Topic Models

Dorota Glowacka
dorota.glowacka@ed.ac.uk
Topic Models

• We want to find themes (or topics) in documents, which is useful for, e.g. searching or visualising themes in document collections.

• We don’t want to do manual annotation – time consuming and relies on availability of human annotators.

• Approach should automatically tease out the topics.

• Essentially a clustering problem where both words and documents are clustered.
Topic Models

Simple intuition: documents exhibit multiple topics
Latent Dirichlet Allocation (LDA)

- Each **topic** is a distribution over **words**
- Each **document** is a **mixture** of corpus-wide topics
- Each **word** is **drawn** from one of these topics
• In reality, we only observe documents
• The other structure are hidden variables
Posterior Distribution

• Our goal is to infer the **hidden variables**
• i.e. compute their distribution conditioned on the documents:  
  \[ p(\text{topics, proportions, assignments}|\text{documents}) \]
LDA: key assumptions

• Documents exhibit multiple topics (but typically not many)

• LDA is a probabilistic model with a corresponding generative process (each document is generated by this process)

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

• Only the number of topics is specified in advance
Document Generation

1. Choose a number of words $N$ the document will have.

2. Choose topic mixture for document according to Dirichlet distribution over a set of $T$ topics.

3. Generate each word $w_i$ in the document by:
   1. Randomly choosing a topic from the distribution over topics.
   2. Randomly choosing a word from the corresponding topic (distribution over vocabulary).
Document Generation

1. Pick 5 to be the number of words in D.
2. Decide D will be ½ about food and ½ about cute animals.
3. Pick first word to come from the food topic: broccoli
4. Pick second word to come from the cute animals topic: panda
5. Pick third word from the cute animals topic: adorable.
6. Pick fourth word from the food topic: cherries.
7. Pick fifth word from food topic: eating.

What is the document generated under the LDA model?
1. Go through each document and randomly assign each word in the document to one of $T$ topics.

2. For each document $d$, go through each word $w$ in $d$ and for each topic $t$ compute:
   1. $p(topic\ t|document\ d)$
   2. $p(word\ w|topic\ t)$

3. Reassign $w$ to a new topic, where we choose topic $t$ with probability: $p(topic\ t|document\ d) \times p(word\ w|topic\ t)$

4. After repeating the previous step a large number of times, you will eventually reach a roughly steady state where your assignments are pretty good.
Generative Process

For $j = 1 \ldots T$ topics
choose $\phi^{(j)} \approx \text{Dirichlet}(\beta)$ $\phi_{V}^{(j)} \ldots \phi_{V}^{(j)}$ : prob. of each wd. in topic $j$

For $d = 1 \ldots D$ documents
choose $\theta^{(d)} \approx \text{Dirichlet}(\alpha)$ $\theta_{1}^{(d)} \ldots \theta_{T}^{(d)}$ : prob. of each topic in doc. $d$

For $i = 1 \ldots N_{d}$ words in doc $d$
choose $z_{i} \approx \text{Multinomial}(\theta^{(d)})$ $z_{i}$: topic of word $i$
choose $w_{i} \approx \text{Multinomial}(\phi_{z_{i}}^{(j)})$ $w_{i}$: identity of word $i$
• Nodes are random variables; edges indicate dependence
• *Shaded nodes* are observed; plates \( \approx \) replicated variables
The Graphical Model

- Proportions parameter $\alpha$ indicates per-document topic proportions.
- Topics parameter $\beta$ leads to $\phi$ and $\theta$. $\phi$ is connected to $W$, representing observed words.
- $\theta$ is related to $Z$, indicating per-word topic assignment.
- $\theta$ and $\alpha$ are connected to the nodes $\alpha$ and $\theta$ respectively, showing dependence.

- Nodes are random variables; edges indicate dependence.
- *Shaded nodes* are observed; plates $\approx$ replicated variables.
The Graphical Model

\[ P(w_i) = \sum_{j=1}^{T} P(w_i | z_i = j) P(z_i = j) \]

- \( P(z_i = j) \) is probability that \( j \)th topic was sampled from \( i \)th word token
- \( P(w_i | z_i = j) \) is probability of word \( w_i \) under topic \( j \)

- \( \phi^{(j)} = P(z_i = j) \)
- \( \theta^{(d)} = P(z) \)
Dirichlet Distribution

\[ p(\theta | \alpha) = \text{Dir}(\alpha_1, \ldots, \alpha_T) \frac{\Gamma \left( \sum_j \alpha_j \right)}{\prod_j \Gamma(\alpha_j)} \prod_{j=1}^{T} \theta_j^{\alpha_j-1} \]

- Dirichlet distribution is an exponential family distribution over the simplex, i.e. positive vectors that sum to one.
- It is conjugate to the multinomial distribution. Given a multinomial observation, the posterior distribution of \( \theta \) is a Dirichlet.
- Parameter \( \alpha \) smoothes topic distribution in the document.
- \( \theta \) smoothes word distribution in every topic.
α = 1
$\alpha = 10$
$\alpha = 100$
$\alpha = 0.1$
$\alpha = 0.001$
Inference with Gibbs Sampling

• An iterative process
• Start with random topic assignment for each word
  – Assume you know (from the previous iteration) the topics
  – Determine the probabilities of each topic-assignment given the rest of data
  – Choose the most probable assignment.
• Iterate until converge
Gibbs Sampling

- Collection of documents is a set of words $w_i$ and document indices $d_i$, for each word token $i$.

\[
P(z_i = j | z_{-i}, w_i, d_i) = \frac{C_{d_i,j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_i,t}^{DT} + T \alpha} \frac{C_{w_i,j}^{WT} + \beta}{\sum_{w=1}^{W} C_{w,j}^{WT} + W \beta}
\]

- From this conditional distribution a topic is sampled and stored as the new topic assignment for this word token.

$z_i = j$: topic assignment of token $i$ to topic $j$
$z_{-i}$: topic assignments of all other word tokens
$C^{WT}$: matrix of counts with dimensions $W \times T$
$C^{DT}$: matrix of counts with dimensions $D \times T$
Posterio Estimates of $\beta$ and $\theta$

\[
\phi_{ij} = \frac{C_{ij}^{WT} + \beta}{\sum_{t=1}^{W} C_{w_{ij}}^{WT} + W \beta}
\]

\[
\theta_{d_{j}} = \frac{C_{d_{ij}}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_{ij}}^{DT} + T \alpha}
\]

- $\phi_{ij}$: normalizes word-topic count matrix (probability of word type $i$ for topic $j$)
- $\theta_{d_{j}}$: normalizes the counts in the documents-topic matrix (probability that document $d$ belongs to topic $j$)
- Technically the posterior mean of the parameters $\phi$ and $\theta$
Example Inference

Seeking Life’s Bare (Genetic) Necessities

How many genes does an organism need to survive? Last week at the genome meeting here, two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today’s organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn’t be enough.

Although the numbers don’t match precisely, those predictions are not all that far apart,” especially in comparison to the 75,000 genes in the human genome, notes Sir Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. “It may be a way of organizing any newly sequenced genome,” explains Aready Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Data: The OCR’ed collection of Science from 1990-2000
17K documents
11M words
20K unique terms (stop words and rare words removed)
Model: 100-topic LDA model
Why does LDA “work”?

• LDA trades off two goals:
  1. For each document, allocate its words to as few topics as possible.
  2. For each topic, assign high probability to as few terms as possible.

• These goals are at odds:
  – Putting a document in a single topics makes 2 hard: all words must have probability under that topic.
  – Putting very few words in each topic makes 1 hard: to cover a document’s words, it must assign many topics to it.

• Trading off these goals finds groups of tightly co-occurring words.
LDA summary

• LDA is a *probabilistic model* of text. It casts the problem of discovering *themes in large document collections* as a posterior inference.
• It allows *visualisation* of *hidden thematic structure* of large document collections.
• LDA is a building block of many applications.
• It is popular because organizing and finding patterns in data has become important in sciences, humanities, industry, etc.
Extending Topic Models

- LDA can be embedded in more complex models, embodying further intuitions about the structure of the text.
- E.g. used in models that account for syntax, authors, correlation or hierarchies.
• Data generating distribution can be changed. We can apply mixed-membership assumptions to many kinds of data.
• E.g. we can build models of images, social networks, music, computer code, genetic data, etc.
Extending Topic Models

- The posterior can be used in creative ways.
- E.g. we can use inferences in information retrieval, recommendation, visualisation, summarization, etc.