Exploring popular topic models

Poonam Tijare and Jhansi Rani P
Department of Computer Science & Engineering (VTU RC), CMR Institute of Technology, Bengaluru, India
Email: poonamtijare@gmail.com, jhansirani.p@gmail.com

Abstract. Information from micro-blogging site such as Twitter is a huge repository of data. A lot of research is happening on sentiments, discovering patterns and prediction. One challenging task is dividing this humongous unstructured data into clusters. Several topic modeling methods are proposed by researchers. This paper presents a brief summary of topic modeling methods LDA, LSI and NMF and their applications. Experiments are conducted on the Twitter based datasets created using tweets on keywords Cauvery river, Lokpal bill and Rahul Gandhi. Paper covers a brief discussion on evaluating the accuracy of topics formed using perplexity, log-likelihood and topic coherence measures. Best topics formed are then fed to the Logistic regression model. The model created is showing better accuracy with LDA.

Keywords: Coherence, LDA, LSA, NMF, Topic Model

1. Introduction
Micro-blogging sites like Twitter, Facebook, etc. generate an enormous quantity of information. This information is brought together, observed and then figure out the answer for several issues well before time. Information Analytics is one such field. Information scientists invariably search for information patterns to create some observations. Although the task seems simple, it is complex and time-consuming. The data on these sites is unstructured due to the use of symbols, short forms, misspelt word and etc. This data is then needed to be reborn into a structured format in order to do any further tasks. A lot of preprocessing is required to be done on this unstructured data[1].
Topic modeling techniques can facilitate to seek out necessary topics from the information. These methods help to uncover data patterns in the corpus and grasp information by providing the outline of necessary topics within the document. Topic models are useful to cluster documents using unsupervised learning approach. Rule-based text mining with the use of regular expressions or wordbook based keyword looking out techniques is different from Topic modeling techniques.

1.1. Topic Modeling
Topics are repeating a pattern of words that occur along or in the collection of written text. Topic modeling can be stated as a way of,
- Dimensionality Reduction
Dimensionality reduction is feasible exploitation of Topic Modeling. Instead of representing text with words with their counts in vector, it represents the text in its topic area with weights on each topic.
- Clustering
Topic modeling may be simply compared to a cluster. The most important criterion in clustering is the number of subjects as is the number of clusters. Topic modeling helps to build word clusters instead of text clusters. So the text is a combination of all the topics with explicit weights.

- A way of Labeling

Document classification is about the assignment of one class to text while topic modeling is assigning multiple labels to a text. An individual skilled will label the ensuing topics with lucid labels and use totally different heuristics to convert the weighted topics to a collection of tags. When model produce topics like: "beauty", "hair care", "shades", "color", "tone", it can conclude that it is about beauty and personal care topic; similarly "agenda", "corruption", "development", "GDP", it's about politics domain.[2-7]

1.2. Uses of Topic modeling Methods

Topic model and feature selection are useful in many applications like organizing emails, revealing inclinations and trend in articles published in the magazine or newspaper. It is conjointly utilized in the companies to extract latent options of job profiles, and then assign it to interested candidates. Topic modeling is found useful in recommender systems. Based on users liking, recommender system can counsel articles. For this, it'll take into account similarity measure on a topic structure and the history of user reads and liked contents. Topic modeling is also used in spatiality reduction.

The paper is organized as follows. Section 2 is about Topic modeling methods LDA, LSI, NMF. Section 3 highlights some of the methods used for checking the accuracy of topics generated. Section 4 presents the results of experiments conducted on the dataset.

2. Topic Modeling Techniques

This paper considers three popular topic modeling techniques. These techniques are:

- Latent Dirichlet Allocation LDA (Blei et al., 2003, 2010)
- Latent Semantic Analysis LSA or Latent semantic Indexing LSI with Singular Value Decomposition (Landaur et al. 1997,1998)
- Latent Semantic Analysis with Non-negative Matrix Factorization (NMF) Method (Lee et al. 2001)

These models use a bag of words concept. Other things common in LDA, LSI and NMF are,

1. In above-stated algorithms, the number of the topic parameter (n_topic) is required.

2. All algorithms take input in the form of Document-term matrix which is the number of occurrences of word x in document y.

3. The above algorithm outputs two matrices, word topic matrix which is words comprising each topic and topic document matrix which is the number of topics in each document.

Python package scikit-Learn has the implementation of LDA, LSI and NMF whereas Gensim has methods to implement LDA and LSI.

2.1. LDA

Latent Dirichlet Allocation is an unsupervised method of topic modeling. LDA uses Bayes theorem. Scikit Learn implementation of LDA uses online variational Bayes algorithm. Assume text corpora in the model with documents D, with K topics. Each document contains N number of words(positive number). LDA process is explained below.

Step 1. Choose the composites i.e. number of documents and parts i.e. words or ngrams. Choose how many unigrams, bigrams..ngrams in each composite.

Step 2. Pick up the number of topics.
Step 3. Build the first matrix on words in each topic. Each Column contains topic. The sample will be drawn from Dirichlet beta distribution value as input. Value of beta is between 0 and 1. It gives the conditional probability of word over the document.

Step 4. Create the second matrix on the topics in each document. This time sample will be drawn using alpha value as input in Dirichlet distribution Dir(α) where α is positive real’s.

Step 5. Based on the above steps, build the composites table containing topic and words with maximum probability.

Step 6. Understand and Interpret results.

Prior probability distribution in Bayesian classifier uses Dirichlet distributions. Categorical distribution and multinomial distributions use Dirichlet distribution based on LDA model. The stated process will automatically generate clusters based on two matrices. Apart from the fixed size of topics, LDA suffers from limitations such as non-correlated topics and static non-evolving topics with time [10].

There are methods which help to find the quality of topics generated by the model discussed in section 3. Despite these odds, LDA is the most chosen method used in Topic generation. Scikit-Learn and Gensim Python packages provide the implementation for LDA [7-8].

2.2. LSA

Latent Semantic Analysis (LSA) uses a bag of word concept. It takes document and terms matrix as input and then divides it into document topic matrix and topic term matrix.

The basic process of LSA is:

Step 1. Input document term matrix X.

Step 2. For m documents with n words in the corpus where m and n are positive real’s, build m x n matrix; each row represents document and column as words in the document with TF-IDF scores and is inversely proportional to the frequency of words in each document.

Step 3: To find the latent topic, need to do dimensionality reduction using SVD (Singular value decomposition).

Decompose X into three matrices U, S and V. Pick value K concepts to keep.

\[
X = U_t S_t V_t^T
\] (1)

where U is the m x k matrix; concepts(column) in each document (row), S is k x k diagonal matrix of singular values to capture the amount of variation in each concept, V is transpose matrix; terms (rows) relating to each concept (column), τ is hyper-parameter; can be adjusted to find the number of topics.

Step 4: Apply cosine similarity to find co-related documents, co-related words. It helps to search the most relevant information in the corpus.

\[
P(m, n) = \sum z P(Z) P(m|Z) P(n|Z)
\] (2)

P(Z) corresponds to matrix S in LSA. P(Z|D) and P(W|Z) can be modeled using multinomial distributions using expectation maximization (EM) algorithm used to maximize log-likelihood in E steps. They correspond to U and V matrix in LSA.

PLSA is same as LSA except it represents everything in terms of probabilities. PLSA suffers from the problem of assignment of probability values. Secondly, there is linear growth; prone to overfitting. [9-11].
2.3. NMF

Non Negative Matrix Factorization (NMF) as its name suggests accepts non-negative values in the input matrix. NMF used for processing Multivariate information. Assume a matrix V of $x \times y$ dimensions, where every element of vector $V_{ij}$ is greater than 0. NMF takes input matrix V and subdivides it into two matrices W with dimension $x \times z$ and H with dimension $z \times n$. Every element of matrices $W_{ij}$ and $H_{ij}$ are greater than or equal to 0 and $z<\min(x,y)$. The relation between W, H, and V is stated as,

$$Z- \text{Inner dimension of W and H } z<\min(x,y) \}\text{. Each column of V i.e. } V_i \text{ can be calculated as,}$$

$$V = W \times H_i$$

Equation 3 states that V is equal to the sum of each column of W after being weighted by its corresponding row in H.

To quantify the quality of factorization cost function is defined. It can be defined using the square of Euclidean distance. Another measure is Kullback-Leibler divergence.

Un-normalized probabilities is updated by multiplicative update rules given by equation 4,

$$W = W^{\frac{MH^TW}{WHH}} \text{ and } H = H^{\frac{MW^TH}{HWW}}$$

NMF is very useful to represent Non Negative data quite well. It works similar to PCA but weights should be positive. It allows only additive combination. NMF is useful for Computer vision to classify images, track objects and reduce feature space for images. It is also found useful in applications like Text mining and speech denoising [12].

3. Topic Accuracy

Once the topic modeling method is applied, data is divided into clusters. There are questions on how to decide the number of topics and make the better assessment of model results. The goodness of topic model can be decided based on the following factors.

3.1. Number of Topics

If quality aspect is taken into consideration, ten topics cover most of the angles of data. If more topics are taken, there are chances of overfitting of the model. Taking too fewer topics will result in underfitting and it will not form proper clusters.

3.2. Log-likelihood score and Perplexity

The simple way to estimate the probabilistic model is to find Log-likelihood scores. They are used to compare different models. Log-likelihood scores are calculated for all the unseen documents with the presented learning rate. The model with the maximum log-likelihood score is considered as a better model. Perplexity or predictive likelihood is the way of measuring in what way model is able to predict a sample. It helps in determining an optimal number of topics. It is calculated by taking log-likelihood of text documents with topics resulted from the topic model[13].

A satisfactory model will have a high likelihood and resultantly low perplexity. Often predictive likelihood and human opinion are least correlated, and even sometimes slightly anti-correlated hence adaptable to some extent in business set up.

3.3. Topic Coherence

Topic coherence observes the collection of words in topics generated by the model and measures how informative is the topic. There are multiple measures that calculate coherence value in diverse ways. The evaluated topic coherence methods take the input of top words of the topic and calculate the sum of confirmation measure over pairs of words.
CUCI (Newman et al. 2010) extrinsic coherence measure is based on pointwise mutual information (PMI) using probabilities. Probabilities are judged based on interconnected word counts. Each individual word is dichotomized with every other word. UMass (Mimno et al. 2011) is an intrinsic coherence measure and needs ordered list. The sum up of UMass coherence will give top words on the topic. This is the fastest method and requires corpus.

Both intrinsic and extrinsic coherence measure calculates the coherence score which is the addition of pair score on the topic describing terms. The highest value for correlation to human topic coherence ratings was found in normalized PMI (NPMI) (Aletras et al. 2013) where elements of vectors are normalized. Confirmation measures taken in pairs of context vectors are vector similarities like cosine, Dice or Jaccard that are averaged over all pairs of a topic top words. Alternatively, topic coherence is computed as average similarity between top word context vectors and their centroid called as Ccen. UCI coherence does even better if done using NPMI. Coherence measure depends on the smoothing factor [13-17].

3.4. Interpretability

Once topic modeling is done post deciding the correct number of topics, applying perplexity and coherence measure, now it's time to visualize the topics. The tool named pyLDAvis [18] will come to help. This tool shows words that make up every topic, can able to visualize the strength of topic and relation between topics. The tool shows topic representation using the circle in the left panel and the horizontal bar chart on the right panel. The bars in the bar chart represents terms making up that topic. The left panel presents a global view of our model. It is possible to view topic term relation, how topics relate to each other and their strength. This tool helps to decide how strong and accurately LDA has performed and if required can make necessary improvements in the model.

4. Experiments and Results

The use of sections to divide the text of the paper is optional and left as a decision for the author. Where the author wishes to divide the paper into sections the formatting shown in table 2 should be used. Experiments consist of a collection of tweets from Twitter. Experiments are done using scikit-learn and Gensim package implementations.

4.1. Dataset

Three datasets Cauvery river, Lokpal bill and Rahul Gandhi consisting tweets on Cauvery, stress, Lokpal bill, and Rahul Gandhi are extracted using twitter API. The Tweepy package is used to access tweets. All dataset together contains total of 5500 tweets.

4.2. Data preprocessing and finding TF-IDF

Preprocessing of data takes the maximum amount of time. Preprocessing includes mainly removing Twitter handles, Punctuations, Numbers, Special characters, stopwords, Stemming and lemmatization. Preprocessed data can be encoded using CountVectorizer, word frequencies with TfidfVectorizer and Hashing Vectors etc. Experiments conducted using TfidfVectorizer. Term Frequency-Inverse Document Frequency uses the bag of the word model. Now each word is assigned with a unique number.

4.3. The Topic model with scikit-learn

The GridSearchCV methods in scikit-learn to find the optimal combination for the LDA model. The method does so by finding log-likelihood and perplexity. Grid search constructs multiple LDA model for all possible combination of presented value. Input will be the number of components/topics and learning decay with the value of less than one. Learning decay controls the learning rate. Log-likelihood scores are calculated with the learning rate of 0.5, 0.7 and 0.9. Figure 1 displays plotting of the number of topics on the x-axis and log-likelihood scores on the y-axis.
The blue line indicated learning decay with 0.5, the red line for 0.7 and green line for 0.9. It is clear from the above plot that the model performs better with the 0.9 decay. LDA model is built considering 0.9 learning rate and the number of topics equal to 10. Experimented LDA model gave best Log-likelihood Score as 64994.24915782703 and Model Perplexity as 26880.570081964765.

**TABLE 1: Top 15 Words in each topic using LDA, NMF and LSA using scikit-learn implementation**

| LDA   | NMF                                      | LSI                                      |
|-------|------------------------------------------|------------------------------------------|
| Topic 0: | boy cave hero black rescue honest | Topic 0: | track prevent corrupt dismal mode | Topic 0: | track protect prevent corrupt mode |
|       | tour human wrong thiacaverscu kid |       | track record corrupt dismal implement |       | record prevent corrupt track record |
|       | effort trust coach strong |       | whistleblow protect dismal |       | prevent corrupt dismal track record |
| Topic 1: | stay rest answer decide | Topic 1: | prevent corrupt dismal |       | implement whistleblow whistleblow protect |
|       | hate mess person decide judge decide |       | record prevent corrupt record prevent |       | Topic 1: | pass anna hazard anna hazard bandwagon |
|       | corrupt complaint office judge corrupt |       | implement whistleblow |       | bandwagon pass hazard bandwagon |
|       | office judge corrupt complaint corrupt |       | whistleblow protect |       | bandwagon pass elect anna hazard |
|       | complaint |       | whistleblower |       | elect anna hazard bandwagon |

Table 1 displays the top 15 words comprising topics using LDA, NMF, and LSA (Truncated SVD) model. These models are implemented using scikit-learn package. The entire dataset is divided into ten topics and each topic contains the following listed words. Topics generated on the dataset using NMF and LSA model shows a lot of similarities compared to topics generated by the LDA model. Next task is to visualize the similarities between various topics. Gensim package does not have NMF implementation at present.

Figure 2 shows a constructed LDA model visualized using pyLDAvis tool with Sklearn. Ten blue circles indicated each topic. Topics are represented according to strength with bigger to smaller circles. The red circle shows the selected topic and its bar chart representation. An advantage of using this tool is, will get a clear idea on topic space and overlapping between topics if present.
4.4. Topic model with Gensim

LSA and LDA topic modeling methods are implemented using Gensim package. Topics modeled using Gensim are shown in table 2. Gensim implementation shows words in each topic with its weight. Word “one” shows 0.007 strength in LDA and 0.308 LSI under Topic 0 which shows the variation in results by both models.

| TABLE 2: Sample topics using LDA and LSA in Gensim |
| LDA Model: | LSA Model: |
| Topic #0: 0.007*"one" + 0.006*"would" + 0.003*"first" + 0.003*"new" + 0.003*"time" + 0.003*"could" + 0.003*"may" + 0.002*"man" + 0.002*"two" + 0.002*"also" | Topic #0: 0.308*"one" + 0.280*"would" + 0.202*"said" + 0.175*"could" + 0.146*"time" + 0.144*"new" + 0.126*"man" + 0.125*"like" + 0.125*"two" + 0.120*"first" |
| Topic #1: 0.005*"would" + 0.004*"said" + 0.004*"new" + 0.003*"two" + 0.003*"man" + 0.003*"like" + 0.002*"even" + 0.002*"may" + 0.002*"could" | Topic #1: -0.294*"said" + 0.219*"may" + 0.179*"state" + - 0.176*"could" + -0.153*"would" + 0.143*"states" + 0.141*"new" + - 0.140*"like" + -0.138*"back" + - 0.105*"man" |

Coherence score will help to approximately calculate the number of topics. Table 3 shows coherence scores for both the models. Coherence score is found out using C_V and U_MASS methods for both models. It is applied to top words in topics and it finds word similarity scores in each topic. A good model will have coherent topics with a high score. C_V method is showing coherent topics. A close to zero UMASS score indicates higher topic coherence. Here negative value indicates coherent topics. Scores for both topic coherence methods show better results for the LDA model compared to the LSI model.

| TABLE 3: Coherence score for LDA and LSI |
| Model | U_MASS | C_V |
| LDA | -13.371590561 | 0.47531674689 |
| LSI | -9.6037764579 | 0.28887862599 |
Next, to know an optimal number of topics, many LDA models are built by changing the number of topic parameter. Coherence score is calculated for LDA with the variable topic count. The graph plotted is shown in figure 3.

The plot in Figure 3 shows that coherence scores increase with the number of topics more than 7. There is a moderate increase in coherence score for topics between 7 and 14. Though coherence scores are high for the number of topics from 15 onwards, it makes the model to underfit or overfit the data. LDA model in Gensim shows the satisfactory value for topic count 10.

![Figure 3. A plot of LDA model with Topics and their coherence score](image3.png)

Scikit-learn package implementation of LDA does not provide methods to measure coherence score. Taking suitable topic count into consideration, the LDA model is constructed and visualized using the pyLDAvis [17] in Gensim. The perplexity of LDA considering 10 topics in Gensim is -8.870244098334238. Figure 4 shows the pyLDAvis visualization of LDA. Blue circles in figure 4 represent words in each topic. Overlapping circles of topic 7, 9 and topic 10 indicates similarity.

![Figure 4. Visual of LDA topic model using pyLDAvis in Gensim with 10 topics](image4.png)

Comparing the results of LDA, LSA and NMF, it is found that topics generated by LDA excelled compared to the other two models. Comparing figure 2 and figure 4, it is clear that the topic formed in scikit-learn LDA is better than Gensim LDA.

Topics resulted from the scikit-learn LDA model is taken as input and the model is constructed using 3819 samples in training data and 1637 sample in test data. Logistic regression is applied to the clustered data. Results are shown for the threshold probability 0.41 in table 4.
TABLE 4: Result of logistic Regression Model

| Accuracy | Precision | Recall | F1 Score | AUC Score |
|----------|-----------|--------|----------|-----------|
| 0.8754   | 0.888     | 0.691  | 0.777    | 0.835295  |

ROC curve stands for Receiver operating characteristics is shown in Figure 5. ROC curve is used to check the performance of the model with respect to false positives rate and true positive rate. The total area of the ROC curve is 1. The curve in Figure 5 is more inclined towards the top left side indicates better performance. An area under the ROC curve is 0.91 which means numbers of positive instances are more with the constructed model.

![Receiver operating characteristics](image)

Figure 5. ROC curve

5. Conclusion and Future Scope
This paper canvassed the basic concept of topic modeling and its importance and applications. Paper presents an explanation of LDA, LSA, and NMF. These methods are implemented using scikit-learn and Gensim package. Results are verified using perplexity, log-likelihood and coherence score calculations. Experiments conducted on the datasets created from Twitter API. Topics constructed are then tested by creating the model using logistic regression, showing the accuracy of 87%. Topic modeling is a very important step and has a huge impact on model accuracy. A bad topic model will never give the desired accuracy. In future work, the coherence measure can be applied to other applications, not just topic models. On the downside, if dataset contents are inclined to negative polarity then coherence score will be almost the same.

6. References

[1] Eickhoff M, Neuss N. Topic modelling methodology: its use in information systems and other managerial disciplines. In Proceedings of the 25th European Conference on Information Systems (ECIS), Guimarães, Portugal, June 5-10, 2017 (pp. 1327-1347). ISBN 978-989-20-7655-3.
[2] Quanzhi Li, Sameena Shah, Xiaomo Liu, ArminehNourbakhsh, Rui Fang: Tweet Topic Classification Using Distributed Language Representations, In: International Conference on Web Intelligence, pp: 81-88, IEEE/WIC/ACM,(2016), ISBN: 978-1-5090-4470-2, doi:10.1109/WI.2016.0022.
[3] Kim J, Caire G, Molisch AF. Quality-aware streaming and scheduling for device-to-device video delivery. IEEE/ACM Transactions on Networking. 2015 Jul 31;24(4):2319-31.
[4] Ligutom C, Orio JV, Ramacho DA, Montenegro C, Roxas RE, Oco N. Using Topic Modelling to make sense of typhoon-related tweets. In 2016 International Conference on Asian Language Processing (IALP) 2016 Nov 21 (pp. 362-365). IEEE. doi: 10.1109/IALP.2016.7876006.
[5] Lim KH, Karunasekera S, Harwood A, Falzon L. Spatial-based topic modelling using wikidata knowledge base. In 2017 IEEE International Conference on Big Data (Big Data) 2017 Dec 11 (pp. 4786-4788). IEEE. doi: 10.1109/BigData.2017.8258542.

[6] Jiang M, Liu R, Wang F. Word network topic model based on word2vector. In 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService) 2018 Mar 26 (pp. 241-247). IEEE. doi: 10.1109/BigDataService.2018.00043.

[7] Barde BV, Bainwad AM. An overview of topic modeling methods and tools. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) 2017 Jun 15 (pp. 745-750). IEEE.

[8] Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Journal of machine Learning research. 2003;3(Jan):993-1022.

[9] Hoffman M, Bach FR, Blei DM. Online learning for latent dirichlet allocation. In advances in neural information processing systems 2010 (pp. 856-864).

[10] Landauer TK, Dumais ST. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review. 1997 Apr;104(2):211.

[11] Landauer TK, Foltz PW, Laham D. An introduction to latent semantic analysis. Discourse processes. 1998 Jan 1;25(2-3):259-84.

[12] Lee DD, Seung HS. Algorithms for non-negative matrix factorization. In Advances in neural information processing systems 2001 (pp. 556-562).

[13] Pritchard JK, Stephens M, Donnelly P. Inference of population structure using multilocus genotype data. Genetics. 2000 Jun 1;155(2):945-59.

[14] Newman D, Lau JH, Grieser K, Baldwin T. Automatic evaluation of topic coherence. In Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics 2010 Jun (pp. 100-108).

[15] Mimno D, Wallach H, Talley E, Leenders M, McCallum A. Optimizing semantic coherence in topic models. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing 2011 Jul (pp. 262-272).

[16] Aletras N, Stevenson M. Evaluating topic coherence using distributional semantics. In Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers 2013 Mar (pp. 13-22).

[17] Röder M, Both A, Hinneburg A. Exploring the space of topic coherence measures. In Proceedings of the eighth ACM international conference on Web search and data mining 2015 Feb 2 (pp. 399-408).

[18] Sievert C, Shirley K. LDAvis: A method for visualizing and interpreting topics. In Proceedings of the workshop on interactive language learning, visualization, and interfaces 2014 Jun (pp. 63-70).