Probabilistic Models for Text and Graphs

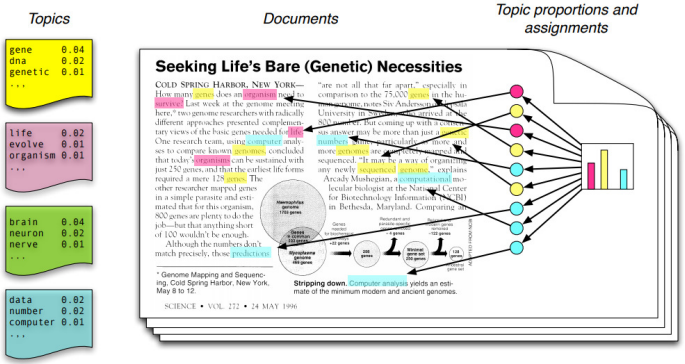
Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

April 5, 2018

Topic Models for Text

- Goal of topic modeling is to learn the topics that underlie the document collection, and represent each document in terms of these topics (fraction of various topics)



- Topic models are based on assigning words in each document to clusters/topics (each cluster can be thought of as representing a "topic"): Essentially word clustering with many documents!

What's a Topic?

- Informally, a topic represents a group of similar words representative of that topic
- Suppose the vocabulary consists of V words
- Formally, each topic is a probability distribution over the V words

Topic 247

word	prob.
DRUGS	.069
DRUG	.060
MEDICINE	.027
EFFECTS	.026
BODY	.023
MEDICINES	.019
PAIN	.016
PERSON	.016
MARIJUANA	.014
LABEL	.012
ALCOHOL	.012
DANGEROUS	.011
ABUSE	.009
EFFECT	.009
KNOWN	.008
PILLS	.008

Topic 5

word	prob.
RED	.202
BLUE	.099
GREEN	.096
YELLOW	.073
WHITE	.048
COLOR	.048
BRIGHT	.030
COLORS	.029
ORANGE	.027
BROWN	.027
PINK	.017
LOOK	.017
BLACK	.016
PURPLE	.015
CROSS	.011
COLORED	.009

Topic 43

word	prob.
MIND	.081
THOUGHT	.066
REMEMBER	.064
MEMORY	.037
THINKING	.030
PROFESSOR	.028
FELT	.025
REMEMBERED	.022
THOUGHTS	.020
FORGOTTEN	.020
MOMENT	.020
THINK	.019
THING	.016
WONDER	.014
FORGET	.012
RECALL	.012

Topic 56

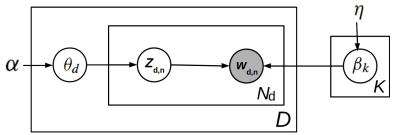
word	prob.
DOCTOR	.074
DR.	.063
PATIENT	.061
HOSPITAL	.049
CARE	.046
MEDICAL	.042
NURSE	.031
PATIENTS	.029
DOCTORS	.028
HEALTH	.025
MEDICINE	.017
NURSING	.017
DENTAL	.015
NURSES	.013
PHYSICIAN	.012
HOSPITALS	.011

- Each topic, say topic number k, can be represented by a probability vector β_k of size V
  - β_{kv} denotes the proability of word v in topic k and $\sum_{v=1}^{V} \beta_{kv} = 1$
  - High-probability words are representative of that topic
- $\{\beta_k\}_{k=1}^{K}$ denotes the set of K topics

Latent Dirichlet Allocation (LDA) - Blei et al, 2003

Total D docs, $w_{d,n}$: n-th word in the d-th document. Assume N_d = no. of words in document d

LDA associates each observed word $w_{d,n} \in \{1, \ldots, V\}$ with a topic $z_{d,n} \in \{1, \ldots, K\}$



The LDA (assuming hyperparams α, η known) generative model is like a mixture model for words

The N_d words in each document are generated via a mixture model, as follows

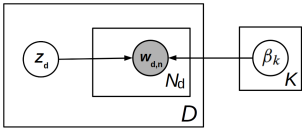
- Draw K topics $\{\beta_k\}_{k=1}^K$ shared by all documents $d = 1, \ldots, D$

$$\beta_k \sim \text{Dirichlet}(\eta, \ldots, \eta) \qquad k = 1, \ldots, K$$

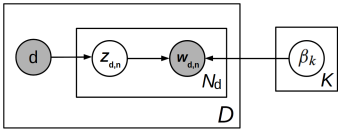
- For each document $d = 1, \ldots, D$
  - Draw its K-dim topic proportion vector $\theta_d \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$
  - For each word $n = 1, \ldots, N_d$ in document d
    - First choose the topic for this word: $z_{d,n} \sim \text{multinoulli}(\theta_d)$
    - Now generate the word form the chosen topic: $w_{d,n} \sim \text{multinoulli}(\beta_{z_{d,n}})$

Some LDA Precursors

- A naïve mixture model: Each document having a single topic, all words from that topic

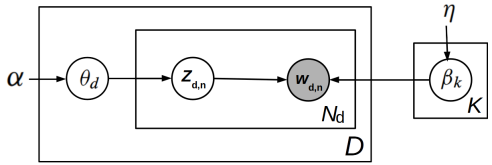


- Each word in a doc has its own topic: Probabilistic Latent Semantic Analysis (Hofmann (2001))



- PLSA has too many parameters $p(w, d) = p(d)p(w|d) = p(d)\sum_{k=1}^{K} p(w|z = k)p(z = k|d)$
  - VK params for $p(w|z = k)$, KD params for $p(z = k|d)$. Grows in the number of documents
  - LDA however models $p(z = k|d)$ via random variables θ_d, not tied to only training data, so can extend to previously unseen documents unlike PLSA. A subtle but important difference (Blei et al, 2003)
- Also, unlike LDA, PLSA is not a Bayesian model

Inference for LDA

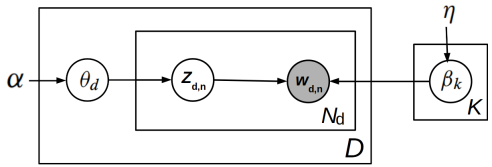


- The goal is to infer the posterior distribution over all the latent variables

$$p(\mathbf{Z}, \theta, \beta | \mathbf{W}, \alpha, \eta) = \frac{p(\mathbf{W}|\beta, \mathbf{Z}) p(\mathbf{Z}|\theta) p(\beta|\eta) p(\theta|\alpha)}{p(\mathbf{W}|\alpha, \eta)} \qquad \text{(assuming hyperparams } \alpha, \eta \text{ are fixed)}$$

- $\mathbf{Z}$: Topic assignments of all words across all documents
- $\theta = \{\theta_1, \ldots, \theta_D\}$: Topic proportion vectors; each θ_d is a K-dim vector
- $\beta = \{\beta_1, \ldots, \beta_K\}$: The K topics; each β_k is a V-dim vector
- $\mathbf{W} = \{\boldsymbol{w}_{d,n}\}, d = 1, \ldots, D, n = 1, \ldots, N_d$: Collection of all words in all the documents

Inference for LDA



- A wide variety of inference methods exist for LDA (MCMC, VB, EP, batch, online, ...)
- LDA is a locally conjugate model (recall that the model uses Dirichlet/multinoulli distr.)
  - The likelihood and priors are all exponential family distributions
- MCMC, in particular Gibbs sampling, is very easy to derive due to local conjugacy
- Local conjugacy also allow deriving a simple VB algorithm (the original LDA paper† used VB)
- Note: Can even collapse some variables and do collapsed Gibbs or collapsed VB

† Latent Dirichlet Allocation (Blei et al, 2003)

LDA Evaluation

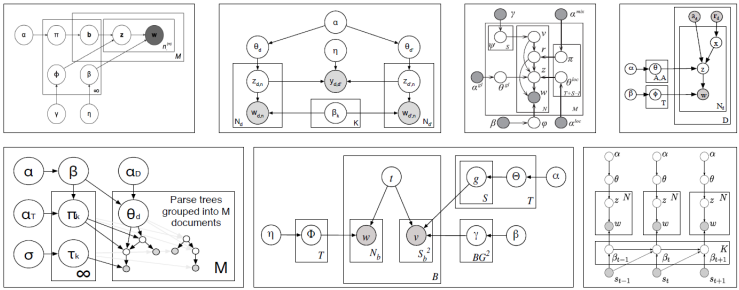
- Many ways to test how well LDA performs on some data
- Extrinsic measures: Perform LDA and measure performance on some external task (e.g., accuracy of a document classification model using the topic based document representation)
- Perplexity is another intrinsic measure

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

.. where $\boldsymbol{w}_d$ collectively denotes the set of words in document d

- We set aside some documents as a test set and compute the average proability the model assigns to the words in these held-out documents
- A high average probability (low perplexity) implies a good fit to the data

LDA is easy to extend

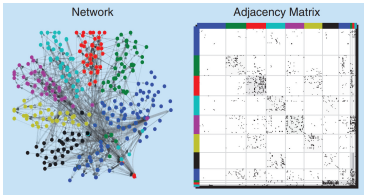


Several extensions of LDA (too many to list here :)). Some include

- LDA with document labels (supervised LDA)
- LDA for multimodal data (images and text) and multilingual data
- LDA with topic hierarchies/correlations
- LDA with time-evolving topics
- LDA with HMM (takes into account word order)
- LDA for graphs

Probabilistic Models for Graphs

- Graphs/networks are ubiquitous in many domains (social networks, computer networks, protein-protein networks, citation networks, etc.)
- A graph between N entities/nodes can be represented by an $N \times N$ adjacency matrix $\mathbf{A}$



- Given a (partially observed) graph, we may be interested in tasks such as
  - Identifying the latent structure (e.g., clusters of similar nodes)
  - Predicting possibility of a link between two node
- We can design probabilistic models for such tasks

Probabilistic Models for Graphs

- Assume each entity/node n to have a latent representation vector ("embedding") $\boldsymbol{z}_n$ of size K
- We can model each link/non-link $A_{nm} \in \{0, 1\}$ via a probability model, e.g.

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta) = f(\boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

  where f is some function of $\boldsymbol{z}_n, \boldsymbol{z}_m$ and params θ, and returns a probability

- The overall probability of the observed graph

$$p(\mathbf{A} | \mathbf{Z}, \theta) = \prod_{n,m} p(A_{nm} | \boldsymbol{z}_n, \boldsymbol{z}_m, \theta)$$

- Various models differ in terms of what the embeddings $\boldsymbol{z}_n$ look like, and what f is defined as
- Some representative models
  - Latent Space Model
  - Stochastic Blockmodel
  - Mixed-Membership Blockmodel (MMSB)

Latent Space Model for Graphs

- LSM assumes each node to have a real-valued embedding vector $\boldsymbol{z}_n \in \mathbb{R}^K$
- The link probability is defined in terms of the Euclidean similarity between nodes in latent space
- One such possible model would be

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m) = \sigma(-||\boldsymbol{z}_n - \boldsymbol{z}_m||)$$

- A reasonable model for link prediction
- However, the real-valued embeddings in LSM don't provide a good interpretability
  - Therefore not very ideal for discovering clusters etc.
- Blockmodels and its variants has such a properties as we will see next

Stochastic Blockmodel

- SBM assumes each node belongs to a cluster/community or "block" (total K clusters)
- The node n's cluster membership denoted by a one-hot vector $\boldsymbol{z}_n$ of size K
- Assume probability of link b/w a node with $\boldsymbol{z}_n = k$ and another node with $\boldsymbol{z}_m = \ell$ to be $\eta_{k\ell}$
- Therefore the probability of a link between nodes n and m will be

$$p(A_{nm} = 1 | \boldsymbol{z}_n, \boldsymbol{z}_m, \eta) = \eta_{\boldsymbol{z}_n, \boldsymbol{z}_m}$$

- The full generative model looks like
  - For $n = 1, \ldots, N$

$$\boldsymbol{z}_n \sim \text{multinoulli}(\pi)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n-1$

$$A_{nm} \sim \text{Bernoulli}(\eta_{\boldsymbol{z}_n, \boldsymbol{z}_m})$$

  - Note: π and η can also be given priors

Mixed-Membership Stochastic Blockmodel

- Unlike SBM, the MMSB assumes each node n to have a probability vector π_n
- π_n denotes the probabilities of memberships of node n in each of the K communities
- The full generative model looks like
- For $n = 1, \ldots, N$

$$\pi_n \sim \text{Dirichlet}(\alpha)$$

  - For $n = 1, \ldots, N$
    - For $m = 1, \ldots, n$

$$\begin{aligned} \boldsymbol{z}_{n \to m} &\sim \text{multinoulli}(\pi_n) \\ \boldsymbol{z}_{m \to n} &\sim \text{multinoulli}(\pi_m) \\ A_{nm} &\sim \text{Bernoulli}(\eta_{\boldsymbol{z}_{n \to m}, \boldsymbol{z}_{m \to n}}) \end{aligned}$$

- Unlike SBM in which node n has a unique one-hot $\boldsymbol{z}_n$ vector, in MMSB, for each link/non-link A_{nm}, node n displays a different community membership $\boldsymbol{z}_{n \to m}$ based on the node m it is interacting with (likewise for node m).
- In contrast to SBM, in MMSB, the unique vector is π_n.