Summary and discussion of: "Latent Dirichlet Allocation"

MATH 5472 project report

Zihao Wang @ IEDA, HKUST

December 10, 2021

1 Summary and discussion of the Paper

1.1 Background and Context

Latent Dirichlet Allocation (LDA) is a celebrating success of empirical Bayes on topic modeling. In the context of topic modeling, a word is denoted by a vector $w \in \{0,1\}^V$, where V is the size of the vocabulary. A document $\boldsymbol{w}$ is a sequence of word $(w_1, \ldots, w_N)$. A corpus D is a collection of M documents $(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M)$. Each document in the corpus is labelled by a topic or a mixture of topics, which is unobservable. The problem of topic modeling is to infer the topic information. This can also be viewed as a dimensionality reduction problem: how to compress a corpus while preserving essential statsitical information. A natural solution is to apply Non-negative Matrix Factorization (NMF). However, the sparsity of frequency table of words make this optimization problem hard to converge. On the other hand, NMF is a discriminative model, lacking the ability to model polysemy. For example, a single word may have multiple meanings, which are related to different topics. Hence, generative model is preferred when modeling topics. LDA not only achieves efficient dimensionality reduction, but is a probabilistic generative model for a corpus. Motivated from topic modeling problem, it also has broad application to other collection modeling problem, including collaborative filtering and bioinformatics.

1.2 Review of Other Generative Models

Before introducing the basic idea of LDA, we briefly review three other simple generative models. This is helpful for us to understand the drawback of these models and how LDA is motivated. For all theses models, we model the number of words in each document N as Poisson(ξ), which is independent of all the other random variables.

Unigram model. Unigram is a *word*-level generative model. We directly sample words in a document from a specified multinomial distribution β.

$$p(\boldsymbol{w}|\beta) = \prod_{n=1}^{N} p(w_n|\beta). \tag{1}$$

There are only $V-1$ parameters for you to specify multinomial distribution β. Obviously, this model can not be used for topic modeling.

Mixture of unigrams model. To incorporate topic information in the generative model, we have to model the *topic*-level generation process. A straightforward way is to consider the topic of each document z is sampled from a multinomial distribution $p(z)$ Then we generate words conditional on this topic from a multinomial distribution $\{\beta\}_{k,V}$.

$$p(\boldsymbol{w}|\beta) = \sum_z p(z) \prod_{n=1}^{N} p(w_n|z, \beta). \tag{2}$$

There are $(k-1)+k(V-1)$ parameters in this model. However, the drawback of the model lies on the assumption that each document is labelled by single topic. Empirical studies show that document is usually labelled by a mixture of topics.

Probabilistic latent semantic indexing. To overcome the limitation of single topic modeling, probabilistic latent semantic indexing (pLSI) endows a mixture weights of topics θ to each document. Instead of generating all the words from a single topic, pLSI allows the words w_n conditionally i.i.d sampled according to the topic mixtures θ.

$$p(\boldsymbol{w}|\theta, \beta) = \prod_{n=1}^{N} \sum_{z_n} p(w_n|z_n, \beta) p(z_n|\theta), \tag{3}$$

$$p(D|\theta, \beta) = \prod_{d=1}^{M} \prod_{n=1}^{N_d} \sum_{z_{dn}} p(w_{dn}|z_{dn}, \beta) p(z_{dn}|\theta_d). \tag{4}$$

Since each document is parametrized by a mixture weights, there are $M(k-1)+k(V-1)$ parameters in pLSI. This reveals that the number of parameters grows linearly with the number of documents in the corpus. Accompanying drawback is that training pLSI is prone to overfitting. Further, this generation model stops at *topic*-level generation, and there is no *document*-level generation.

1.3 Basic Ideas for LDA

That's one small step of empirical Bayes, one big step for topic modeling.

In pLSI, the mixture weights θ of all the document are required to be estimated. Can the documents "borrow information from each other"? Indeed, empirical Bayes approach provide a principle way to tackle the problem.

In an empirical Bayes' view, it is quite natural to put another prior distribution on the mixture weights θ and then optimize the parameters in the prior. In particular, the original paper adopt Dirichlet distribution $\mathrm{Dir}(\alpha)$, because it is conjugate to multinomial distribution of θ. Now we are ready to write down the marginal distribution of a document and a corpus:

$$p(\boldsymbol{w}|\alpha, \beta) = \int p(\theta|\alpha) \prod_{n=1}^{N} \sum_{z_n} p(w_n|z_n, \beta) p(z_n|\theta) d\theta, \tag{5}$$

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \prod_{n=1}^{N_d} \sum_{z_{dn}} p(w_{dn}|z_{dn}, \beta) p(z_{dn}|\theta_d) d\theta_d. \tag{6}$$

The number of parameters to be estimated reduce to $k + k(V-1)$. Compared with pLSI, the number of parameters keeps constant with the number of documents. Thus, training LDA is free of overfitting. Furthermore, LDA indeed capture how to generate a document in the corpus.

Remark 1. Although LDA and its extensions have made great success on modeling collections of discrete data, the basic idea, as we can see, is so simple. As long as we keep the empirical Bayes principle in the mind, we can derive LDA based on pLSI easily and get rid of the cumbersome *exchangeability assumption*, *graphical* and *geometrical* interpretation in the original paper. For completeness, I also show the graphical representation as follows:

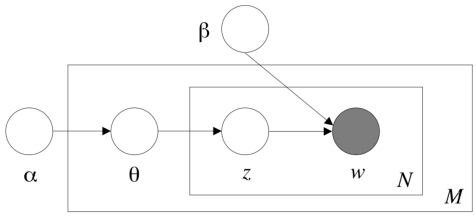

Figure 1: Graphic representation of generative model. Source: LDA. [1]

1.4 Statistical Inference and Parameter Estimation

Before proceeding, we plug in the expression of Dirichlet distribution and multinomial distribution to the LDA model. Recall that z_n is an indicator vector with $\beta_{i,j} = p(w_n^j = 1|z_n^i = 1)$, the marginal distribution of a document can thus be simplified as:

$$p(\boldsymbol{w}|\alpha,\beta) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma\left(\alpha_i\right)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} \left(\theta_i \beta_{ij}\right)^{w_n^j}\right) d\theta \tag{7}$$

Unfortunately, due to the coupling of θ and β, this integration on θ is intractable, cumbering a closed-form of $p(\boldsymbol{w}|\alpha,\beta)$.

1.4.1 Inference of hidden variables

In empirical Bayes, we are always interested in the posterior distribution of the hidden variables. LDA has a three-level hierarchical structure. The mixture weights θ and topic of words $\boldsymbol{z}$ are hidden while only words $\boldsymbol{w}$ are observable. Write down the posterior:

$$p(\theta, \boldsymbol{z}|\boldsymbol{w},\alpha,\beta) = \frac{p(\theta,\boldsymbol{z},\boldsymbol{w}|\alpha,\beta)}{p(\boldsymbol{w}|\alpha,\beta)}. \tag{8}$$

One have to develop approximation approach to evaluate it due to the intractability of $p(\boldsymbol{w}|\alpha,\beta)$. In the original paper, the authors adopt variational inference.

We use mean-field to approximate $p(\theta, \boldsymbol{z}|\boldsymbol{w},\alpha,\beta)$ by

$$q(\theta, \boldsymbol{z}|\gamma,\phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{9}$$

Remark 2. We can also first factorize $p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = p(\theta|\boldsymbol{w}, \alpha)p(\boldsymbol{z}|\boldsymbol{w}, \theta)$. Then (9) is equivalent to use $q(\theta|\gamma)$ and $q(z|\phi)$ to approximate $p(\theta|\boldsymbol{w}, \alpha)$ and $p(\boldsymbol{z}|\boldsymbol{w}, \theta)$ respectively.

Here we omit the tedious calculation details but point out that, after mean-field approximation, $q(\theta|\gamma)$ is actually Dirichlet distribution with parameter γ and $q(z_n|\phi_n)$ is actually multinomial distribution with parameter ϕ_n. Maximizing the evidence lower bound, or minimizing the KL divergence between $p(\theta, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta)$ and $q(\theta, \boldsymbol{z}|\gamma, \phi)$ equivalently gives the following iteration rule:

$$\begin{aligned} \phi_{ni} &\propto \beta_{iw_n}(\Psi(\gamma_i) - \Psi(\sum_{j=1}^{k}\gamma_j), \\ \gamma_i &= \alpha_i + \sum_{n=1}^{N}\phi_{ni}, \end{aligned} \tag{10}$$

where Ψ is the digamma function.

Variational procedure also helps to build a lower bound of the log-likelihood. In particular, for each document $\boldsymbol{w}$,

$$\begin{aligned} \log p(\boldsymbol{w}|\alpha, \beta) &\geq L(\gamma, \phi; \alpha, \beta) \\ &= \log\Gamma\left(\sum_{j=1}^{k}\alpha_j\right) - \sum_{i=1}^{k}\log\Gamma\left(\alpha_i\right) + \sum_{i=1}^{k}\left(\alpha_i - 1\right)\left(\Psi\left(\gamma_i\right) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right)\right) \\ &+ \sum_{n=1}^{N}\sum_{i=1}^{k}\phi_{ni}\left(\Psi\left(\gamma_i\right) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right)\right) \\ &+ \sum_{n=1}^{N}\sum_{i=1}^{k}\sum_{j=1}^{V}\phi_{ni}w_n^j\log\beta_{ij} \\ &- \log\Gamma\left(\sum_{j=1}^{k}\gamma_j\right) + \sum_{i=1}^{k}\log\Gamma\left(\gamma_i\right) - \sum_{i=1}^{k}\left(\gamma_i - 1\right)\left(\Psi\left(\gamma_i\right) - \Psi\left(\sum_{j=1}^{k}\gamma_j\right)\right) \\ &- \sum_{n=1}^{N}\sum_{i=1}^{k}\phi_{ni}\log\phi_{ni}, \end{aligned} \tag{11}$$

Here we omit the calculation details and present the result for reference.

1.4.2 Parameter estimation

Following the same spirit as EM, we maximize the evidence lower bound $\sum_{d=1}^{M} L_d(\gamma_d, \phi_d; \alpha, \beta)$ with respect to α and β to do one step updating. We update α and β until the evidence lower bound converges. We also omit the calculation details and present the updating rule for coding implementation reference.

At each step:

$$\beta_{i,j} \propto \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}w_{dn}^j \tag{12}$$

Run Newton-Raphson algorithm for α with gradient and Hessian:

$$\begin{aligned}
\frac{\partial L}{\partial \alpha_i} &= M\left(\Psi\left(\sum_{j=1}^{k}\alpha_j\right) - \Psi\left(\alpha_i\right)\right) + \sum_{d=1}^{M}\left(\Psi\left(\gamma_{di}\right) - \Psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right), \\
\frac{\partial L}{\partial \alpha_i \alpha_j} &= \delta(i,j) M \Psi'\left(\alpha_i\right) - \Psi'\left(\sum_{j=1}^{k}\alpha_j\right).
\end{aligned} \tag{13}$$

1.5 Extensions

In the original paper, the authors also mention some possible extensions of LDA.

Smoothing. Recall that $\beta_{i,j} = p(w_n^j = 1 | z_n^i = 1)$. If there exists a certain word j which does not appear in the training set, then standard LDA parameter estimation yields $\hat{\beta}_{i,j} = 0$ for all i, which is not desirable. To "smooth" β, one can put another prior on β. The authors adopt i.i.d. exchangeable Dirichlet distribution with parameter η on each row β_i. Then it remains to use variational inference EM to estimate α and η. This approach can be considered as a fuller Bayesian model compared to standard LDA.

Partial exchangeability. Although in this report we didn't explicitly mention exchangeability, LDA indeed relies on the assumption that the topic variables in a document are infinitely exchangeable. Mathematically, this means the joint distribution of every subsequence of random variables $(z_1, \cdots, z_n)$ is invariant to permutation π,

$$p(z_1, \ldots, z_n) = p(z_{\pi(1)}, \ldots, z_{\pi(n)}). \tag{14}$$

This assumption can be relaxed to partial exchangeability, which allows us to characterize the topics in a time series or dependent on the different part structure of a document, such as paragraph.

Continuous data modeling. LDA is designed for modeling collections of discrete data. Conditional on a topic variable z, word is sampled from a multinomial distribution $p(w|z$. Allowing this distribution to be continuous, we obtain a LDA model for continuous data.

Although the authors mention many extensions of LDA, they are all falling under the broad umbrella of empirical Bayes. The differences between these extensions only concern how to choose latent variable and which kind of prior distribution you impose.

2 Simulations and Results

In this section, we show some simulation result of LDA. In the original paper, the authors model the TREC AP corpus and apply LDA to document modeling, document classification and collaborative filtering. However, these simulation tasks are too timely and computationally demanding to reproduce. To deliver the key feature of LDA, here we only present two simpler illustrative examples. But interested readers can also use my code to deal with larger scale problem.

2.1 Simulation data

To make this simulation task not so boring, I borrow the idea from others and adopt the following setting:

- $M = 300$, $k = 10$, $V = 30$, N_d uniformly sampled in $[150, 200]$
- Documents come from *three different copra*, i.e., sampled from $\text{Dir}(\alpha_1)$, $\text{Dir}(\alpha_2)$ and $\text{Dir}(\alpha_3)$.
  - $\alpha_1 = [20, 15, 10, 1, \ldots, 1]$
  - $\alpha_2 = [1, 1, 1, 10, 15, 20, 1, \ldots, 1]$
  - $\alpha_3 = [1, \ldots, 1, 10, 12, 15, 18]$
- β is chosen such that for each topic, there are 3 common words. Here is the heat-map of β matrix:

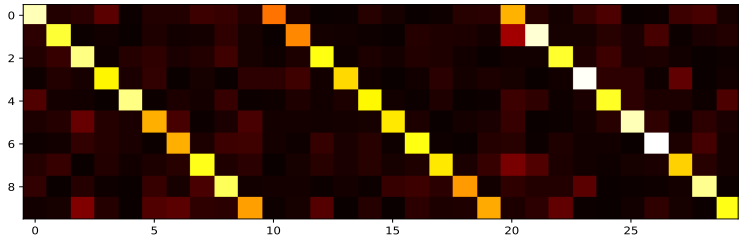

Figure 2: Heatmatp of true β

This is motivated by thinking what is a "corpus" after all. In reality, the corpus is just a messy collection of documents while these documents may be clustered as different groups. In this situation, is LDA still robust and powerful? My simulation shows that LDA can still capture the pattern of the corpus. After 100 step training, we can obtain estimated β. By artificially re-arranging rows, we can plot the following heat-map, which indeed capture the essential pattern of true β.

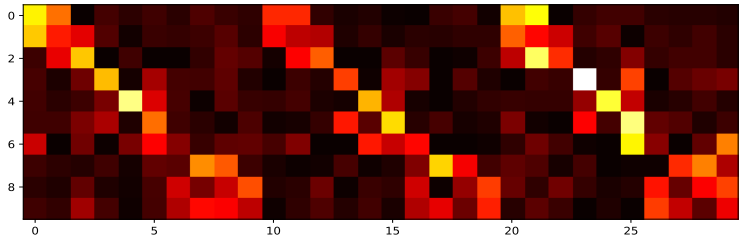

Figure 3: Heatmatp of estimated β

2.2 Real document topic modeling

We examine the effeteness of LDA topic modeling on real data. We choose a subset of the Associated Press corpus downloaded from Professor Blei's Lab https://github.com/blei-lab/lda-c/tree/master/example. As an illustrative example, we choose $M = 200$, $k = 7$. Vocabulary list with $V = 1072$ is random sampled from the file provided by Blei's lab. Here we list 7 top words for each topic by LDA.

1	new	year	union	president	last	state	officials
2	party	year	predident	goverment	political	monday	two
3	police	people	state	man	two	city	arrested
4	new	year	last	billion	government	people	gold
5	prices	trade	market	dollar	new	cents	late
6	soviet	president	union	people	two	committee	government
7	central	president	people	rating	new	year	officials

Table 1: Top words of 7 topics

From above Table, one can find the pattern within each topic. Human can further infer the topic according to these top words. For example, topic 3 may be related with politics or criminal events because "police", "state", "arrested" are top words; topic 5 is very likely to be related with financal market, because the top words include "prices", "trade", "market" and so on. Moreover, within one topic, there is no too much conflict between these top words. For example, "soviet" in topic 6 and "market" in topic 5 rarely appear in the same article together.

Note added 1. — There is a typo of the linear time Newton-Raphson iteration formula in the appendix. Interested readers can refer the correct iteration formula in my code.

Note added 2. — All the contents in this report is written based on my personal understanding after reading the original paper. The code is available online:
https://github.com/zihaophys/latent_dirichlet_allocation. You can reproduce all the results in this report.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.