Chapter 1

Introduction

To motivate my paper, indulge me with the following scenario. Imagine having a stack of documents about yay high to review tonight. Your boss wants a document-by-document summary tomorrow and you are not even sure where the stack is. Luckily, there is an online PDF, but you are scrolling down to no end. While I cannot promise you that end-of-year bonus, if you read along, you can learn how to review those documents automatically and in no time.

They say we are in the Information Age, a historical stamp characterized by economic, political, and psychological domination of "information". Advancements in hardware and software engineering have increased accessibility to the world of ideas, but handling the throughput of information is an increasingly difficult endeavor. Moreover, the continued digital connectivity of our globe has made the management and dissemination of information all the more critical. So while the engineers and bloggers work at the front end to increase access and throughput, some mathematicians work at the back end of processing knowledge out of information. "Information retrieval" is a term encapsulating this endeavor. In this paper, I study the mathematical properties of one particular information retrieval model commonly known as Topic Modeling. I focus on modeling the "topics" of digitized text and image data. While this study uses tools from the mathematical and statistical disciplines, the consequences of Topic Modeling are far, far reaching. They say.

This paper is organized as follows. The first chapter takes a birds eye view of Topic Modeling as a concept and introduces the Latent Dirichlet

Allocation (LDA) algorithm. Chapter 2 dives into the statistical foundations underpinning Topic Modeling, building up to the LDA algorithm. Chapter 3 expands on the implementation of the LDA model. Chapter 4 then presents some applications of LDA on textual and imagery data.

1.1 Modeling "Topics"

Topic Modeling is a statistical technique for analyzing data and decomposing it into "topics" [1]. The framework built around Topic Modeling is the endeavor to parse large dimensional data with a lot of information into a collection of smaller, representative information sources or "topics." Each "topic" is a distribution of a few data points capturing some meaningful aspect of a larger set of data points. In the context of textual data, topics are distributions of words capturing the semantics of an underlying document. Below is an illustrative example of three topics. The numbers next to each word is the probability of finding that word in the corresponding topic.

T_1	T_2	T_3
women (0.08)	strawberry (0.05)	lead (0.09)
glass (0.03)	integrate (0.05)	market (0.06)
freedom (0.01)	invest (0.03)	system (0.05)
wage (0.01)	freedom (0.02)	health (0.01)

Table 1.1: An illustration of three topics

Note that topic models do not necessarily produce an explicit "topic" in the traditional sense. Instead of a phrase or a grammatically structured sentence, "topics" are defined by the distribution of words assigned to them. It is then up to the human (or artificial intelligence as it may soon be) to interpret the collection of words. Can you guess what these three topics could be? A possible "solution" is given below. The numbers next to each candidate topic is the probability of finding that topic in the target document.

Wage Gap (0.47)	Sales Ad for Blender (0.28)	Lead Pollution (0.16)
women (0.08)	strawberry (0.05)	lead (0.09)
glass (0.03)	integrate (0.05)	market (0.06)
freedom (0.01)	invest (0.03)	system (0.05)
wage (0.01)	freedom (0.02)	health (0.01)

Table 1.2: A potential interpretation of three topics

The statistical framework behind Topic Modeling is based on hierarchical Bayesian models. Hierarchical models are used on data believed to have structurally different sub-components [6]. For instance, suppose you want to estimate the mean of student GPA's across the Claremont Consortium. You could randomly sample some number of students and get an average GPA, but that statistic would be uninformative about each college. For one, the students in your sample may not be representative of the relative sizes of each of the 5 colleges. Moreover, data on, say, Harvey Mudd may skew the average GPA one way and Pomona the other (I leave it up to the reader to guess which way). A hierarchical model sampling from the Claremont Consortium would take into account the structural differences between the 5 colleges while permitting an overall dependency across them. This hierarchical framework is useful for retrieving information from documents believed to be a mixture of inter-related themes.

The convenience of topic modeling algorithms becomes apparent as information sources become more digitized and centralized on the web. Software programs with topic models like *Gensim* in Python can scrape large amounts of digitized data and produce better manageable information sources in the form of topics. The applications of topic models are therefore readily welcomed in fields like bioinformatics, policy research, and the social sciences. This thesis explores its applications in textual and visual data, building the conceptual framework of topic models on the former and extending its capabilities with the latter. While many algorithmic variations of topic models exist, I primarily focus on the Latent Dirichlet Allocation algorithm for its popularity in the field.

1.2 Latent Dirichlet Allocation

The general idea behind LDA can be broken down by nomenclature; (1) the outputs of the algorithm are the *latent* or hidden topics assumed to be present in a document collection; (2) these latent topics are distributed according to a *Dirichlet* probability function. This function is discussed in greater detail in the subsequent chapter. (3) the principle mechanics of LDA concern the *allocation* of words to topics and topics to documents.

LDA is a generative model; it generates the relationship between words and their corresponding topics rather than relying on pre-labelled topics [1]. Consequently, the entire operation of the algorithm hinges on the assumptions made about the relationship between words and their topics. Note that the generative model does not make an assumption about the topics themselves, but how documents are generated from those "ghost" topics. The actual topics outputed by LDA ultimately depend on reverse engineering the generative assumption. An example illustrating this process is in order.

Let's say you are working with a corpus consisting of the 5C newspaper *The Student Life*, and you want to automatically produce meaningful topics from each document. The inputs into the LDA model are the words in each document.


THE STUDENT LIFE

The newspaper of the Claremont Colleges, est. 1889

April 3, 2019 8:04 pm

OPINION: Don't use astrology as an excuse for personal failures

By Eamon Morris

I don't believe in astrology. I don't believe in a lot of things, actually, but astrology definitely ranks in my top ten. I don't like the concept of my life being controlled by anything, and I definitely don't like the concept of my life being controlled by the positioning of celestial bodies light-years away.

There is no scientific evidence of any kind that supports astrology. This isn't to say astrology can't be helpful. Astrology may provide a placebo effect that provides direction and comfort in difficult times. During periods of high stress, people are most likely to turn to religion or spirituality — which includes astrology.

I don't have a problem with people who believe in astrology. But astrology can be detrimental when it's used as an excuse for failure or as a concrete indicator of future relationships.

One of the common ice-breakers is the classic "what's your sign" question. Replying gives people a chance to find things in common and to bond over a shared label. This in itself isn't harmful. However, it becomes an issue when the answer closes people off to pursuing relationships with the respondent.

For example, my birthday is Sept. 1, which means I fall under the label of "Virgo." If I believed in astrology, it would be wise for me to avoid relationships with people who are Geminis, as we are, apparently, of very different temperaments.

Figure 1.1: A document collection

Being a generative model, LDA assumes that each word you see in a document originates from a topic with a probability determined by the topic structure of the corpus. Note that the "topic structure" - the distribution of topics across the corpus - is assumed to already exist. Below is an illustration of this process with words color-coded to their presumed latent topics.

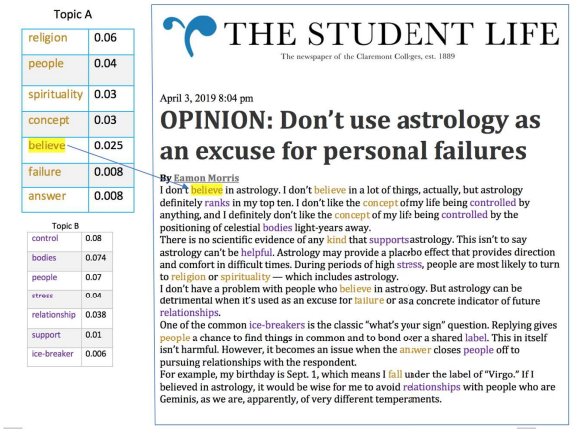

Figure 1.2: An illustration of document generation by topic

The goal of LDA is to *infer* the topic structure of a corpus given just the words in each document. Loosely speaking, this entails "reverse engineering" the topic structure from the corpus.

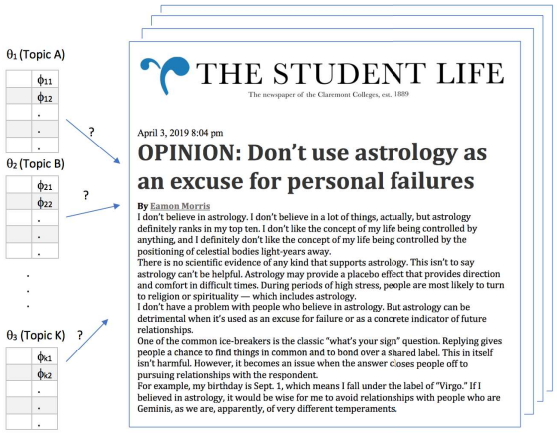

Figure 1.3: Inferring unknown topics from a document collection

Now, you may be wondering at this point: how does one reverse engineer a "ghost" feature that did not exist in the first place? The answer lies in Bayes rule. More detail will be discussed in the next section, but the main idea relies on a back and forth inferential process of making a hypothesis and testing it with data, then making a new hypothesis based on the previous result and testing that with data... and so on. In the context of LDA, this amounts to making an initial hypothesis about the topic structure of a corpus and then repeatedly testing and updating that hypothesis until some probabilistic stability is reached. The next section briefly discusses LDA's conceptual predecessors, LSA and pLSA, to foreground the contribution of LDA within the field.

1.3 Other Document Classification Schemes: LSA and pLSA

Latent Semantic Analysis (LSA) is another document classification technique used for information retrieval. Like LDA, LSA represents large information in smaller dimensions with no foreknowledge about the structure (or content) of documents. The input into LSA is a collection of (textual) documents with the goal of automatically grouping words and documents based on semantic similarity. LSA makes two main assumptions about the semantic environment of documents: like LDA, LSA assumes that the semantic dependencies between words are latent in a document and not explicit [7]; but while LDA constructs "semantic similarity" from probability distributions, LSA assumes that the meaning of a word is a linear combination of the words surrounding it. Note that with LDA and LSA, the grammatical structure of a phrase or sentence is not considered, only the main words comprising them.

To illustrate this linear dependency, consider the three-dimensional input data $[(orange), (apple), (table)]$. Under LSA, the input could be transformed into the two-dimensional output $[(0.3 * orange + 1.2 * apple), (table)]$, the semantic similarity of "apple" and "orange" uncovered from a larger set of text. But interpreting the outputs of LSA is not always intuitive. An input of $[(orange), (apple), (cyan), (rice)]$ could yield the output $[(1.6 * orange + 0.4 * apple), (2.1 * cyan + 0.8 * rice)]$. One is left wondering what "cyan" has to do with "rice."

Instead of dealing directly with these lower dimensional representations of

the original input data, the outputs of LSA are used to determine correlations between word and document vectors using cosine similarity [7]. Quantifying similarity allows documents to be systematically classified using clustering algorithms like K-Nearest Neighbors.

Even though this modeling technique can be used to classify a corpus into clusters of similar documents, LSA does not provide a helpful heuristic about the content of the document clusters as illustrated above with "cyan rice." Another limitation is LSA's inability to capture the different meanings that a word can have. In the previous example, the dual semantics of "orange" was not identified. Since words are not vectorized in LDA but tokenized, every occurrence of the same word is treated distinctly.

Instead of the linear algebraic framework of LSA, probabilistic LSA (pLSA) uses a probabilistic framework to process documents [5]. This allows more flexibility in how words can be related to each other and how documents can be grouped. For instance, a word like "orange" can be related to "apple" with a certain probability and to "cyan" with another probability depending on the distribution of concepts in a given document. Both pLSA and LDA reduce the dimensionality of a corpus by inferring lower dimensional topics from probability distributions. pLSA is based on the Bayesian hierarchical model briefly discussed earlier. However, the two models differ in their generalizability. To illustrate this point, let's resume with the example of estimating student GPA's across the Claremont Consortium.

In the hierarchical model of the student body, each of the 5 colleges in the consortium are given some weight corresponding to the size of a college's student body. Perhaps some parameter indicating college "difficulty" may be added to the model. In the pLSA model, the weights given to each college are fixed. But in the LDA model, these weights are stochastic and are dependent on a *Dirichlet* probability distribution. What difference does this make? Let's say you wanted to estimate the average GPA of students at UCLA using data from the Claremont Consortium as a reference. The weights of the 5 liberal arts colleges used in the pLSA model remain the same when estimating the UCLA statistic. In LDA, however, these weights can be adjusted to emphasize (or understate) the effects of a large (or small) student body. Essentially, the LDA model is more flexible in uncovering topics from a document collection, making inference on a new document more robust than with the pLSA model.

Interlude: Is Topic Modeling Simply a Glorified Word Cloud?

Word clouds (or tag clouds) in their most basic implementation are visual representations of textual data that output different sizes of words as a function of their relative frequencies in a document. Below is a dummy example of a word cloud for *The Student Life* article shown earlier.

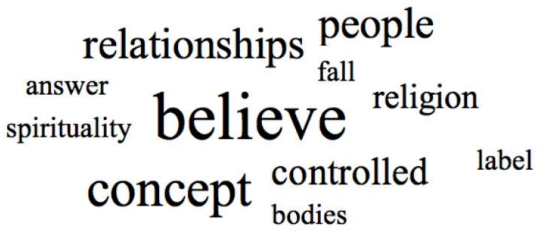

Figure 1.4: A word cloud for a *TSL* article

How much more *informative* information does the topic model output in figure 1.2 provide than the word cloud above? For one, a topic model categorizes its output into discrete topics. The equivalent in a word cloud is thinking of a single word as its own topic - not a very useful semantic source. Moreover, the "topics" of a word cloud are not clearly transferable to new, unseen documents. Like any other mathematical model used to approximate real life phenomena, using a word cloud over a topic model depends on the target application. For larger text collections and for inference on new documents, topic models can be highly effective tools for retrieving information.