Machine Learning and Computational Statistics

David Sontag

New York University

Lecture 13, April 29, 2014

Expectation maximization

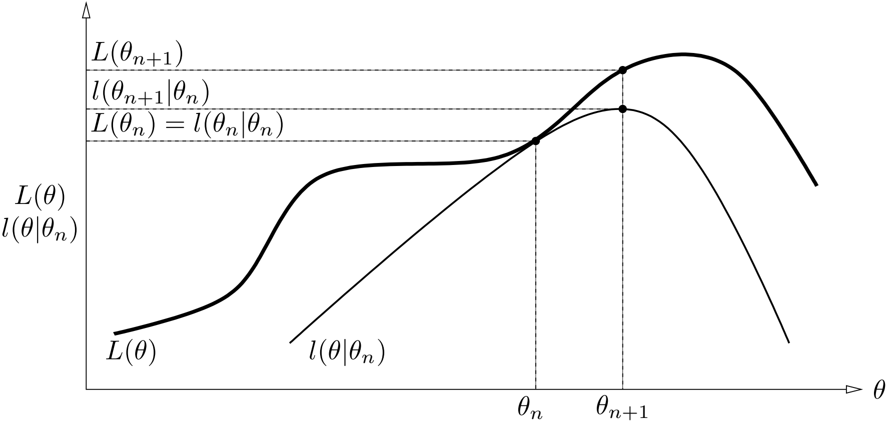
Algorithm is as follows:

1. Write down the **complete log-likelihood** $\log p(\mathbf{x}, \mathbf{z}; \theta)$ in such a way that it is linear in $\mathbf{z}$
2. Initialize θ_0, e.g. at random or using a good first guess
3. Repeat until convergence:

$$\theta_{t+1} = \arg\max_{\theta} \sum_{m=1}^{M} E_{p(\mathbf{z}_m|\mathbf{x}_m;\theta_t)}[\log p(\mathbf{x}_m, \mathbf{Z}; \theta)]$$

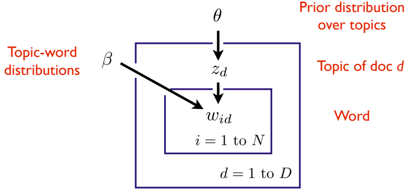
- Notice that $\log p(\mathbf{x}_m, \mathbf{Z}; \theta)$ is a random function because $\mathbf{Z}$ is unknown
- By linearity of expectation, objective decomposes into expectation terms and data terms
- "E" step corresponds to computing the objective (i.e., the **expectations**)
- "M" step corresponds to **maximizing** the objective

Derivation of EM algorithm



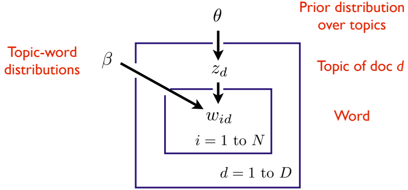
(Figure from tutorial by Sean Borman)

Application to mixture models



- This model is a type of (discrete) **mixture model**
    - Called *multinomial* naive Bayes (a word can appear multiple times)
    - Document is generated from a single topic

EM for mixture models



- The complete likelihood is $p(\mathbf{w}, \mathbf{Z}; \theta, \beta) = \prod_{d=1}^{D} p(\mathbf{w}_d, Z_d; \theta, \beta)$, where

$$p(\mathbf{w}_d, Z_d; \theta, \beta) = \theta_{Z_d} \prod_{i=1}^{N} \beta_{Z_d, w_{id}}$$

- Trick #1: re-write this as

$$p(\mathbf{w}_d, Z_d; \theta, \beta) = \prod_{k=1}^{K} \theta_k^{1[Z_d=k]} \prod_{i=1}^{N} \prod_{k=1}^{K} \beta_{k, w_{id}}^{1[Z_d=k]}$$

EM for mixture models

- Thus, the complete log-likelihood is:

$$\log p(\mathbf{w}, \mathbf{Z}; \theta, \beta) = \sum_{d=1}^{D} \left(\sum_{k=1}^{K} 1[Z_d = k] \log \theta_k + \sum_{i=1}^{N} \sum_{k=1}^{K} 1[Z_d = k] \log \beta_{k,w_{id}} \right)$$

- In the "E" step, we take the expectation of the complete log-likelihood with respect to $p(\mathbf{z} \mid \mathbf{w}; \theta^t, \beta^t)$, applying linearity of expectation, i.e.

$$E_{p(\mathbf{z}|\mathbf{w};\theta^t,\beta^t)}[\log p(\mathbf{w}, \mathbf{z}; \theta, \beta)] =$$

$$\sum_{d=1}^{D} \left(\sum_{k=1}^{K} p(Z_d = k \mid \mathbf{w}; \theta^t, \beta^t) \log \theta_k + \sum_{i=1}^{N} \sum_{k=1}^{K} p(Z_d = k \mid \mathbf{w}; \theta^t, \beta^t) \log \beta_{k,w_{id}} \right)$$

- In the "M" step, we maximize this with respect to θ and β

EM for mixture models

- Just as with complete data, this maximization can be done in closed form
- First, re-write expected complete log-likelihood from

$$\sum_{d=1}^{D}\left(\sum_{k=1}^{K} p(Z_d = k \mid \mathbf{w}; \theta^t, \beta^t)\log\theta_k + \sum_{i=1}^{N}\sum_{k=1}^{K} p(Z_d = k \mid \mathbf{w}; \theta^t, \beta^t)\log\beta_{k,w_{id}}\right)$$

to

$$\sum_{k=1}^{K}\log\theta_k\sum_{d=1}^{D} p(Z_d = k \mid \mathbf{w}_d; \theta^t, \beta^t) + \sum_{k=1}^{K}\sum_{w=1}^{W}\log\beta_{k,w}\sum_{d=1}^{D} N_{dw}\, p(Z_d = k \mid \mathbf{w}_d; \theta^t, \beta^t)$$

- We then have that

$$\theta_k^{t+1} = \frac{\sum_{d=1}^{D} p(Z_d = k \mid \mathbf{w}_d; \theta^t, \beta^t)}{\sum_{\hat{k}=1}^{K}\sum_{d=1}^{D} p(Z_d = \hat{k} \mid \mathbf{w}_d; \theta^t, \beta^t)}$$

Latent Dirichlet allocation (LDA)

- **Topic models** are powerful tools for exploring large data sets and for making inferences about the content of documents

Documents

Topics

politics	religion	sports
president	hindu	baseball
obama	judiasm	soccer
washington	ethics	basketball
religion	buddhism	football
...	...	...

- Many applications in information retrieval, document summarization, and classification

New document

Words $w_1, \ldots, w_N$

What is this document about?

weather	.50
finance	.49
sports	.01

Distribution of topics θ

- LDA is one of the simplest and most widely used topic models

Generative model for a document in LDA

1. Sample the document's **topic distribution** θ (aka topic vector)

$$\theta \sim \mathrm{Dirichlet}(\alpha_{1:T})$$

   where the $\{\alpha_t\}_{t=1}^T$ are fixed hyperparameters. Thus θ is a distribution over T topics with mean $\theta_t = \alpha_t / \sum_{t'} \alpha_{t'}$

2. For $i = 1$ to N, sample the **topic** z_i of the i'th word

$$z_i | \theta \sim \theta$$

3. ... and then sample the actual **word** w_i from the z_i'th topic

$$w_i | z_i \sim \beta_{z_i}$$

   where $\{\beta_t\}_{t=1}^T$ are the *topics* (a fixed collection of distributions on words)

Generative model for a document in LDA

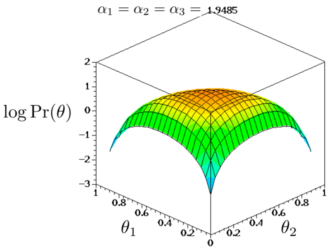
1. Sample the document's **topic distribution** θ (aka topic vector)

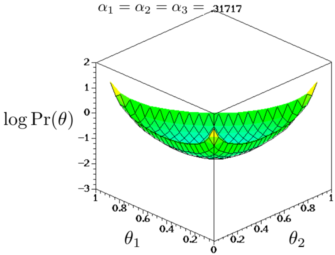
$$\theta \sim \mathrm{Dirichlet}(\alpha_{1:T})$$

where the $\{\alpha_t\}_{t=1}^T$ are hyperparameters.The Dirichlet density, defined over $\Delta = \{\vec{\theta} \in \mathbb{R}^T : \forall t\ \theta_t \geq 0, \sum_{t=1}^T \theta_t = 1\}$, is:

$$p(\theta_1, \ldots, \theta_T) \propto \prod_{t=1}^T \theta_t^{\alpha_t - 1}$$

For example, for T=3 ($\theta_3 = 1 - \theta_1 - \theta_2$):





Generative model for a document in LDA

3. ... and then sample the actual **word** w_i from the z_i'th topic

$$w_i | z_i \sim \beta_{z_i}$$

where $\{\beta_t\}_{t=1}^{T}$ are the *topics* (a fixed collection of distributions on words)

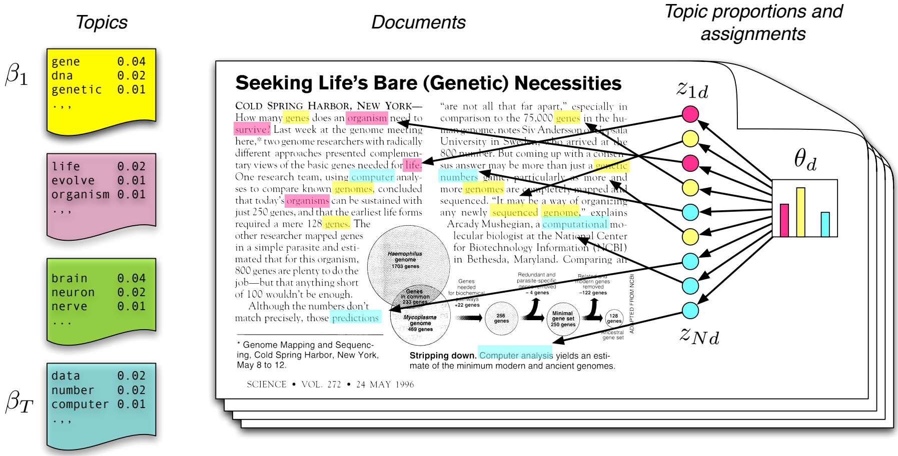
Documents

Topics

politics .0100	religion .0500	sports .0105
president .0095	hindu .0092	baseball .0100
obama .0090	judiasm .0080	soccer .0055
washington .0085	ethics .0075	basketball .0050
religion .0060	buddhism .0016	football .0045
...	...	...

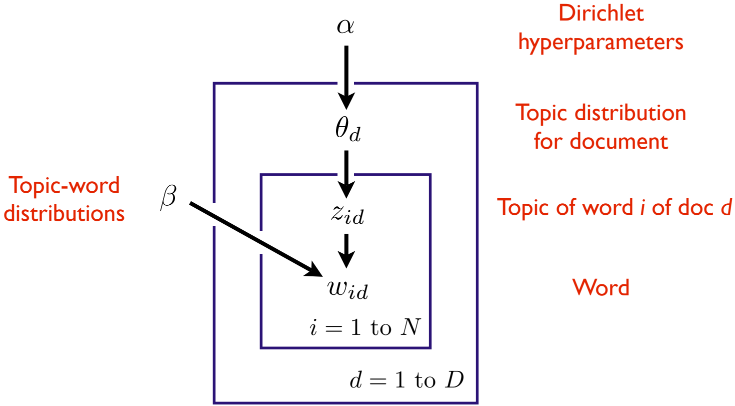
$$\beta_t = \{\, p(w \mid z = t) \,\}$$

Example of using LDA



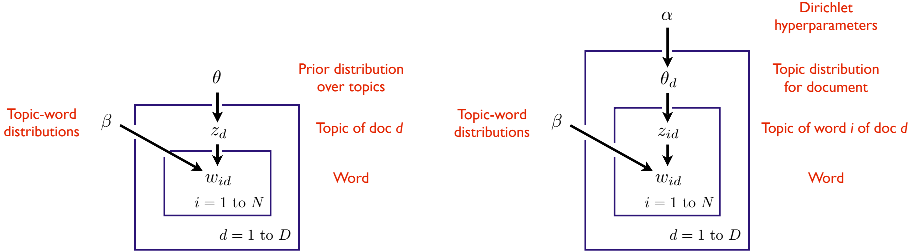
(Blei, *Introduction to Probabilistic Topic Models*, 2011)

"Plate" notation for LDA model



Variables within a plate are replicated in a conditionally independent manner

Comparison of mixture and admixture models



- Model on left is a **mixture model**
  - Called *multinomial* naive Bayes (a word can appear multiple times)
  - Document is generated from a single topic
- Model on right (LDA) is an **admixture model**
  - Document is generated from a distribution over topics

Two steps

- Can typically separate out these two uses of topic models:
  1. *Learn* the model parameters (α, β)
  2. Use model to make *inferences* about a single document
- Step 1 is when topic discovery happens. Since the topic assignments z are never observed, one can use EM to do this
- Exact inference is intractable: approximate inference (typically Gibbs sampling) is used
- Another common approach is to put a prior distribution over β and to do MAP inference over β and z, in which case the whole learning algorithm can be performed with Gibbs sampling