Introduction to Probabilistic Topic Models

• We want to find themes (or topics) in documents
  – useful for e.g. search or browsing

• We don’t want to do supervised topic classification
  – rather not fix topics in advance nor do manual annotation

• Need an approach which automatically teases out the topics

• This is essentially a clustering problem - can think of both words and documents as being clustered
Key Assumptions behind the LDA Topic Model

• Documents exhibit multiple topics (but typically not many)

• LDA is a probabilistic model with a corresponding *generative process*
  – each document is assumed to be generated by this (simple) process

• A *topic* is a distribution over a fixed vocabulary
  – these topics are assumed to be generated first, before the documents

• Only the number of topics is specified in advance
The Generative Process

To generate a document:

1. Randomly choose a distribution over topics

2. For each word in the document
   a. randomly choose a topic from the distribution over topics
   b. randomly choose a word from the corresponding topic (distribution over the vocabulary)

- Note that we need a distribution over a distribution (for step 1)
- Note that words are generated independently of other words (unigram bag-of-words model)
The Generative Process more Formally

- Some notation:
  - $\beta_{1:K}$ are the topics where each $\beta_k$ is a distribution over the vocabulary
  - $\theta_d$ are the topic proportions for document $d$
  - $\theta_{d,k}$ is the topic proportion for topic $k$ in document $d$
  - $z_d$ are the topic assignments for document $d$
  - $z_{d,n}$ is the topic assignment for word $n$ in document $d$
  - $w_d$ are the observed words for document $d$

- The joint distribution (of the hidden and observed variables):

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{K} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \prod_{n=1}^{N} p(z_{d,n}|\theta_d)p(w_{d,n}|\beta_{1:K}, z_{d,n})$$
Plate Diagram of the Graphical Model

- Note that only the words are observed (shaded)
- $\alpha$ and $\eta$ are the parameters of the respective Dirichlet distributions (more later)
- Note that the topics are generated (not shown in earlier pseudo code)
- Plates indicate repetition

Picture from Blei 2012
Multinomial Distribution

- **Multinomial** distribution: \( x_i \in \{0, \ldots, n\} \)

\[
P(x|\theta) = \frac{n!}{\prod_{i=1}^{d} x_i!} \prod_{i=1}^{d} \theta_i^{x_i}, \quad n = \sum_{i=1}^{d} x_i, \quad \sum_{i=1}^{d} \theta_i = 1, \quad \theta_i \geq 0
\]

- When \( n = 1 \) the multinomial distribution simplifies to

\[
P(x|\theta) = \prod_{i=1}^{d} \theta_i^{x_i}, \quad \sum_{i=1}^{d} \theta_i = 1, \quad \theta_i \geq 0
\]

- a unigram language model with 1-of-V coding (\( d = V \) the vocabulary size)
- \( x_i \) indicates word \( i \) of the vocabulary observed, \( x_i = \begin{cases} 1, & \text{word } i \text{ observed} \\ 0, & \text{otherwise} \end{cases} \)
- \( \theta_i = P(w_i) \) the probability that word \( i \) is seen
The Dirichlet Distribution

- **Dirichlet** (continuous) distribution with parameters $\alpha$

  \[ p(x | \alpha) = \frac{\Gamma(\sum_{i=1}^{d} \alpha_i)}{\prod_{i=1}^{d} \Gamma(\alpha_i)} \prod_{i=1}^{d} x_i^{\alpha_i - 1}; \quad \text{for "observations": } \sum_{i=1}^{d} x_i = 1, \quad x_i \geq 0 \]

- $\Gamma()$ is the Gamma distribution

- **Conjugate prior** to the multinomial distribution
  (form of posterior $p(\theta | D, M)$ is the same as the prior $p(\theta | M)$)
Dirichlet Distribution Example

- Parameters: \((\alpha_1, \alpha_2, \alpha_3)\)
Parameter Estimation

- Main variables of interest:
  - $\beta_k$: distribution over vocabulary for topic $k$
  - $\theta_{d,k}$: topic proportion for topic $k$ in document $d$

- Could try and get these directly, e.g., using EM (Hoffmann, 1999), but this approach not very successful

- One common technique is to estimate the posterior of the word-topic assignments, given the observed words, directly (whilst marginalizing out $\beta$ and $\theta$)
Gibbs Sampling

- Gibbs sampling is an example of a Markov Chain Monte Carlo (MCMC) technique

- Markov chain in this instance means that we sample from each variable one at a time, keeping the current values of the other variables fixed
Posterior Estimate

- The Gibbs sampler produces the following estimate, where, following Steyvers and Griffiths:

  - $z_i$ is the topic assigned to the $i$th token in the whole collection;
  - $d_i$ is the document containing the $i$th token;
  - $w_i$ is the word type of the $i$th token;
  - $z_{-i}$ is the set of topic assignments of all other tokens;
  - $\cdot$ is any remaining information such as the $\alpha$ and $\eta$ hyperparameters:

$$P(z_i = j|z_{-i}, w_i, d_i, \cdot) \propto \frac{C_{w_{ij}}^{WT} + \eta}{\sum_{w=1}^{W} C_{w_{j}}^{WT} + W\eta} \frac{C_{d_{ij}}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_{it}}^{DT} + T\alpha}$$

where $C^{WT}$ and $C^{DT}$ are matrices of counts (word-topic and document-topic)
Posterior Estimates of $\beta$ and $\theta$

$$\beta_{ij} = \frac{C_{ij}^{WT} + \eta}{\sum_{k=1}^{W} C_{kj}^{WT} + W \eta}$$

$$\theta_{dj} = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T \alpha}$$

- Using the count matrices as before, where $\beta_{ij}$ is the probability of word type $i$ for topic $j$, and $\theta_{dj}$ is the proportion of topic $j$ in document $d$. 

MPhil in Advanced Computer Science
References

- David Blei’s webpage is a good place to start
