A Comparison of Posterior Approximation Algorithms for Latent Dirichlet Allocation.

Cindy Cook

STAT 540 Final Project

December 14, 2015

0.1 Introduction

With an increase in the amount of digitized text, topic modeling has become an important area of research. One of the most popular methods for topic modeling is Latent Dirichlet Allocation (LDA) first described in [3]. LDA, a probabilistic hierarchical generative model, uses the Bayesian framework to discover hidden structure, or *topics*, within a set of documents referred to as a corpus. Given a document, the distribution of latent topics is found through estimation of the posterior distribution. However, this is very challenging and (computationally demanding) since the posterior distribution of the latent variables given a document is intractable. Approximation of this posterior is a valid option, and there has been a heavy development of fast and accurate approximation methods applied to LDA, which provide a user with several options including collapsed Gibbs sampling (CGS) [8], collapsed variational Bayesian inference [13], maximum likelihood estimation [10], and maximum a posteriori estimation [7]. In [3], a variational EM (VEM) algorithm was applied to solve the problem. These methods were coded in C by Blei (refer to [4]) and have since been utilized in the *topicmodels* package in R [9]. Another package, *lda*, in R has been developed by Chang, which uses his own C code with collapsed Gibbs sampling for posterior approximation [6]. There has been no uniform discussion under what situations should VEM or CGS be used. On the website Cross Validated, there is a discussion of which package is better for users, with no definitive answer [1].

This paper begins by reviewing LDA, VEM and CGS. It then conducts a simulation study, which will focuses on a comparison of the CGS and VEM algorithms as applied to LDA. The simlation study is modeled after Blei's comparison of VEM to collapsed VEM in [12]. He found both algorithms to perform computationally efficiently and fast, but that the collapsed version of the VEM algorithm was less biased. Two other studies have been done comparing different methods of approximating the posterior for LDA [2] and [13], but both focused on real datasets and did not consider varying model parameters.

0.2 Latent Dirichlet Allocation

0.2.1 Generative Process

Given a set of K topics $k = 1, ...K$ and a set of D documents $d = 1, ..., D$, also referred to as a corpus, the vocabulary V is the set of unique words in the Corpus say $w_{1,1}, ..., w_{d,n_d}, ..., w_{D,n_D}$, where n_d is the number of uniques words in document d and $|V| = \sum_{d=1}^{D} n_d$ or the size of the vocabulary. The goal of this algorithm as stated by Blei is, "to find a probabilistic model of a corpus that not only assigns high probability to members of the corpus, but also assigns high probability to other 'similar' documents" ([3], pp.996). The generative process for a corpus is as follows:

- For each topic,
  1. Draw $\beta^k \sim Dir(\eta)$, a distribution over the vocabulary.

- For each document,
  2. Draw $\theta^d \sim Dir(\alpha)$, a distribution over the topics.
  - For each word in document d,
    3. Draw $z_{d,n} \sim Mult(\theta^d)$, a topic assignment.
    4. Draw $w_{d,n} \sim Mult(\beta^{z_{d,n}})$, the nth word in document d.

Assuming this model, we can define the joint probability distribution of a corpus as:

$$p(\beta, \theta, z_{d,n}, w_{d,n}|\alpha, \eta) = \prod_{k=1}^{K} p(\beta^k|\eta) \prod_{d=1}^{D} \left(p(\theta^d|\alpha) \prod_{n=1}^{n_d} p(z_{d,n}|\theta^d) p(w_{d,n}|\beta^1, ..., \beta^K, z_{d,n}) \right), \tag{1}$$

This process first requires that we know the number of topics K a priori, and second places no emphasis on word order allowing the joint distributions for both $z_{d,n}$ and $w_{d,n}$ be exchangeable meaning that they are invariant to permutations. All that is considered to matter are the word counts or number of times a word is assigned a particular topic and/or is drawn for a certain document. We will use the following dot notation to denote specific word counts:

$$c_{k,d,v} = \sum_{n=1}^{n_d} I(z_{d,n} = k \& w_{d,n} = v); \qquad \# \text{ of times word } v \text{ is assigned to topic } k \text{ in document } d$$

$$c_{k,.,v} = \sum_{d=1}^{D} c_{k,d,v}; \qquad \# \text{ of words in document } d \text{ assigned to topic } k$$

$$c_{k,d,.} = \sum_{v=1}^{V} c_{k,d,v}; \qquad \# \text{ of times word } v \text{ is assigned to topic } k \text{ in any document}$$

$$c_{k,.,.} = \sum_{d=1}^{D} \sum_{v=1}^{V} c_{k,d,v}; \qquad \# \text{ of words in corpus assigned to topic } k$$

In practice, we do not know $z_{d,n}, \beta^k$ or θ^d. All we are given is a set of documents providing the $w_{d,n}$ parameters. Most often, a researcher is interested in the latent variables β^k and θ^d. To estimate these variables, we need to consider the following posterior distribution:

$$\begin{aligned} p(\beta, \theta|w, \eta, \alpha) &= \frac{p(\beta, \theta|\eta, \alpha) * p(w|\theta, \beta)}{p(w|\alpha, \eta)} \\ &= \frac{p(\beta|\eta) p(\theta|\alpha) \sum_x (p(w|z, \beta) p(z|\theta)}{\int_\beta \int_\theta \left[(p(\beta|\eta) p(\theta|\alpha) \sum_z (p(w|z, \beta) p(z|\theta))) \right] d\beta d\theta} \end{aligned} \tag{2}$$

Not only has this distribution been proven to be intractable [3], so is the likelihood function. In both cases we have to sum over all possible values of z for every word. We will explore both Gibbs sampling and variational EM methods to approximate 2 in the following two sections.

0.3 Gibbs Sampling

Gibbs sampling is a special form of the Metropolis Hasting MCMC algorithm. The idea, as applied to lda, is to approximate 2 by repeatedly sampling from the conditional distributions of the latent variables given all other known and estimated variables at each step. It is known that the Gibbs sampler constitutes a Markov Chain whose stationary distribution is the function in question. The Markov Chain described below asymptotically converges to 2:

1. Initialize $\beta^{k^{(0)}}, \theta^{d^{(0)}}$, and all of the topic assignments $z_{d,n}^{(0)}$ for each word in the vocabulary.
2. For t in $1, 2, ...,$ sample:
   (a) $\beta^{k^{(t)}}$ from $p(\beta^{k^{(t)}}|\theta^{d^{(t-1)}}, z_{d,n}^{(t-1)}, w_{d,n}, \eta, \alpha) = Dir(\eta + c_{k,.,v})$
   (b) For every document d
      i. $\theta^{d^{(t)}}$ from $p(\theta^{d^{(t)}}|\beta^{k^{(t)}}, z_{d,n}^{(t-1)}, w_{d,n}, \eta, \alpha) = Dir(\alpha + c_{k,d,.})$
      ii. $z_{d,i}^{(t)}$ from $p(z_{d,i}^{(t)}|z_{d,-i}^{(t-1)}\beta^{k^{(t)}}, \theta^{d^{(t)}}, w_{d,n}, \eta, \alpha)$ for all $i = 1, ...n_d$
3. Continue to update and sample each of the latent variables until convergence.

Although this algorithm will eventually converge to the distributions of the latent variables, it is slow. Due to the conjugacy of the priors, each step is fast and easy to compute, but there are a large number of latent variables that need to be sampled. One way of speeding this algorithm up is to 'collapse' the parameter space by integrating out specific parameters. This method is referred to collapsed Gibbs sampling and has been implemented for LDA in [8].

Here, the algorithm integrates out both θ and β from the Markov Chain's state space. Then it uses a Gibbs sampler to iteratively sample the topic assignments. After solving for converged estimates, both θ and β can be easily estimated. For more information on the collapsed gibbs sampler, refer to [11]. After integrating out these variables, [5] showed that for word n in document d, the probability of word $w_{d,n}$ being assigned topic $z_{d,n} = k$ is

$$p(z_{d,n}^t = k|z_{-(d,n)}^{(t-1)}, \theta, \beta, w) = c^{-1} * \frac{(c_{z_{d,n},d,.}^{-(d,n)} + \alpha_{z_{d,n}}) * (c_{z_{d,n},.,w_{d,n}}^{-(d,n)} + \beta_{w_{d,n}})}{c_{z_{d,n},.,.}^{-(d,n)} + \sum_{v=1}^{V} \beta_v} \tag{3}$$

where c is the normalizing constant equal to

$$\sum_{k=1}^{K} \frac{(c_{k,d,.}^{-(d,n)} + \alpha_k) * (c_{k,.,w_{d,n}}^{-(d,n)} + \beta_{w_{d,n}})}{c_{k,.,.}^{-(d,n)} + \sum_{v=1}^{V} \beta_v}$$

The update equations for θ and β are as follows [5]:

$$\hat{\theta}_{d,k} = \frac{\alpha_k + n_{d,k,.}}{D\alpha + n_{.,d,.}} \qquad \hat{\beta}_{v,k} = \frac{\beta_{k,v} + n_{k,.,v}}{J\beta + n_{k,.,.}} \tag{4}$$

Psuedo Code for the collapsed Gibbs sampler is as follows:

1. Initialize the topic assignments for each word in the vocabulary $z^{(0)}$.
2. For each w_i and $t = 1, 2, 3, ...$:
   (a) For each topic $k = 1, 2, 3, ...$ compute 3.
   (b) Draw the new topic assignment from the computed discrete distribution $z_i^{(t)}$.
3. Continue until convergence.
4. Calculate $\hat{\theta}$ and $\hat{\beta}$ from 4.

0.4 Variational EM

In order to infer the parameters, a second approach using variational methods was first described in [3]. This approach uses Jensen's inequality to obtain a lower bound on the log likelihood of the data given α, η: $l(\alpha, \eta) = \sum_{d=1}^{D} (\log (w_d|\alpha, \eta))$. In general, a family of tractable surrogate distributions defined by their unique parameters is found. These parameters are referred to as variational parameters. The lower bound is found by minimizing the Kulllback-Leibler (KL) divergence between the variational distribution and the true posterior. A simple variational distribution to consider, the one that Blei considers, just removes the dependence between θ and β, which causes the intractability of the true posterior. Here we will use the following surrogate function:

$$\tilde{q}(z, \beta, \theta) = \prod_k \tilde{q}(\beta_k|\tilde{\eta}_k) \prod_d \tilde{q}(\theta_d|\tilde{\alpha}_d) \prod_{d,n} \tilde{q}(z_{d,n}|\tilde{\gamma}_{d,n}).$$

Minimizing the KL distance leads to the following update equations:

$$\tilde{\gamma}_{n,d,k}^{(t+1)} = \exp\left(\Psi(\tilde{\alpha}_{d,k}^t) + \Psi(\tilde{\eta}_{k,x_{d,n}}^t) - \Psi\left(\sum_n (\tilde{\eta}_{k,n}^t)\right)\right) \qquad \tilde{\alpha}_{d,k}^t = \alpha + \sum_n (\tilde{\gamma}_{d,k,n}^{(t+1)}) \qquad \tilde{\eta}_{k,n}^{(t+1)} = \eta \tag{5}$$

where $\Psi(x) = \frac{d \log(\Gamma x)}{dx}$ and I is the indicator function. Blei goes a step further and proposes the following EM algorithm to estimate the β and α parameters:

1. (E-step) Find the variational parameters by using a fixed point iterative algorithm to minimize the KL distance.
2. (M-step) Maximize the resulting lower bound on the log likelihood with respect to β and α.

For the M-step, β can be found analytically using

$$\beta^{(t+1)} \propto \sum_{d=1}^{D} \sum_{n=1}^{N_d} (\tilde{\eta}_{d,k,n} w_{d,k,n}), \tag{6}$$

but $\alpha^{(t+1)}$ must be found using a Newton Raphson algorithm. Here, the estimate for β is proportional because we need to normalize so that the β's sum to one.

Below outlines Pseudo Code for the variational Bayes approximation algorithm:

1. Initialize:

$$\begin{aligned} \text{For each topic } k, \qquad & \tilde{\eta}_n^0 = \eta_n + fracNk \\ \text{For each document } d, \qquad & \tilde{\alpha}_k^0 = \alpha_k + fracNk \\ \text{For each word in document } d, \qquad & \tilde{\gamma}_k^0 = frac1k \end{aligned}$$

2. (E-step) For each document d in $1, ..., D$,
   Repeat for $t = 0, 1, 2, ...$:
   (a) For n in $1, ..., N_d$
   (b) For k in $1, ..., K$
   (c) Update $\tilde{\gamma}_{n,d,k}^{(t+1)}$ using 5.
   (d) Normalize $\tilde{\gamma}_{n,d}^{(t+1)}$ to sum to one.
   (e) Update $\tilde{\alpha}^{(t+1)}$ and $\tilde{\eta}^{(t+1)}$ using 5
   (f) Continue until convergence.
3. Solve the log likelihood given the data and variational estimates $\tilde{\alpha}$ and $\tilde{\eta}$.
4. (M-step) For each document d in $1, ..., D$
5. For each k in $1, ..K$
6. For each n in $1, ..., N_d$
7. Solve for $\beta_{n,k}$ using 6
8. Normalize the β so they sum to one.
9. Solve for α.
10. If likelihood converged using new parameters then end.
11. Else go back to the E-step.

1.

0.5 Simulation Study

I will start by first gaining an understanding of what these algorithms are theoretically doing. Then I will explore their performances empirically. To this aim, I will simulate data following the generative process defined in [3]. I will vary the number of topics, i.e. groups: $5, 10, 25, 50$, the hyperparameters , i.e. values: $0.001, 0.01, 0.1$, and the length of words in the vocabulary, i.e. 0 to $5,000$ by $1,000$ to follow the simulation set up in [12]. Mukherjee and Blei focused their study on comparing the collapsed and non-collapsed variational Bayes algorithms. I will focus on comparing an algorithm I develop to compute LDA with collapsed Gibbs sampling and compare it to the existing results from the LDA function in the *topicmodels* package in R. I will use both perplexity and precision/recall measures to compare the two algorithms as shown in [2]. I will also take into account the computational costs. With the research I have done so far, it is known that the collapsed Gibbs algorithm will take longer to gain convergence, but should be more accurate. In what situations this very general statement remains true is the ultimate goal of this research project.

0.5.1 Results

0.6 Conclusions

0.6.1 Discussion

0.6.2 Future Directions

Bibliography

Bibliography

[1] Anonymous. (2012), "Two R Packages for Topic Modeling, lda and topicmodels?" *Cross Validated*: http://stats.stackexchange.com/questions/24441/two-r-packages-for-topic-modeling-lda-and-topicmodels.

[2] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. (2009), "On Smoothing and Inference for Topic Models." *UAI*: 27-34.

[3] Blei, D., Ng, A., and Jordan, M. (2003), "Latent Dirichlet Allocation." *Journal of Machine Learning Research*: 3 993-1022.

[4] Blei, D. (2004),"LDA-C."

[5] Carpenter, Bob. (2010), "Integrating out Multinomial Parameters in Latent Dirichlet Allocation and Naive Bayes for Collapsed Gibbs Sampling." *LingPipe*: carp@lingpipe.com.

[6] Chang, J. (2015), "Package 'lda'." *CRAN*.

[7] Chien, J., and Wu, M. (2008), "Adaptive Bayesian Semantic Analysis." *Audio, Speech, and Language Processing*, IEEE Transactions on: 16(1), 198-207.

[8] Griffiths, L. and Steyvers, M. (2004), "Finding Scientific Topics." *PNAS*: 1(Suppl 1), 5228-5235.

[9] Grun, B. and Hornik, K. (2015), "Package 'topicmodels'." *CRAN*.

[10] Hofmann, T. (2001), "Unsupervised Learning by probabilistic Latent Semantic Analysis." *Machine Learning*: 42(1), 177-196.

[11] Liu, Jun. (1994), "The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem." *ASA Jorunal*: Vol.89, No.427 Theory and Methods, 958-966.

[12] Mukherjee, I. and Blei, D. (2009), "Relative Performance Grantees for Approximate Inference in Latent Dirichlet Allocation." *NIPS*: 21, 1129âĂŞ1136.

[13] Teh, Y. W., Newman, D., and Welling, M. (2007), "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation."*NIPS*: 3, 1353-1360.