Latent Dirichlet Allocation: Topic Modelling on the 20 Newsgroups Corpus

Kevin Yan kevin.yan@mail.utoronto.ca
Department of Statistical Sciences
University of Toronto
Lawrence Toh lawrence.toh@mail.utoronto.ca
Department of Statistical Sciences
University of Toronto

1 Introduction

The ability to properly classify or represent documents, whether it be emails, websites, or papers, is extremely important in the information age. Having proper classifications will allow us to better organize, and search for relevant documents. In addition, it's also important for machines to be able to do this task automatically in an unsupervised fashion because there is a large amount of documents, and performing the task manually is impractical. We also rarely know what the possible topics are in a given corpus/collection, therefore it is essential for the machine to learn these topics itself. In the Latent Dirichlet Allocation (LDA) model, we assume that each document is constructed by an amalgamation of topics. We also make assumptions on the number of topics (K), that we believe to be present in our corpus. As an output, LDA generates a distribution of those K topics, in an unsupervised fashion. We will experiment with the model on the 20 Newsgroups dataset.

2 Latent Dirichlet Allocation

2.1 The model

The Latent Dirichlet Allocation model (LDA) is designed to represent the generative process in which documents in a given corpus/collection are constructed. The underlying assumption in LDA, is that each word in a given document comes from a different topic. The variant of LDA that we will use in the paper is a fully Bayesian LDA model. α and β are the only non-stochastic parameters in the model, and in general can be fitted during the M-Step of the EM algorithm, however these will be chosen instead. Φ is a vector with V elements, where V is the size of the vocabulary, and they represent the probabilities of each word in the vocabulary (there is one ϕ_k for each of the k topics). Each ϕ_k is going to be the same for all documents. Θ are the topic probabilities for each document where θ_m represents the topic distribution for the m-th document. The main components in this model are $W_{m,n}$ *and* $Z_{m,n}$ which represent the n-th word in the m-th document, and the topic associated with that word respectively. Notably, only the words in all the documents are observed, thus it's the only shaded node in the model.

In order to generate the n-th word in document m, we first generate the topic-word probabilities for the entire corpus: $\phi_k \sim Dir(\beta_k)$ for all k. Then, we select the topic distribution for that particular document by $\theta_m \sim Dir(\alpha)$. For a particular word, we first choose the relevant topic $Z_{m,n} \sim Multinomiall_K(\theta_m)$, and then we choose the word. Assuming $Z_{m,n} = k$, we choose $W_{m,n} \sim Multinomiall_V(\phi_k)$. Thus, we see that each word in a document could originate from a different topic entirely, under the LDA model. This is illustrated in Figure 1.

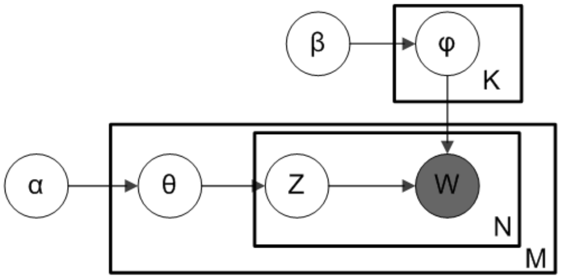


Figure 1: Fully Bayesian LDA model.

2.2 Inference Overview

We assume that α *and* β are hyperparameters, which are specified. Most importantly, after observing the documents, we want to make some inferences about the model parameters. In particular, we like to know, which topic did each word most likely come from (eg: $P(Z_{m,n} = k| W))$, what the distribution of topics is for a particular document (eg: the MAP estimate of $P(\theta_m | W)$) , and lastly, what the most popular words are for a particular topic (eg: MAP estimate $P(\phi_k | W))$.

The importance of finding a good posterior estimate of θ_m for each document is that we will take these estimates as latent representations of the model. We are then able to use these representations for multiple tasks, like plotting/visualizing documents, as well as fitting a classifier to predict the topic of new documents.

The importance of finding a good posterior estimates of ϕ_k is that they will provide us with information on the range of topics that are present in our corpus. For the model, we set the number of topics based on our qualitative judgment, despite not knowing what these topics are. To uncover them, we will look for the most probable words for a particular topic, meaning the words with the highest values in ϕ_k , and this will give us an idea of what topic k is all about.

2.3 Performing Inference

The two methods that we will be using to perform inference in this model are Variational Bayes, and Monte Carlo Markov Chain. We will be using two LDA packages in order to make inferences using these two methods, for Variational Bayes, we'll being the Gensim package, and for Gibbs sampling, we'll be using the LDA package in python.

The Gensim package uses an online approach to Variational Bayes [2]. In all variational inference techniques, we want to find a way to approximate the posterior distribution of the latent variables given the observed variables. This is the case because it's not analytically feasible, or computationally expensive to calculate the true posterior, so we approximate it with a new distribution q (equation 1), which minimizes the KL divergence between the two distributions.

$$q(\theta, \phi, Z) = \prod_k q(\phi_k) \cdot \prod_m q(z_m) q(\theta_m) \qquad (1)$$

$$\textit{where}\ \ q(z_m) \text{\textasciitilde} \textit{Multinomial}\ (\eta_m), q(\theta_m) \text{\textasciitilde} \textit{Dirichlet}(\gamma_m),\ q(\phi_k) \text{\textasciitilde} \textit{Dirichlet}(\lambda_k)$$

We can then use q, in order to make point estimates of quantities discussed in 2.2.

The LDA package uses collapsed Gibbs sampling to make inferences on the attributes we are interested in [3]. Under this approach we marginalize out θ *and* ϕ from the joint distribution, and use Gibbs sampling to sample from $P(Z|W)$. The resulting samples are used to make point estimates of θ_m and φ_k (2), given z *and* w (*ie* : *using samples from the posterior* $P(z|w)$).

$$\hat{\varphi}_{k,v} = (N_{vk} + \beta)/(N_k + V \cdot \beta) \ , \quad \hat{\theta}_{m,k} = (N_{m,k} + \alpha)/(N_m + K \cdot \alpha) \qquad (2)$$
$$\textit{where } N_{v,k} = \# \textit{ of times word v has been assigned to topic k}$$
$$N_{m,k} = \# \textit{ of times a word in document m has been assigned to topic k}$$
$$N_k = \textit{total \# of words assigned to topic k, and } N_m = \textit{total number of words in document m}$$

This shows that we can find all the assignments, N’s, by using the samples we got from the posterior of Z’s, and these results can be found in [3], and [4].

3 Experiment

For this paper, we will be comparing the effectiveness of the LDA model on the 20 Newsgroups data set. We will look at the topics that are generated by the LDA model, and use features generated by the model in order to plot and visualize the LDA representations of documents in 2-d space. We will also fit logistic regression classifiers using these features. This allows us to access the learned features’ relevance in predicting the correct topic for each document. Moreover, we will perform these tasks using both sets of features, derived by the inference techniques described in section 2.3. Indirectly, we are comparing the effectiveness of the algorithms, as well as the LDA model.

3.1 Data

The data that we have decided to use for this paper is the 20 Newsgroups data collected by Ken Lang. The data contains 20,000 messages/posts from 20 different newsgroups, each relating to a different subject. This particular dataset was chosen mainly because of its popularity in the text/topic modelling community. Moreover, there are labels that indicate the subject of the messages. This allows us to be able to perform classification, and then analyze the relevance of the topics generated by LDA. Lastly, an assumption of the LDA model is that each document is generated based on a distribution of topics. Hence, it is important that some documents in our data are relevant to several subjects in the corpus, something that is reasonably evident in the 20 Newsgroups corpus.

3.2 Pre-Processing

Data was pre-processed using functions available in the scikit-learn library in python. As the 20 Newsgroups data was available in scikit-learn, we were able to use the function fetch_20newsgroups to import the dataset without headers, footers, or quotes. Next, we had to generate a document-term matrix (a size $M \cdot V$ matrix, describing the frequency of each word in a document) as input to the LDA model. We excluded all popular English stop-words from the vocabulary. These English words are repeated frequently in the documents, and uninformative in topic modelling. Examples of such words are ‘the’, ‘he’, ‘she’, ‘there’, ‘those’, etc. Next, words that occurred overly frequent in each document were removed. The frequency threshold chosen was 0.7, meaning if a word occurred in more than 70% of the documents, we would remove it and deem it irrelevant. The exclusion of words was done using the function CountVectorizer, which returned a sparse-matrix representation of the document term matrix. We also chose to use a limited vocabulary size of 60,000, out of the 120,000+ words in the actual corpus. This was chosen to make the model simpler and easier to train. It was also our belief that some words were incredibly rare, and were not incredibly useful in discovering topics, thus we only used the 60,000 most frequent words in the corpus, after the frequency threshold reduction mentioned earlier.

3.3 Process

After processing the data, we perform inference on the dataset. We will perform all the following tasks using both inference algorithms. Firstly, we will fit LDA models using different number of topics (K=6, 20, 25). These values were chosen because there are 6 general topics in which the 20 newsgroups can be categorized. 25 was arbitrarily chosen in order to see the effect of increasing the number of topics. Secondly, each document will then have a latent representation, which will be the estimate of θ_m, which is the estimate of the distribution of topics for the particular document (which is a 1 x K vector of probabilities that sum up to 1). We will then use PCA to reduce the dimensionality of this "latent representation" to 2 dimensions. This enables us to visualize the representations. Thirdly, we will fit logistic regression classifiers to assess the relevance of the representations discovered by LDA in predicting the correct topic for each document. Overall, the results for both inference algorithms will be compared

4 Results

Initially, the hyperparameters of the model α *and* β were set to the default values used by the LDA package, 0.1 and 0.01 respectively for α and β. This ensures consistency between both inference algorithms. Also, we do not have any prior beliefs on the topic distributions, or the distributions of words in any topics. Therefore, we assumed that the default values would be a good starting point. Also, the hyperparameters are treated as scalars. This is consistent with the papers we referenced, which assumed that the Dirichlet parameters are equal and multiplied by a scalar [1,2,3]. As our goal is to compare the performance of the two inference algorithms, it was difficult to perform cross validation on all parameters, because the parameters would need to agree for both algorithms. Given this goal, we believe it makes sense to choose α, *and* β as mentioned.

4.1 PCA Plots

The number of features generated by the LDA model correspond to the number of topics that have been specified in the model (K). Specifically, the features are estimates of $\hat{\theta}_m$ *for all documents m*, which we can interpret as the estimate of the distribution of topics for document m, provided that we've observed the corpus. In order to visualize these representations, principal component analysis is performed to reduce the dimensions of the LDA-derived features to 2 components. The first and second principal components are plotted on scatter plots for models with 6, 20 and 25 topics. Plots were generated using features discovered using both Gibbs sampling, and Variational Bayes.

Observing the plots, we're able to discern some separation between the 6 main topics, regardless of the amount of features we have used in our model. However, it appears that increasing the number of topics has improved the separation in the space of spanned by the first two principal components. Moreover, it appears that separation is better when using Gibbs sampling as an inference technique compared to Variational Bayes. Although the separation isn't entirely perfect, we can attribute this mainly to the fact that we have projected onto a different dimensional space, and so it would not necessarily be the case that the structure in the original feature space holds in this subspace. (Note: in Figure 2-7, blue = religion, green=computers, red=sports, magenta=science, yellow = for sale, and finally, black=politics).

Moreover, for the 6 topic case, we compared the topics discovered by the model using both the Variational Bayes, and Gibbs sampling techniques. We used our posterior estimates of ϕ_k *for all k*, in order to find the most probable words per topic. Remember that ϕ_k represents the probability distribution of all words in our vocabulary conditional on the k-th topic. Thus, we use our posterior estimate of these quantities to find the 5 most probable words for each topic k. These words will be used to manually identify the most relevant topic.

Graphs corresponding to Gibbs Sampling:

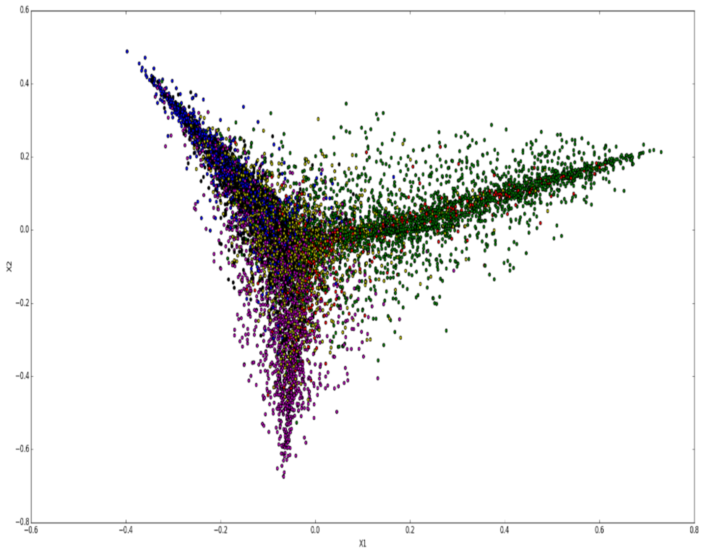


Figure 2: 25 topics, Gibbs Sampling

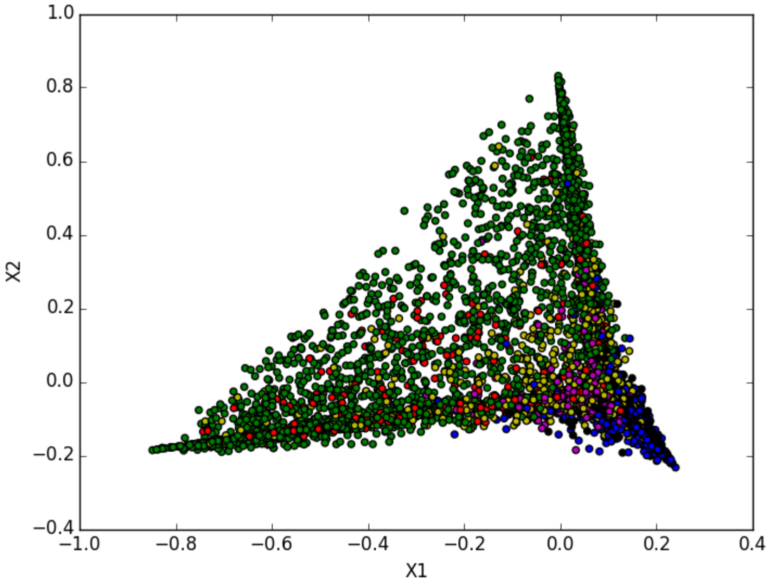
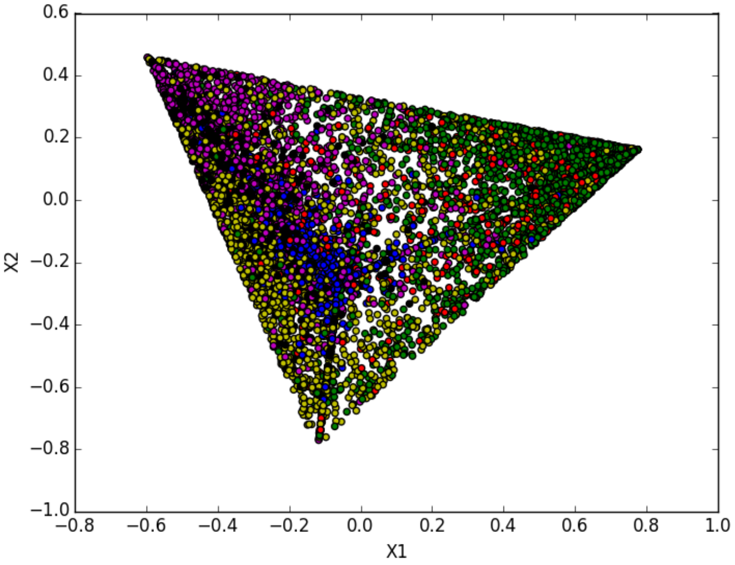


Figure 3: 20 topics Gibbs Sampling



s

Figure 4: 6 Topic, Gibbs Sampling

Graphs corresponding to Variational Bayes:

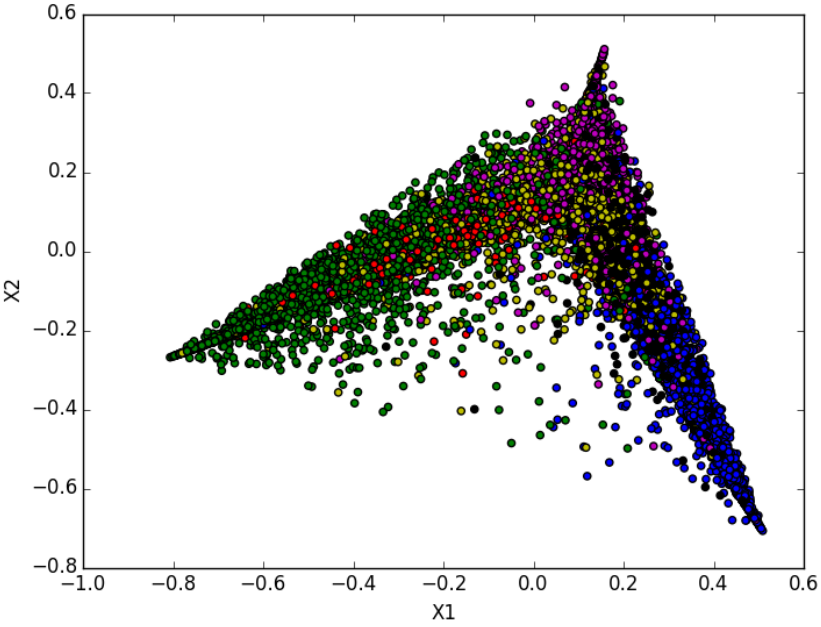


Figure 5: 25 topics, VB

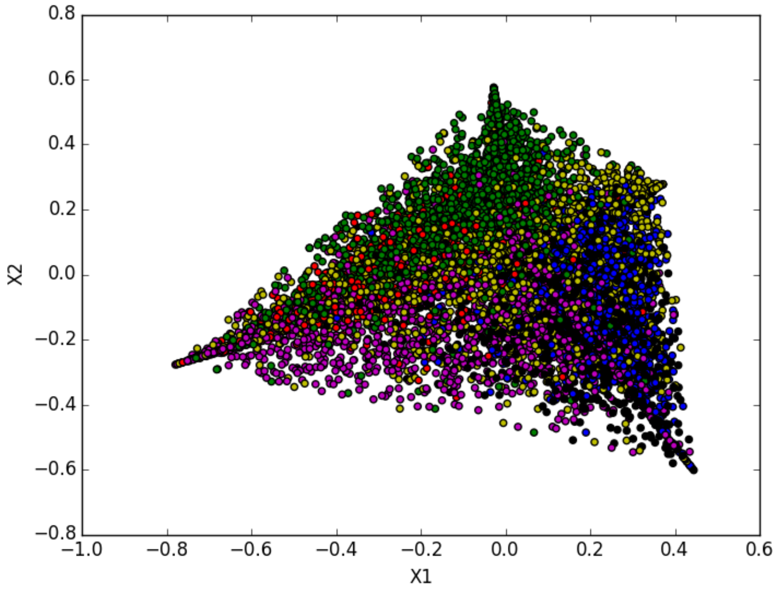


Figure 6: 20 topics,VB

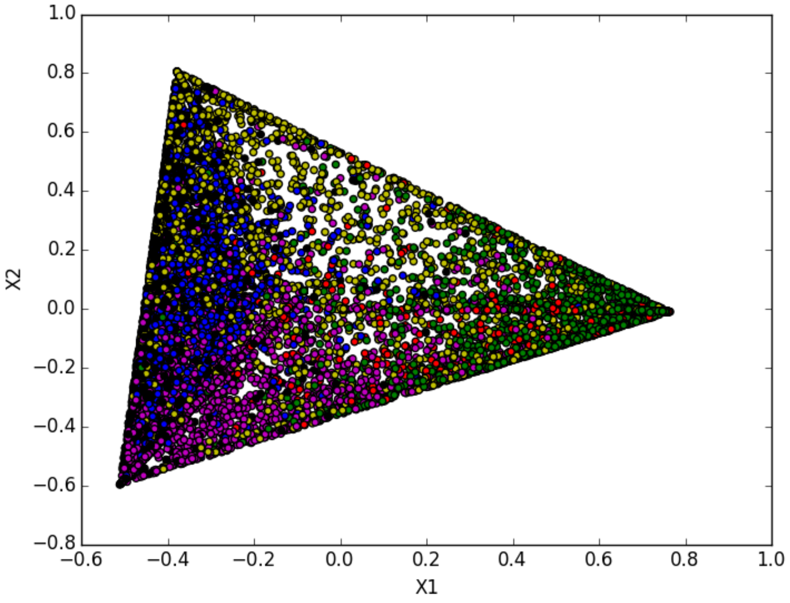


Figure 7: 6 Topics, VB

Variational Bayes (Gensim)
Topic 1: people, said, right, car, gun (Politics)
Topic 2: god, people, think, does, say (Religion)
Topic 3: use, edu, file, windows, com (Computers)
Topic 4: key, space, use, government, public (Science)
Topic 5: 10, 00, 25, team, game (Sports)
Topic 6: ax, max, g9v, b8f, a86 (Computers)

Gibbs sampling (LDA)
Topic 1: 10 00 year 20 15 (Sports)
Topic 2: edu windows file available data (Computers)
Topic 3: db edu power turkish second (Politics)
Topic 4: god jesus believe true read (Religion)
Topic 5: government going mr president ll (Politics)
Topic 6: ax max g9v b8f a86 (Computers)

Notice that both sets of words, derived using different inference algorithms, are quite similar. However, it appears that the Variational Bayes method was better able to capture the topics than GIbbs Sampling. We see that Gibbs Sampling repeated both computers and politics, thus identifying 4 out of 6 topics, where as variational bayes was able to capture 5 out of 6, and only repeating computers. The repetition in computers is more understandable as there were relatively more subgroups involved with computers. Thus, it appears that variational bayes was better able to uncover the correct topics underlying our corpus. By increasing the number of topics, we were able uncover a wider range of topics, which became more specific towards the subtopics.

4.2 Logistic Regression Classification

For classification purposes, the corresponding labels of the documents are mapped to a set of 6 topics (ie: each subtopic is grouped into its parent group, of which there are 6 in total). Based on the original labels, it is intuitive to map the wide array of topics to just these 6 main topics. The plots in the prior section and Logistic Regression in this section make use of the grouped topics.

The features, derived from the LDA model, are then used to fit logistic regression models for classification. The following table shows the training accuracy for both methods of inference. It can be observed that features from Gibbs sampling results in lower training error, compared to the Variational Bayes method when the number of topics is are 20, and 25. However the opposite holds true when the number of topics is 6. We can also see that increasing the pre-defined number of topics for the LDA model results in better features for the logistic regression to predict the documents' topic with higher accuracy. Thus this shows that in general, the features derived from Gibbs Sampling scale better than those of VB when we increase the quantity of features. In other words, it seems to indicate that Gibbs Sampling is better able to estimate the distribution of topics for a given document when we include more topics in our model, and may be the better choice for more complex LDA models. This is quite possible because Variational Bayes may make over simplistic assumptions about the variables/stochastic values in the model, and the errors from the simplistic assumptions may accumulate as we increase the complexity of the distribution we're estimating by increasing the number of topics K. Additionally, we varied the vocabulary size of the model in order to see how well the algorithms responded to an increased size of $\phi's$. It seemed like a reduced vocabulary size (ie: removing less common words in the corpus), increased the performance of the LR model. This indicates that including rarer words into the vocabulary, and having to make inferences on them (hence further spreading out the probabilities to these words in each ϕ_k) has resulted in poorer classification accuracy. However, regardless of vocabulary size, the relationship between results for Gibbs Sampling and Variational Bayes remain the same.

Vocabulary Size	Number of Topics	Training Accuracy		Testing Accuracy
		Gibbs Sampling	Variational Bayes	Variational Bayes
30000	6	0.59103765	0.61666961	0.60023898
30000	20	0.74748100	0.69038360	0.69569835
30000	25	0.74544812	0.69471451	0.70140733
60000	6	0.54781686	0.62922043	0.62785449
60000	20	0.71265689	0.64875376	0.65201806
60000	25	0.73422309	0.69144423	0.69490175

Note that testing accuracy for features derived through Gibbs sampling are not presented due to a malfunctioning transform method of the LDA package in Python, thus we were unable to use the fitted model to transform the test data. Notice however, that the test accuracy was in fact higher than the training accuracy for VB. In general this is usually not the case, this indicates that the features learned by LDA are fairly good, as the performance using these features on unseen documents is relatively good, in fact better than the performance on the training features. Hence, this may seem to indicate that we may need to increase the complexity of the LDA model.

Looking at the results, it appears as we increase the complexity of the LDA model, the features learned are more representative of the actual documents, and thus we see an increase in test accuracy.

5 Conclusion

Ultimately we see that the LDA model was able to identify relevant topics that were in the 20 newsgroups corpus. Thus in general LDA would still be a useful tool if you were to identify key topics within a given corpus. However, the performance of the LDA features on the topic classification was not exceptional. So although LDA is able to identify topics within a corpus, the task of actually grouping/classifying documents may not be its strong suit; though this may be a consequence of LDA assuming that each document is constructed from multiple topics, and not just one. In addition, its lackluster performance can also be associated with the need to perform approximate inference in order to derive these features, as well as to identify words associated with the "unnamed" topics in LDA. However in terms of inference algorithms for LDA, we've shown that variational inference did a better job than MCMC on discovering what certain topics are about (ie: using the most probable words per topic), as well as had the upper hand on the classification task for smaller topic sizes, which is consistent with the prior statement. However, for larger topic sizes Gibbs Sampling outperformed and scaled better than Variational Bayes primarily because of the simplifying assumptions made by VB, which are not required by MCMC. Overall, LDA is interesting model in the assumptions that it makes regarding the relationship between topics and documents, and may be more useful in other datasets with longer documents (e.g: essays and newspaper articles), which actually contain a larger variety of topics per document, compared to the messages in the 20 newsgroups, which ultimately may have not contained as much overlap between topics as we originally believed.

References

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

[2]Hoffman, M., Bach, F. R., & Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems* (pp. 856-864).

[3] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*(suppl 1), 5228-5235.

[4] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., & Welling, M. (2008, August). Fast collapsed gibbs sampling for latent dirichlet allocation. In*Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 569-577). ACM.

[5] Grun, B. & Hornik, K. (2011). Topic Models: An R Package for Fitting Topic Models. Journal of Statistical Software, **40**(13), 1-30

[6] Wallace, H. M., Murray, I., Salakhutdinov, R. & Mimno, D. (2009). Evaluation Methods for Topic Models. Proceedings of the 26th International Conference on Machine Learning (ICML 2009), pp. 1105-1112. ACM Press.