta, \theta) = \sum_{k=1}^{K} P(x_{dn} = i, z_{dn} = k | \beta, \theta_d)
\]

\[
= \sum_{k=1}^{K} P(x_{dn} = i | \beta, z_{dn} = k) P(z_{dn} = k | \theta_d)
\]

\[
= \sum_{k=1}^{K} \beta_{ik} \theta_{dk} = (B\Theta)_{id}
\]

For \( B = [\beta_1...\beta_K] \), \( \Theta = [\theta_1...\theta_D] \)

### References


