A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling

William M. Darling
School of Computer Science
University of Guelph

December 1, 2011

Abstract

This technical report provides a tutorial on the theoretical details of probabilistic topic modeling and gives practical steps on implementing topic models such as Latent *Dirichlet* Allocation (LDA) through the Markov Chain Monte Carlo approximate inference algorithm Gibbs Sampling.

1 Introduction

Following its publication in 2003, Blei et al.'s Latent *Dirichlet* Allocation (LDA) [3] has made topic modeling – a subfield of machine learning applied to everything from computational linguistics [4] to bioinformatics [8] and political science [2] – one of the most popular and most successful paradigms for both supervised and unsupervised learning. Despite topic modeling's undisputed popularity, however, it is for many – particularly newcomers – a difficult area to break into due to its relative complexity and the common practice of leaving out implementation details in papers describing new models. While key update equations and other details on inference are often included, the intermediate steps used to arrive at these conclusions are often left out due to space constraints, and what details are given are rarely enough to enable most researchers to test the given results for themselves by implementing their own version of the described model. The purpose of this technical report is to help bridge the gap between the model definitions provided in research publications and the practical implementations that are required for performing learning in this exciting area. Ultimately, it is hoped that this tutorial will help enable the reader to build his or her own novel topic models.

This technical report will describe what topic modeling is, how various models (LDA in particular) work, and most importantly, how to implement a working system to perform learning with topic models. Topic modeling as an area will be introduced through the section on LDA, as it is the "original" topic model and its modularity allows the basics of the model to be used in more complicated topic models.[1] Following the introduction to topic modeling through LDA, the problem of *posterior inference* will be discussed. This section will concentrate first on the theory of the stochastic approximate inference technique Gibbs Sampling and then it will discuss implementation details for building a topic model Gibbs sampler.

2 Latent *Dirichlet* Allocation

LDA is a generative probabilistic model for collections of grouped discrete data [3]. Each group is described as a random mixture over a set of latent topics where each topic is a discrete distribution over the collection's vocabulary. While LDA is applicable to any corpus of grouped discrete data, from now on I will refer to the standard NLP use case where a corpus is a collection of documents, and the data are words. The generative process for a document collection $\mathbf{D}$ under the LDA model is as follows:

1. For $k = 1...K$:
    (a) $\phi^{(k)} \sim Dirichlet(\beta)$
2. For each document $d \in \mathbf{D}$:
    (a) $\theta_d \sim Dirichlet(\alpha)$
    (b) For each word $w_i \in d$:
        i. $z_i \sim Discrete(\theta_d)$
        ii. $w_i \sim Disctete(\phi^{(z_i)})$

where K is the number of latent topics in the collection, $\phi^{(k)}$ is a discrete probability distribution over a fixed vocabulary that represents the kth topic distribution, θ_d is a document-specific distribution over the available topics, z_i is the topic index for word w_i, and α and β are hyperparameters for the symmetric Dirichlet distributions that the discrete distributions are drawn from.

The generative process described above results in the following joint distribution:

$$p(\mathbf{w}, \mathbf{z}, \theta, \phi|\alpha, \beta) = p(\phi|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|\phi_z) \tag{1}$$

The unobserved (latent) variables $\mathbf{z}$, θ, and ϕ are what is of interest to us. Each θ_d is a low-dimensional representation of a document in "topic"-space, each z_i represents which topic generated the word instance w_i, and each $\phi^{(k)}$ represents a $K \times V$ matrix where $\phi_{i,j} = p(w_i|z_j)$. Therefore, one of the most interesting aspects of LDA is that it can learn, in an unsupervised manner, words that

[1] While LDA is an extension to probabilistic latent semantic analysis [12] (which in turn has ideological routes in the matrix factorization technique LSI), the topic modeling "revolution" really took off with the introduction of LDA likely due to its fully probabilistic grounding.

“environment”	“travel”	“fantasy football”
emission	travel	game
environmental	hotel	yard
air	roundtrip	defense
permit	fares	allowed
plant	special	fantasy
facility	offer	point
unit	city	passing
epa	visit	rank
water	miles	against
station	deal	team

Table 1: Three topics learned using LDA on the Enron Email Dataset.

we would associate with certain topics, and this is expressed through the topic distributions ϕ. An example of the top 10 words for 3 topics learned using LDA on the Enron email dataset[2] is shown in Figure 1 (the topic labels are added manually).

3 Inference

The key problem in topic modeling is *posterior inference.* This refers to reversing the defined generative process and learning the posterior distributions of the latent variables in the model given the observed data. In LDA, this amounts to solving the following equation:

$$p(\theta, \phi, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \phi, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{2}$$

Unfortunately, this distribution is *intractable* to compute. The normalization factor in particular, $p(\mathbf{w}|\alpha, \beta)$, cannot be computed exactly. All is not lost, however, as there are a number of approximate inference techniques available that we can apply to the problem including variational inference (as used in the original LDA paper) and Gibbs Sampling (as we will use here).

3.1 Gibbs Sampling

3.1.1 Theory

Gibbs Sampling is one member of a family of algorithms from the Markov Chain Monte Carlo (MCMC) framework [9]. The MCMC algorithms aim to construct a Markov chain that has the target posterior distribution as its stationary distribution. In other words, after a number of iterations of stepping through the chain, sampling from the distribution should converge to be close to sampling

[2]`http://www.cs.cmu.edu/~enron/`.

from the desired posterior. Gibbs Sampling is based on sampling from conditional distributions of the variables of the posterior.

For example, to sample $\mathbf{x}$ from the joint distribution $p(\mathbf{x}) = p(x_1, ..., x_m)$, where there is no closed form solution for $p(\mathbf{x})$, but a representation for the conditional distributions is available, using Gibbs Sampling one would perform the following (from [1]):

1. Randomly initialize each x_i
2. For $t = 1, ..., T$:

   2.1 $x_1^{t+1} \sim p(x_1 | x_2^{(t)}, x_3^{(t)}, ..., x_m^{(t)})$

   2.2 $x_2^{t+1} \sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, ..., x_m^{(t)})$

   2.m $x_m^{t+1} \sim p(x_m | x_1^{(t+1)}, x_2^{(t+1)}, ..., x_{m-1}^{(t+1)})$

This procedure is repeated a number of times until the samples begin to converge to what would be sampled from the true distribution. While convergence is theoretically guaranteed with Gibbs Sampling, there is no way of knowing how many iterations are required to reach the stationary distribution. Therefore, diagnosing convergence is a real problem with the Gibbs Sampling approximate inference method. However, in practice it is quite powerful and has fairly good performance. Typically, an acceptable estimation of convergence can be obtained by calculating the log-likelihood or even, in some situations, by inspection of the posteriors.

For LDA, we are interested in the latent document-topic portions θ_d, the topic-word distributions $\phi^{(z)}$, and the topic index assignments for each word z_i. While conditional distributions – and therefore an LDA Gibbs Sampling algorithm – can be derived for each of these latent variables, we note that both θ_d and $\phi^{(z)}$ can be calculated using just the topic index assignments z_i (*i.e.* $\mathbf{z}$ is a sufficient statistic for both these distributions).[3] Therefore, a simpler algorithm can be used if we integrate out the multinomial parameters and simply sample z_i. This is called a *collapsed* Gibbs sampler.

The collapsed Gibbs sampler for LDA needs to compute the probability of a topic z being assigned to a word w_i, given all other topic assignments to all other words. Somewhat more formally, we are interested in computing the following posterior up to a constant:

$$p(z_i | \mathbf{z}_{-i}, \alpha, \beta, \mathbf{w}) \tag{3}$$

where $\mathbf{z}_{-i}$ means all topic allocations *except* for z_i. To begin, the rules of conditional probability tell us that:

$$p(z_i | \mathbf{z}_{-i}, \alpha, \beta, \mathbf{w}) = \frac{p(z_i, \mathbf{z}_{-i}, \mathbf{w} | \alpha, \beta)}{p(\mathbf{z}_{-i}, \mathbf{w} | \alpha, \beta)} \propto p(z_i, \mathbf{z}_{-i}, \mathbf{w} | \alpha, \beta) = p(\mathbf{z}, \mathbf{w} | \alpha, \beta) \tag{4}$$

[3] $\theta_{d,z} = \frac{n(d,z)+\alpha}{\sum_{|Z|} n(d,z)+\alpha}$, $\phi_{z,w} = \frac{n(z,w)+\beta}{\sum_{|W|} n(z,w)+\beta}$.

We then have:

$$p(\mathbf{w},\mathbf{z}|\alpha,\beta) = \int\int p(\mathbf{z},\mathbf{w},\theta,\phi|\alpha,\beta)\mathrm{d}\theta\mathrm{d}\phi \tag{5}$$

Following the LDA model defined in equation (1), we can expand the above equation to get:

$$p(\mathbf{w},\mathbf{z}|\alpha,\beta) = \int\int p(\phi|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p(\mathbf{w}|\phi_z)\mathrm{d}\theta\mathrm{d}\phi \tag{6}$$

Then, we group the terms that have dependent variables:

$$p(\mathbf{w},\mathbf{z}|\alpha,\beta) = \int p(\mathbf{z}|\theta)p(\theta|\alpha)\mathrm{d}\theta \int p(\mathbf{w}|\phi_z)p(\phi|\beta)\mathrm{d}\phi \tag{7}$$

Both terms are multinomials with Dirichlet priors. Because the Dirichlet distribution is conjugate to the multinomial distribution, our work is vastly simplified; multiplying the two results in a Dirichlet distribution with an adjusted parameter. Beginning with the first term, we have:

$$\begin{aligned}\int p(\mathbf{z}|\theta)p(\theta|\alpha)\mathrm{d}\theta &= \int \prod_i \theta_{d,z_i} \frac{1}{B(\alpha)} \prod_k \theta_{d,k}^{\alpha_k} \mathrm{d}\theta_d \\ &= \frac{1}{B(\alpha)} \int \prod_k \theta_{d,k}^{n_{d,k}+\alpha_k} \mathrm{d}\theta_d \\ &= \frac{B(n_{d,\cdot}+\alpha)}{B(\alpha)} \end{aligned} \tag{8}$$

where $n_{d,k}$ is the number of times words in document d are assigned to topic k, a $\cdot$ indicates summing over that index, and $B(\alpha)$ is the multinomial beta function, $B(\alpha) = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$. Similarly, for the second term (calculating the likelihood of words given certain topic assignments):

$$\begin{aligned}\int p(\mathbf{w}|\phi_z)p(\phi|\beta)\mathrm{d}\phi &= \int \prod_d \prod_i \phi_{z_{d,i},w_{d,i}} \prod_k \frac{1}{B(\beta)} \prod_w \phi_{k,w}^{\beta_w} \mathrm{d}\phi_k \\ &= \prod_k \frac{1}{B(\beta)} \int \prod_w \phi_{k,w}^{\beta_w+n_{k,w}} \mathrm{d}\phi_k \\ &= \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(\beta)} \end{aligned} \tag{9}$$

Combining equations (8) and (9), the expanded joint distribution is then:

$$p(\mathbf{w},\mathbf{z}|\alpha,\beta) = \prod_d \frac{B(n_{d,\cdot}+\alpha)}{B(\alpha)} \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(\beta)} \tag{10}$$

The Gibbs sampling equation for LDA can then be derived using the chain rule (where we leave the hyperparameters α and β out for clarity).[4] Note that the superscript $^{(-i)}$ signifies leaving the ith token out of the calculation:

$$\begin{aligned}
p(z_i|\mathbf{z}^{(-i)},\mathbf{w}) &= \frac{p(\mathbf{w},\mathbf{z})}{p(\mathbf{w},\mathbf{z}^{(-i)})} = \frac{p(\mathbf{z})}{p(\mathbf{z}^{(-i)})} \cdot \frac{p(\mathbf{w}|\mathbf{z})}{p(\mathbf{w}^{(-i)}|\mathbf{z}^{(-i)})p(w_i)} \\
&\propto \prod_d \frac{B(n_{d,\cdot}+\alpha)}{B(n_{d,\cdot}^{(-i)}+\alpha)} \prod_k \frac{B(n_{k,\cdot}+\beta)}{B(n_{k,\cdot}^{(-i)}+\beta)} \\
&\propto \frac{\Gamma(n_{d,k}+\alpha_k)\Gamma(\sum_{k=1}^K n_{d,k}^{(-i)}+\alpha_k)}{\Gamma(n_{d,k}^{(-i)}+\alpha_k)\Gamma(\sum_{k=1}^K n_{d,k}+\alpha_k)} \cdot \frac{\Gamma(n_{k,w}+\beta_w)\Gamma(\sum_{w=1}^W n_{k,w}^{(-i)}+\beta_w)}{\Gamma(n_{k,w}^{(-i)}+\beta_w)\Gamma(\sum_{w=1}^W n_{k,w}+\beta_w)} \\
&\propto (n_{d,k}^{(-i)}+\alpha_k)\frac{n_{k,w}^{(-i)}+\beta_w}{\sum_{w'} n_{k,w'}^{(-i)}+\beta_{w'}} \qquad (11)
\end{aligned}$$

3.1.2 Implementation

Implementing an LDA collapsed Gibbs sampler is surprisingly straightforward. It involves setting up the requisite count variables, randomly initializing them, and then running a loop over the desired number of iterations where on each loop a topic is sampled for each word instance in the corpus. Following the Gibbs iterations, the counts can be used to compute the latent distributions θ_d and ϕ_k.

The only required count variables include $n_{d,k}$, the number of words assigned to topic k in document d; and $n_{k,w}$, the number of times word w is assigned to topic k. However, for simplicity and efficiency, we also keep a running count of n_k, the total number of times any word is assigned to topic k. Finally, in addition to the obvious variables such as a representation of the corpus ($\mathbf{w}$), we need an array $\mathbf{z}$ which will contain the current topic assignment for each of the N words in the corpus.

Because the Gibbs sampling procedure involves sampling from distributions conditioned on all *other* variables (in LDA this of course includes all other current topic assignments, but not the current one), before building a distribution from equation (11), we must remove the current assignment from the equation. We can do this by decrementing the counts associated with the current assignment because the topic assignments in LDA are *exchangeable* (i.e. the joint probability distribution is invariant to permutation). We then calculate the (unnormalized) probability of each topic assignment using equation (11). This discrete distribution is then sampled from and the chosen topic is set in the $\mathbf{z}$ array and the appropriate counts are then incremented. See Algorithm 1 for the full LDA Gibbs sampling procedure.

[4] For the full, nothing-left-out derivation, please see [5] and [11].

```
Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}, n_{k,w}$, and $n_k$
begin
    randomly initialize $\mathbf{z}$ and increment counters
    foreach iteration do
        for $i = 0 \to N - 1$ do
            $word \leftarrow w[i]$
            $topic \leftarrow z[i]$
            $n_{d,topic}$-=1; $n_{word,topic}$-=1; $n_{topic}$-=1
            for $k = 0 \to K - 1$ do
                $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta\times W}$
            end
            $topic \leftarrow$ sample from $p(z|\cdot)$
            $z[i] \leftarrow topic$
            $n_{d,topic}$+=1; $n_{word,topic}$+=1; $n_{topic}$+=1
        end
    end
    return $\mathbf{z}$, $n_{d,k}, n_{k,w}, n_k$
end
```

Algorithm 1: LDA Gibbs Sampling

4 Extensions To LDA

While LDA – the "simplest" topic model – is useful in and of itself, a great deal of novel research surrounds extending the basic LDA model to fit a specific task or to improve the model by describing a more complex generative process that results in a better model of the real world. There are countless papers delineating such extensions and it is not my intention to go through them all here. Instead, this section will outline some of the ways that LDA can and has been extended with the goal of explaining how inference changes as a result of additions to a model and how to implement those changes in a Gibbs sampler.

4.1 LDA With a Background Distribution

One of the principal problems with LDA is that for useful results, stop-words must be removed in a pre-processing step. Without this filtering, very common words such as *the, of, to, and, a*, etc. will pervade the learned topics, hiding the statistical semantic word patterns that are of interest. While stop-word removal does a good job at solving this problem, it is an *ad hoc* measure that results in a model resting on a non-coherent theoretical basis. Further, stop-word removal is not without problems. Stop-word lists must often be domain-dependent, and there are inevitably cases where filtering results in under-coverage or over-coverage, causing the model to continue being plagued by noise, or missing patterns that may be of interest to us.

One approach to keep stop-words out of the topic distributions is to imagine all stop-words being generated by a "background" distribution [6, 7, 10]. The background distribution is the same as a topic – it is a discrete probability distribution over the corpus vocabulary – but every document draws from the background as well as the topics specific to that document. [7] and [10] use this approach to separate high-content words from less-important words to perform multi-document summarization. [6] uses a similar model for information retrieval where a word can either be generated from a background distribution, a document-specific distribution, or one of T topic distributions shared amongst all the documents. The generative process is similar to that of LDA, except that there is a multinomial variable x associated with each word that is over the three different "sources" of words. When $x = 0$, the background distribution generates the word, when $x = 1$, the document-specific distribution generates the word, and when $x = 2$, one of the topic distributions generates the word.

Here, we will describe a simpler model where only a background distribution is added to LDA. A binomial variable x is associated with each word that decides whether the word will be generated by the topic distributions or by the background. The generative process is then:

```
1. $\zeta \sim Dirichlet(\delta)$
2. For $k = 1...K$:
   (a) $\phi^{(k)} \sim Dirichlet(\beta)$
3. For each document $d \in \mathbf{D}$:
   (a) $\theta_d \sim Dirichlet(\alpha)$
   (b) $\lambda_d \sim Dirichlet(\gamma)$
   (c) For each word $w_i \in d$:
       i. $x_i \sim Discrete(\lambda_d)$
       ii. If $x = 0$:
           A. $w_i \sim Discrete(\zeta)$
       iii. Else:
           A. $z_i \sim Discrete(\theta_d)$
           B. $w_i \sim Disctete(\phi^{(z_i)})$
```

where ζ is the background distribution, and λ_d is a document-specific binomial sampled from a Dirichlet prior γ.

Developing a Gibbs sampler for this model is similar to the LDA implementation, but we have to be careful about when counts are incremented and decremented. We only adjust the background-based counts when the background was sampled as the word generator, and we only adjust the topic counts when it is the converse. We must update the x-based counts each time, however, because we sample the *route* that led to the word being generated each time. The sampler must compute the probability not only of a topic being chosen for the given document and the probability of that topic generating the given word, it must also compute the probability that the model is in the topic-model state. This too, however, is straightforward to implement. A distribution of $T + 1$ components can be created for each word (on each iteration) where the first component corresponds to the background distribution generating the word and the other T are the probabilities for each topic having generated the word.

5 Conclusion

LDA and other topic models are an exciting development in machine learning and the surface has only been scratched on their potential in a number of diverse fields. This report has sought to aid researchers new to the field in both understanding the mathematical underpinnings of topic modeling and in implementing algorithms to make use of this new pattern recognition paradigm.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] David M. Blei and Sean Gerrish. Predicting legislative roll calls from text. In *International Conference on Machine Learning*, 2011.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[4] Jordan Boyd-Graber, David Blei, and Xiaojin Zhu. A topic model for word sense disambiguation. In *Empirical Methods in Natural Language Processing*, 2007.

[5] Bob Carpenter. Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling. Technical report, Lingpipe, Inc., 2010.

[6] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, pages 241–248, 2006.

[7] Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[8] Georg K Gerber, Robin D Dowell, Tommi S Jaakkola, and David K Gifford. Automated discovery of functional generality of human gene expression programs. *PLoS Comput Biol*, 3(8):e148, 2007.

[9] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC, 1999.

[10] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[11] Gregor Heinrich. Parameter estimation for text analysis. Technical report, University of Leipzig, Germany, 2008.

[12] Thomas Hofmann. Probabilistic latent semantic analysis. In *In Proc. of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.