News Topic Modeling with Latent Dirichlet Allocation

Alex Jian

April 21, 2023

1 Introduction

In this report, I investigate a Bayesian hierarchical model called Latent Dirichlet Allocation (LDA) and use it to build a topic model for a collection of news articles. News articles are written independently, but they generally share common topics and themes - news can be grouped into categories such as politics, health, and finance, and two distinct articles may cover the the same news event from different perspectives or multiple events from the same developing story. Humans reading news articles can easily identify those that cover the same topic, and Latent Dirichlet Allocation allows programs to approximate the same classifications automatically by analyzing the statistical properties of the observed words. This report will both explain how LDA works and apply it to construct a topic model over a large set of news articles.

2 Latent Dirichlet Allocation

2.1 Overview

Latent Dirichlet Allocation is a generative statistical model which explains a large number of observations through a smaller set of unobserved groups. In the context of machine learning and natural language processing, it is used to discover common topics underlying a collection of documents and then classify each individual document based on which of these topics it contains. In particular, we define a topic as a set of words that collectively suggest a common theme and use LDA to discover these topics based on the repeated appearance of these words. For example, in a news dataset, the words *election, president* or *law* might suggest a politics-related topic, while the words *technology, breakthrough* or *experiment* may suggest a science-related theme. Note that LDA only finds the words that best represent each topic - assigning meaningful labels to these topics requires domain knowledge on the user's part.

2.2 Probabilistic Graph Model

As a generative model, Latent Dirichlet Allocation assumes a statistical process by which documents are generated, then conducts Bayesian inference to learn what the parameters of this statistical process are most likely to be. The parameters of the process define the per-document topic and word distributions, which allow for the construction of the topic model over a full document collection.

The probabilistic graph model below shows the generative, topic-based process by which LDA assumes documents are generated. Before showing the diagram, we define variables as follows:

M is the total number of documents in the collection
N_i is total number of words in document i
α is a vector specifying the Dirichlet prior for the per-document topic distribution
β is a vector specifying the Dirichlet prior for the word distribution per topic
θ_i is the topic distribution for a single document (document i)
ψ_k is the word distribution for a single topic (topic k)
z_{i_j} is the topic referred to by the jth word in document i
w_{i_j} is the jth observed word appearing in document i

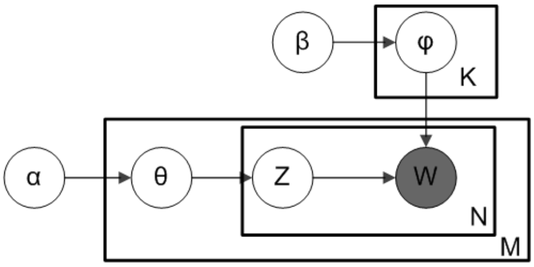


Figure 1:
Graph Model of LDA document generation process

Based on this diagram, we assume that the words for each document in a collection are generated as follows:

1) For each document i, choose $\theta_i \sim Dir(\alpha)$. Intuitively, we can think of this as the author deciding which topics the document will cover.

2) For each topic k, choose $\psi_k \sim Dir(\beta)$. Intuitively, once the topic is chosen,

certain words are much more likely to appear than others, a property specified by this per-topic word distribution.

3) We assume that each word in each document refers to one singular topic. For each word of each document, first choose its topic, $z_{ij} \sim Categorical(\theta_i)$. Then, choose the word from the topic's word distribution, $w_{i_j} \sim Categorical(\psi_{z_{i_j}})$.

Figure 1 illustrates this process at a high level. Note that W is grayed out, emphasizing that the words are the only observable variable, and the rest are latent variables which must be inferred through Bayesian Inference. Since each node in the graph is conditionally dependent only on its immediate predecessors, we can treat LDA as a Hierarchical Model and conduct inference using Bayesian methods. Note that this model assumes that all words within a document can be treated as exchangable, independent and identically distributed random variables. This assumption is largely untrue - words in sentences depend on each other and have context - but for the purposes of topic modeling, this assumption has still led to good results.

2.3 Prior Distributions

In LDA, it is important that the prior distributions accurately reflect our existing knowledge of text documents and their topic and word distributions. In this case, we observe that most documents (especially news articles) are rather focused, and contain only a few topics at most. We can account for this by a constructing sparse symmetric prior ($Dir(\alpha)$, $\alpha < 1$) for the per-document topic distribution, so that while there is no bias against any specific topic, there is bias against more uniform distributions that treat many topics as reasonably likely. The same holds true for the per-topic word distribution ($Dir(\beta)$, $\beta < 1$): for a given topic, certain words are more likely than others, a fact the prior should reflect.

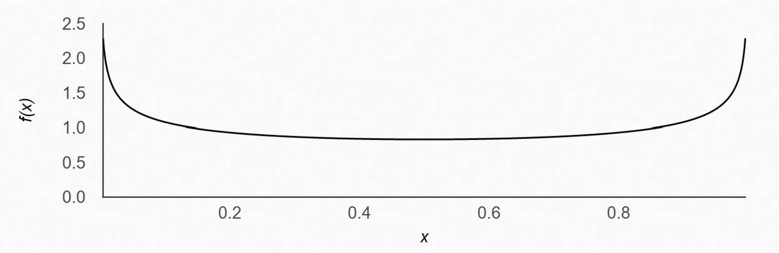


Figure 2:
Sparse Symmetric Prior for 1D Dirichlet Distribution

2.4 Inference

The mathematical problem that LDA tries to solve is computing the posterior distribution of the model's latent variables. We condition on the observed words ($\mathbf{w}$) as well as the α and β Dirichlet priors to compute the posterior per-document topic and topic-word distributions, θ and $\mathbf{z}$:

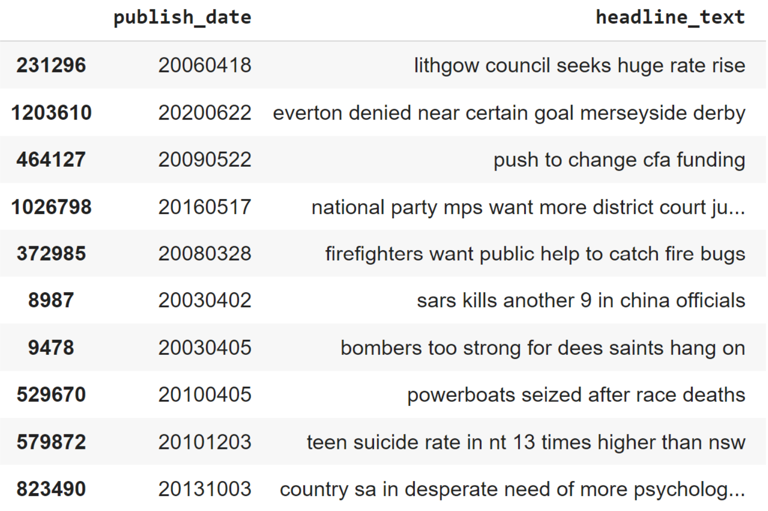
$$p(\theta, \mathbf{z}|\mathbf{w},\alpha,\beta) = p(\theta, \mathbf{z}, \mathbf{w}|\alpha,\beta)/p(\mathbf{w}|\alpha,\beta)$$

As is often the case for Bayesian posteriors, it is impractical to analytically compute the denominator this distribution. To account for this, we can use several numerical/computational methods, including MCMC with Gibbs Sampling or Variational Bayes. Since the LDA model is typically trained on large datasets, Variational Bayes is the usual choice due to its advantage in efficiency.

3 News Headlines Data Analysis

3.1 Dataset

I apply Latent Dirichlet Allocation to build a topic model over a set of over 1 million news headlines. The dataset is obtained from ABC news [link] and consists of roughly 1.2 million headlines along with their dates of publication.



	publish_date	headline_text
231296	20060418	lithgow council seeks huge rate rise
1203610	20200622	everton denied near certain goal merseyside derby
464127	20090522	push to change cfa funding
1026798	20160517	national party mps want more district court ju...
372985	20080328	firefighters want public help to catch fire bugs
8987	20030402	sars kills another 9 in china officials
9478	20030405	bombers too strong for dees saints hang on
529670	20100405	powerboats seized after race deaths
579872	20101203	teen suicide rate in nt 13 times higher than nsw
823490	20131003	country sa in desperate need of more psycholog...

Figure 3:
News Headlines Dataset

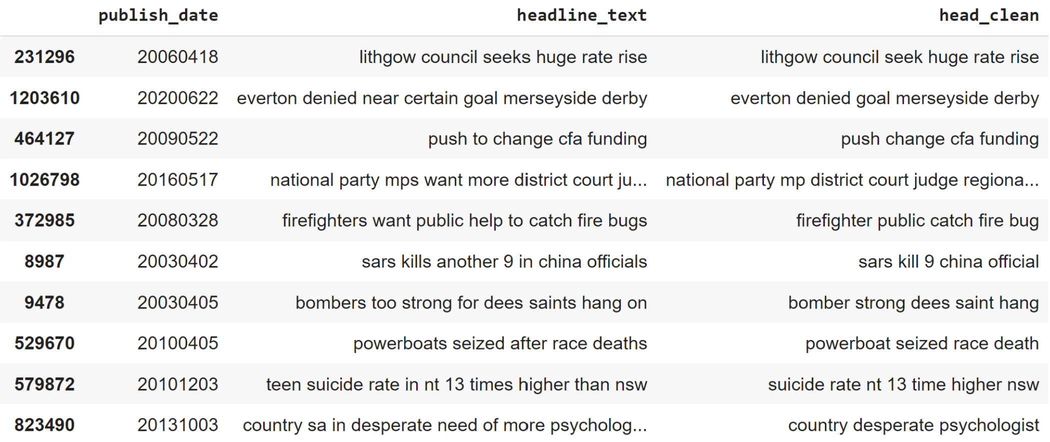
The goal is to use LDA to both identify the underlying topics and classify each headline by the topics they are most relevant to.

3.2 Data Preprocessing

Proper data preprocessing can significantly improve performance for the LDA model. The preprocessing I perform focuses on removing noise and allowing LDA to work with only the most important features of the data.

I begin by dropping the dates of publication from the dataset since they are incompatible with the LDA model. Then, I remove common words from each headline, known as stopwords, which carry low predictive power and will most likely not distinguish between topics. Stopwords are words like "I", "the", "and", "as", and so on.

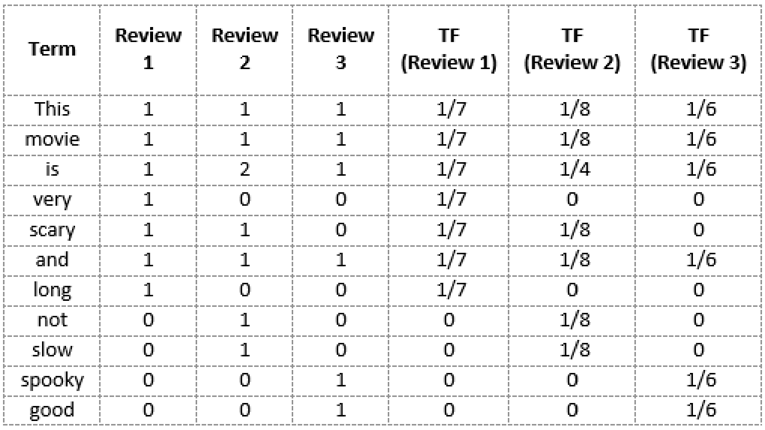
To reduce noise further, I use a technique called lemmatization. Many english words have multiple forms that roughly carry the same meaning. For example, the words "election", "elections", and "elect" are distinct words but likely refer to similar things. To ensure that the model recognizes this, I replace all words with their simplest (lemmatized) form so that the model treats multiple forms of the same word as the same. Below, I show the results of both the stopword removal and lemmatization preprocessing steps, where the column "head- clean" is the result of applying preprocessing on "headline-text":



	publish_date	headline_text	head_clean
231296	20060418	lithgow council seeks huge rate rise	lithgow council seek huge rate rise
1203610	20200622	everton denied near certain goal merseyside derby	everton denied goal merseyside derby
464127	20090522	push to change cfa funding	push change cfa funding
1026798	20160517	national party mps want more district court ju...	national party mp district court judge regiona...
372985	20080328	firefighters want public help to catch fire bugs	firefighter public catch fire bug
8987	20030402	sars kills another 9 in china officials	sars kill 9 china official
9478	20030405	bombers too strong for dees saints hang on	bomber strong dees saint hang
529670	20100405	powerboats seized after race deaths	powerboat seized race death
579872	20101203	teen suicide rate in nt 13 times higher than nsw	suicide rate nt 13 time higher nsw
823490	20131003	country sa in desperate need of more psycholog...	country desperate psychologist

Figure 4:
Stopwords and Lemmatization Preprocessing

The last preprocessing step is to apply the tf-idf transformation to the entire collection of headlines. Since tf-idf is not the focus of this report, I will not go deep into details, but tf-idf essentially weights words in a single headline by their frequency within that headline as well as their inverse frequency throughout the rest of the collection. Words with high tf-idf values are highly specific to their individual headlines and carry large predictive power. The end result of preprocessing produces a document-word matrix similar to the right half of the figure below.



Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

Figure 5:
Tf-Idf Matrix for a vocabulary and set of documents

3.3 Data Analysis Procedure

I train the LDA model on the preprocessed news headlines dataset. As described above, I set sparse symmetric parameters on the Dirichlet priors for per-document topic and per-topic word distributions. The number of topics must be chosen beforehand as a hyperparameter; After manual testing, I choose n = 6 topics. By default, I choose α and $\beta = \frac{1}{6}$ for my $Dir(\alpha)$ and $Dir(\beta)$ sparse symmetric priors. To fit the model, I use variational Bayes as explained above. The results are the posterior distributions for both the topic-word distributions and per-document topic distributions, θ and ψ.

4 Results

Below are the results for training an LDA model over this news headlines dataset. We have full posterior distribution for both the topic-word and the per-headline topic distributions. We can also directly visualize how some headlines were classified in order to check our model performance by hand.

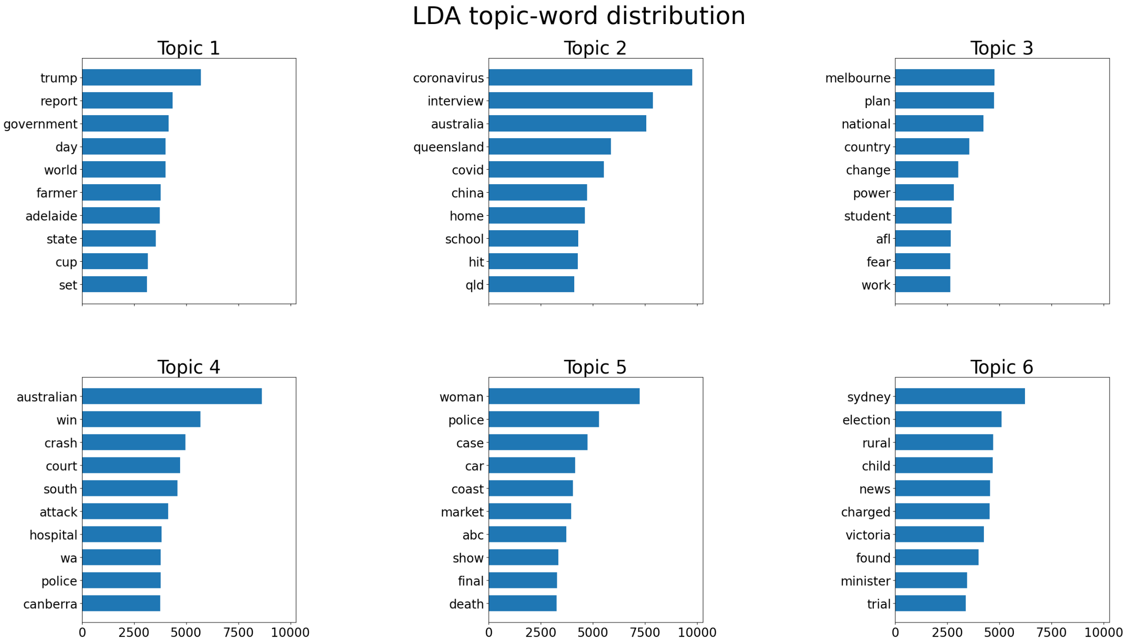


Figure 6:
Posterior Topic-Word Distribution for Top 10 Most Frequent Words

4.1 Topic-Word Distribution

The posterior topic-word distribution is shown above for each of the six topics for the top 10 highest likelihood words. We can see that most of the headlines are related to government/elections, COVID-19, local police/law enforcement issues or business/work. Again, note that LDA provides the word distributions but not meaningful labels for the topics.

By conjugacy, the posterior is still a Dirichlet, so we can interpret the values of the distribution as pseudocounts for the number of times the given word has appeared under the given topic. The normalized values can be interpreted as the probability that the word will appear at an arbitrary position in a document with that topic.

The pseudocount interpretation of the distribution's values also gives a measure of uncertainty for the posterior. Since high pseudocount values for a Dirichlet distribution translate roughly to a high sample size, they indicate low variance and high confidence in the posterior mean. Here, the pseudocount values are high - all larger than 2500 for the top 10 words. As a result, for each topic's top 10 words, we can be reasonably certain that the estimate is accurate as long as the training data is representative.

Below, we have the same topic-word distributions, but showing the values for some of the lowest-probability words. We can see that the pseudocounts are all around 0.2, which is a product of the Python implementation arbitrarily assigning positive probability to words in the vocabulary that were never observed under the given topic. This means the real observed probability was zero. Words that were never observed create major problems for the LDA and significantly increase posterior uncertainty, since future observations of previously unseen words will alter the distribution drastically.

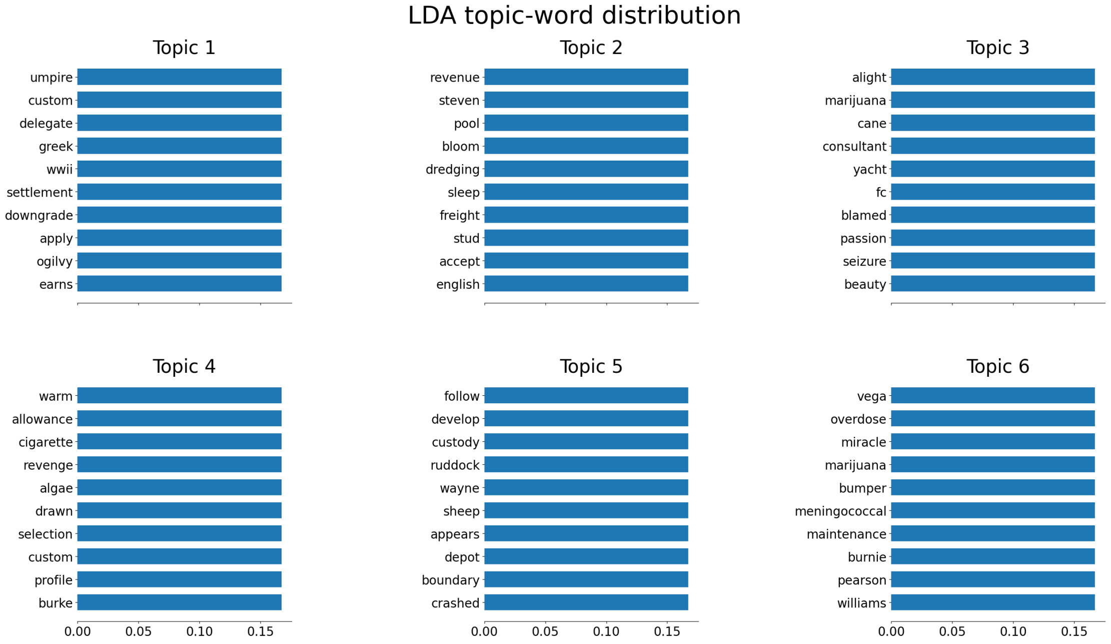
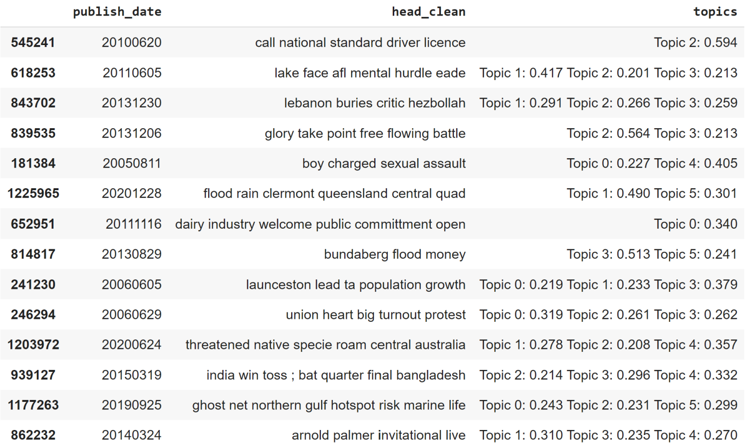


Figure 7:
Posterior Topic-Word Distribution for Infrequent Words

The arbitrary assignment of positive probability (called smoothing) can mitigate this problem, since zero probabilities will not collapse the likelihood function entirely, but the noisiness of language data in general causes this to remain a problem. The biggest issues are caused by rare terms that falsely indicate a certain topic. For example, the term ”Mark Cuban” appearing in a sports headline about the Dallas Mavericks can often be misclassified as a business-related topic, since the term has a low probability under sports but high probability under business, a difference that can overwhelm the contributions of remaining terms. In this case, for the lower-probability regions of our posterior, we have higher uncertainty for the results.

4.2 Per-Headline Topic Distribution

A sample of the per-headline topic distributions is shown below for selected headlines. Since this distribution is used for classification, I show the headline itself and the associated topics with high probabilities (> 0.2). By reading the headlines and comparing to the top-10 word distributions for their most likely topics, we can see that the model's classifications are reasonable.



	publish_date	head_clean	topics
545241	20100620	call national standard driver licence	Topic 2: 0.594
618253	20110605	lake face afl mental hurdle eade	Topic 1: 0.417 Topic 2: 0.201 Topic 3: 0.213
843702	20131230	lebanon buries critic hezbollah	Topic 1: 0.291 Topic 2: 0.266 Topic 3: 0.259
839535	20131206	glory take point free flowing battle	Topic 2: 0.564 Topic 3: 0.213
181384	20050811	boy charged sexual assault	Topic 0: 0.227 Topic 4: 0.405
1225965	20201228	flood rain clermont queensland central quad	Topic 1: 0.490 Topic 5: 0.301
652951	20111116	dairy industry welcome public committment open	Topic 0: 0.340
814817	20130829	bundaberg flood money	Topic 3: 0.513 Topic 5: 0.241
241230	20060605	launceston lead ta population growth	Topic 0: 0.219 Topic 1: 0.233 Topic 3: 0.379
246294	20060629	union heart big turnout protest	Topic 0: 0.319 Topic 2: 0.261 Topic 3: 0.262
1203972	20200624	threatened native specie roam central australia	Topic 1: 0.278 Topic 2: 0.208 Topic 4: 0.357
939127	20150319	india win toss ; bat quarter final bangladesh	Topic 2: 0.214 Topic 3: 0.296 Topic 4: 0.332
1177263	20190925	ghost net northern gulf hotspot risk marine life	Topic 0: 0.243 Topic 2: 0.231 Topic 5: 0.299
862232	20140324	arnold palmer invitational live	Topic 1: 0.310 Topic 3: 0.235 Topic 4: 0.270

Figure 8:
Posterior per-document Topic Distribution, showing topics which contributed significantly to the observed words (>20%)