Latent Dirichlet Allocation

Jérôme DOCKÈS (jerome@dockes.org), Pascal LU (pascal.lu@centraliens.net)

École Normale Supérieure de Cachan — January 5, 2016

Objectives

We consider the problem of modeling text corpora. The goal is to find short descriptions of the members of a collection. Our work is mainly based on:

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

Notations

- $\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$ is a corpus.
- $\mathcal{V}$ is the vocabulary of size V.
- k is the number of topics.

For a document $d \in \mathcal{D}$,

- $d = (w_1^{(d)}, \ldots, w_{N_d}^{(d)})$ represents the document d. N_d is the number of words in the document d.
- $w^{(d)}$ is a matrix whose element $w_{ni}^{(d)} = 1$ if the word n in the document is the word i in the vocabulary. Size of $w^{(d)} = N_d \times V$.
- $\theta^{(d)}$ represents a probability mass function over the topics. Size of $\theta^{(d)} = k$.
- $z^{(d)}$ is the set of topics : $z_{ni}^{(d)} = 1$ if the word n was generated by the topic i. Size of $z^{(d)} = N_d \times k$.

Presentation of the model

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus.

- Documents = random mixtures over latent topics,
- Topic = a distribution over words.

Output: corpus $\mathcal{D}$

1: **for** each document $d \in \mathcal{D}$ **do**
2: Choose the length $N_d \sim \text{Poisson}(\xi)$
3: Choose $\theta^{(d)} \sim \text{Dir}(\alpha)$
4: **for** each of the N_d words $w_n^{(d)}$ **do**
5: Choose a topic $z_n^{(d)} \sim \text{Multinomial}(\theta^{(d)})$
6: Choose a word $w_n^{(d)}$ from $p(w_n^{(d)}|z_n^{(d)}; \beta)$, a multinomial probability conditioned on $z_n^{(d)}$.
7: **end for**
8: **end for**

We have:

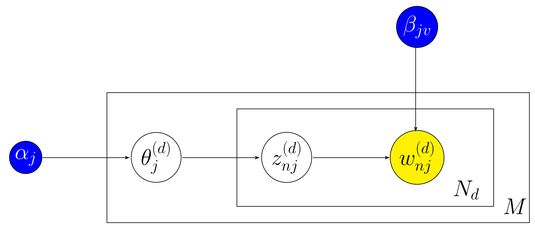
$$p(d|\alpha,\beta) = \int p(\theta^{(d)}|\alpha) \left(\prod_{n=1}^{N_d} p(w_n^{(d)}|\theta^{(d)},\beta) \right) \mathrm{d}\theta$$

$$= \int p(\theta^{(d)}|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_n^{(d)}} p(z_n^{(d)}|\theta^{(d)}) p(w_n^{(d)}|z_n^{(d)},\beta) \right) \mathrm{d}\theta$$

Generative model

The goal is to determine:

- α is an estimate of the parameter of the Dirichlet distribution which generates the parameter for the (multinomial) probability distribution over topics in the document. Size of $\alpha = k$.
  Exchangeable Dirichlet distribution assumption:
  $$\forall i \in \{1, \ldots, k\}, \quad \alpha_i = \alpha$$
- β is a matrix of size $k \times V$ which gives the estimated probability that a given topic will generate a certain word: $\beta_{ij} = p(w^j = 1 | z^i = 1)$.



Variational inference

We need to compute the posterior distribution of the hidden variables given a document d:

$$p(\theta^{(d)}, z^{(d)}|d,\alpha,\beta) = \frac{p(\theta^{(d)}, z^{(d)}, d|\alpha,\beta)}{p(d|\alpha,\beta)}$$

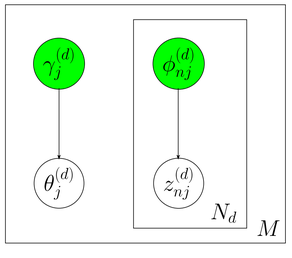
For a document $d \in \mathcal{D}$:

- $\gamma^{(d)}$ is the variational parameter for the Dirichlet distribution. Size of $\gamma^{(d)} = k$.
- $\phi^{(d)}$ is the variational parameter for the multinomial distribution. Size of $\phi^{(d)} = N_d \times k$.
  $\phi_{ni}^{(d)}$ depends on the relation between the word in position n of the document and the topic i of the list of topics.

$$q(\theta^{(d)}, z^{(d)}|\gamma^{(d)},\phi^{(d)}) = q(\theta^{(d)}|\gamma^{(d)}) \prod_{n=1}^{N_d} q(z_n^{(d)}|\phi_n^{(d)})$$

$\Rightarrow$ Estimate $\gamma^{(d)}, \phi_n^{(d)}$ instead of $\theta^{(d)}$ and $z_n^{(d)}$.

$$(\gamma^{(d)}, \phi^{(d)}) = \underset{(\gamma,\phi)}{\text{argmin}}\, \mathrm{D}\left(q(\theta^{(d)}, z^{(d)}|\gamma,\phi) \,\middle\|\, p(\theta^{(d)}, z^{(d)}|d,\alpha,\beta) \right)$$



Variational Inference Procedure (E-step for a document d)

Input: a document d defined by its $w^{(d)}$, α, β
Output: $\gamma^{(d)}$, $\phi^{(d)}$

1: Initialize $\phi_{ni}^{(d)} = \frac{1}{k}$ for all i and n.
2: Initialize $\gamma_i^{(d)} = \alpha + \frac{1}{k}\sum_{n=1}^{N_d} w_n^{(d)}$ for all i.
3: **while** the expected log-likelihood $L(\gamma^{(d)}, \phi^{(d)}, \alpha, \beta)$ for the document d has not converged **do**
4: **for** $n = 1 \ldots N_d$ **do**
5: **for** $i = 1 \ldots k$ **do**
6: $\phi_{ni}^{(d)} = \beta_{iw_n^{(d)}} \exp(\Psi(\gamma_i^{(d)}))$
7: **end for**
8: Normalize $\phi_n^{(d)}$ to sum to 1.
9: **end for**
10: $\gamma^{(d)} = \alpha + \sum_{n=1}^{N_d} w_n^{(d)} \phi_n^{(d)}$
11: Update $L(\gamma^{(d)}, \phi^{(d)}, \alpha, \beta)$
12: **end while**

EM-algorithm

Input: corpus $\mathcal{D}$, number of topics k
Output: α, β

1: **for** each $d \in \mathcal{D}$ **do**
2: Compute $w^{(d)}$.
3: **end for**
4: Initialize α and β.
5: **while** the expected log-likelihood $L(\alpha, \beta)$ has not converged **do**
6: Initialize $\Sigma_\gamma = 0$, $\beta_{\text{new}} = 0$ and $L(\alpha, \beta) = 0$.
7: **for** each $d \in \mathcal{D}$ **do**
8: $(\gamma^{(d)}, \phi^{(d)}) =$ **variational-inference**$(w^{(d)}, \alpha, \beta)$
9: $\beta_{\text{new}} \leftarrow \beta_{\text{new}} + (\phi^{(d)})^\top w^{(d)}$
10: $\Sigma_\gamma \leftarrow \Sigma_\gamma + \sum_{i=1}^{k} \Psi(\gamma_i^{(d)}) - \Psi\left(\sum_{j=1}^{k} \gamma_j^{(d)}\right)$
11: $L(\alpha, \beta) \leftarrow L(\alpha, \beta) + L(\gamma^{(d)}, \phi^{(d)}, \alpha, \beta)$
12: **end for**
13: Normalize β_{new} and set $\beta = \beta_{\text{new}}$
14: **while** α has not converged **do**
15: $\frac{\partial L}{\partial \alpha}(\alpha) = |\mathcal{D}|k\left[\Psi(k\alpha) - \Psi(\alpha)\right] + \Sigma_\gamma$
16: $\frac{\partial^2 L}{\partial \alpha^2}(\alpha) = |\mathcal{D}|k[k\Psi'(k\alpha) - \Psi'(\alpha)]$
17: $\alpha \leftarrow \alpha - \frac{\frac{\partial L}{\partial \alpha}(\alpha)}{\frac{\partial^2 L}{\partial \alpha^2}(\alpha)}$
18: **end while**
19: **end while**

Implementation issues

- **Document and vocabulary preprocessing**.
- **Initialization**: $\alpha > 0$ and β were initialized randomly.
- **Representation of** $w^{(d)}$: N_d is the number of **distinct** words in the document d and $w^{(d)}$ is a dictionary (with N_d entries) containing for each distinct word (indexed by its position in the vocabulary) the number of times it appears in the document.

Results

We tested our LDA on real data from the Reuters21578 database (`reut2-000.sgm`).

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
devices	prolonged	zestril	seasons	withdrawn
disk	council	annesthetic	hotels	expiration
megabyte	forum	hypertension	VMS	clearances
expandable	dissident	oth	Biltmore	expire
megabytes	flying	statil	Marriott	Willemijn
equipped	sparks	diabetic	rename	BV
monochrome	talks	complications	hotel	Rotterdam
peripheral	outweighed	Barbara	228	licensed
color	accomplishments	definitive	DH	NCR

Table 1: Results for 5 topics ($k = 20$)

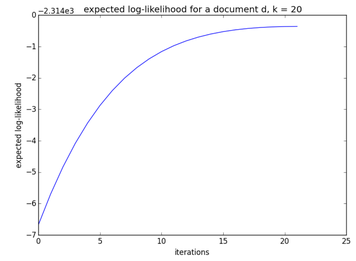


Figure 1: Expected log-likelihood for a document ($k = 20$)

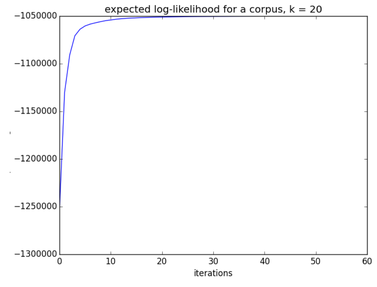


Figure 2: Expected log-likelihood for the corpus ($k = 20$)

Conclusion

- LDA = a way to apply graphical models to information retrieval = group words in the same categories.
- Key idea = variational inference.
- LDA is a non-convex optimization problem: estimates for α, β depend on their initialization.
- Other interesting applications: biology (DNA sequence), content-based image retrieval...