Latent Dirichlet Analysis for Document Topic Modelling

Xiang Li
xngli@umich.edu

Zheng Luo
zluo@umich.edu

Yan Chen
yanchenm@umich.edu

Sajan Patel
sajanptl@umich.edu

Abstract

Latent Dirichlet Allocation (LDA) is a generative probabilistic model for collections of discrete data and text corpora proposed by Blei et al. in their 2003 *Journal of Machine Learning Research* paper. A common application of LDA is the unsupervised learning of document topic models. In this paper, we explore the LDA model according to Blei et al., 2003, and implement it using Gibbs sampling and variational inference methods. We evaluate our implementation using the topic modeling experiment done by Blei et al., 2003, applying it to the TREC AP corpus. We also compare LDA to the simple unigram topic model and discuss the subsequent conclusions gained from our experimentation.

1 Introduction

The problem of modeling text documents and extracting the topics within them given unsupervised training data is document topic modeling. Each topic is a short description of the members of a collection of documents it represents. Applications of this problem can range from extracting topics from lengthy journals and news articles to short social media posts or business reviews.

Latent Dirichlet Allocation (LDA) is a generative model proposed by Blei et al., 2003 [4], that can be used to solve this problem. It is a three-level hierarchical Bayesian model in which document is modeled as a mixture of underlying topics, and each topic is modeled by underlying probabilities of topics. LDA treats topics as the latent variables of the document model that are learned throughout the training process. There are other applications of LDA as well, such as in document classification and collaborative filtering; however, in this paper, we focus on the application of LDA to topic models.

We implement the model using both the variational inference method proposed in Blei et. al, 2003, as well as using the collapsed Gibbs Sampling method of Griffiths and Steyvers, 2004 [1]. Both methods offer a different perspective on how to train the LDA model using unsupervised training data, and each method has its advantages and disadvantages in terms of performance and implementation. To evaluate each method, we adapt the topic modeling experiment Blei et al., 2003, to compare the convergence of the training phase for each method using the TREC AP corpus [6] as our unlabeled training and testing dataset. We present our findings from this experiment and the subsequent conclusions gained from them.

1.1 Related Works

It is important to note that LDA is not the only probabilistic document model that exists. Other works in topic and document modeling using latent variables have been done using the unigram model,

mixture of unigrams, and probabilistic latent semantic indexing (pLSI). In the unigram model, words in each document are independently generated from a single multinomial probability distribution for the entire vocabulary of words. This model is highly limited in that each word, essentially, is its own topic.

In the mixture of unigrams model proposed by Nigam et al., 2000 [7], documents are generated by first choosing a topic from a distribution of all topics and then independently generating the words from a distribution of the vocabulary conditioned on the chosen topic. This model also highly overfits to the training set. Each document is limited to being generated from one topic and overfitting to the training document set is a major issue.

The pLSI model proposed by Hofmann, 1999 [8], attempts to relax the assumption in the mixture of unigrams to allow documents to be generated from multiple topics with varying probabilities. However, it is not a well-defined model since it there is no intuitive way to assign probabilities to previously unseen documents in the testing set and it greatly overfits the model to the training set due to conditioning the model based only on the training set.

LDA overcomes both the problems these three other models encounter with overfitting and limited document models. It allows for documents to be generated from distribution of topics with varying probabilities from which the words are then generated. There is more flexibility in learning the hidden or latent topics, and the model generalizes better to new documents.

2 Notation

We used the same notations used by topic modeling literature throughout this paper as shown in Table 1.

Table 1: Table of Notation

K	=	number of topics
D	=	number of documents
V	=	vocabulary size
N	=	total number of words in the text corpus
N_d	=	number of words in document d
d_i	=	i-th document
z_j	=	j-th topic
w_i	=	i-th word in the document
α, β	=	hyperparameters that defines the Dirichlet prior
C_{dk}	=	number of type v which are assigned with topic k
C_{vk}	=	number of topics k in document d
I_{di}	=	sampling times of w_{di} in one Gibbs Sampling iteration
S^t_{di}	=	sampling rate of type w_{di} in iteration t
Γ_{di}	=	average sampling rate in iteration t
S^t_{di}	=	parameter vector of the multinomial distribution ¶$(I_{di}\|\Gamma_{di})$, it has $N_d i$ entries: [Γ_{di1}, ..., $\Gamma_{diN_{di}}$]
α, β	=	Dirichlet priors
γ	=	dumping factor

3 Latent Dirichlet Allocation

The latent Dirichlet allocation (LDA) model explains the generation of text documents, which can be viewed as samples from a mixture of multinomial distributions over a vocabulary of words. Each multinomial mixture component is called a "topic". The general process of the model to write a document is the following:

1. The number of words N in document $\sim$ Poisson (ζ)
2. The topic mixture θ for the document $\sim$ Dir (α) (with a fixed set of K topics)

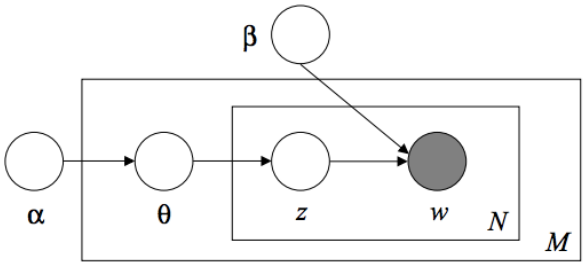

Figure 1: Graphic model of LDA. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

3. Then for each word w_i in the document:
   - Choose a topic t_i based on multinomial distribution with parameter θ from step (2)
   - Use this topic z_i to generate word itself by using the existing probability for each word in this topic (i.e. $p(w_i|z_i, \beta)$)

This is the generative model for a collection of documents. LDA then tries to backtrack from the (training) documents to find a set of topics that are likely to have generated the collection. Now the question is how does LDA backtrack to find the parameters in this model. Suppose we have a set of documents D, and we set the number of topics to be K. What we want is to use LDA to learn two things 1) the topic representation of each document and 2) the words associated to each topic.

There are two commonly used methods to learn the latent parameters for a text corpus: collapsed Gibbs sampling [1] and variational inference [2, 3]. We have implemented both methods in our paper.

3.1 Collapsed Gibbs Sampling

- For each document, randomly assign each word in the document to one of the K topics.
  1. this step gives topic representation of all the documents
  2. this step gives word distributions of all the topics
  3. since randomly assign topics to each word is very native, so we need to improve it
- For each word w_k in document d_i
  1. For each topic t_j that this word belongs to, compute:

  (a) $p(z_j|d_i) = \frac{\text{number of words assigned to } z_j \text{ in } d_i}{\text{total number of words in } d_i}$

  (b) $p(w_k|z_j) = \frac{\text{number of words assigned to} z_j \text{ in } d_i}{\text{number of words assigned to } z_j \text{for all docs}}$
- we compute the product of i) and ii) above which gives the new topics to assign to this word.
- repeating step 2 over and over until it reaches a steady state where the assignments make good sense.
- Use this model to estimate the topic mixtures of each document and words associated to each topic, which are the two things we want to learn.

3.2 Inference Method

With the same Dirichlet distribution model assumption shown in the previous section, variational parameters (γ and ϕ) are introduced. The way to do this is to place a distribution q over hidden variables (θ and Z) with free parameters which are the so called variational parameters. In order to make the problem intractable, Jensen's inequality is applied to yield an evidence lower bound for the log-likelihood. Then, an optimization process can be performed to make the placed distribution

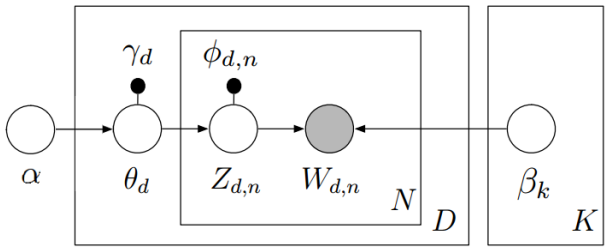

Figure 2: Graphical model representation of the variational distribution used to approximate the posterior in LDA.

Estimation
Initializatize: $\phi_{ni}^0 = 1/K$ for all i and n
Initializatize: $\gamma_i^0 = \alpha_i + N/K$ for all i
repeat
for $n = 1$ to N **do**
  for $i = 1$ to K **do**
    $\phi_{ni}^{t+1} = \beta_{iw_n} \exp\{\varphi(\gamma_i^t)$
    $-\varphi(\sum_{j=1}^K \gamma_j^t)\}$
  end forend
end forend
$\gamma^{t+1} := \alpha + \sum_{n=1}^N \phi_n^{t+1}$
$t \leftarrow t + 1$
until convergence

(a) E-step

Maximization
Update β
for $i = 1$ to K **do**
  for $j = 1$ to V **do**
    $\beta_{ij} \propto \sum_{d=1}^D \sum_{n=1}^{N_d} \phi_{dni} w_{dn}^j$
  end forend
end forend
Update α
repeat
Caculate gradient $g(\alpha^t)$ and Hessian $H(\alpha^t)$
$\alpha^{t+1} = \alpha^t - H^{-1}(\alpha^t) g(\alpha^t)$
$t \leftarrow t + 1$
until convergence

(b) M-step

Alg. 2: EM algorithm

close to the posterior in KL divergence:

$$D(q(\theta, Z|\gamma, \phi)||p(\theta, Z|W, \alpha, \beta)) = \log p(W|\alpha, \beta) - L(\gamma, \phi; \alpha, \beta) \tag{1}$$

where

$$L(\gamma, \phi; \alpha, \beta) = E_q[\log p(\theta, Z, W|\alpha, \beta)] - E_q[\log q(\theta, Z)] \tag{2}$$

Minimizing the KL divergence is equivalent to maximizing the evidence lower bound [5]. The posterior distribution can be obtained with those variational parameters using EM algorithm.

When specifying the initial values of α and β, a symmetric Dirichlet distribution is assumed. In the maximization step, the gradient $g(\alpha^t)$ and Hessian $H(\alpha^t)$ are calculated using the evidence lower bound:

$$\frac{\partial L}{\partial \alpha_i} = D(\Psi(\sum_{j=1}^K \alpha_j) - \Psi(\alpha_i)) + \sum_{d=1}^D (\Psi(\gamma_{di}) - \Psi(\sum_{j=1}^K \gamma_{dj})) \tag{3}$$

$$\frac{\partial L}{\partial \alpha_i \alpha_j} = D\Psi'(\sum_{j=1}^K \alpha_j) - \delta(i, j) D\Psi'(\alpha_i) \tag{4}$$

When applying LDA to training documents, the estimation and maximization steps are iteratively applied until the convergence of the evidence lower bound (or equivalently, until the convergence of perplexity $e^{-L/N}$). In addition, Laplace smoothing is used when updating the value of β to avoid zero entries.

4 Evaluations and Empirical Results

4.1 Dataset and Preprocessing

To evaluate our algorithm, we replicated one of the experiments implemented by Blei et al., 2003, on document modeling. We used AP dataset which contains 2246 news articles from the Associated Press [6]. The dataset was partitioned into 80 percent training set and 20 percent test data. In preprocessing of the data, we removed a list of stop words from the text. We further removed words that appear less than 20 times (rare words), as well as words that appear in more than 90 percent of the documents (common words, including articles and common adjectives). We ended up with a vocabulary size of 3564 words.

4.2 Convergence of Algorithms

In AP dataset, each document is unlabeled. Therefore, here we are doing unsupervised learning with a purpose to estimate the likelihood of the test dataset. In natural language processing, the likelihood of a document is usually called *perplexity*, which is the inverse of the average per-word log likelihood. The perplexity of on the text corpus is defined by

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p(w_d)}{\sum_{d=1}^{M} N_d}\right\} \tag{5}$$

$$= \exp\left\{-\frac{1}{N}\sum_{d=1}^{D}\sum_{i=1}^{N_d} N_{di}\log\sum_{k=1}^{K}\theta_{w_{di},k}\phi_{d,k}\right\} \tag{6}$$

In Gibbs sampling LDA, $\theta_{v,k}$ and $\phi_{d,k}$ can be estimated by

$$\theta_{v,k} = \frac{C_{vk}+\beta}{\sum_{v'} C_{v'k}+\beta}, \phi_{d,k} = \frac{C_{dk+\alpha}}{\sum_{k'} C_{dk'}+K\alpha}$$

While implementing our algorithms, we first monitors the convergence of the algorithm. This was done by monitoring the perplexity of training data, as shown Figure 3. We can see that in both cases the algorithm converges after enough number of iterations, although the numbers of iterations required to reach convergence are very different. Variational inference LDA usually converges after about 10 iteration, while Gibbs sampling LDA needs several hundred of iterations. This is a primary reason that the variational inference algorithm is much faster and more commonly used in practice.

4.3 Comparison to Unigram Model

The unigram model represents the words in each document with the following distribution:

$$p(w_i) = \prod_{j=1}^{N} p(w_j) \tag{7}$$

The counts of each word occurrence in each documents are summed up and then normalized across all the counts seen in the training data. These normalized counts then become the probabilities for each word in the vocabulary after applying smoothing.

We compared the perplexities of both implementations of LDA to the simple unigram model. Since perplexity is inversely proportional to the log-likelihood of the word distribution, lower perplexities indicate better performance. Perplexity was calculated in the same manner as in Equation **??**. Note that unlike LDA, the unigram model is trained in only one iteration. The results of the final testing and training perplexities for the unigram, LDA with Gibbs Sampling, and LDA with Variational Inference models are shown in Table **??**. As seen in the table, both LDA methods have much lower perplexities in for both the training and testing datasets. All models exhibit a higher perplexity on the testing set than the training set. The perplexity increase for the variational inference method of LDA is the lowest because it has less overfitting to the training set than the Gibbs Sampling method.

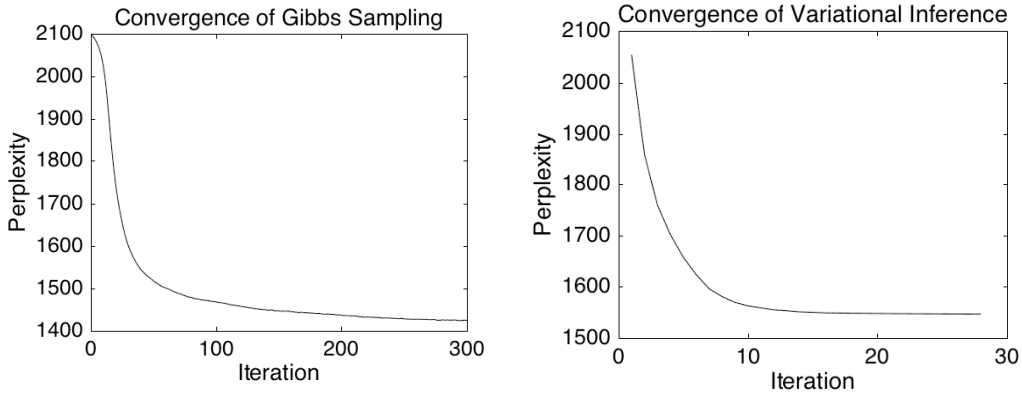

Figure 3: Convergence of LDA using Gibbs Sampling (*left*) and Variational Inference (*right*)

Table 3: Training and Testing Perplexities

METHOD	**TRAINING SET**	**TESTING SET**
Unigram	5315	5839
LDA with Gibbs Sampling	1427	2537
LDA with Variational Inference	1576	1880

High overfitting to the training set and the lack of incorporating latent topic variables in the topic model are why the simple unigram model has such a high perplexity. LDA, in both cases, overcomes both of these issues, and these results confirm the claim by Blei et al., 2003, that LDA outperforms the unigram model.

4.4 Extracted Subtopics

We trained a 10-topic LDA model on the training dataset using both Gibbs sampling and variational inference, and 5 of the generated subtopics in shown in Figure 4. For each topic, the 10 words that are most likely to occur is listed. As we expected, LDA automatically generated meaning full subtopics from the AP corpus, with similar words more likely to appear in the same topic. We also note that the choice and order of words in the topic generated by Gibbs sampling and variational inference are different, although the topics they represented are similar. We think this discrepancy is because the two algorithms we used might converge into different local opitma, and each algorithm used (different) random initialization.

5 Conclusion

Discuss the subsequent conclusions we gained from this reimplementation of LDA. Summarize advantages and disadvantages as well.

Author Contributions

All authors contributed to the overall study of LDA and its implementation. Xiang preprocessed AP dataset in Python to generate input matrices for LDA, and implemented Gibbs Sampling LDA in MATLAB. Zheng implemented variational inference LDA in MATLAB. Sajan implemented the Unigram model in MATLAB. Yan wrote MATLAB functions for calculating perplexity and extracting subtopics from LDA outputs. All authors participated in final data analysis and interpretation, as well as writing of the manuscript.

“Law”	**“War”**	**“Trade”**	**“Politics”**	**“Company”**
court	united	percent	government	million
state	states	market	soviet	year
federal	military	prices	party	billion
case	war	stock	union	company
department	president	dollar	south	new
law	american	trade	gorbachev	workers
attorney	iraq	year	political	based
judge	officials	late	west	last
office	aid	oil	country	corp
former	israel	higher	president	co

(a) List of topics generated by Gibbs sampling.

“Law”	**“War”**	**“Trade”**	**“Politics”**	**“Company”**
court	president	dollar	dukakis	percent
year	bush	late	bush	million
years	united	new	new	year
one	states	one	year	billion
two	government	yen	campaign	market
new	new	air	people	new
state	year	london	president	stock
people	soviet	two	state	prices
case	military	york	one	company
last	house	bid	democratic	last

(b) List of topics generated by variational inference.

Figure 4: List of topics generated by LDA using two difference algorithms.

References

[1] Griffiths, TL., Steyvers, M. "Finding scientific topics." Proceedings of the National Academy of Sciences. 101(suppl 1) (2004): 5228-35.

[2]. Jordan, Michael I., et al. "An introduction to variational methods for graphical models." Machine Learning 37.2 (1999): 183-233.

[3]. Teh, Yee Whye, et al. "Hierarchical Dirichlet Processes." Journal of the American Statistical Association 101.476 (2006).

[4]. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." the Journal of Machine Learning Research 3 (2003): 993-1022.

[5]. D. Blei and J. Lafferty. "Topic Models." In A. Srivastava and M. Sahami, editors, Text Mining: Classification, Clustering, and Applications. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009.

[6]. `http://www.cs.columbia.edu/~blei/lda-c/`. It should be noted that the size of the dataset here is about one eighth of the dataset used by Blei et al. So there might be some discrepancy between our result and those presented in Blei et al.

[7]. Nigam, K., McCallum A., Thrun S., and Mitchell, T. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–104, 2000.

[8]. Hoffman, T. Probabilistic latent semantic indexing. *Proceedings fo the Twenty-Second Annual International SIGIR Conference*, 1999.