



Topic Modeling

Using Latent Dirichlet Allocation (LDA)

EE401 Pattern Recognition and Machine Learning Course Project



Abstract

Topic Modeling is a type of statistical modeling for discovering the abstract “topics” that occur in a collection of documents. We used Latent Dirichlet Allocation to find the topics in the IIT Dharwad broadcast E-mail. We also explored the mathematics behind LDA and other applications like using LDA for images. To better understand the Variational EM algorithm used LDA we implemented a simplified version of it to estimate parameters of a Gaussian Mixture Model. We also used LDA to model topics in the popular 20 newsgroups dataset.

By:

Akshat Karani (170010003)

Hritik Kumar (170010013)

Arpit Agrawal (170010040)



Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is an example of a topic model and is used to classify text in a document to topics. In this project, we applied LDA to a set of documents that we collected from the IIT Dharwad broadcast E-mail.

Input given to LDA is a document term matrix and the number of topics, each document is represented as a Bag of Words. LDA is a generative model and the generative process is as follows:

1. Each document is modeled as a multinomial distribution of topics and each topic is modeled as a multinomial distribution of words.
2. It also assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution.
3. Now, a document is assumed to be generated as follows: first, you select a distribution over the topics.
4. Then draw a topic from this distribution, then draw a word from the distribution over words corresponding to the topic. This is the first word of the document.
5. This is repeated for all the words in the document.

LDA then uses Variational EM algorithm to estimate the topic distributions and the distribution of words for each topic during training.



Extracting Topics using LDA

Following process involved in building the LDA model for IIT Dharwad dataset

1. Collection of Data
2. Preprocessing the raw text:

- a. Removing emails newline characters and punctuation and character using regular expressions.
 - b. Tokenization - Convert a document into a list of lowercase tokens, ignoring tokens that are too short or too long. We used the gensim library for this.
 - c. All stopwords are removed.
 - d. Making bigrams of a document - Bigrams are two words frequently occurring together in the document. We used the gensim library for this.
 - e. Lemmatization - Words in the third person are changed to first person and verbs in past and future tenses are changed into the present. We used the spacy library for this.
3. Converting text of bag of words:
 - a. Prior to topic modeling, we convert the tokenized and lemmatized text to a bag of words — which you can think of as a dictionary where the key is the word and value is the number of times that word occurs in the entire corpus.
 - b. We can further filter words that occur very few times or occur very frequently. Now for each pre-processed document we use the dictionary object just created to convert that document into a bag of words. i.e for each document we create a dictionary reporting how many words and how many times those words appear.
4. After this, the data is fed to the LDA model. We used gensim's implementation of Latent Dirichlet Allocation. As said above the output given is document-topic distribution and topics-word distribution.



Evaluating the model - Coherence Score

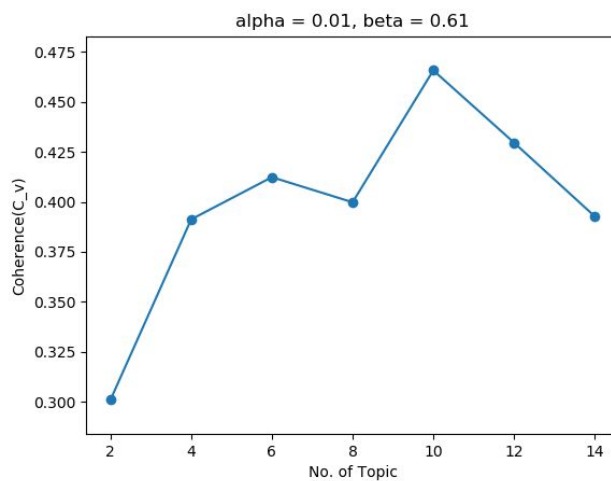
We want to see if the trained model is objectively good or bad. But it can be difficult to evaluate such a model because natural language is ambiguous. We use coherence score to quantify the performance of our model.

A set of statements or facts is said to be coherent if they support each other. Topic Coherence measures score a single topic by measuring the degree of semantic similarity between high scoring words in the topic.

- There are different coherence measures like C_v , C_{umass} , C_p , C_{uci} , and many others.
- We used C_v measure to find the optimal number of topics.
- **C_v measure** is based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosine similarity.

Tuning for Hyperparameters

To find the optimal number of topics we modeled with varying number of topics (2, 4, ..., 14) and choose the one for which the coherence score was maximum.



Coherence score is maximum when the number of topics is equal to 10



Visualizing the result - pyLDAvis

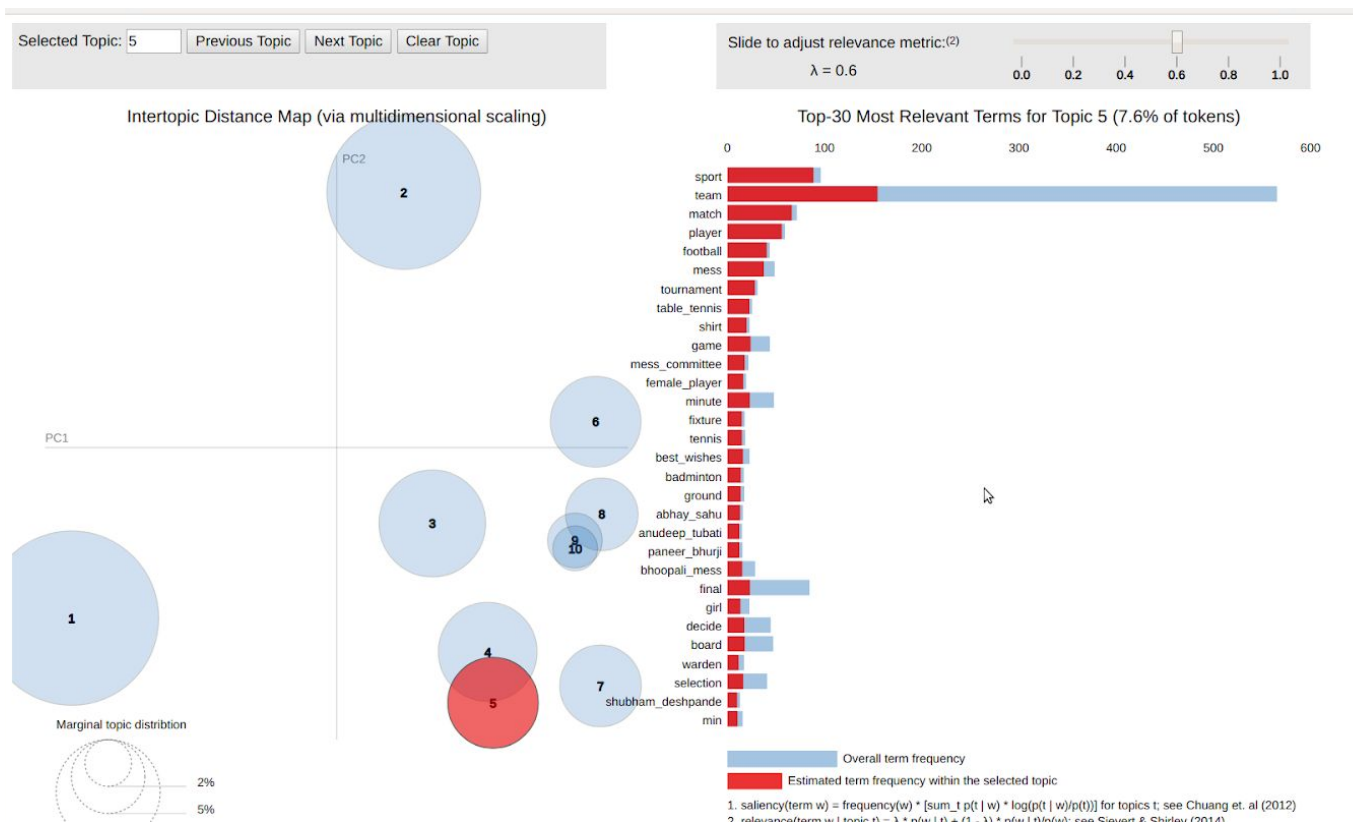
As we know the output LDA gives is

1. Document-Topic distribution *i.e.*
2. Topic-Word distribution

Below is the topic-word distribution for model with number of topics equal to 10. The numbers from 1 to 10 represents the index of the topic where each topic is a combination of keywords and the coefficient of each word represents the probability of the word occurring in that topic.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
event	intended recipient	team	hostel	course
team	strictly prohibited	sport	ashrith adepu	academic
club	confidential	match	election	office
registration	unauthorized review	player	campus	slot
workshop	disclosure dissemination	football	vote	notice
competition	copying	table tennis	general	credit course
participate	unlawful	tournament	manifesto	general
register	print	game	contestant	schedule
cultural	notify sender	final	wifi	rule
technical	action taken	schedule	cycle	staff

Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
talk	music	water	course	analysis
research	internship	library	internet	seminar
speaker	program	journal	speed	mechanical
university	company	omprakash bhendigeri	rajshekar	wind farm
design	yourdost	questions panel	wifi	mathematic
development	apply	cube	network	simulation
science	expert	fearless step	computer science	patil
application	skill	rubik	sandeep	amlan
communication	saurav dosi	interspeech	moodle	research
program	scholarship	corpus	password	turbulence





Retrieval of similar documents

Given a new document, we can use LDA to find similar documents in our corpus.

We can do this as follows:

1. Firstly we predict the topics in the new document.
2. We run Gibbs sampling just on the words in the new document, this will give a topic assignment for every word in the new document.
3. This will give the distribution of topics in that document.
4. After we have a topic distribution of the new document we can find documents with similar topic distributions in our corpus. We used Jensen-Shannon similarity to find documents with similar topic distribution.



Using TF-IDF

We also tried to model LDA by giving input as tf-idf matrix.

$$idf = \log(N/D)$$

where N is the total number of documents and D is the number of documents in which the term has occurred.

tf-idf is calculated by multiplying idf with term frequency:

But the results using this were not very good.

Gaussian Mixture Model

To better understand the Variational EM algorithm used LDA to estimate model parameters we used implemented a simple version of the EM algorithm to estimate the parameters of a Gaussian mixture model.

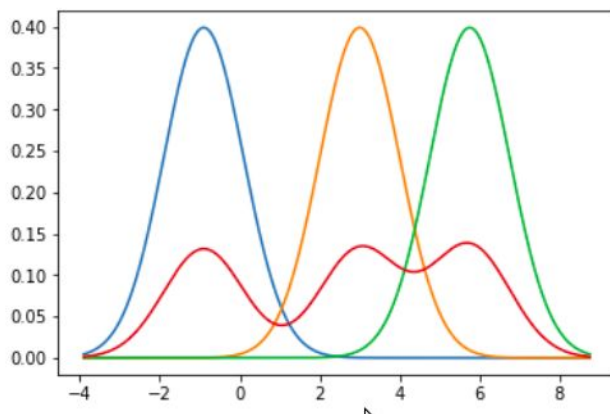
We started with a mixture of 3 Gaussians with the following parameters

```
Weights = [0.33, 0.33, 0.34]
Means = [-0.9, 3, 5.75]
Standard_Deviations = [1, 1, 1]
```

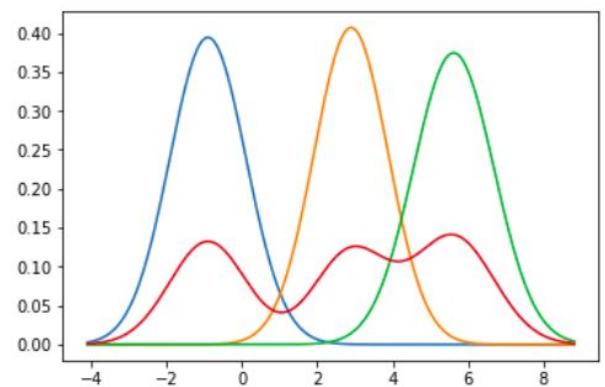
Then we generated data from this and using this data tried to estimate the parameters of the mixture. The result that we got is

```
Weights = [0.33541291, 0.29463832, 0.36994877]
Means = [-0.90089807, 2.89497223, 5.61521725]
Standard_Deviations = [1.01186431481449, 0.9805002015171135, 1.06558400188086]
```

Actual



Calculated





LDA for Images

We also explored the area of LDA for images. LDA for images tries to define an analogy with the text-domain. It defines a word to be a segment of the image. Images are segmented using the N-cut algorithm. But when a segment is used as a word, it is not possible to define the vocabulary as being the set of all possible segments of images, because it could be intractably large. So, to reduce the size of the vocabulary, we cluster the entire set of image segments. Once all the segments are clustered, then any two segments within the same cluster are considered equivalent. And each image is considered as a document.


LDA for images has some great applications like object categorization & segmentation.

Steps involved:

1. For each image, segment it into 'p' segments.
2. For each such segment compute a set of features.
3. Once this is done for all the images, cluster the set of all segments using the feature vector, to reduce the size of the vocabulary.
4. For each image, compute a frequency representation of how many times a member of a cluster of segments occurs in the image.
5. Feed this information as input to LDA.
6. Use the topic-simplex representation of the image to cluster the image data set.

Problem faced:

Even after going through various research papers and understanding the theoretical concept behind it, we were not able to implement because we did not find any suitable libraries that can help us run LDA for images.



Other Applications of Topic Modeling

- **Opinion summarization:**

We can use LDA for opinion summarization which will not only tell us about opinions are positive or negative but also LDA will help identifying which events correlated with such opinions.

- **Recommender System:**

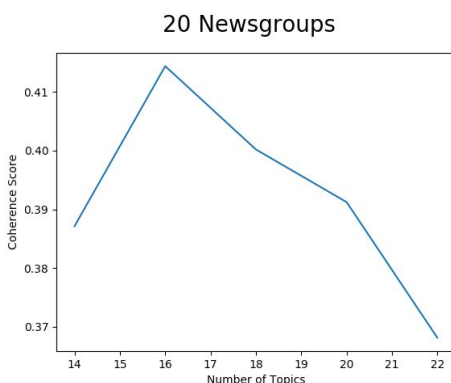
LDA gives topic distribution of all the documents across corpus. We can fetch documents similar to query document by using Jensen-Shannon similarity.

Recommender system of various types such as movie(based on plot), book etc. can be built.

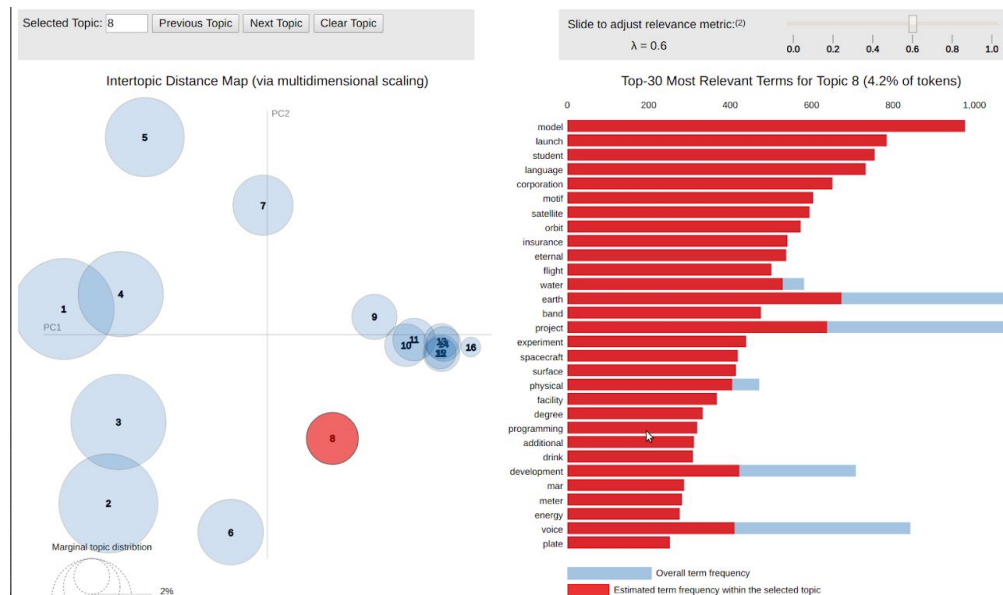
- Many other applications such as Sentiment analysis, chatbot, Q&A etc can be built using LDA.

LDA on 20-newsgroups Data

In IIT Dharwad dataset number of documents were small and number of topics were also not very large. LDA gives better results when number of topics are high. So we also used LDA to model topics in 20 newsgroup dataset. We found that optimal number of topics were 16



The results for this are



References

- [1] David M. Blei, Andrew Y. Ng and Michael I. Jordan.
Latent Dirichlet Allocation, Journal of Machine Learning Research 3
- [2] Carson Sievert and Kenneth E. Shirley. LDAvis: A method for
visualizing and interpreting topics
- [3] Jason Chuang, Daniel Ramage, Christopher D. Manning and Jeffrey
Heer. 2012a. Interpretation and Trust: Designing Model-Driven
Visualizations for Text Analysis. CHI.
- [4] Pradheep K Elango and Karthik Jayaraman
Clustering Images Using the Latent Dirichlet Allocation Model
- [5] pyLDAvis(<https://pyldavis.readthedocs.io/en/latest/index.html>)
- [6] genism (<https://radimrehurek.com/gensim/index.html>)

Link to the Code: <https://github.com/agrawalarpit14/lda>

