My Comprehensive Evaluation

A Comprehensive Evaluation Report
Presented to
The Statistics Faculty
Amherst College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts
in
Statistics

Cassidy W. Maher

February 2019

Acknowledgements

First off, I'd like to thank Nick Horton and Chris Wild for their constant mentorship and support through all of my text analysis adventures over the past year. Both have been wonderful advisors to me, constantly showing support for my ideas and helping me to grow as a person both in and out of the classroom.

Secondly, I'd like to thank the entirety of the Statistics faculty for this wonderful education and support system that I've built over the past four years. They have provided me with so many wonderful opportunities and lessons that I am forever grateful for. I am so happy to have had access to so many great minds.

Lastly, I'd like to thank my mom for always helping to keep me grounded and her endless support in my academic pursuits. I could not have gotten through these four years without her constant presence. And thank you to my dad, grandma, and brother for providing many laughs and memories throughout my years at Amherst.

Table of Contents

List of Tables

List of Figures

Abstract

As an extension to a previous text analysis project, this study is an expository review of the theoretical basis of Latent Dirichlet allocation and how it is used to create latent topics within documents. This report seeks to provide a basic explanation to the theory behind the generative process of Latent Dirichlet allocation and how inference is then used to model data. A data set consisting of over 400 Emily Dickinson poems is used to show how Latent Dirichlet allocation can be applied to text data.

Introduction

Our daily lives are consumed by large amounts of text - we have hundreds of emails, access to numerous news sources that publish stories everyday, and a world of information appears at the click of a button. This can become overbearing, which is where the field of text analysis becomes very handy. Different methodologies within text analysis can boil large amounts of data into more bite sized summaries.

This study introduces one text analysis method, namely Latent Dirichlet allocation, first through a review of the basic information needed to understand the algorithm and then through an applied example. Latent Dirichlet allocation uses a generative probabilistic model to construct topics within a group of documents. By looking at the most common words within those designated topics, we can see a simplified version of the original text data and identify what different themes might be present.

The remainder of this report is organized as follows:

- Chapter One is an introduction to the basic assumptions and facts of Latent Dirichlet allocation and a glimpse at the generative model behind the method. It also covers a simple conceptual example and a look at how inference is used to generate topics. Finally, it covers the advantages of LDA over previous models as well as an explanation of its useful applications.
- Chapter Two is an applied example of Latent Dirichlet allocation to a subset of Emily Dickinson's poems. This example is a deeper dive into the information you can gather from the results of LDA.

Chapter 1

Latent Dirichlet Allocation

Latent Dirichlet Allocation (referred to as LDA from now on) is a specific type of topic model. The main idea of topic models is to "cluster" text data into groups to get a sense of different topics or sets of similar words that exist within a set of documents (Silge & Robinson, 2017). If multiple documents are partially made up from the same topic, they might share some commonalities. Further, if two documents do not share any of the same topics, they likely have completely different content.

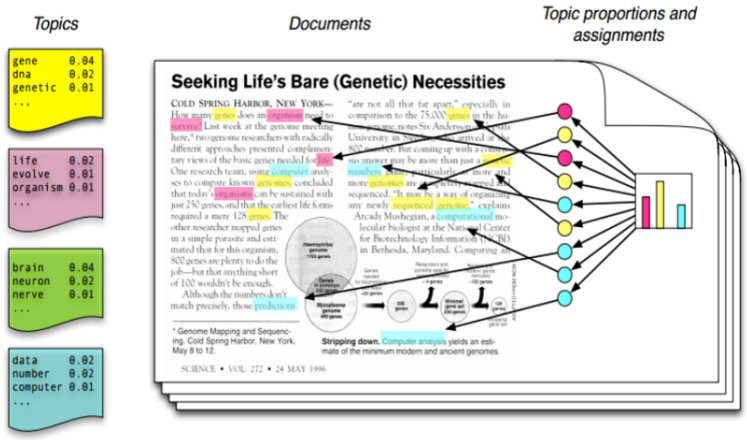


Figure 1.1: Building Blocks of LDA

There are two basic assumptions to remember about LDA (they are visualized in Figure 1.1):

1. Each document within a corpus is made up of a mixture of topics.
2. Each topic is made up of a distribution of words.

These two facts are the basis behind LDA as a mixture model. Mixture models use probability distributions to identify subgroups (topics) within a larger population

(documents) where the subgroups do not have to be explicitly identified in the data (Bonakdarpour, 2016). In our case, the subgroups (topics) have multinomial distributions within the larger distribution of the population (document). As we will see later, these topics and their distributions are assigned through a generative process using LDA.

1.1 Background

It is important to understand what each word in Latent Dirichlet Allocation means before introducing the theory of the model. Let's take a moment to break it down.

First off, a *latent variable* is essentially "hidden". It is not directly observed in a data set but, instead, can be inferred from variables that *do* exist in the data (Blei, Ng, & Jordan, 2003). In our case, topics are latent variables that can be inferred from the words in our documents. A few other parameters described later on are unobserved within the data itself.

Secondly, the Dirichlet distribution is a distribution over distributions. Although this may seem confusing at first, let's go back to the basics of LDA described above. Each document is made up of its own distribution of topics. The probability distributions of each topic within a document will sum to 1. In turn, each topic is made up of a distribution of words. The probability of each word over all topics will also sum to 1 (Blei et al., 2003).

The Dirichlet distribution has a few nice properties that make it particularly useful. First off, it is a member of the exponential family of functions (Blei et al., 2003). As we learned in STAT495, this makes it easy to work with when estimating parameters using methods such as maximum likelihood estimation. Additionally, the Dirichlet distribution is conjugate to a multinomial distribution (Blei et al., 2003). This means that if words and topics with multinomial distributions have a Dirichlet prior, then their posterior is also Dirichlet. This fact is also important for the inference portion of LDA. As you iterate over data during inference, the updated distributions remains Dirichlet. Mathematically, being a member of the exponential family and being part of a conjugate family are both very useful facts for inference procedures discussed later on.

A k-dimensional Dirichlet distribution has the following probability density function (the parameters θ and α will be covered later on) (Blei et al., 2003):

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k}(\alpha_i))}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta^{\alpha_1-1}...\theta_k^{\alpha_k-1}$$

Last in the term LDA, *allocation* references the process of assigning words to topics with certain probabilities. This also encompasses allocating topics to documents.

Before getting into the generative process of LDA, it is important to understand a key assumption of the model. The most important assumption of LDA is that each document is a "bag of words", meaning that the order in which the words appear in a document does not matter while using LDA. Within probability theory, this concept is termed *exchangeability* (Blei et al., 2003). This exchangeability is not equivalent to saying that the random variables involved are independent and identically distributed (iid). Instead, it means that they are *conditionally* iid when you condition on a latent parameter (in this case, topics) of a distribution (Blei et al., 2003).

1.2 LDA: A Generative Process

1.2.1 Terminology

In order to understand the process of LDA, we must first understand what parameters go into the model. First off, some basic parameters of LDA have been defined (Blei et al., 2003):

- A **word** is one item from a vocabulary that is indexed {1, . . . , V}. Words are stored in vectors where one element is equal to one (where the word is stored) and all others are set equal to zero (where all other words are stored). To put this in notation, the vth word has a V-vector where $w_v = 1$ and $w_u = 0$ for $u \neq v$.

- A **document**, denoted by $w = (w_1, w_2, ..., w_N)$, is a sequence of N words where w_n is the nth word in the sequence.

- A **corpus**, denoted by $D = w_1, w_2, ..., w_M$, is a set of M documents.

1.2.2 Generative Process

LDA is a *generative probabilistic model*. This means that it approximates the process that created the documents in a corpus by *generating* artificial documents. From these artificial documents, we can identify which parameter values best fit our data. The

generative structure of LDA proposed by Blei, Ng, and Jordan goes as follows for each document $\mathbf{w}$ in a corpus D:

1. $N \sim Poisson(\xi)$
   - This step samples the number of words that a document will have, where N is the number of words that document contains.
2. Choose $\theta \sim Dirichlet(\alpha)$.
   - $\boldsymbol{\theta}$ is the topic distribution for a document. As we learned in the overview, each document is made up of one or more topics.
   - α is the shape parameter for the Dirichlet distribution where values under one are sparse and towards the edges, one is uniform, and more than one is more dense in the middle.
3. For each of the N words w_n in the document,
   - choose a topic $z_n \sim Multinomial(\theta)$
     - Choose the topic that the word will come from.
   - choose a word w_n from $P(w_n|z_n, \beta)$.
     - Once the topic has been picked, choose a word given what you know about the topic distribution. This is a multinomial probability conditioned on the topic z_n.

Additional Notes: Word probabilities are parameterized by a k x V matrix β where $\beta = P(w_j = 1|z_i = 1)$ where k is the dimensionality of the Dirichlet distribution, which is assumed to be known and fixed, and V is the number of words in the vocabulary (i.e. the number of unique words across all of the documents). The β matrix contains the probability of a certain word being in a certain topic.

The following distributions are now defined (Blei et al., 2003):

1. On the topic level, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by:

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha)\Pi_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta)$$

2. On the document level, the marginal distribution of a document is given by:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha)(\Pi_{n=1}^{N} \sum_{z_n} p(z_n|\theta)p(w_n|z_n), \beta))d\theta$$

3. On an entire corpus level, the probability of a corpus is given by:

$$p(D|\alpha,\beta) = \Pi_{d=1}^{M} \int p(\theta_d|\alpha)(\Pi_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta))d\theta_d$$

1.2.3 Graphical Representation

One way to better understand the generative process of LDA is to look at its graphical representation displayed in Figure 1.2 (Blei et al., 2003). Each of the plates (rectangles) means something - the outer plate represents *documents* whereas the inner plate represents *the choice of topics and words within a document*. It should be noted here that the choice of topics and words within a document is *repeated*. This repetition is what allows a single document to made up of *multiple* topics and, consequently, words from those different topics (Blei et al., 2003). Another aspect of the figure to note is that the shaded circle above w represents the only observed variable while all other variables (α, β, θ, and z) are latent.

Figure 1.2 includes the following parameters:

- α: parameter of the Dirichlet prior on per-document topic distributions
- β: parameters of the Dirichlet prior on per-topic word distribution
- θ: document-topic distribution where θ_m is the distribution for document m
- z: a topic where z_{mn} is the topic of the nth word in document m
- w: a word where w_{mn} is the specific word
- M: the number of documents in the corpus
- N: the number of words in a document

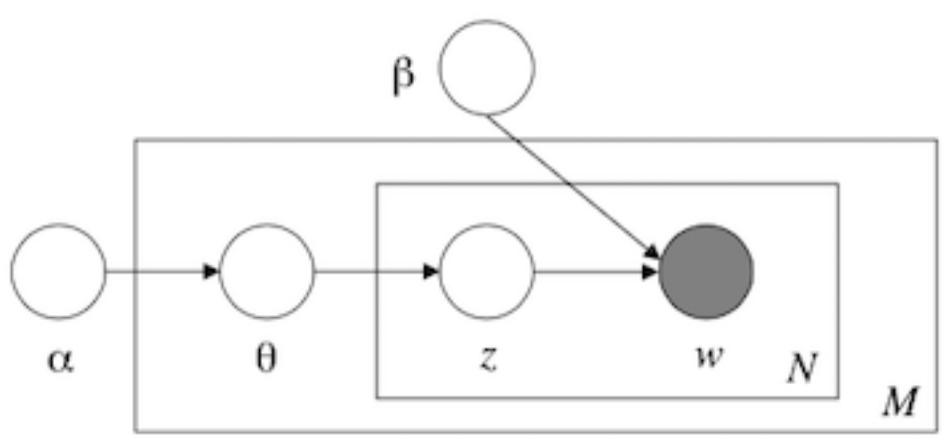


Figure 1.2: Graphical model of LDA

This diagram is a great way to visualize the steps that are happening during the generative process. Moving from inside out, we see that words (w) are determined by the topic (z) and the per-topic word distribution (β). Topics are determined by the document-topic distribution (θ). And finally, the document-topic distriibution is determined by α, the Dirichlet shape parameter that determines per-document topic distributions.

1.3 Conceptual Example: Farm Animals

To clarify what the generative process is really doing, let's walk through a simple example. This example uses animal emojis (specifically pigs, sheep, and cows) as representations of words and is inspired by an example in the book titled *The Little Book of LDA* (Tufts, 2018).

To start, let's think about the objective. LDA assumes that documents are created via the generative process outlined above. This assumes that documents are a mixture of topics and that topics are a mixture of words. We will start by specifying that we will generate five documents, $M = 5$, that are made up of a mix of two topics ($k = 2$). Since the example involves farm animal emojis, let's pretend that these documents are farms. As was stated earlier, there are three words/emojis in this example: pigs, sheep, and cows. The vocabulary is three words, so $V = 3$.

Next, we set the values of the Dirichlet parameter vector $\boldsymbol{\alpha}$ all to one and all of the values of the $\boldsymbol{\beta}$ to one as well. Setting each value to one means that both the document-topic distribution (related to α) and per-topic word distribution (related to β) are symmetric.

For this example, here are the topic-word distributions that were generated:

Table 1.1: Topic-word distributions

Topic	Pigs	Sheep	Cows	Total
1	0.39	0.13	0.48	1
2	0.16	0.51	0.33	1

After this information has been specified, it is time to generate documents! For step one of the generative process, the length of each of the five documents is drawn from a Poisson distribution with a mean of 10. We now have $N \sim Poisson(10)$ where N is the number of words in the document.

Once we have the length of the document N from step one, we choose $\theta \sim Dirichlet(\alpha)$ where the α values have equal weight. The α parameter values determine where the weight of the distribution is located (in the middle, at the corners, etc). Since the specified α values are both one, this Dirichlet distribution is uniform and each topic has an equal chance of being picked from the probability simplex (which is simply a vector of probabilities that adds to one) (Blei et al., 2003).

The last step in the generative process has two parts. For each word in a document, a topic is picked from $z_n \sim Multinomial(\theta)$ first. Once the topic has been picked, a word is picked from that topic's distribution. Remember that this distribution was specified earlier on and is shown in Table 1.1. These two steps repeat for each word in each document.

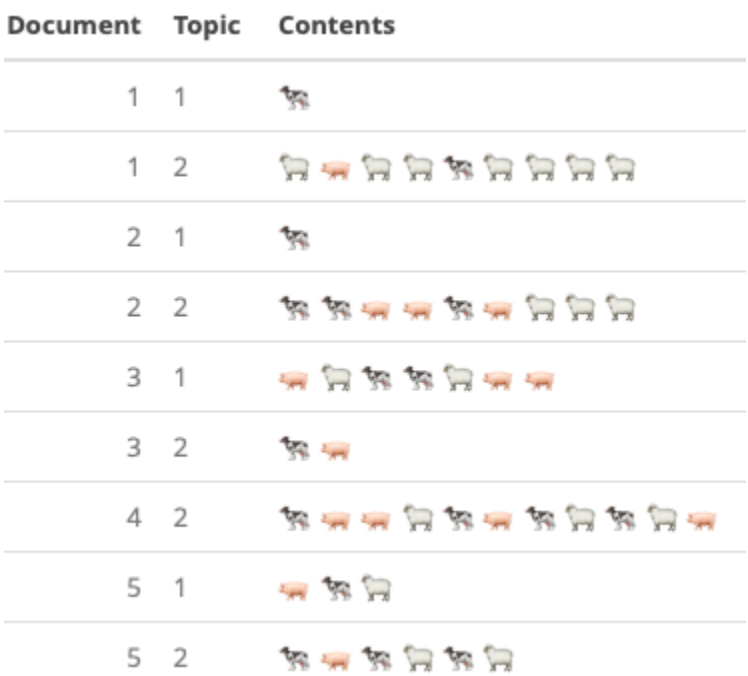
Before looking at the contents of the documents created, let's get a sense of the document-topic distributions. Table 1.2 shows this distribution.

Table 1.2: Document-topic distributions

Topic	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
1	0.25	0.16	0.85	0.02	0.49
2	0.75	0.84	0.15	0.98	0.51
Total	1	1	1	1	1

Notice that the two topic probabilities θ for each document sum to one. Some of these distributions are quite different, with some documents coming entirely from a singular topic.

Let's take a look at the generated documents in Figure 1.3 and work through the thought process again. Each document has two lines in the table - one for the words/animals generated from topic one and one for those generated from topic two. Document one, shown in the first two lines in Figure 1.3, has ten elements. This was determined by sampling from the Poisson distribution with a mean of ten and is simply a coincidence that it has exactly ten words. One of the words/animals came from topic one and nine came from topic two. The first topic generated one cow, which is not super surprising because, when drawing from topic one, there is a 48% chance that the word picked is a cow. The second topic generated seven sheep, one pig, and one cow. Remember the topic two distribution: 16% of words are pigs, 51% are sheep, and 33% are cows. These results were fairly expected given that sheep should be the most common, but they are a bit off of the theoretical distribution (shown in Table 1.1).



Document	Topic	Contents
1	1	🐄
1	2	🐑🐖🐑🐑🐄🐑🐑🐑🐑
2	1	🐄
2	2	🐄🐄🐖🐖🐄🐖🐑🐑🐑
3	1	🐖🐑🐄🐄🐑🐖🐖
3	2	🐄🐖
4	2	🐄🐖🐖🐑🐄🐖🐄🐑🐄🐑🐖
5	1	🐖🐄🐑
5	2	🐄🐖🐄🐑🐄🐑

Figure 1.3: Documents created via generative process of LDA

1.4 Inference

You may have noticed that the process outlined above is the generative process that *creates* a document based on probability distributions created for words and topics. The next step in LDA is to navigate *backwards* through this process to find the parameter values that best fit the data.

This process is called *learning* and is the inference step of LDA. The goal of inference in LDA is to identify the posterior distribution (shown below) of the hidden variables when presented with documents. This distribution is intractable and cannot be calculated directly, so inference is used as an approximation of the distribution.

The posterior distribution of the data is given by (Blei et al., 2003):

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

and creates problems when you marginalize over the hidden variables, which results in:

$$p(\mathbf{w}|\alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\Pi_i \Gamma(\alpha_i)} \int (\Pi_{i=1}^{k} \theta_i^{\alpha_i - 1})(\Pi_{n=1}^{N} \sum_{i=1}^{k} \Pi_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j}) d\theta$$

This is intractable because we cannot compute the summation over latent topics with θ and β together (Blei et al., 2003).

The inference procedure proposed by Blei, Ng, and Jordan to deal with this problem is variational expectation-maximization (VEM). The first step is to simplify the model and get rid of the issues that make it intractable - in this case, a new family of distributions q is created to replace the intractable posterior distribution p. The topic distribution for a document θ is replaced with a new *variational parameter* γ. Additionally, the multinomial distributions of words over topics are replaced with $(\phi_1 ... \phi_N)$.

The new variational distribution is given by:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma)\Pi_{n=1}^{N} q(z_n|\phi_n)$$

VEM has two steps - expectation and maximization - briefly outlined below (Blei et al., 2003):

- E-step: After specifying the new parameters, a procedure is used to minimize the Kullback-Leibler divergence between the new distributions q and the true posterior p. This process is done at the document level and produces the optimal values for γ and ϕ that approximate the posterior.
- M-step: Once the distributions have converged and the variational parameters have been identified, their estimates are used to update the model estimates of the actual topic parameters $\boldsymbol{\theta}$.

The E and M steps are iterated until the lower bound of the log likelihood converges.

1.5 Advantages and Applications

There are three key advantages covered by Blei, Ng, and Jordan that LDA has over previous generative topic models:

- LDA is a three level model that allows the topic node to be sampled repeatedly, and therefore a single document can be comprised of *multiple* topics. This is an advantage over two-level models that only sample one Dirichlet distribution for a corpus, one multinomial clustering variable for each document, and a set of words for the document conditional on the cluster variable. This limits the number of topics assigned to each document to just one.
- Many of LDA's predecessors have issues with overfitting. LDA doesn't have the same issue because it treats the topic mixture weights as a k-parameter hidden

random variable as opposed to a set of individual parameters that are explicitly linked to the training set.

- LDA generalizes easily and well to new documents because it assigns nonzero probabilities to words that have never been seen before. Most of its predecessors assigned zero probability to words not in the vocabulary, so they didn't perform well on unseen documents that contained new words.

Beyond recognizing the advantages, it is important to think about the possible applications of LDA within the field of text analysis as well as in the real world. Topics generated from sets of words similar to each other can be indicative of what a document or set of documents is all about. Topics, though not named automatically, should be defined to a point where a human can identify what that group of words might be about. If we know the topic(s) present in a document, we can summarize what might be included in a document. For example, if I have 20,000 unread emails that I run LDA on, I can identify the main themes of what people are contacting me about without having to read them all.

Here are a few real world examples of LDA at work:

- The New York Times has used LDA to make content recommendations based on what articles are about and who is reading them (`https://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/`).
- LDA has been used to trace the evolution of topics in *Science* and how they change over time. The researchers grouped *Science* articles by year and then analyzed that year's topics based on which topic's were most prominent in the previous year (`https://mimno.infosci.cornell.edu/info6150/readings/dynamic_topic_models.pdf`).
- ...and here is an *entire* book of applications of topic models (`https://mimno.infosci.cornell.edu/papers/2017_fntir_tm_applications.pdf`).

Chapter 2

LDA with Emily Dickinson Poems

Figure 2.1: Emily Dickinson Poems

The farm example demonstrated in Section 1.3 showed how the assumed generative process of LDA "creates" topics and documents. As we learned in Section 1.4, what LDA really does is try to trace backwards through the generative process to determine the document-topic and topic-word distributions. The following section is a more complex example on a subset of just over 400 poems written by Emily Dickinson that demonstrates the inference process LDA uses to identify distributions within a corpus of documents ("Project Gutenberg," 2004). According to the Emily Dickinson Museum website, she wrote on topics including nature, religion, law, music, and medicine ("Emily Dickinson Museum," 2009). Each of these topics is fairly distinct from the rest – we will see this arise in LDA analysis.

2.1 Data Preparation

Before starting an analysis, it is important to understand what data you are working with and what structure it is. Each line in the Dickinson data set holds the contents of one document. Here is an example of one of Dickinson's poems used in this analysis:

Table 2.1: Example Poem

MAY-FLOWER.
Pink, small, and punctual, Aromatic, low, Covert in April, Candid in May, Dear to the moss, Known by the knoll, Next to the robin In every human soul. Bold little beauty, Bedecked with thee, Nature forswears Antiquity.

2.1.1 Exploratory Analysis

To get a sense of what words Emily Dickinson used the most, I identified the top five most frequently used words in this set of poetry. This list has "stop words" removed - these are words like "the", "like", and "about" - that don't carry any significant contextual meaning in a piece of text.

Table 2.2: Top 5 most frequent words in our data

Word	Count
day	86
life	53
night	51
time	50
heaven	47

We also want to have a sense of how many words comprise Emily Dickinson's vocabulary in our data set. After stop word removal, there were 4920 *unique* words in the poems. In this set of vocabulary, $V = 4920$. Emily Dickinson uses at lot of different words! This means some words may only appear once or a handful of times.

2.1.2 Preprocessing and Data Structure

We are interested in categorizing the words in each document into topics. Recall the "bag of words" assumption where it is assumed that the order of words within a document does not affect the analysis. Therefore, nothing about the structure of the text matters other than the words themselves. All punctuation and numbers are removed before running any analysis. Additionally, every letter is transformed to lowercase and any extra white space is stripped out.

The most important step to preprocessing the data is to remove all stop words. Stop words are common words in the English language such as "the" and "about" that don't tell us much about the unique topics in a document. I also removed some older words such as "thee" and "thou" often used by Dickinson since the stop words list didn't have them. If we were to keep stop words in, most words in each topic would be terms that don't tell us anything insightful about the topics in the document.

The data is transformed into a document term matrix after preprocessing. A document term matrix has one row for each document and one column for each word in the vocabulary of the corpus. Remember that the vocabulary is the set of all unique words leftover after preprocessing the data. The values in the matrix count how many times a specific word appears in each document. Since most documents don't contain the vast majority of the words in the vocabulary, this matrix is sparse. See an example of the first three rows and 5 columns of the document term matrix for our data set below:

Table 2.3: Example document term matrix

Document	day	death	god	heaven
14	1	0	0	0
294	1	0	0	1
55	2	0	0	1

2.2 Application of LDA to Dickinson Poems

After preprocessing and exploring the data, LDA analysis is run on the 400+ poems. Remember that the topics created are latent - "unobserved" - so we decide how many should be created. I arbitrarily decided to create three topics for this data set.

Once the topics have been generated, we can see what they are about by looking at the words with the highest probability of being in that topic. We get the word-topic probabilities by looking at the β matrix. The higher the β value, the higher the probability of that term will be in that topic.

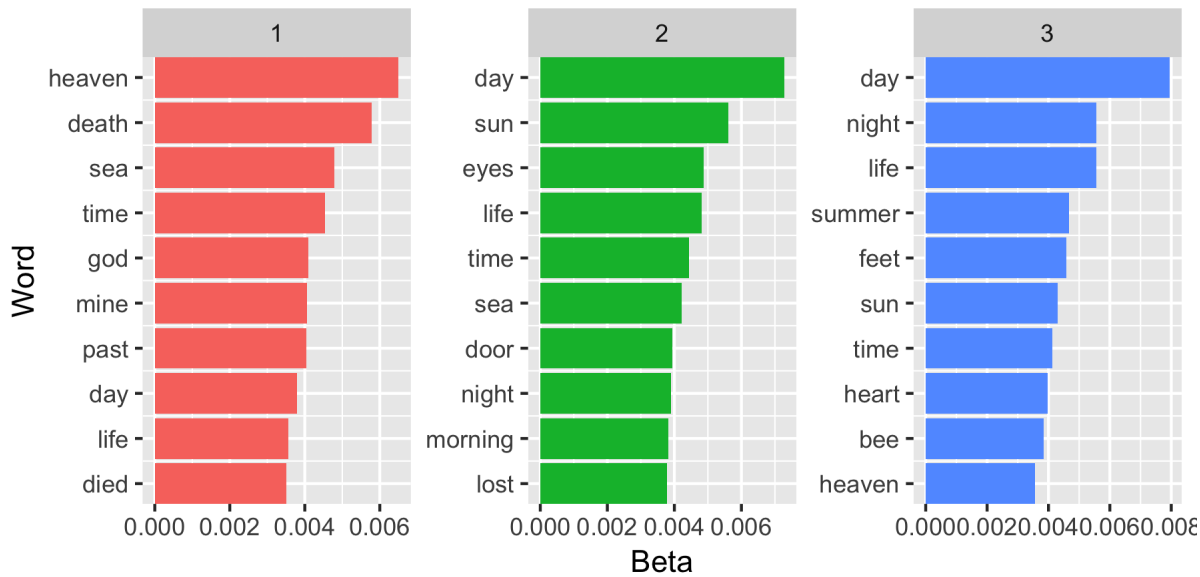


Figure 2.2: Top Word-Topic Probabilities

Figure 2.2 shows the top ten words for each topic. After some inspection, it seems that these groups of words aren't very different. In fact, all three topics contain the words "time", "day", and "life" within their top ten words. Additionally, only 50% of the top ten words in topic one are unique while just 40% are unique in topics two and three. The ambiguity between topics may be due to Emily Dickinson's use of vocabulary and writing style as a whole but it may also be attributed to choosing the wrong number of topics to create. There are better ways to choose k than arbitrarily doing so.

That being said, let's try to parse out some underlying themes that may be present in each topic. The unique words from topic one are "death", "god", "mine", "past", and "died". This topic seems to be on the negative side – maybe it focuses on themes of death. In topic two, there are words like "life", "morning", and "night" that all

reference time in some way. Lastly, topic three has “summer”, “bee”, and “sun”, which might relate to being in nature and happier times.

To check that the LDA worked properly, we can check whether the probabilities of each word being in a topic sum to one as they should. I won’t print that table here, but the math did work out properly. We also want to check that the probability of a topic being in a document sums to one over all three topics. This can be see in Table 2.4.

To make this more clear, let’s take a look at the probability distributions within the poem titled May-Flower that was introduced in the beginning of Chapter 2. LDA assigned a probability for each topic to be in this document and these can be found in Table 2.4. A quick inspection tells us that this poem is made up of about 34% of words from topic one and 66% of words from topic two (with topic three probability being nearly zero). Based on what we guessed about topic one, the poem might contain themes surrounding death or negative thoughts. Similarly, based on topic two, the poem might contain themes relating to time, whether that be a time of day or a time during one’s life.

Table 2.4: Topic distribution for May-Flower

topic	gamma
1	0.338
2	0.660
3	0.00149

Let’s look at the poem again (shown in Table 2.5) and see what it’s about. Like the title might give away, this poem is about a flower in May. It has some elements of time when April transitions to May but it appears to focus more on nature. Topic three, the one that seems most related to nature, had the smallest probability for this document. This might be another sign that the topics created in this specific analysis aren’t very meaningful.

Table 2.5: Example poem pt. 2

MAY-FLOWER.
Pink, small, and punctual, Aromatic, low, Covert in April,

MAY-FLOWER.

Candid in May,
Dear to the moss,
Known by the knoll,
Next to the robin
In every human soul.
Bold little beauty,
Bedecked with thee,
Nature forswears
Antiquity.

This analysis of Emily Dickinson poems didn't turn out to be very meaningful in terms of the three topics LDA created. This might be a sign that Dickinson writes on abstract topics that aren't easily differentiated or it could be that I simply didn't pick a good number of topics to create. Although the topics didn't work out well, we can see that the basics of LDA were upheld. We saw that the topic distribution over one document summed to one. Additionally, although I didn't include the output, I confirmed that the word probabilities within each topic summed to one. These two facts confirm the two basic tenets of LDA: documents are made up of a mixture of topics and topics are made up of a mixture of words.

Conclusion

In this extension of previous coursework and research, I move from simple text analysis to a more complex process of using Latent Dirichlet allocation as a method of topic modeling with text data. For my first task, I explored the generative process and graphical model involved in LDA. Then I did a review of the basic inferential methods that can be used to model the data given the generative structure assumed for a document. Secondly, I used a subset of Emily Dickinson's poems as an aid to demonstrate the building blocks and results of LDA. I focus on how modeling text data into smaller, "hidden" subsets can be useful for summarizing and understanding large amounts of qualitative data.

Appendix A

Chapter One Appendix

```r
# Determine the number of topics and documents to create
k <- 2 # number of topics
M <- 5 # number of documents

# Create the vocabulary: pig, sheep, cow emojis (farm theme)
vocab <- c(ji("pig2"), ji("sheep"), ji("cow2"))
```

```r
alphas <- rep(1,k) # topic document dirichlet parameters

betas <- rep(1,length(vocab)) # dirichlet params for topic-word dists

set.seed(13)
phi <- gtools::rdirichlet(k, betas)
```

```r
xi <- 10 # average document length
N <- rpois(M, xi) # sample words in each document using poisson
```

```r
# Create dataset with documents, words, and topics
ds <- tibble::tibble(doc_id = rep(0,sum(N)),
            word   = rep('', sum(N)),
            topic  = rep(0, sum(N)),
            theta_a = rep(0, sum(N)),
            theta_b = rep(0, sum(N))
)

row_index <- 1
for(m in 1:M){ # number of documents
  theta <-  gtools::rdirichlet(1, alphas)
  for(n in 1:N[m]){ # for how many words will be in that document,
    # select topic from multinomial distribution
```

```
    topic <- which(rmultinom(1, 1, theta) == 1)
    # sample word from topic
    new_word <- vocab[which(rmultinom(1, 1, phi[topic, ]) == 1)]
    # add to row that represents that document
    ds[row_index,] <- c(m, new_word, topic, theta)
    row_index <- row_index + 1
  }
}
```

```
ds %>%
  mutate(doc_id = as.numeric(doc_id)) %>%
  group_by(doc_id, topic) %>%
  rename(Document = doc_id, Topic = topic) %>%
  summarise(Contents = paste(word, collapse = ' '),
            Topic1 = round(as.numeric(unique(theta_a)), 2),
            Topic2 = round(as.numeric(unique(theta_b)), 2)) %>%
  arrange(Document) %>%
  knitr::kable() %>%
  kable_styling()
```

Appendix B

Chapter Two Appendix

```r
# Get all files in the folder
files <- list.files("gutenberg")
files <- paste("gutenberg/", files, sep = "")

# Read data in and store in a data frame
tbl_list <- lapply(files, function(x)
  {paste0(readLines(x), collapse = " ")})
data_set <- Reduce(rbind, tbl_list)

# Remove some extraneous info / stuff in wrong format
data_set <- data_set[-c(26, 116:118, 281),]

# Remove the roman numerals before each poem
data_set <- gsub("^.*?\\.","", data_set)
```

```r
# Unnest tokens (we're just doing unigrams here)
# Remove stop words
exploratory_data <- as.data.frame(data_set) %>%
  mutate(data_set = as.character(data_set)) %>%
  unnest_tokens(word, data_set) %>%
  anti_join(stop_words)
```

```
Joining, by = "word"
```

```r
# Find top 10 words
top5 <- exploratory_data %>%
  filter(word != "thee" & word != "till" & word != "thou" &
           word != "thy" & word != "thine") %>%
  group_by(word) %>%
  summarise(count = n()) %>%
```

```
  arrange(desc(count)) %>%
  filter(row_number() <= 5)
```

```
exploratory_data %>%
  filter(word != "thee" & word != "till" & word != "thou" &
           word != "thy" & word != "thine") %>%
  group_by(word) %>%
  summarise(count = n())
```

```
data <- as.data.frame(data_set)

# Process data
dickinson <- Corpus(VectorSource(data$data_set))
dickinson <- tm_map(dickinson, removePunctuation)
dickinson <- tm_map(dickinson, removeNumbers)
dickinson <- tm_map(dickinson, tolower)
dickinson <- tm_map(dickinson, removeWords,
                    c(tidytext::stop_words$word, "thee", "till",
                      "thou", "thy", "thine"))
dickinson <- tm_map(dickinson, stripWhitespace)
```

```
dickinson_dtm <- DocumentTermMatrix(dickinson)
inspect(dickinson_dtm)
```

```
# Dickinson LDA - 3 topics

# Had issues due to some documents containing zero words
# Find documents with zero words and remove them
row_sum <- apply(dickinson_dtm, 1, FUN=sum) #sum by row each row
dickinson_dtm <- dickinson_dtm[row_sum != 0, ]

dickinson_lda <- LDA(dickinson_dtm, k = 3, method = "VEM",
                     control = list(seed = 15))
```

```
# Find topics

# One topic per term per row format
dickinson_topics <- tidy(dickinson_lda, matrix = "beta")
```

```
# We can check that the topics sum to 1 as they should
dickinson_topics %>%
  group_by(topic) %>%
```

```r
  summarise(total = sum(beta))

# Document-topic probabilities
dickinson_documents <- tidy(dickinson_lda, matrix = "gamma")
dickinson_documents

dickinson_documents %>%
  group_by(document) %>%
  summarise(prob = sum(gamma))
```

```r
dickinson_documents %>%
  filter(document == 45)
```

References

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 993–1022.

Bonakdarpour, M. (2016). Introduction to mixture models. *fiveMinuteStats*. `https://stephens999.github.io/fiveMinuteStats/intro_to_mixture_models.html`; GitHub.

Emily Dickinson Museum. (2009). *Emily Dickinson Museum*. Trustees of Amherst College. Retrieved from `https://www.emilydickinsonmuseum.org/`

Project Gutenberg. (2004, May). *Emily Dickinson Poems*. Project Gutenberg. Retrieved from `https://www.gutenberg.org/files/12242/12242-h/12242-h.htm`

Silge, J., & Robinson, D. (2017). *Text Mining with R*. Sebastopol, CA: O'Reilly Media.

Tufts, C. (2018, January). The little book of lda. *The Little Book of LDA*. Retrieved from `https://ldabook.com/index.html`