663 Final Project: LDA Collapsed Gibbs Sampling

David Chester Bernardo Dionis Altamash Rafiq

1. Abstract

We approach Latent Dirichlet Allocation (LDA), a hierachical Bayesian model, via collapsed Gibbs sampling on text corpora in order to make inferences on the parameters of interest (i.e. the latent variables). In this hierarchy, words within a document are generated by a corresponding document-specific mixture distribution of finite topics, and documents themselves are generated by a mixture of topics. We are interested in obtaining samples from the topic specific full conditionals in order to estimate the latent variables that serve as the parameters dictating the probabilities of a word occuring within a particular topic (or more generally the distribution words in a topic), and the probability of a topic occuring in a document (or more generally the mixture distribution of topics within a given document). We describe the theoretical work presented by Griffiths and Steyvers, but utilize the equivalent algorithmic respresentation illustrated by Porteous et. al. We increase the speed of the collapsed Gibbs sampler by ***

2. Background

We are primarily using "Finding Scientific Topics" (Griffiths and Styevers) [2] to understand the theoretical basis of the collapsed Gibbs sampler, but we also reference "Latent Dirichlet Allocation" (Blei et. al.)[1] and "Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation" (Porteous et. al.)[3]. We code our algorithm using the more intutive representation from Porteous et. al., but note it is theoretically equivalent to Griffiths et. al. To provide some context, the focus of Porteous et. al. was then to speed up the collapsed Gibbs sampler via their "Fast LDA" algorithm and was not reiterate the theoretical work presented by Griffiths et al. in detail. Please note that the notation within this section matches that of Griffiths et. al., while the subsequent sections covering the algorithm match the notation utilized by Porteous et al. In general, the concept of collapsed Gibbs sampling is to introduce latent variables which we can then integrate out, and sample from the full conditionals of the topic specific distributions via a relatively cost-effecient updating procedure via summing counts. We can then utilize these samples to obtain estimates of the latent parameters of interest. By collapsing, or integrating out the latent parameters, the algorithm significantly cuts down the number of steps within the markov chain.

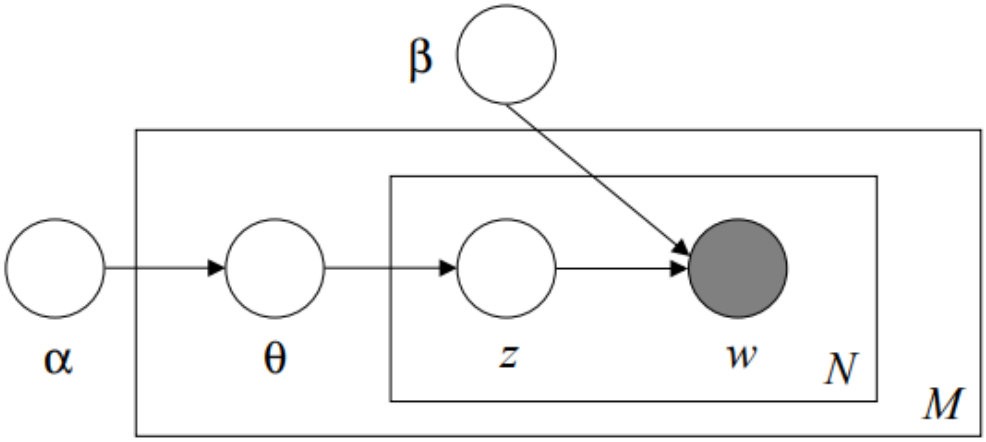
To begin, we are primarily interested in garnering samples from:

$$p(\mathbf{z}|\mathbf{w}) = \frac{p(\mathbf{z},\mathbf{w})}{\Sigma_{\mathbf{z}} p(\mathbf{z},\mathbf{w})}$$

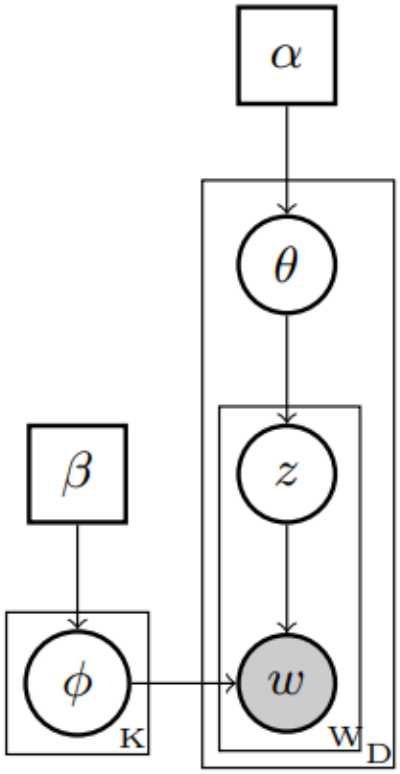
Where $p(\mathbf{z}|\mathbf{w})$ represents the posterior distribution over the assigments of words to topics. However, the denominator cannot be factorized and this quantity cannot be computed directly. The approach by Griffiths et. al. is to convert this to an MCMC set up on top of a slightly altered LDA setting first provided by Blei et. al.

We discuss the hierarchical Bayesian model and the use of conjugacy before examining the results provided by Griffiths and Steyvers. We also attempt to fill in holes not originally provided by the authors.

Consider the graphical model representation of LDA from Blei et. al.:



We contrast this to the graphical LDA model used by Griffiths et. al. (and Porteous et. al.):



Where we noticably see that the two constructions differ by ϕ over K topics.

Seen in a different light, the complete probability model by Griffiths et. al. is:

$$
\begin{aligned}
w_i|z_i, \phi^{(z_i)} &\sim Discrete(\phi^{(z_i)}) \\
\phi &\sim Dirichlet(\beta) \\
z_i|\theta^{(d_i)} &\sim Discrete(\theta^{(d_i)}) \\
\theta &\sim Dirichlet(\alpha)
\end{aligned}
$$

However, we can view this in the multivariate setting where $\mathbf{w}|\mathbf{z}, \phi \sim Multinomial(\phi)$, and similarly, $\mathbf{z}|\theta \sim Multinomial(\boldsymbol{\theta})$, however, the above depiction provides a nice relationship between each particular $w_i, z_i, \phi^{(z_i)}, \theta^{(d_i)}$. Also note that while α and β could be vector specified, here we assume one value for $\alpha = \mathbf{1}\alpha$ and similarly $\beta = \mathbf{1}\beta$, i.e. we have symmetric Dirichlet distributions where α and β are fixed hyperparameters and each vector component α_i represents the equivalent prior weight placed on each topic k in a document and each β_i represents the equivalent prior weight on each word in a topic. The approach by Griffiths et. al. was to integrate out the latent variables ϕ and θ allowing us to avoid a much longer chain structure. However, they leave much of the calculations to the reader, of which we attempt to spell out below. I.e. we can note that:

$$p(\mathbf{z}|\mathbf{w}) = \frac{p(\mathbf{z}, \mathbf{w})}{\Sigma_{\mathbf{z}} p(\mathbf{z}, \mathbf{w})}$$

where

$$p(\mathbf{w}, \mathbf{z}) = p(\mathbf{w}|\mathbf{z})p(\mathbf{z})$$

where

$$p(\mathbf{z}, \mathbf{w}, \phi) = p(\mathbf{w}|\mathbf{z}, \phi)p(\mathbf{z}|\phi)p(\phi)$$

where integrating ϕ out of $p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)$ is equivalent to $p(\mathbf{w}|\mathbf{z})$. However, we know via Multinomial-Dirichlet conjugacy, that $p(\phi|\mathbf{w}, \mathbf{z}) \; \alpha \; \frac{p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)}{p(\mathbf{w}, \mathbf{z})} \; \alpha \; p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)$ and we can then note that the right most term has the kernel of an updated Dirichlet distribution $\phi \sim Dir(\beta + \mathbf{\Sigma w_i})$, and consequently combine this information with the fact that it is a probability distribution that must integrate to one and as such must take the normalizing constant of this updated Dirichlet form, or stated differently, we can conclude $p(\phi|\mathbf{w}, \mathbf{z}) = Dir(\beta + \Sigma \mathbf{w}_i)$. By integrating ϕ we have equivalently (using slight of hand in the second integral):

$$p(\mathbf{w}|\mathbf{z}) = \int_{\phi} p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)d\phi \; \alpha \; \int_{\phi} Dir(B + \Sigma \mathbf{w}_i)d\phi = \int_{\phi} \frac{\Gamma(\Sigma(\beta_i + \Sigma \mathbf{w_i}))}{\Pi \, \Gamma(\beta_i + \Sigma \mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma \mathbf{w_i}) - 1} d\phi$$

where $p(\mathbf{w}|\mathbf{z}, \phi)p(\phi) \; \alpha \; Dir(\beta + \Sigma \mathbf{w}_i)$ but then since the left hand side is a probability distribution as we just noted, , then we know that the proportionality constant is precisely the normalizing factor of the updated Dirichlet to which $p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)$ is proportional, and as such the chained equation above can replace the proportionaly sign with an equality. or

$$p(\mathbf{w}|\mathbf{z}) = \int_{\phi} p(\mathbf{w}|\mathbf{z}, \phi)p(\phi)d\phi = \int_{\phi} Dir(B + \Sigma \mathbf{w}_i)d\phi = \int_{\phi} \frac{\Gamma(\Sigma(\beta_i + \Sigma \mathbf{w_i}))}{\Pi \, \Gamma(\beta_i + \Sigma \mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma \mathbf{w_i}) - 1} d\phi$$

Griffiths et. al. does not provide the above steps, rather the authors provide the final form (below) with ϕ integrated out, i.e.

$$P(\mathbf{w} \mid \mathbf{z}) = \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \right)^T \prod_{j=1}^{T} \frac{\prod_w \Gamma\left(n_j^{(w)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)},$$

A number of steps seem to missing which we attempt to uncover below by reverse engineering from the results to find forms that might be of interest.

$$
\begin{aligned}
P(w \mid z) &= \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^{\top} \prod_{j=1}^{T} \frac{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)} \\
&= \int_{\phi} \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi \\
&= \underbrace{\prod_{j=1}^{T} \frac{\Pi_w \Gamma\left(n_j^{(w)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)} \cdot \prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)}}_{\text{multiply by 1.}} \int_{\phi} \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi \\
&= \prod_{j=1}^{T} \frac{\Pi_w \Gamma\left(n_j^{(w)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)} \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})} \int_{\phi} \prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi
\end{aligned}
$$

and if $\left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^{\top} = \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})}$ then this would imply that $\int_{\phi} \prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi$ integrates to 1, but this seems potentially reasonable since $\prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)}$ seems to resemble a product that would yield $\frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})}$, and the integral looks almost identical to that of the updated $\phi \sim Dir(\beta + \mathbf{\Sigma w_i})$ where expanding the product in the numerator would yield the necessary summation within the gamma function we need in order for the full term to integrate to one. However in the interest of not speculating, we can again "multiply by 1".

$$
\begin{aligned}
P(w \mid z) &= \prod_{j=1}^{T} \frac{\Pi_w \Gamma\left(n_j^{(w)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)} \cdot \prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)} \int_{\phi} \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi \\
&= \prod_{j=1}^{T} \frac{\Pi_w \Gamma\left(n_j^{(w)} + \beta\right)}{\Gamma\left(n_j^{(\cdot)} + W\beta\right)} \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^{\top} \int_{\phi} \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^{-T} \prod_{j=1}^{T} \frac{\Gamma\left(n_j^{(\cdot)} + W\beta\right)}{\Pi_w \Gamma\left(n_j^{(\omega)} + \beta\right)} \frac{\Gamma(\Sigma(\beta_i + \Sigma\mathbf{w_i}))}{\Pi\,\Gamma(\beta_i + \Sigma\mathbf{w_i})} \Pi \phi_i^{(\beta_i + \Sigma\mathbf{w_i})-1} d\phi
\end{aligned}
$$

implying that everything in the integral integrates to one. We do not find very much intuition to what we've found, but felt an attempt to fill in the holes not presented within the literature mentioned might be useful, and could be worth returning to at a later point.

Note that $n_j^{(\cdot)}$ is the number of times word w was assigned to topic j in vector of assignments $\mathbf{z}$ above.

Similarly, we note that $p(\mathbf{z})$ is the same as integrating ϕ out of the joint $p(\mathbf{z}, \theta) = p(\mathbf{z}|\theta)p(\theta)$, but just as before we have a Multinomial-Dirichlet conjugacy and we can procede in the same manner as before to obtain

$$
P(\mathbf{z}) = \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \prod_{d=1}^{D} \frac{\prod_j \Gamma\left(n_j^{(d)} + \alpha\right)}{\Gamma\left(n^{(d)} + T\alpha\right)},
$$

The authors then jump to provide the form of the full conditional for $p(z_i|\mathbf{z_{-i}}, \mathbf{w})$ such that

$$
P(z_i = j \mid \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,n}^{(d_i)} + T\alpha}
$$

Where at first glance we can notice a blend of the forms from $P(\mathbf{w}|\mathbf{z})$ and $P(\mathbf{z})$ on a univariate-like scale as we might expect since the full conditional is proportionate to the joint, and the joint is equal to the product of $P(\mathbf{w}|\mathbf{z})P(\mathbf{z})$. Given all of the $\mathbf{z_{-i}}$ are constants, we note that many terms / counts can be cancelled out through the proportionate sign, however we see information from the words still included. Of course, this would make sense as we are thinking of the words as a likelihood function within our

Bayesian machinery. Moreover, we note that the jth marginal of a Dirichlet distribution is a Beta distribution, and in this vein our jth topic full conditional might be seen as proportionate to the product of two Beta distributions (where again, we have cancelled out terms through the proportionate sign). Note that $n_{-i}^{(.)}$ is the count that does not include current assignment ot z_i. The authors note that this is actually a relatively effecient calculation since we just have to sum a relatively small number of nonzero counts.

Moreover, the full conditional has an intuitive nature such that the left term can be interpreted as the probability of w_i under topic j, and the right term can be interpreted as the probability of topic j in document d_i. Of course, once our chain has converged, we can take our samples to be independent samples of the posterior topic marginals. Despite the theoretical development provided by Griffiths and Steyvers, we employ the equivalent algorithm developed by Porteous et. al. as the terms are slightly more intuitive.

Lastly, Griffiths et. al. then discuss the estimation of θ_j and ϕ_j. The authors note their estimates

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)}+\beta}{n_j^{(\cdot)}+W\beta}$$
$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)}+\alpha}{n^{(d)}+T\alpha}$$

correspond to the predictive distributions over new words w and new topics z, conditioned on $\mathbf{w}$ and $\mathbf{z}$, but leaves it for the reader to infer how. It's not particularly clear what they mean, but we postulate they could be describing finding the mean of the posterior predictive draws. If so, we illustrate the idea noting that $p(w_{+1}|\mathbf{w})$ is the same as integrating $\mathbf{z}$ out of $p(w_{+1}, \mathbf{z}|\mathbf{w})$ which is the same as integrating out $\mathbf{z}$ from $p(w_{+1}|\mathbf{z}, \mathbf{w})p(\mathbf{z}|\mathbf{w})$. Since we have the samples from $p(\mathbf{z}|\mathbf{w})$, and since we can presumably sample from w_{+1} now that we are conditioning on $\mathbf{z}$, we can apply Monte Carlo sampling to approximate the posterior predictive distribution for w_{+1}. From here we can take the average of these samples to represent the estimate of the posterior predictive mean as $\hat{\phi}$. Of course, the estimates provided above are with respect to the jth $\hat{\phi}_j$ which requires essentially the same procedure. Note that we can follow the same method to obtain the estimate $\hat{\theta}_j$.

Backtracking in time, Blei et. al. also ran into issues of intractability and resorted to variational bayes, whereas others have resorted to expectation propogation. We still note that Blei et. al. discuss a key component surrounding the concept of exchangeability when calculating $p(\mathbf{w}, \mathbf{z})$. These authors claim that since the topics are infinitely exchangeable within a document in LDA, then by de Finetti's representation theorem, the joint distribution of the topics conditioned on θ are then *independent and identically distributed*, such that since

$$p(z_1, \ldots, z_N) = p\big(z_{\pi(1)}, \ldots, z_{\pi(N)}\big)$$

then

$$p(\mathbf{w}, \mathbf{z}) = \int p(\theta)\left(\prod_{n=1}^{N} p(z_n \mid \theta)p(w_n \mid z_n)\right)d\theta$$

In other words, why we are able to express the joint as a product in the manner presented above. This notion of exchangeability and di Finetti's theorem was not mentioned within Griffiths, but we thought it important to discuss as it was a critical tool in developing the LDA model within Blei et. al. from which Griffiths et. al. built upon.

Clearly, the approach in Griffiths et. al. provides a huge speed up as we can integrate out the latent variables and simply sum a relatively small number of nonzero counts in our quest to obtain posterior samples. However, one of the disadvantages to this algorithm is the number of full conditionals that need to be sampled. Following the work from Griffiths et. al. we note that Porteous et. al. have developed a way to "reduce the inner-loop" or reducing the number of K topics to sample from to a much smaller number of full conditionals. The argument is based on the notion that "for any particular word and

document, the sampling distributions of interst are frequently skewed such that most of the probability mass is concentrated on a small fraction of the total number of topics K." The authors provide an alogorithm "Fast LDA", which takes advantage of this concentrated probability notion mentioned above.

We draw slightly on the works of Porteous et. al., noting that the authors claim that this concentration occurs "for most real data sets after several iterations of the Gibbs sampler", and note their experiment with the New York Times Corpus, and the very different PubMed Corpus. Despite the very different nature of the Corpora, the findings were quite similar as illustrated below

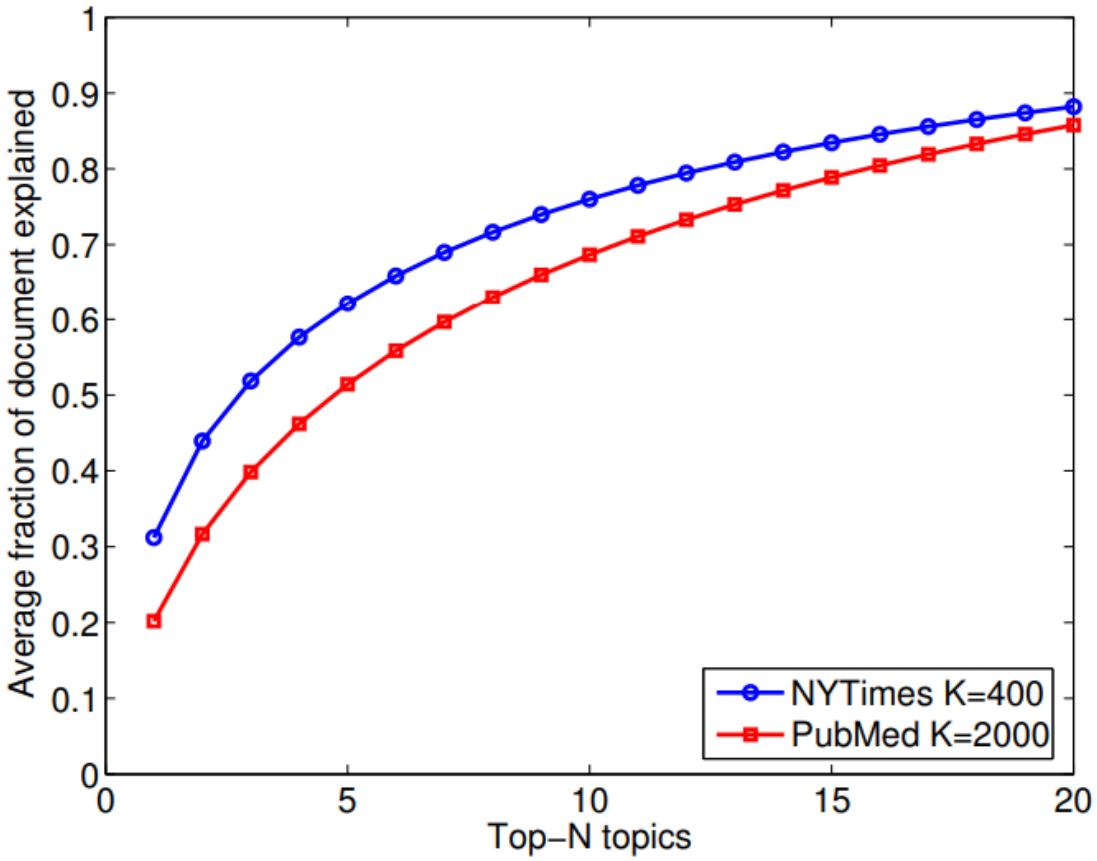


Figure 4: Average fraction of a document explained by top-20 topics, for NYTimes (K=400 topics) and PubMed (K=2000 topics). We see that, on average, the top-20 topics in any document account for approximately 90% of the words in the document.

We note that while our model draws slightly on Porteous et. al., we are manually setting our topic number which may not be the most advantageous aspect of our model. One might look into the BNP approaches that allow for allow for the number of topics to grow. Moreover, we still have to sample each topic whereas the Porteous et. al. model is drastically faster since it does not have to do this (and peforms even better with even larger K).

For Griffiths et. al. the purpose of the algorithm was centered on the title of the paper. That is, they were focused on "Finding Scientific Topics". Griffiths et. al. used Bayesian model selection to establish the number of topics (where as we choose 20), and they illustrated that the extracted topics from a corpora of abstracts from PNAS was capable of providing meaningful structure in the data consistent with the class designations provided by the authors of the articles. This is quite useful for experts as a "first-order approximation to the kind of knowledge avialable" within a corpora of documents. In other words, it can help scientists decide whether or not to invest valuable time investigating a particular corpora.

In a similar vein, we might also like to apply the same approach to single papers. Knowing in advance what topics largely explain a particular topic can also save time.

Known possible applications also include identifying "hot topics" as Griffiths and Steyvers point out by examining dynamics and tagging abstracts to illustrate semantic content.

These are concepts that could help anyone hoping to better expend their valuable research time. In this vein, whatever the topic area might be, it might be wise to begin the research process by first running an LDA model on a few relevant corpora before digging in one's specified research.

LDA methods are of particular relevance to genomic research more broadly.

3. Algorithm

3.1. High Level Description

We now switch gears and move towards Porteous et. al. where we find the description of the collapsed Gibbs sampling algorithm to have a slightly nicer representation.

The general goal is to generate samples from posterior topic marginals through collapsed Gibbs sampling in order to then find estimates of $\hat{\phi}_{wk}$ and $\hat{\theta}_{kj}$ where the former is the probability of of word w occuring in topic k and the later is the probability of topic k occuring in the jth document. We attempted to elaborate on the commentary of predictive distributions very briefly stated in Griffiths et. al., whereas no elaboration was provided in Porteous et. al. That said, the forms provided clearly provide a gateway into their underlying intuition as we are weighting in the prior α and β respectively in combination with the counts of interest to update the underlying vector components of the parameters generating words and topics. Clearly the updated parameters would need to blend these new counts and priors.

In general, the first idea behind the Porteous et. al. formulation was to underscore the idea of reinforcing a certain "stickyness" for similar topics within the same document - i.e. to place a higher probability on a new word having an assignment to a topic already popular within the same document.

The second idea was to encourage a "stickyness" for words belonging to the same topic - i.e. words often assigned to the same topic should have more probability of being assigned to said topic (for new instances of the same word).

In sum, the overal goal was getting samples from the posterior marginals via collapsed Gibbs sampling, and in the process utilize counts to estimate the latent variables of interest. We spell out the specifics below.

3.2. Detailed Description

A full run down of the notation is provided below:

- $x_{ij} = \mathbf{x} =$ set of all words in doc j
- $N_{wkj} = \#\ \{i : x_{ij} = w, z_{ij} = k\}$
- $N_{kj} = \Sigma_w N_{wkj} = \#$ times a word in doc j assigned to topic k
- $N_{wk} = \Sigma_j N_{wkj} = \#$ times word w is assigned to topic k
- $N_k = \Sigma_w \Sigma_j N_{wkj} = \#$ times topic k appears in the corpus

and the full conditional topic posterior is described as:

$$p\big(z_{ij} = k \mid \mathbf{z}^{\neg ij}, \mathbf{x}, \alpha, \beta\big) = \frac{1}{Z} a_{kj} b_{wk}$$

where

- $a_{kj} = N_{kj}^{\neg ij} + \alpha$
- $b_{wk} = \frac{N_{wk}^{\neg ij} + \beta}{N_k^{\neg ij} + W\beta}$
- $Z = \sum_k a_{kj} b_{wk}$

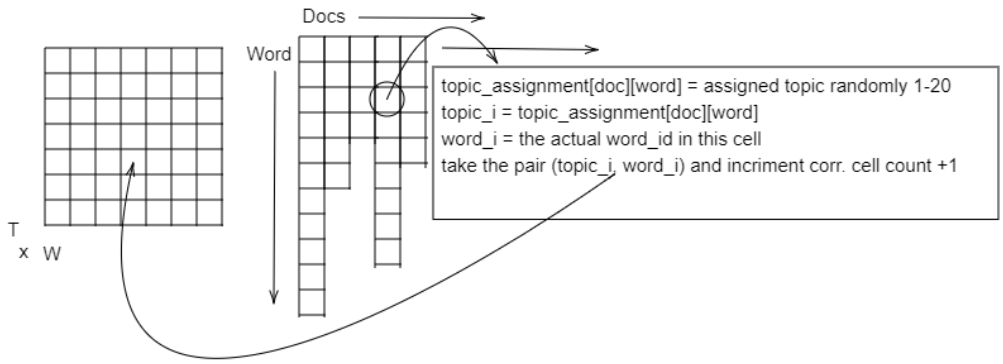
where Z ensures that our posterior topic multinomial still has a probability vector summing to one.

Before moving onto the sampler, we should discuss *** (tokenizing etc...)

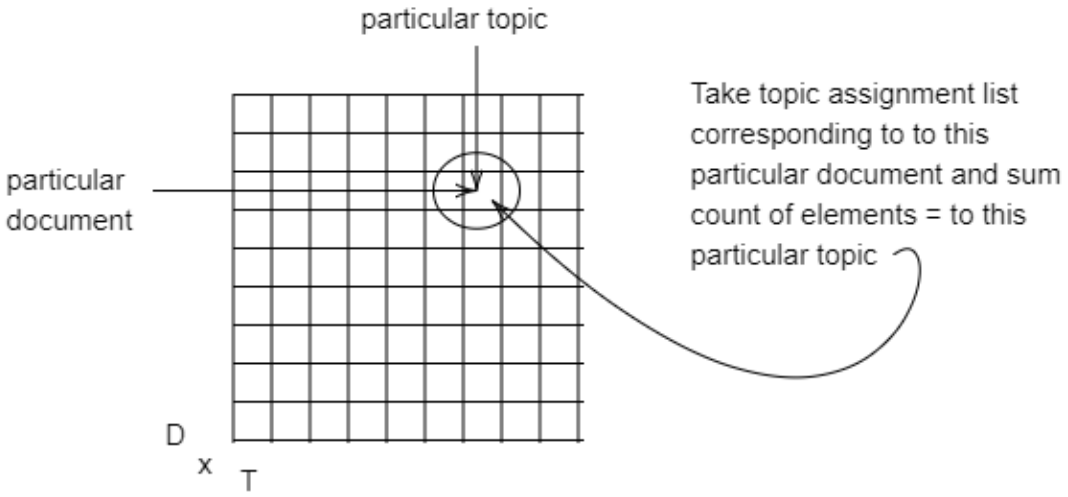
Now we discuss the algorithm. Before running the sampler, we want to note the structure of the "word-topic count matrix" and the "document-topic count matrix".

- T is total topics
- W is total words

Here is a diagram of how the **word − topic** count matrix works (left image)



Whereas the **document − topic** matrix functions as:



and the algorithmic details are such that:

Algorithm 1: Setting up

Result: pre-Gibbs preparation
initialization;
- clean data (lower case, remove punctuation etc.)
- docs ← list of lists: sublists correspond to document-specific words
- assign an id to each word:
vocabulary ← empty set
i ← 0
for *d in documents* **do**
  for *word in document* **do**
    if *word in vocabulary* **then**
      continue
    end
    vocabulary[*word*] ← i
    i ← i + 1
  end
end
docs id ← replace all words with their id's
ta ← list of lists: sublists correspond document word-topic assign.
W ← number of words in vocabulary
D ← number of documents
T ← number of topics = 20 **change to K**
- Create **word − topic** count matrix:
wt ← T by W empty matrix
for *document in D* **do**
  for *word in document* **do**
    ta[document][word] ← random choice 1-20
    ti = temp topic id ← ta[document][word]
    wi = temp word id ← docs id[document][word]
    wt[ti, wi] ← +1
  end
end
- Create **document − topic** count matrix:
dt ← D by T empty matrix
for *document in D* **do**
  for *topic in T* **do**
    dt[document,word] ← Σ#{elements of ta[document] == t}
  end
end

Algorithm 2: Collapsed Gibbs Sampling

Result: full conditional topic samples
initialization;
$\alpha = 1$ $\beta = .001$ iterations = ?
RUN SAMPLER : iterate *1000(?)* times through the chain:
for *i in iterations* **do**
  for *document in D* **do**
    for *word in document* **do**
      initialize topic assignment to token w:
      t0 ← ta[d][w]
      token id:*
      wid ← docs id[document][word]
      remove token w (when sampling for token w):*
      dt[document, t0] ← dt[document,t0] - 1
      wt[t0,wid] ← wt[t0,wid] - 1
      $p_{dt} \leftarrow \frac{(dt[document,:]+\alpha)}{\Sigma dt[document,:]+T*\alpha}$
      $p_{wt} \leftarrow \frac{(wt[:,wid]+\beta)}{wt.sum(axis=1)+W*\beta}$
      - where:
      unnormalized full conditional $p(z)_{ij}^{unnormalized} \leftarrow p_{dt} * p_{wt}$
      - and:
      $p(z)_{ij} \leftarrow \frac{p(z)_{ij}^{unnormalized}}{\Sigma p(z)_{ij}^{unnormalized}} = \frac{a_{kj}b_{wk}}{\Sigma_k^K a_{kj}b_{wk}}$
      - draw topic for word n from multinomial using probabilites just calculated:
      t1 ← np.random.choice(range(T),p = $p(z)_{ij}$
      ta[document][word] ← t1
      - re-increment with new topic assignment:
      dt[document,t1] ← dt[document,t1] + 1 wt[t1,wid] ← wt[t1,wid] + 1
    end
  end
end

Lastly, we can estimate the parameters of interest:

Algorithm 3: Estimation

Result: obtain estimates
$\theta \leftarrow \frac{(dt+\alpha)}{(dt+\alpha).sum(axis=1[:,None]}$
$\phi \leftarrow \frac{wt+\beta}{(wt+\beta).sum(axis=1[:,None]}$
- # words in each topic:
$\phi_{\textbf{rounded}} \leftarrow round(\phi)$
for *t, topic* **in** *enumerate($\phi_{rounded}$)* **do**
  print("Topic .....???....")
end
- # topics in each document
...

(next page)

Where we ultimately have obtained estimates:

$$\hat{\phi}_{wk} = \frac{N_{wk} + \beta}{N_k + W\beta}$$
$$\hat{\theta}_{kj} = \frac{N_{kj} + \alpha}{N_j + K\alpha}$$

4. Optimization for Performance

5. Applications to Simulated Data Sets

6. Applications to Real Data Sets

7. Comparative Analysis with Competing Algorithms

8. Discussion / Conclusion

9. References / Bibliography

[1] Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003) J. Machine Learn. Res. 3, 993-1022.

[2] T. L. Griffiths and M. Steyvers. Finding scientific topics. Proc Natl Acad Sci $USA, 101$ Suppl $1 : 5228 - 5235$, April 2004.

[3] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages $569 - -577$, New York, NY, USA, 2008 . ACM. 8

[4] F. Zhao, X. Ren, S. Yang, Q. Han, P. Zhao, and X. Yang, "Latent dirichlet allocation model training with differential privacy," IEEE Transactions on Information Forensics and Security, vol. 16, pp. $1290 - 1305, 2021$.

10. Appendix

In our original plans, we hoped to make this algorithm differentially private. However, limited time and certain issues within the algorithms forced us to abandon this angle for now.

We refrain from a deeper discussion of differential privacy, but note how Zhao et. al.[4] refer to the definition as a randomized mechanism $\mathcal{M} : D \to Y$ such that for any neighboring datasets D, D' that differ by only one record, and for any output $S \subseteq Y$ there is

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \cdot \Pr\left[\mathcal{M}\left(D'\right) \in S\right]$$

constitutes achieving ϵ-differential privacy, where ϵ controls the privacy level (this is the common definition). The authors then proceed to illustrate how to achieve ϵ-differential privacy wthin a collapsed Gibbs sampler by injecting Laplacian noise on each iteration (also protecting against word count leakage), and by exploiting the fact that collapsed Gibbs sampling has some inherent privacy (helping to protect sampled topics). The full set up is

α, β	hyper-parameters for LDA
$\mathcal{W}$	word space
$\mathcal{K}$	topic space
ϕ_k	topic-word distribution for topic k
ϕ_k^t	probability that word t is generated by topic k
n_m^k	count of words with topic k in document d_m
n_k^t	count of word t with topic k in corpus D
n_t	total word count of $t \in D$
p_k	probability that topic k is sampled
w_r	replaced word
$\mathbf{w}^+$	set of related words
$\mathbf{w}^-$	set of unrelated words
ϵ_i^r	inherent privacy loss on w_r in the i -th iteration
$\epsilon_i^t, \epsilon_i^{t'}$	inherent privacy loss on related words t or t'

Where the algorithm is below

Algorithm 1: HDP-LDA

Input: Document corpus D, Prior parameters α, β, Topic number K, Clipping bound C

Output: Trained topic-word distribution Φ, Privacy loss $\epsilon = T \cdot (\epsilon_L + \epsilon_I)$

// **Initialization**

1 **for** $d_m \in D$ **do**
2 **for** $w = t \in d$ **do**
3 Sample topic: $k \sim Mult(\frac{1}{K} \cdot \mathbf{I}_K)$;
4 Initialize word counts n_k^t and n_m^k;
5 **end**
6 **end**

// **Collapsed Gibbs Sampling**

7 Set $iter = 0$;
8 **while** $iter < T$ **do**
9 Add noise to each n_k^t independently: $n_k^t \leftarrow n_k^t + \eta, \quad \eta \sim Lap(2/\epsilon_L)$;
10 **for** $d \in D$ **do**
11 **for** $w = t \in d$ **do**
12 Clip: $(n_k^t)^{temp} \leftarrow \min\{n_k^t, C\}$;
13 Compute sampling distribution $\mathbf{p}$: $p_k \propto \frac{(n_k^t)^{temp} + \beta}{\sum_{t=1}^{V}(n_k^t + \beta)} \cdot \frac{n_m^k + \alpha}{\sum_{k=1}^{K}(n_m^k + \alpha)}$;
14 Compute inherent privacy loss: $\epsilon_I \leftarrow 2 \log(\frac{C}{\beta} + 1)$;
15 Sample topic and update word count n_k^t;
16 **end**
17 **end**
18 $iter \leftarrow iter + 1$;
19 **end**
20 Compute the trained model Φ;

where noticably we see a difference in line 9 where Laplacian noise is added, and line 12 and 14 concerning clipping and inherent privacy loss (on each iteration, summed in total = ϵ_I). Testing utilized defending against *Topic-based Attacks* below.

Algorithm 3: Topic-based Attack Algorithm

Input: The attacked word w_{mn}
Output: The inferred result $\bar{t}$

Set $i = 0$;
while $i < T$ **do**
  Add noise: $(\hat{n_k^t})_i \leftarrow n_k^t + \eta, \quad \eta \sim Lap(2/\epsilon_L)$;
  Record $(\hat{n_k^t})_i$;
  for $w \in D$ **do**
    Sample topic: $k_i \sim \mathbf{P}$;
    if $w = w_{mn}$ **then**
      Record k_i;
    end
  end
  $i \leftarrow i + 1$;
end
Compute $\bar{t} = \arg\max_t P[w_{mn} = t | k_1, ..., k_i, ..., k_T]$ acc. to Equation (24);

Where:

$$\begin{aligned}
&\Pr(w_{mn} = t \mid k_1, \ldots, k_i, \ldots, k_T) \\
&= \tfrac{\Pr(w_{mn}=t, k_1, \ldots, k_i, \ldots, k_T)}{\Pr(k_1, \ldots, k_i, \ldots, k_T)} \\
&= \tfrac{\prod_i \Pr(w_{mn}=t, k_i)}{\prod_i \Pr(k_i)} = \prod_i \tfrac{\Pr(w_{mn}=t, k_i)}{\Pr(k_i)} \\
&\approx \prod_i \frac{\left(n_{k_i}^t\right)_i / \sum_{k,t} \left(\hat{n}_k^t\right)_i}{\sum_t \left(n_{k_i}^t\right)_i / \sum_{k,t} \left(\hat{n_k}^t\right)_i} = \prod_i \frac{\left(n_{k_i}^t\right)_i}{\sum_t \left(n_{k_i}\right)_i}
\end{aligned}$$

And while the above may seem relatively reasonable, how the authors arrived at their clipping bound C is slightly unclear. However, they did claim that for for a small number of topics, we can let C be n_t, the total count of word t in D. Moreover, the use of $Lap(2/\epsilon_I)$ is slightly confusing since one of the parameters is implicit, but not explicitly mentioned in the surrounding text. If the assumption is that $\mu = 0$, which seems reasonable since ϵ_I would likely control the overall noise injected into the counts, then it is possible to sample negative values when adding this to the original counts. We believe these negative counts contribute to a negative probability vector for the topic multinomial. To circumvent this, we thought of shifting the Laplace right, taking the absolute value of the noise, drawing extra counts from a Poisson distribution (which also seems more in line with adding noise to a discrete variable), however, everything broke our algorithm when we adjusted the counts via adding noise. Of course, this was a good experience to build some familiarity with techniques capable of altering algorithms to satisfy differential privacy definitions.