Topic Modelling using Latent Dirichlet Allocation

CSE 250B Project 3

Suvir Jain, Gaurav Saxena

27 February,2014

Abstract

Abstract about LDA, our data, brief results.

1 Introduction

The objective of the project is to learn a Latent Dirichlet Allocation (LDA) model to predict dominant topics associated with a collection documents. This process is called topic modelling.

We discuss the LDA framework for topic modelling in section 2. We, then, describe our implementation of algorithm based on the framework in section 3. Furthermore, we discuss our experiments and code optimizations in 4 and finally present results in 5 and lessons learnt in section 6

We use two data sets classic400 and hookah data set for topic modelling (described in section 2.1). We use Latent Dirichlet Allocation to discover latent topics in these data set. The latent variables are discovered using Collapsed Gibbs (CGS) Sampling.

We found 3 main themes in the classic400 data set. They are about biology, aeronautical science and a collection of research related words.

In the hookah data set, we discovered that hookah related tweets have a social aspect to them. People commonly used words like 'party','bar','lol','tonight' in their tweets.

2 Framework

2.1 Dataset

Classic400 This dataset is a part of classic3[1] dataset. Classic3 dataset is a collection of 3893 documents, which in turn contains 1400 CRANFIELD documents from aeronautical systems papers, 1033 MEDLINE documents from medical journals and 1460 CISI documents from information retrieval papers. Classic400 is a collection of 400 documents randomly chosen from classic3 dataset by [1]. We use the same dataset for our analysis.

This data set contains a total of 6505 words and hence each document is converted to a vector of 6505 integers, each showing the frequency a word in a document. The dataset also contains true labels of the dataset which we use for sanity check of our model. The dataset also contains the vocabulary used in the classic400. We use the vocabulary to find the top words for each topic.

Hookah data set This data set consists of a set of tweets about Hookah usage. These tweets are drawn from a larger collection of tweets. The main set of tweets was collected between November 2011 and August 2013 using the Twitter Application Programming Interface (API). These tweets represented 1% of all tweets on Twitter during this time interval. On average, this data set consisted of 1.3 million tweets per day. Hookah-related tweets were extracted from this set using the keywords: hookah,hooka,waterpipe,shisha,sheesha and water pipe.The hookah data set consisted of 95,738 tweets.

To allow for experiments to be done in a short amount of time, we could not use all 95,738 tweets. Hence, we extracted 3000 random tweets from this data set and used that for this project. This data set has a vocabulary size of 8021 words.

What did we expect to find?

Classic400 data set Our aim in processing this data set was to discover the dominant topics. Before running LDA, we did not have any idea about the themes and topics that characterize this data set.

Hookah data set This data set consists of tweets about hookah usage. We were interested in finding out the context in which people use hookah. Is it something people associate with negative emotions? Is it something that people use socially? Do people perceive hookah positively? Results are presented in section 5.2.2.

2.2 Model

Representation LDA uses a vector of length equal to the vocabulary to represent each document. Each element is represent the frequency of words found in that document. As discussed in class notes, this model is called bag of words. This representation doesn't preserve the word order and consequently loses information. However, as discussed in class notes, this representation may be sufficient to deduce topics from the documents. Therefore, for a document d and a vector $\bar{x}$ then x_j is the number of word j in d and the length of a document n is

$$n = \sum_{j=1}^{m} x_j \tag{1}$$

Multinomial Distribution over documents Now, given a set of documents and their representation as vectors, we need a model to represent them. A model is a probability distribution over the set of documents. We calculate the parameters of this model such that the training documents have a higher probability. We can, then, use this model to find the probability of a test document.

LDA uses multinomial distribution. Mathematical expression for a multinomial distribution is given below

$$p(x;\theta) = (\frac{n!}{\prod_{j=1}^{m} x_j!})(\prod_{j=1}^{m} \theta_j^{x_j}) \tag{2}$$

where x is is a vector of non-negative integers and parameter θ are parameters of the model. In this expression, θ can be argued as a probability of a word j while x_j represents its count.

As discussed in the class notes, the first term in the equation 2, the first term is the number of sentences which results in the same x vector. The second term is the probability of an equivalence class of $\bar{x}$.

Also, given a set of training documents, the maximum likelihood estimate of the j_{th} parameter is

$$\theta_j = \frac{1}{T} \sum_{x} x_j \tag{3}$$

where the sum is over all documents x to the training set and $T = \sum_x \sum_j x_j$. However, this expression may lead to 0 probabilities. Therefore, we add a constant to the expression called a pseudocount. It is a notional number of occurences of each word in each document. The formula, thus, becomes,

$$\theta_j = \frac{1}{T'}(c + \sum_{x} x_j) \tag{4}$$

where $T' = mc + T$ and $0 < c \leq 1$

Generative Process As discussed in the notes, the generative process assumes that the data points are generated by a probabilistic process. It helps to find the parameters of this process by based on the concept of maximum likelihood. The algorithm for the generative process is given in the notes[5] and not reproduced here for brevity. The generative process gives the following global probability distribution which represents a mixture distribution

$$f(x) = \sum_{k=1}^{K} f(x; \phi_k) \tag{5}$$

where x is a document, ϕ_k is a parameter value of the k_{th} multinomial and α_k is the proportion of component number k.

In LDA we use Dirichlet Distribution which acts as a prior for multinomial distribution discussed above. It takes the following form [5]

$$p(\gamma|\alpha) = \frac{1}{D(\alpha)} \prod_{s=1}^{m} \gamma_s^{\alpha_s - 1} \tag{6}$$

where $D(\alpha) = \int_\gamma \prod_{s=1}^{m} \gamma_s^{\alpha_s - 1}$

2.2.1 Training Methods for LDA

We use the words of the documents as our training data. We assume that the number of topics K, α and β are fixed. Our aim is to learn θ values for documents and ϕ values topics. We use collapsed Gibbs Sampling for this process.

Collapsed Gibbs Sampling As discussed in the class notes [5], we use a variant of Gibbs Sampling called collapsed Gibbs Sampling (CGS) for learning the topics from the documents. In this process we do not try to learn θ and ϕ directly, but try to learn the hidden topic label z for each word in each document.

We represent each word by its position in the document vector and its count as the value of the element at the position. However, CGS is based on the occurrence of the words. Therefore, the length of the z is taken as the sum of all the elements of the vector (or the length of the document).

Gibbs Sampling assumes that we know the z value for an occurrence of word i. It, then, draws a random value of topic for every word except i according to a distribution. We use the following distribution for Gibb's Sampling

$$p(z_i|\bar{z}', \bar{w}) = \frac{p(\bar{w}|\bar{z})p(\bar{z})}{p(w_i|\bar{z}')p(\bar{w}'|\bar{z}')p(\bar{z}')} \tag{7}$$

where $\bar{w}$ is the sequence of words of the vocabulary, $\bar{z}$ is the corresponding sequence of z values, $\bar{w}'$ represents $\bar{w}$ without the word w_i and similarly, $\bar{z}'$ represents $\bar{z}$ without the word w_i. This expression can be reduced in terms of the counts of topics as given below

$$p(z_i = j|\bar{z}', \bar{w}) = \frac{q_{jw_i'} + \beta_{w_i}}{\sum_t q_{jt}' + \beta_t} \frac{n_{mj'} + \alpha_j}{\sum_k n_{mk}' + \alpha_k} \tag{8}$$

where q_{jw_i} is the count of word w_i occurs with topic j, q_{jw_i}' is the count of word w_i occurs with topic j except for the word w_i, n_{mk} is the count of the number of times topic $z_i = k$ in the document m and n_{mk}' is the count of the number of times topic $z_i = k$ in the document m except for the word w_i

2.2.2 Topic Metrics from LDA

The aim of the CGS is to come up with words which together give shape to a topic. This is done by first finding dominant topics using θ values and then finding the words which provides context for this topic using ϕ values.

θ values are a measure of probabilities that a document aligns with a topic. Therefore, dominant topics can be found by finding the topics which have the most documents associated with it. This is done by finding the maximum θ value for each document. Dominant topics can be found by keeping a count of the documents associating with a topic. This is discussed in detail in section 3.

Similarly, words with highest ϕ values for a topic are expected to be associated with a topic. We pick top 10 words with the highest ϕ values for each topic. The details can be found in 3.

3 Design and Analysis of Algorithms

Mention complexities of algorithms.

3.1 LDA Initialization

We run topic modelling on two aforementioned datasets to find the dominant topics for each. As discussed in the class notes [5], we initialize z as a $M \times N_m$ array. A z_i represents a position of a word in a sentence. Therefore, each word may occur multiple times in z. We assign topics randomly to all z values. Subsequently, using the corpus we initialize q as a $K \times V$ based on z and the corpus. Similarly, we also initialize n counts as a $K \times M$ using z and the corpus. The complexity of this process is $O(MN_m)$.

We also initialized the number of topics K, pseudocounts α and β before the modeling.

Chosing K, α, β Choice of K, α, β is critical a directly affects the quality of topics and the words association. High β values tend to decrease the number of topics as it reduces the impact of sparsity [6]. This is a important for our case, in particular, as we are dealing with small number of themes in both classic400 datasets and hookah. Therefore, as advised by [6] we use $\beta = 0.1$.

Similarly, α is used to for smoothing the θ distribution. It adds contribution from each topic to each document. Therefore, high values for α can reduce the separation between different topics and can lead to smudging of boundaries between them. We chose $\alpha = 50/K$ as suggested in the class and in [6].

Although, we assume both α and β to be constant but they can be found using EM algorithm. However, this increases the run time of the algorithm and therefore we have not implemented it.

Knowing α and β, the values of K can be found by maximizing the likelihood of data, as suggested by [6]. In their experiments they found that $K = 300$ to be a value at which likelihood peaks. However, we did not follow this procedure as we are not expecting a large number of topics and calculating likelihood computation is not computationally trivial. We experimented with several value of K and qualitatively found that noise words reduced with increasing K in dominant topics.

3.2 Collapsed Gibb's Sampling

We ran Collapsed Gibbs Sampling to convergence to find the dominant topics of the corpus. To achieve this we used equation 8 for an estimate of probability [5]. It is reproduced here for clarity.

$$p(z_i = j|\bar{z}', \bar{w}) = \frac{q_{jw_i'} + \beta_{w_i}}{\sum_t q_{jt}' + \beta_t} \frac{n_{mj'} + \alpha_j}{\sum_k n_{mk}' + \alpha_k} \tag{9}$$

For each document, and each position (a word could take) in the document, we calculated probability of each topic being assigned to the position. We assumed that the current topic assigned to the position was fixed and calculated the probability estimate using equation 8 of all the topics for this position. We, then, randomly selected a topic using the distribution of the these probability estimates.

We did this for each position and a number of iterations until the z values stopped changing by a wide margin at which point we assumed that convergence is reached

3.3 Obtaining Topics

Once we the final values of z, we calculated θ and ϕ values to get the topics and words associated with the them. We used the following equations [5] to do it

$$\hat{\theta_k} = \frac{\sum_{i=1}^{M} I(z_i = k)}{n} \tag{10}$$

$$\hat{\phi_k j} = \frac{\sum_{d=1}^{M} \sum_{i=1}^{n_d} I(w_i = j \wedge z_i = k)}{\sum_{d=1}^{M} \sum_{i=1}^{n_d} I(z_i = k)} \tag{11}$$

where z_i is the topic-word association vector, w_i is a word, k is a topic, M is the number of documents, n_d is the length of a document d.

Topics with the highest θ values were taken as dominant topics. Although we ran for various K values but we considered only top 3 dominant topics as we conjectured that both classic400 and hookah datasets have a small number of dominant themes and as we present in the results, probabilities of other topics was too low to be considered.

3.4 Topic Metrics

Furthermore, we collected more information for reporting like top words, document and word coverage for each topic. Top 10 words were calculated by taking the *argmax* of the top 10 ϕ value for a topic. *Document coverage* is a ratio of the number of documents with a specific topic to the total number of topics. Similarly, *word coverage* is a ratio of the number of words associated with a topic to the total number of topics.

4 Design of Experiments

4.1 Dataset Preprocessing

Pre-processing was required only for the second data-set. The classic400 dataset was used as provided in the MATLAB file format.

The following steps were taken to pre-process the hookah data set :

1. The text is tokenized using the Punkt Tokenizer [8] which is part of the Python NLTK library [2]. This process splits the tweets into individual words.
2. Then, the tokens are stemmed using the Porter Stemmer [11], also part of the Python NLTK library. This helps map similar words to the same stem. Word variations like plurals and tense-based conjugates are replaced by the root word.
3. All stopwords were deleted. Stopwords are common words like 'I' ,'me',can' etc. We used a list of 127 stopwords that is provided with NLTK[2].
4. All words are mapped to unique integer values and represented in the same format as classic400 i.e. a matrix of dimensions K(number of documents) x V(size of vocabulary).

Mention the stopwords used for processing second dataset. Cite the link : http://cseweb.ucsd.edu/users/elkan/1

4.2 Experiments with LDA

We ran the LDA implementation on both data sets for different number of topics. Specifically, we experimented with K = 3,10,50,100.

For each experiment, the θ vector and ϕ vector were recorded. From these, we inferred the top 3 topics for the entire data set. The top 3 topics are the 3 topics with which the highest number of words and documents were labeled.

We also tracked the number of topic-word associations that remained the same between consecutive iterations of LDA. This measure served as a indicator of convergence of the distribution. Related plot are presented in the results section.

4.3 Implementation

We implement LDA in MATLAB.

The pre-processing code was written in Python. Specifically, the NLTK python library was used for tokenizing, stemming and removing stop words. Details of these procedures are covered in section 4.1.

Some more implementation details that we discovered while developing the code are :

1. Z is specific to a document. Therefore it will have a different length for different document.
2. Also each word can have a different topic, therefore there will be a different entry for each word in Z. However, this is not true about q and n as they are counts. They will have K * V and K * M sizes.

4.4 Code Optimization

We optimized the following equation in 4 ways :

$$p(z_i = j|\bar{z}', \bar{w}) = \frac{q_{jw_i'} + \beta_{w_i}}{\sum_t q_{jt}' + \beta_t} \frac{n_{mj}' + \alpha_j}{\sum_k n_{mk}' + \alpha_k} \tag{12}$$

1. $q_{jw_i'} + \beta_{w_i}$ In this part of the calculation, β_{w_i} is constant for any fixed value of j. Therefore, for each j, β_{w_i} were added it to all $q_{jw_i'}$ outside the loop.
2. $n_{mj}' + \alpha_j$ For each value of j, α_j is a constant. Therefore, for each j, α_j was added to all n_{mj}' outside the loop.
3. $\sum_t q_{jt}' + \beta_t$ For any value of t, β_t is a constant value. Therefore, for all t, β_t was added to $\sum_t q_{jt}'$.
4. $\sum_k n_{mk}' + \alpha_k$ For any value of k, α_k is a constant. Therefore, for all k, α_k is added to all $\sum_k n_{mk}'$.

One more code optimization saved us a lot of time during experiments. This was in the calculation of the ϕ vector. Calculating the ϕ vector for all topics is an O(K*V*M) operation. V and M cannot really be optimized. However, we did optimize the K value. We calculated ϕ vector for only the top 3 topics. These top 3 topics were the topics with which most documents were associated as per the θ vector.

4.5 Sanity Checks

We used the following checks during the development of the code to ensure that our implementation was correct.

1. For the classic400 dataset, we had access to the true labels which were grounded in real-world truth. We compared the topic distribution of our learned model to these true labels.
2. After initializing q and n, the sum of these two vectors should be the same and it should equal the total number of words in the corpus.

5 Results of Experiments

5.1 Convergence of LDA

In order to measure the convergence of LDA, we tracked the word-topic associations recorded by vector z. After each iteration, we measured the number of z values which had stayed the same.

This measure(Δz) stabilized after some iterations. This experiment was repeated for number of topics K = 3,10,50,100.

The 1 shows this convergence for classic400 data set. The 2 shows this convergence for hookah data set.

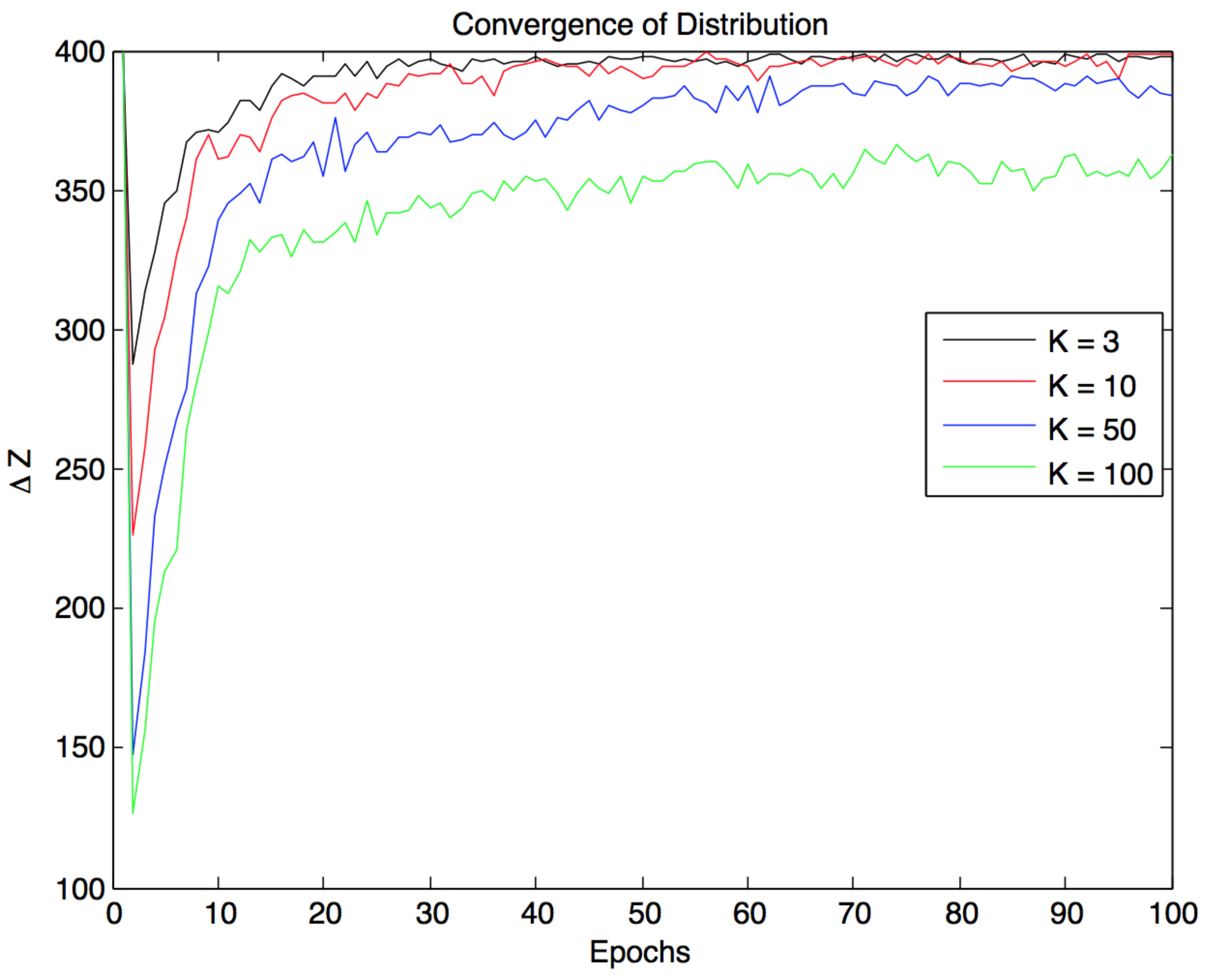

Figure 1: Tracking the convergence of LDA over 100 iterations for classic400 data set.

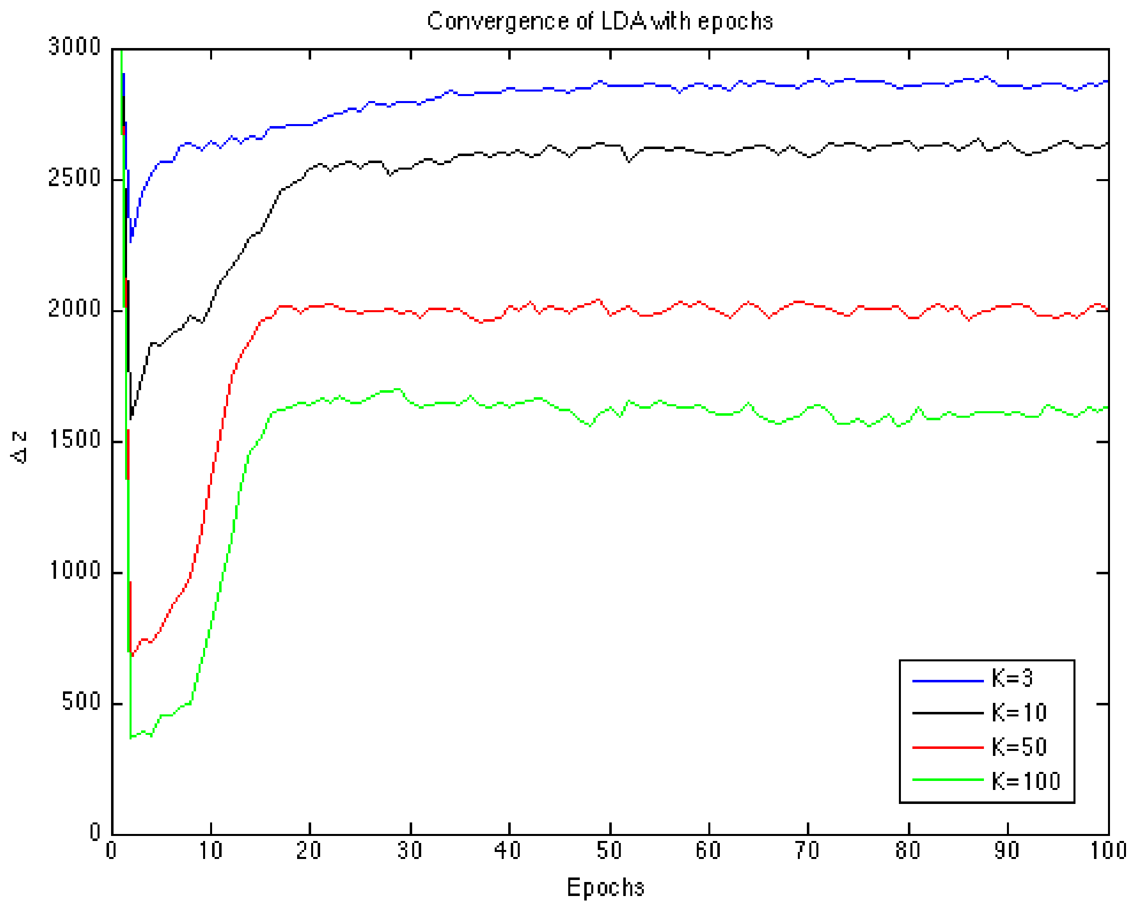

Figure 2: Tracking the convergence of LDA over 100 iterations for Hookah data set.

5.2 Topic Models discovered

5.2.1 Classic400 data set

The tables 1,2,3,4 shows the topic models discovered by LDA for the classic 400 data set with K = 3, 10, 50 and 100, respectively.

We found 3 dominant topics in the classic400 data set.

1. The first topic has words related to aviation and aeronautical engineering. For example - mach, supersonic, wings, and velocity.
2. The second topic has words like scientific, research, language, and methods. These keywords seem to be related to scientific research and pulications.
3. The third topic has medical terms like ventricular, left, aortic, and septal. All these terms are commonly used in medical science.

5.2.2 Hookah data set

The tables 5,6,7,8 shows the topic models discovered by LDA for the Hookah data set with K = 3, 10, 50 and 100, respectively.

We found that the hookah tweets were dominated by a words like smoke, bar, 'lol', 'like','tonight'.

This was in line with our hypotheses that Hookah is normally associated with positive emotions and is used in a social context.

Some interesting points to note about the discovered keywords in this data set :

Topic	Words	Document Coverage (%)	Word Coverage(%)
1	boundary,layer,wing,mach,supersonic, ratio,wings,velocity,effects,shock	50.25	38.07
2	patients,ventricular,left,retrieval,cases, system,language,aortic,septal,research	15.75	38.53
3	fatty,acids,glucose,journals,acid ffa,free,blood,nickel,infants	34	23.40

Table 1: Topics Discovered in classic400 data set for K=3

Topics	Words	Document Coverage(%)	Word Coverage(%)
1	boundary,layer,wing,mach,supersonic,ratio wings,velocity,effects,shock	0.4975	56.15
2	system,scientific,retrieval,research,language science,journals,systems,methods,subject	0.2475	23.38
3	patients,ventricular,left,aortic,septal,cases, defect,regurgitation,ventricle,pulmonary	0.1025	20.47

Table 2: Topics Discovered in classic400 data set for K=10

Topics	Words	Document Coverage(%)	Word Coverage(%)
1	boundary,layer,wing,mach,supersonic, ratio,wings,effects,velocity,shock	50.00	58.97
2	system,scientific,retrieval,research,language, science,systems,journals,subject,requests	23.75	19.47
3	patients,ventricular,left,cases,aortic,septal, defect,regurgitation,ventricle,pulmonary	12.75	21.56

Table 3: Topics Discovered in classic400 data set for K=50

Topics	Words	Document Coverage(%)	Word Coverage(%)
1	boundary,layer,wing,mach,supersonic,ratio wings,effects,velocity,shock	56.75	68.51
2	patients,ventricular,left,cases,aortic,septal, visual,defect,regurgitation,ventricle	14.00	20.27
3	retrieval,language,system,subject,systems, library,requests,descriptor,basic,search	10.50	11.22

Table 4: Topics Discovered in classic400 data set for K=100

Topic	Words	Document Coverage(%)	Word Coverage(%)
1	hookah,http,shisha,smoke,bar, hooka,lol,like,night,tonight	79.23	56.89
2	shisha,party,que,hookah,con, http,una,die,fumando,fumar	11.77	24.21
3	shisha,haha,aku,rokok,tak, nak,hisap,sheesha,hahaha,yang	9.00	18.90

Table 5: Topics Discovered in Hookah data set for K=3

Topics	Words	Document Coverage(%)	Word Coverage(%)
1	hookah,http,shisha,smoke,bar, hooka,lol,like,night,tonight	76.57	71.41
2	shisha,haha,aku,tak,rokok, hahaha,die,nak,hisap,dia	10.67	0.1905
3	party,que,con,una,vamoc, party,por,fumando,esta,sheesha	3.90	0.954

Table 6: Topics Discovered in Hookah data set for K=10

1. 'http' is a word in the dominant topic. This is used in tweets when people 're-tweet' or RT. RTs typically have a short URL.

2. Words like 'combinado', 'fumando', 'una' are seen in the keyword list. These are spanish words. The data set has a mixture of english and non-english tweets.

   These results reinforce the findings of prior work by Myslin et al [9]. Their work had indicated that some tobacco products are perceived positively. These products provide opportunities for tobacco control education.

5.3 Plot of θ vector

We evaluated the theta vector as described in section 2.2.

This was done for both the data sets and visualized in plots. These plots help visualize the documents clustered by their most likely topics.

It is interesting to see that most documents align themselves in 3 clusters. The topics which dominate have dense clusters and other topics have a sparse cluster.

Fig. 3 and Fig. 4 are plots of the θ vector for the classic400 and hookah data set, respectively. They helps in visualizing the hyperplane between topics.

Fig. 5 and Fig. 6 are plots of the θ vector but in scatter plot form. These plots helps to visualize the documents clustered by topics.

Topics	Words	Document Coverage(%)	Word Coverage(%)
1	hookah,shisha,http,smoke,bar, hooka,lol,like,night,tonight	70.60	94.22
2	party,hooka,thank,para,que, pra,combinado,right,power,move	1.07	3.07
3	club,hai,grandma,haram,hahaha, go,homi,super,till,good	0.87	2.72

Table 7: Topics Discovered in Hookah data set for K=50

Topic	Words	Document Coverage(%)	Word Coverage(%)
1	hookah,shisha,http,smoke,bar, hooka,lol,like,night,loung	61.10	97.24
2	shisha,hookah,boy,ligero,rene,fri, offer,time,party,catchaflight	0.73	1.45
3	hookah,heart,http,talk,hooka,wine, satelit,dgib,shower,mypoorlung	0.7	1.31

Table 8: Topics Discovered in Hookah data set for K=100

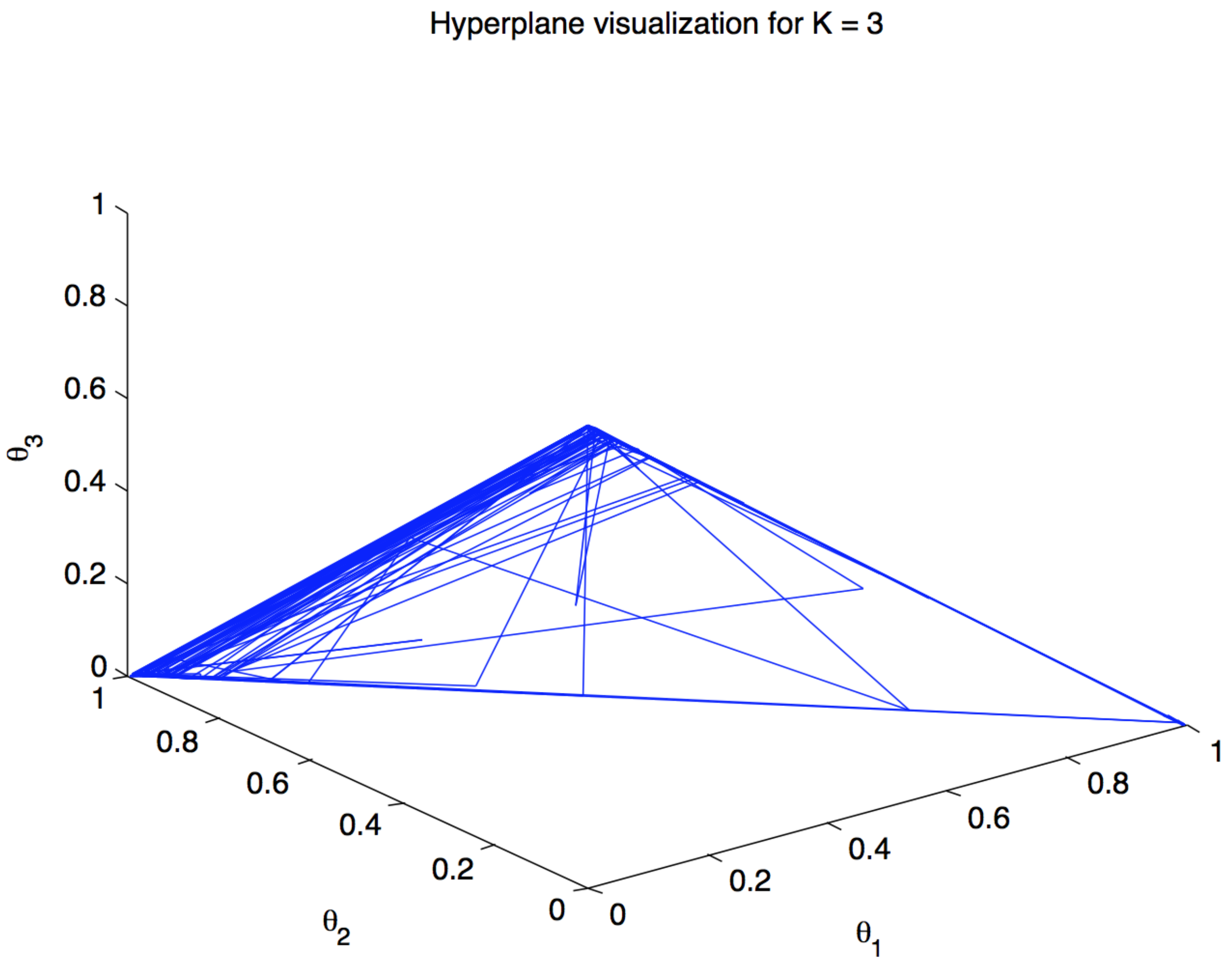

Figure 3: classic400 : Plot of θ vector helps in visualizing the documents clustered by topics

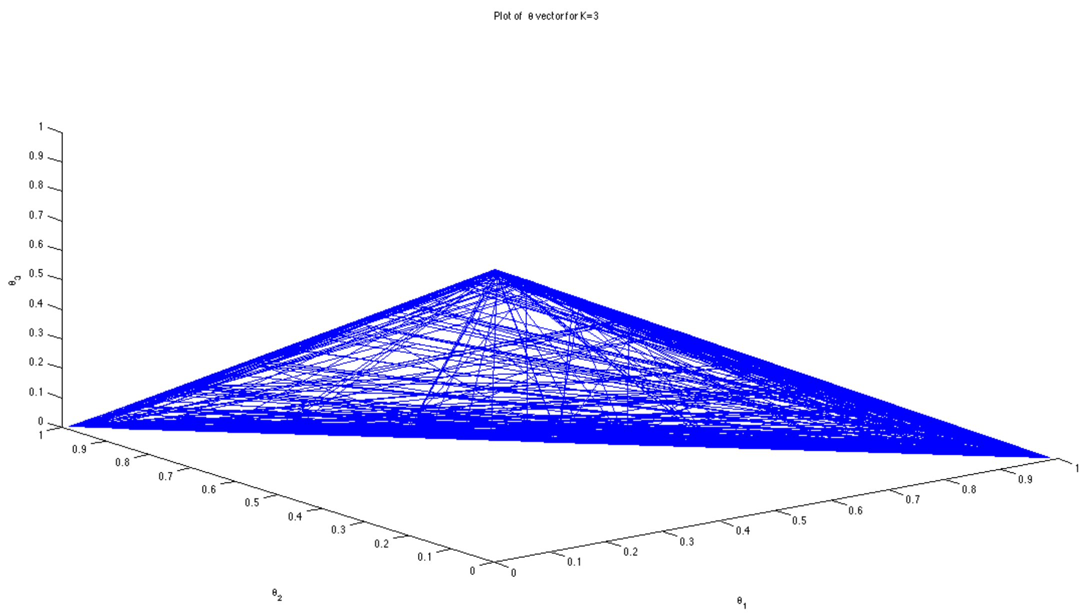

Figure 4: Hookah data set : Plot of θ vector helps in visualizing the documents clustered by topics

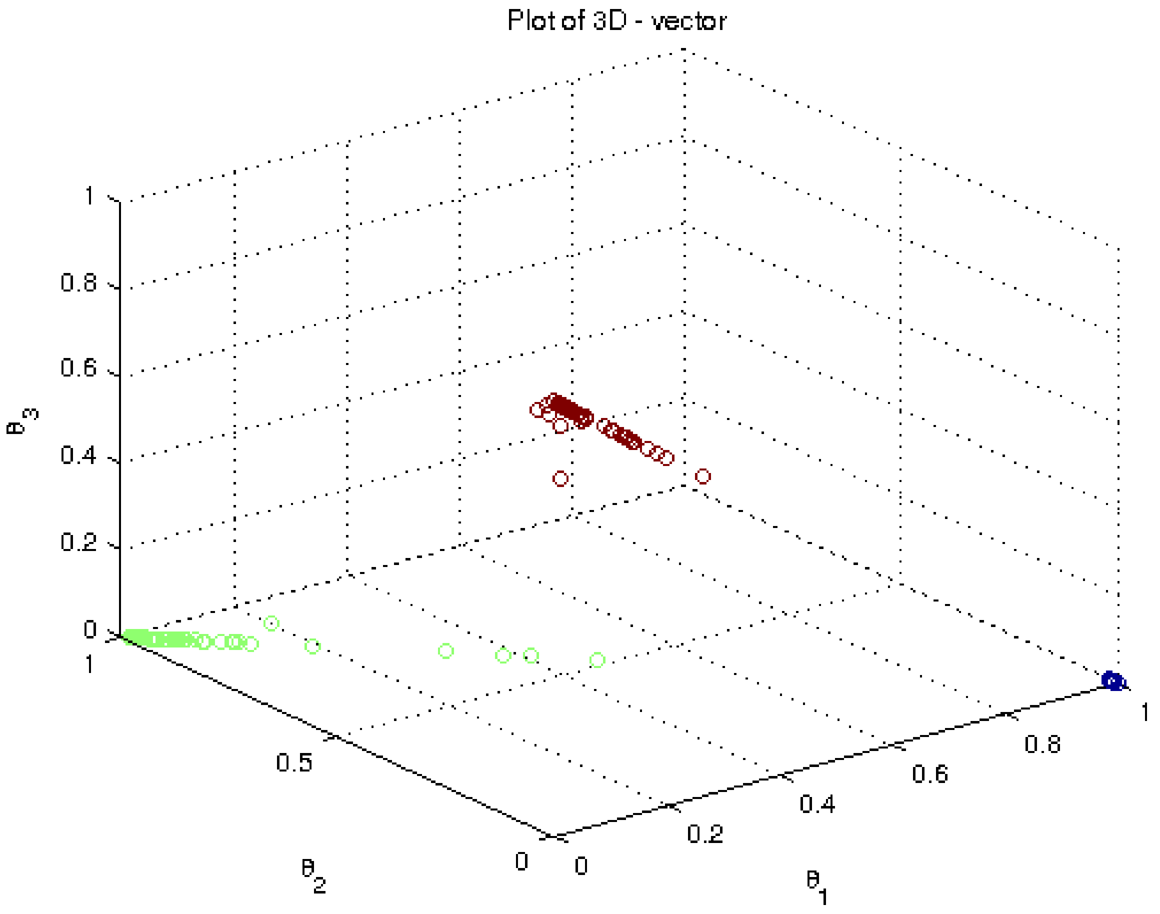

Figure 5: Classic400 : Visualizing documents clustered by topics

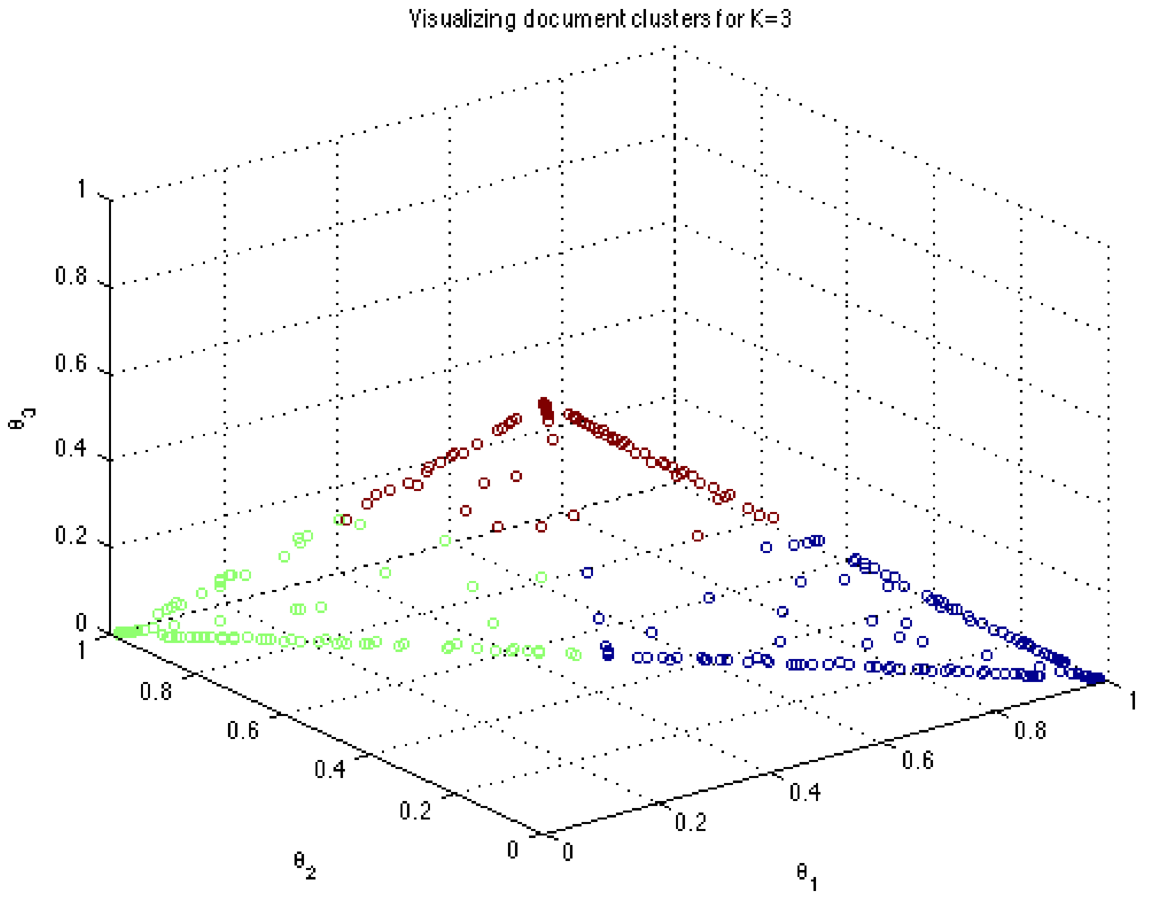

Figure 6: Hookah Data Set : Visualizing documents clustered by topics.

5.4 Comparison with true labels in classic400 data set

For the classic400 data set, we have access to the set of true labels based on real-world knowledge. Fig. 7 plots the number of documents labelled by each topic. This plot is superimposed over the number of documents labelled by the true labels. We see that discovered topics closely track the true labels.

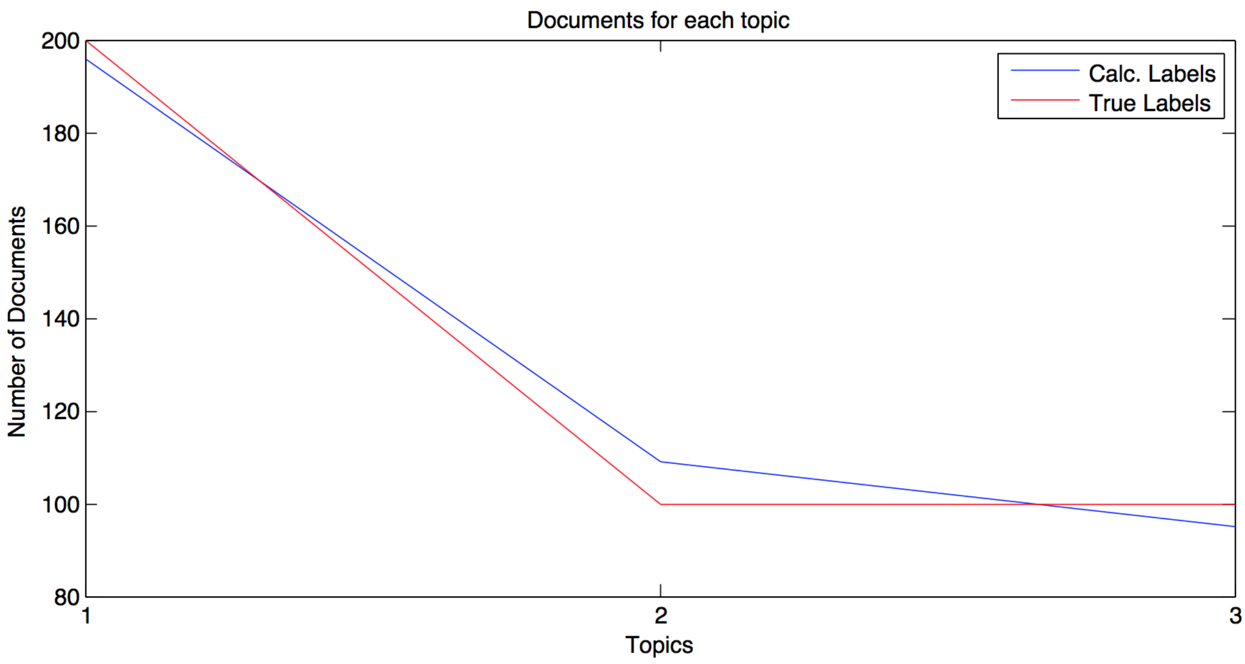

Figure 7: Classic400 : Topics discovered by LDA vs True Labels

6 Lessons Learned

- The main bottleneck in running the code was its slow speed of Collapsed Gibbs Sampling. We started with a running time of more then 2 hours for 100 iterations on Haskel i5, 4GB machine. After the optimizations done in 6 we could complete 100 iterations in about 30 minutes.
- We found that calculation of LCL is computationally expensive. Therefore, as a measure of convergence we tracked change in the distribution during the iterations of CGS. We found that when the number of z values stop changing, there was a good separation in the topics.
- Evaluating quality of a topic model is a subjective thing. [3]
- We also found that having less number of topics than the actual gives incoherent word-topic matches. However, as the number of topics increase this incoherence decreases. Even when the number of topics is more than true topics, the dominant topics still emerge and we found that with increasing K the word-topic associations still stay the same.

7 Goodness of Fit of LDA

We did a brief survey of literature about goodness of fit of LDA models. It appears that defining goodness of fit for LDA is a non-trivial and rather tricky task [7],[10].

7.1 Defining goodness of fit

As per [10], if the topics ϕ_1 to ϕ_k are given, the goodness-of-fit can be modeled as $p(\bar{w}|\Phi, \alpha)$. If the topics ϕ_1 to ϕ_k are not available, the author suggests the goodness-of-fit can be modeled as $p(\bar{w}|\alpha, \beta)$, where α and β are the hyperparameters of the LDA.

As per the Elkan [4], perplexity can be used as a measure of goodness of fit. As per [7] perplexity is the reciprocal geometric mean of the token likelihoods in the test corpus. It can be calculated as :

$$P(\tilde{W}|M) = exp - \frac{\sum_{m=1}^{M} logp(\tilde{\vec{w}}_{\tilde{m}}|M)}{\sum_{m=1}^{M} N_m} \tag{13}$$

As per Heinrich, this measure can be intuitively interpreted as the expected size of a vocabulary with uniform word distribution that the model would need to generate a token of the test data. A model (or parameter set) that better captures co-occurrences in the data requires fewer possibilities to choose tokens given their document context. Thus lower values of perplexity indicate a lower misrepresentation of the words of the test documents by the trained topics.

7.2 Computing goodness of fit

As per [10],[7] and [6], only the approximate likelihood of data can be computed. Given this likelihood, we then compute perplexity. As per our understanding, since perplexity is the reciprocal geometric mean of likelihoods, it is also only an approximation.

7.3 Determining overfitting of LDA

As per [7], perplexity is a measure of clustering quality. Higher value of perplexity indicates higher misrepresentation of the words of the test documents, thus indicating overfitting.

Based on our research, overfitting can be detected as follows :

1. Train LDA for a given number of topics (K) on the training set.
2. For different values of K, compute perplexity on the validation set.
3. Pick $\tilde{K}$, for which the perplexity is minimum.
4. If $\tilde{K} < K$, then the model has likely overfit the training data.

As per [10], this approach is justified because larger number of topics will result in a larger parameter space. A larger parameter space has more expressive power and can hence better fit the data. In case of $\tilde{K} < K$, it means that it is possible to get a good fit with lesser topics. Hence, that makes overfitting more likely.

References

[1] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Sra, and G. Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

[2] S. Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[3] J. Chang, J. Boyd-Graber, C. Wang, S. Gerrish, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.

[4] C. Elkan. Word burstiness, topic models, and perplexity, June 2009.

[5] C. Elkan. Lecture notes, cse 250b: Principles of artificial intelligence: Learning, 2014.

[6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[7] G. Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[8] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006.

[9] M. Myslín, S.-H. Zhu, W. Chapman, and M. Conway. Using twitter to examine smoking behavior and perceptions of emerging tobacco products. *Journal of medical Internet research*, 15(8), 2013.

[10] P. Nguyen. Experiments with latent dirichlet allocation, Mar. 2012.

[11] M. F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.