Applications of Topic Models

Jordan Boyd-Graber
Department of Computer Science, UMIACS, Language Science
University of Maryland
jbg@umiacs.umd.edu

Yuening Hu
Google, Inc.
ynhu@google.com

David Mimno
Information Science
Cornell University
mimno@cornell.edu

Full text available at: http://dx.doi.org/10.1561/1500000030
Editorial Scope

Topics

Foundations and Trends® in Information Retrieval publishes survey and tutorial articles in the following topics:

- Applications of IR
- Architectures for IR
- Collaborative filtering and recommender systems
- Cross-lingual and multilingual IR
- Distributed IR and federated search
- Evaluation issues and test collections for IR
- Formal models and language models for IR
- IR on mobile platforms
- Indexing and retrieval of structured documents
- Information categorization and clustering
- Information extraction
- Information filtering and routing
- Metasearch, rank aggregation, and data fusion
- Natural language processing for IR
- Performance issues for IR systems, including algorithms, data structures, optimization techniques, and scalability
- Question answering
- Summarization of single documents, multiple documents, and corpora
- Text mining
- Topic detection and tracking
- Usability, interactivity, and visualization issues in IR
- User modelling and user studies for IR
- Web search

Information for Librarians

Foundations and Trends® in Information Retrieval, 2017, Volume 11, 5 issues. ISSN paper version 1554-0669. ISSN online version 1554-0677. Also available as a combined paper and online subscription.

Full text available at: http://dx.doi.org/10.1561/1500000030
Applications of Topic Models

Jordan Boyd-Graber
Department of Computer Science, UMIACS, Language Science
University of Maryland\(^1\)
jbg@umiacs.umd.edu

Yuening Hu
Google, Inc.\(^2\)
ynhu@google.com

David Mimno
Information Science
Cornell University
mimno@cornell.edu

\(^1\)Work completed while at University of Colorado
\(^2\)Work completed while at Yahoo!
Contents

1 The What and Wherefore of Topic Models 2
  1.1 Tell Me about Your Haystack ............................... 2
  1.2 What is a Topic Model ...................................... 5
  1.3 Foundations ................................................. 6
  1.4 Latent Dirichlet Allocation ................................. 11
  1.5 Inference ...................................................... 13
  1.6 The Rest of this Survey ...................................... 19

2 Ad-hoc Information Retrieval 21
  2.1 Document Language Modeling ............................... 23
  2.2 Topic-based Document Language Models ..................... 26
  2.3 Query Expansion ............................................. 27
  2.4 Beyond Relevance—Search Personalization ................... 34
  2.5 Summary ...................................................... 36

3 Evaluation and Interpretation 38
  3.1 Displaying Topics ........................................... 38
  3.2 Labeling Topics ............................................. 40
  3.3 Displaying Models ........................................... 43
  3.4 Evaluation, Stability, and Repair ............................ 45
  3.5 Summary ...................................................... 47

Full text available at: http://dx.doi.org/10.1561/1500000030
4 Historical Documents

4.1 Newspapers ........................................ 50
4.2 Historical Records .................................. 54
4.3 Scholarly Literature ................................ 56
4.4 Summary ........................................... 58

5 Understanding Scientific Publications .......... 60

5.1 Understanding Fields of Study .................. 62
5.2 How Fields Change ................................ 64
5.3 Innovation ........................................ 66
5.4 Summary ........................................... 68

6 Fiction and Literature ............................. 69

6.1 Topic Models in the Humanities ............... 69
6.2 What is a Document? ............................. 71
6.3 People and Places ................................ 72
6.4 Beyond the Literal ................................ 76
6.5 Comparison to Stylometric Analysis .......... 78
6.6 Operationalizing "Theme" ....................... 78
6.7 Summary ........................................... 79

7 Computational Social Science .................... 81

7.1 Topic Models for Qualitative Analysis ....... 84
7.2 Sentiment Analysis ................................ 84
7.3 Upstream and Downstream Models .......... 86
7.4 Understanding Stance and Polarization ....... 87
7.5 Social Networks and Media .................... 88
7.6 Summary ........................................... 91

8 Multilingual Data and Machine Translation .... 92

8.1 Document-level Alignment from Multilingual Corpora .... 94
8.2 Word-level Alignment from Lexical Data ........ 96
8.3 Alignment from Parallel Corpora and Lexical Information .... 98
8.4 Topic Models and Machine Translation .......... 99
8.5 The Components of Statistical Machine Translation ..... 100
8.6 Topic Models for Phrase-level Translation .......... 102
Abstract

How can a single person understand what’s going on in a collection of millions of documents? This is an increasingly common problem: sifting through an organization’s e-mails, understanding a decade worth of newspapers, or characterizing a scientific field’s research. Topic models are a statistical framework that help users understand large document collections: not just to find individual documents but to understand the general themes present in the collection.

This survey describes the recent academic and industrial applications of topic models with the goal of launching a young researcher capable of building their own applications of topic models. In addition to topic models’ effective application to traditional problems like information retrieval, visualization, statistical inference, multilingual modeling, and linguistic understanding, this survey also reviews topic models’ ability to unlock large text collections for qualitative analysis. We review their successful use by researchers to help understand fiction, non-fiction, scientific publications, and political texts.
Imagine that you are an intrepid reporter with an amazing scoop: you have twenty-four hours of exclusive access three decades of e-mails sent within a corrupt corporation. You know there’s dirt and scandal there, but it has been well-concealed by the corporation’s political friends. How are you going to understand this haystack well enough to explain it to your devoted readers under such a tight deadline?

1.1 Tell Me about Your Haystack

Unlike the vignette above, interacting with large text data sets is often posed as a needle in a haystack problem. The poor user—faced with documents that would take a decade to read—is looking for a single needle: a document (or at most a handful of documents) that matches what the user is looking for: a “smoking gun” e-mail, the document that best represents a concept \cite{Salton1968} or the answer to a question \cite{Hirschman2001}.

These questions are important. The discipline of information retrieval is built upon systematizing, solving, and evaluating this problem. Google’s search service is built on the premise of users typing a few
keywords into a search engine box and seeing quick, consistent search results. However, this is not the only problem that confronts those interacting with large text datasets.

A different, but related problem is understanding large document collections, common in science policy [Talley et al. 2011], journalism, and the humanities [Moretti 2013a]. The haystack has more than one precious needle. At the risk of abusing the metaphor, sometimes you care about the straw. Instead of looking for a smoking gun alerting you some crime that was committed, perhaps you are looking for a sin of omission: did this company never talk about diversity in its workforce? Instead of a single answer to a question, perhaps you are looking for a diversity of responses: what are the different ways that people account for rising income inequality? Instead of looking for one document, perhaps you want to provide population level statistics: what proportion of Twitter users have ever talked about gun violence?

At first, it might seem that answering these questions would require building an extensive ontology or categorization scheme. For every new corpus, you would need to define the buckets that a document could fit into, politely ask some librarians and archivists to put each document into the correct buckets, perhaps automate the process with some supervised machine learning, and then collect summary statistics when you are done.

Obviously, such laborious processes are possible—they have been done for labeling congressional speeches[1] and understanding emotional state [Wilson and Wiebe 2005]—and remain an important part of social science, information science, library science, and machine learning. But these processes are not always possible, fast, or even the optimal outcome if we had infinite resources. First, they require a significant investment of time and resources. Even creating the list of categories is a difficult task and requires careful deliberation and calibration. Even if it were possible, a particular question might not warrant the time or effort: the œuvre of a minor author (only of interest to a few), or the tweets of a day (not relevant tomorrow).

[1] www.congressionalbills.org/
Table 1.1: Five topics from a twenty-five topic model fit on Enron e-mails. Example topics concern financial transactions, natural gas, the California utilities, federal regulation, and planning meetings. We provide the five most probable words from each topic (each topic is a distribution over all words).

<table>
<thead>
<tr>
<th>Topic</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>trading financial trade product price</td>
</tr>
<tr>
<td>6</td>
<td>gas capacity deal pipeline contract</td>
</tr>
<tr>
<td>9</td>
<td>state california davis power utilities</td>
</tr>
<tr>
<td>14</td>
<td>ferc issue order party case</td>
</tr>
<tr>
<td>22</td>
<td>group meeting team process plan</td>
</tr>
</tbody>
</table>

This survey explores the ways that humans and computers make sense of document collections through tools called topic models. Topic models allow us to answer big-picture questions quickly, cheaply, and without human intervention. Once trained, they provide a framework for humans to understand document collections both directly by “reading” models or indirectly by using topics as input variables for further analysis. For readers already comfortable with topic models, feel free to skip this chapter; we will mostly cover the definitions and implementations of topic models.

The intended audience of this book is a reader with some knowledge of document processing (e.g., knows what “tokens” and “documents” are), basic understanding of some probability (e.g., what a distribution is), and interested in many application domains. We discuss the information needs of each application area, and how those specific needs affect models, curation procedures, and interpretations.

By the end of the book (Chapter 9), we hope that readers will be excited enough to attempt to embark on building their own topic models. In this chapter, we go deeper into more of the implementation details. Readers who are already topic model experts will likely not learn much technically, but we hope our coverage of diverse applications will expose a topic modeling expert to models and approaches they had not seen before.
1.2. What is a Topic Model

Yesterday, SDG&E filed a motion for adoption of an electric procurement cost recovery mechanism and for an order shortening time for parties to file comments on the mechanism. The attached email from SDG&E contains the motion, an executive summary, and a detailed summary of their proposals and recommendations governing procurement of the net short energy requirements for SDG&E’s customers. The utility requests a 15-day comment period, which means comments would have to be filed by September 10 (September 8 is a Saturday). Reply comments would be filed 10 days later.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.42</td>
</tr>
<tr>
<td>11</td>
<td>0.05</td>
</tr>
<tr>
<td>8</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Figure 1.1: Example document from the Enron corpus and its association to topics. Although it does not contain the word “California”, it discusses a single California utility’s dissatisfaction with how much it is paying for electricity.

1.2 What is a Topic Model

Returning to our motivating example, consider the e-mails from Enron, the prototypical troubled corporation of the turn of the century. A source has provided you with a trove of emails, and your editor is demanding an article by yesterday. You know that wrongdoing happened, but you do not know who did it or how it was planned and carried out. You have suspicions (e.g., around the California energy spot market), but you are curious about other skeletons in the closet and you are highly motivated to find them.

So you run a topic model on the data. True to its name, a topic model gives you “topics”, each of which is a ranking of all the distinct words in the e-mails by relevance to a topic. Taking the top five most relevant words in each topic results in collections of words that make sense together (Table 1.1). For example, one topic seems to have words relating to finance and trading. Another seems to involve to gas pipelines, their capacity, and deals or contracts relating to those pipelines. This all makes sense: Enron was an energy trading company. Others seem to involve language used in any business, such as meetings and plans.
The first half of a topic model connects topics to a jumbled “bag of words”. When we say that a topic is about \( X \), we are manually assigning a \textit{post hoc} label (more on this in Chapter 3.1). It remains the responsibility of the human consumer of topic models to go further and make sense of these piles of straw (we discuss labeling the topics more in Chapter 3).

Making sense of one of these word piles by itself can be difficult. The second half of a topic model links topics to individual documents. For example, the document in Figure 1.1 is about a California utility’s reaction to the short-term electricity market and exemplifies Topic 9 from Table 1.1. Considering examples of documents that are strongly connected to a topic, along with the words associated with the topic, can give us a more complete representation of the topic. If we get a sense that Topic 9 is of interest, we can explore deeper to find other documents.

### 1.3 Foundations

You might notice that we are using the general term “topic model”. There are many mathematical formulations of topic models and many algorithms that learn the parameters of those models from data. Although we will focus on particular models and algorithms, we choose
1.3. Foundations

our terminology to emphasize that the similarities between formulations, models, and algorithms are often greater than their differences.

Topic modeling began with a linear algebra approach [Deerwester et al., 1990] called latent semantic analysis (LSA): find the best low rank approximation of a document-term matrix (Figure 1.2). While these approaches have seen a resurgence in recent years [Anandkumar et al., 2012, Arora et al., 2013], we focus on probabilistic approaches [Hofmann, 1999a, Papadimitriou et al., 2000, Blei et al., 2003], which are intuitive, work well, and allow for easy extensions (as we see later in many of our later chapters).

The two foundational probabilistic topic models are latent Dirichlet allocation [Blei et al., 2003, LDA] and probabilistic latent semantic analysis [Hofmann, 1999a, pLSA]. We describe the former in significant detail in Chapter 1.4, but we want to take a moment to address some of the historical connection between these two models.

pLSA was historically first and laid the foundation for LDA. pLSA was used extensively in many applications such as information retrieval. However, this survey focuses on LDA because more researchers have not just used LDA—they have also extended it. LDA is not just widely used, but it is also widely modified. Because of these prolific modifications, we focus on the mechanics of LDA, which many researchers have used as the foundations of new models. However, as we explain below (Chapter 1.5.4), the similarities between pLSA and LDA outweigh the differences.

In any technical field it is common for general terms to take on specific, concrete meanings, and this can be a source of confusion. In topic modeling the word “topic” takes on the specific meaning of a probability distribution over words, while still alluding to the more general meaning of a theme or subject of discourse. Because other areas of information retrieval have similarly developed specific meanings for the word “topic”, we distinguish them here. The most common definition is a specific information need, as in the TREC evaluation corpora developed by NIST [Voorhees and Harman, 2005]. TREC topics are generally much more specific than topic model topics, and may relate to particular aspects or perspectives on a subject. An example from
the 2003 TREC Robust Track is “Identify positive accomplishments of the Hubble telescope since it was launched in 1991” [Voorhees, 2003]. Similarly to information retrieval, the related field of topic detection and tracking also has a specific technical definition of “topic” [Allan, 2002]. In TDT, a “topic” is usually closer to an event or an individual story. In contrast, topic models tend to identify more abstract latent factors. For example, a TDT topic might include an earthquake in Haiti, whereas a topic model might represent the same event as a combination of topics such as Haiti, natural disasters, and international aid.

There has been some work on using topic models to detect emerging events by searching for changes in topic probability [AlSumait et al., 2008]. But these methods tend to identify mainly the fact that an event has occurred, without necessarily identifying the specific features of that event. Other work has found that more lexically specific methods than topic models are best for identifying memes and viral phrases [Leskovec et al., 2009].

1.3.1 Probabilistic Building Blocks

In probabilistic models we want to find values for unobserved model variables that do a good job of explaining the observed data. The first step in inference is to turn this process around, and assert a way to generate data given model variables. Probabilistic models thus begin with a generative story: a recipe listing a sequence of random events that creates the dataset we are trying to explain. Figure 1.3 lists some of the key players in these stories, how they are parameterized and what samples drawn from these distributions look like. We will briefly discuss them, as we will use them to build a wide variety of topic models later.

**Gaussian** If you know any probability distribution already, it is (probably) the Gaussian. This distribution does not have a role in the most basic topic models that we will discuss here, but it will later (e.g., Chapter 7). We include it because it is a useful point of comparison against the other distributions we are using (since it is perhaps the easiest to understand and best known). A Gaussian is a distribution over all real numbers (e.g., 0.0, 0.5, −4.2, π, ...). You can ask it to spit
1.3. Foundations

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Density</th>
<th>Example Parameters</th>
<th>Example Draws</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>( \frac{1}{\sqrt{2\pi\sigma_2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} )</td>
<td>( \mu = 2, \sigma^2 = 1.1 )</td>
<td>( x = 2.21 )</td>
</tr>
<tr>
<td>Discrete</td>
<td>( \prod_i \phi_i^{</td>
<td>w=i</td>
<td>} )</td>
</tr>
<tr>
<td>Dirichlet</td>
<td>( \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} )</td>
<td>( \alpha = \begin{bmatrix} 1.1 \ 0.1 \end{bmatrix} )</td>
<td>( \theta = \begin{bmatrix} 0.8 \ 0.15 \end{bmatrix} )</td>
</tr>
</tbody>
</table>

Figure 1.3: Examples of probability distributions used in the generative stories of topic models. In the case of the discrete draw, \( w = 2 \) denotes that the second element (the one with probability 0.6) was drawn.

Out a number, and it will give you some real number between negative infinity and positive infinity. But not all numbers have equal probability. Gaussian distributions are parameterized by a mean \( \mu \) and variance \( \sigma^2 \). Most samples from the distribution will be near the mean \( \mu \); how close is determined by the variance: higher variances will cause the samples to be more spread out.

Discrete While Gaussian distributions are over a continuous space, documents are combinations of discrete symbols, usually word tokens. Thus, we need a distribution over discrete sets.

A useful metaphor for thinking about discrete distributions is a weighted die. The number of faces on the die is its dimension, and each face is associated with a distinct outcome. Each face has its own probability of how likely that outcome is; these probabilities are the parameters of a discrete distribution (Figure 1.3).

Topic models are described by discrete distributions (sometimes called multinomial distributions) that describe the connection between words and topics (the first half) and topics and documents (the second half). A distribution over words is called a topic distribution; each of

\[ \text{An emerging trend in natural language processing research is to view words as embedded in a continuous space. We discuss these “representation learning” approaches and their connection to topic modeling in Chapter 10 but even then models are still defined over a discrete set of words.} \]
the topics gives higher weights to some words more than others (e.g., in Topic 9 from the Enron corpus, “state” and “california” have higher probability than other words). Each document also has an “allocation” for each topic: documents are about a small handful of topics, and most documents have very low weights for most of the possible topics.

**Dirichlet** Although discrete distributions are the star players in topic models, they are not the end of the story. We often begin with Dirichlet distributions. Just as Gaussians produce real numbers and discrete distributions produce symbols from a finite set, Dirichlet distributions produce probability vectors that can be used as the parameters of discrete distributions. Like the Gaussian distribution, they have parameters analogous to a mean and variance. The mean is called the “base measure” \( \tau \) and is the expected value of the Dirichlet distribution: the values you would get if you averaged many draws from the Dirichlet. The concentration parameter \( \alpha_0 \) controls how far away individual draws...
1.4. Latent Dirichlet Allocation

are from the base measure. We often combine these parameters into a single value for each dimension: \( \alpha_k = \alpha_0 \tau_k \).

If \( \alpha_0 \) is very large, then the draws from a Dirichlet will be very close to \( \tau \) (Figure 1.4, left). If \( \alpha_0 \) is small, however, the discrete distributions become sparse (Figure 1.4, right). A sparse distribution is a distribution where only a few values have high probability and all other values are small.

Because topic models are meant to reflect the properties of real documents, modeling sparsity is important. When a person sits down to write a document, they only write about a handful of the topics that they could potentially use. They do not write about every possible topic, and the sparsity of Dirichlet distributions is the probabilistic tool that encodes this intuition.

There are several important special cases of the Dirichlet distribution. If the base measure \( \tau \) is the same for every dimension, we call the resulting distribution symmetric. This case is appropriate when we do not expect any one element to be, on average, more likely than any other element across all samples from the distribution. In the symmetric case the distribution has only one parameter, the concentration \( \alpha_0 \). If the base measure is uniform and the concentration parameter \( \alpha_0 \) is equal to the number of dimensions \( K \) (or, equivalently, \( \alpha_k = 1.0 \) for all \( k \)), the distribution is uniform, placing equal probability on all \( K \)-dimensional probability distributions.

1.4 Latent Dirichlet Allocation

We now have all the tools we need to tell the complete story of the most popular topic model: latent Dirichlet allocation [Blei et al., 2003, LDA]. Latent Dirichlet allocation\(^3\) posits a “generative process” about how the data came to be. We assemble the probabilistic pieces to tell this

\(^3\)The name LDA is a play on LSA, its non-probabilistic forerunner (latent semantic analysis). Latent because we use probabilistic inference to infer missing probabilistic pieces of the generative story. Dirichlet because of the Dirichlet parameters encoding sparsity. Allocation because the Dirichlet distribution encodes the prior for each document’s allocation over topics.
story about generating topics and how those topics are used to create diverse documents.

**Generating Topics**  The first part of the story is to create the topics. The user specifies that there are $K$ distinct topics. Each of the $K$ topics is drawn from a Dirichlet distribution with a uniform base distribution and concentration parameter $\lambda$: $\phi_k \sim \text{Dir}(\lambda u)$. The discrete distribution $\phi_k$ has a weight for every word in the vocabulary.

However, when we summarize topics (as in Figure 1.1), we typically only use the top (most probable) words of a topic. The lower probability words are less relevant to the topic and thus are not shown.

**Document Allocations**  Document allocations are distributions over topics for each document. This encodes what a document is about; the sparsity of the Dirichlet distribution’s concentration parameter $\alpha_0$ ensures that the document will only be about a few topics. Each document has a discrete distribution over topic: $\theta_d \sim \text{Dir}(\alpha u)$.

**Words in Context**  Now that we know what each document is about, we create the words that appear in the document. We assume that there are $N_d$ words in document $d$. For each word $n$ in the document $d$, we first choose a topic assignment $z_{d,n} \sim \text{Discrete}(\theta_d)$. This is one of the $K$ topics that tells us which topic the word token is from, but not what the word is.

To select which word we will see in the document, we draw from a discrete distribution again. Given a word token’s topic assignment $z_{d,n}$, we draw from that topic to select the word: $w_{d,n} \sim \phi_{z_{d,n}}$. The topic assignment tells you what the word is about, and then this selects which distribution over words we use to generate the word.

For example, consider the document in Figure 1.1 To generate it, we choose a distribution over all of the topics. This is $\theta$. For this document, the distribution favors Topic 9 about California. The value for this topic

---

4We can model this in the generative story as well, e.g., with a Poisson distribution. However, we often do not care about document lengths—only what the document is about—so we can usually ignore this part of the story.
1.5. Inference

is higher than any other topic. For each word in the document, the generative process chooses a topic assignment \( z_n \). For this document, any topic is theoretically possible, but we expect that most of those will be Topic 9.

Then, for each token in the document, we need to choose which word type will appear. This comes from Topic 9’s distribution over words (multiple topics have word distributions shown in Figure 1.1). Each is a discrete draw from the topic’s word distribution, which makes words like “California”, “state”, and “Sacramento” more likely.

It goes without saying that the generative story is a fiction [Box and Draper, 1987]. Nobody is sitting down with dice to decide what to type in on their keyboard. We use this story because it is useful. This fanciful story about randomly choosing a topic for each word can help us because if we assume this generative process, we can work backwards to find the topics that explain how a document collection was created: every word, every document, gets associated with these underlying topics.

This simple model helps us order our document collection: by assuming this story, we can discover topics (which certainly do not exist) so we can understand the common themes that people use to write documents. As we will see in later chapters, slight tweaks of this generative story allow us to uncover more complicated structures: how authors prefer specific topics, how topics change, or how topics can be used across languages.

1.5 Inference

Given a generative model and some data, the process of uncovering the hidden pieces of the probabilistic generative story is called inference. More concretely, it is a recipe for generating algorithms to go from data to topics that explain a dataset.

There are many flavors of algorithms for posterior inference: message passing [Zeng et al., 2013], variational inference [Blei et al., 2003], gradient descent [Hoffman et al., 2010], and Gibbs sampling [Griffiths and Steyvers, 2004]. All of these algorithms have their advocates and
The What and Wherefore of Topic Models

reasons you should use them. In this survey, we focus on Gibbs sampling, which is simple, intuitive, and—with some clever tricks specific to topic models—fast [Yao et al., 2009]. (We discuss variational inference in Chapter 9.)

We present the results of Gibbs sampling without derivation, which—along with the history of its origin in statistical physics—are well described elsewhere. We use a variety of Gibbs sampling called collapsed Gibbs sampling, which allows inference to side-step some of the pieces of the generative story: instead of explicitly representing the parameters of a discrete distribution, distinct from any observations drawn from that distribution, we represent the distribution solely through those observations. We can then recreate the topic and document distributions through simple formulas.

1.5.1 Random Variables

Topic Assignments Since every individual token is assumed to be generated from a single topic, we can consider the topic assignment of a token as a variable. For example, an instance of the word “compilation” might be in a Computer topic in one document and in an Arts topic in another document. Because each token has its own topic assignment, the same word might be assigned to different topics in the same document.

To estimate global properties of the topic model we use aggregate statistics derived from token-level topic assignments.

Document Allocation The document allocation is a distribution over the topics for each document; in other words, it says how popular each topic is in a document. If we count up how often a document uses a topic, this gives us its popularity. We define \( N_{d,i} \) as the number of times document \( d \) uses topic \( i \). This is larger for more popular topics; however, it is not a probability because it is larger than one. We make it a probability by dividing by the number of words in a document

\[
\frac{N_{d,i}}{\sum_k N_{d,k}},
\]  

(1.1)

\(^5\)We recommend Resnik and Hardisty [2009] for additional information on derivation.
but this is problematic because it can sometimes give us zero and ignores the influence of the Dirichlet distribution; a better estimate is

$$\theta_{d,i} \approx \frac{N_{d,i} + \alpha_i}{\sum_k N_{d,k} + \alpha_k}. \quad (1.2)$$

This must never become zero because we do not want it to rule out the possibility that a topic is used in a particular document (hence, each $\alpha_i$ must be non-zero). This helps the sampler explore more of the possible combinations.

**Topics** Each topic is a distribution over words. To understand what a topic is about, we look at the profile of all of the tokens that have been assigned to that topic. We estimate the probability of a word in a topic as

$$\phi_{i,v} \approx \frac{V_{i,v} + \beta_v}{\sum_w V_{i,w} + \beta_w}, \quad (1.3)$$

where $\beta$ is the Dirichlet parameter for the topic distribution.

### 1.5.2 Algorithm

The collapsed Gibbs sampling algorithm for learning a topic model is only based on the topic assignments, but we will use our estimates for the topics $\phi_k$ and the documents $\theta_d$ discussed above. We begin by setting topic assignments randomly: if we have $K$ topics, each word has equal chance to be associated with any of the topics. These topics will be quite bad, looking like noisy copies of the overall corpus distribution. But we will improve them one word at a time.

The algorithm proceeds by sweeping over all word tokens in turn over and over. At each iteration we change the topic assignments for each word in a way that reflects the underlying probabilistic model of the data. On average, each pass over the data makes the topics slightly better until the model reaches a steady state. There is no easy way to tell when such a steady state has been reached, but eventually the topics will “converge” to reasonable themes and you can consider yourself done.

---

6To be technical, Equation 1.1 is a maximum likelihood estimate and Equation 1.2 is the maximum \textit{a posteriori}, which incorporates the influence of both the prior and the data.
The equation for the probability of assigning a word to a particular topic combines information about words and about documents:

$$p(z_{d,n} = i | \ldots) = \theta_d \phi_i = \left( \frac{N_{d,i} + \alpha_i}{\sum_k N_{d,k} + \alpha_k} \right) \left( \frac{V_{i,w_{d,n}} + \beta_v}{\sum_w V_{i,w} + \beta_w} \right).$$ \hfill (1.4)

Computing this value for each topic will result in a probability distribution over the topic assignment for this word token, given all the other topic assignments. The next step is to randomly choose one of those indices with probability proportional to the vector value. You now assign that word to the topic, update $N_{d,\cdot}$ and $V_{\cdot,w_{d,n}}$, and move on to the next word and repeat. The two terms provide two “pressures”, for global and local coherence. Sparsity in the topic-word distributions encourages tokens of the same word type to be assigned to a small number of topics, regardless of where they occur. Sparsity in the document-topic distributions encourages tokens in the same document to be assigned to a small number of topics, regardless of what type they are. For example, knowing that a word is “compilation” narrows down the number of potential topics considerably, but leaves ambiguity: is it program compilation or a music compilation? Knowing that the word occurs in a document with many other words in the Arts topic resolves this ambiguity, leaving the Arts topic as the most probable assignment.

At the very end of the algorithm, we can use the estimates of each topic (Equation 1.3) to summarize the main themes of the corpus and the estimates of each document’s topic distribution (Equation 1.2) to start exploring the collection automatically (Chapter 2) or with a human in the loop (Chapter 3).

The algorithm that we have sketched here is the foundation of many of the more advanced models that we will discuss later in the survey. While we will not describe the algorithms in detail, we will occasionally reference this sketch to highlight challenges or difficulties in implementing topic models.

\footnote{To be theoretically correct, it is important not to include the count associated with the token you are sampling in these counts, which becomes more clear if the probability is written as $p(z_{d,n} = j | z_{d,1} \ldots z_{d,n-1}, z_{d,n+1} \ldots z_{d,N_d}, w_{d,n})$ to show the dependence on the topic assignments of all other tokens but not this token.}
1.5. Inference

1.5.3 Plate Diagrams

Plate diagrams provide a shorthand for quickly explaining which random variables are associated with each other. If you look up many of the references used in this survey, you will likely see plate diagrams (we also use a plate diagram later in Figure 2.1b).

Let’s begin with a plate diagram for LDA (Figure 1.5). You can compare these to the generative story in Chapter 1.4. All of the random variables are there, each in its own circle. The lines between random variables tell more of the story. You can see that if a random variable is conditioned on another, there is a line going from the variable that is conditioned on to the variable that is conditionally dependent. For example, a word depends on the token assignment $z_{d,n}$ and a topic $\phi_k$, so we draw lines from both.

You can think about the rectangular boxes as repetition. The letter in the bottom right of the box shows how often what is inside the box is replicated. There is a box for each document (there are $M$ in total) and each token (the box of words is inside the box for documents).

When a variable is shaded, this means that it is observed. These are the data we start with. The unshaded variables must either be inferred (e.g., topics $\phi$) or are hyperparameters that must be set or inferred (e.g., Dirichlet parameter $\alpha$).

Figure 1.5: Plate diagram for LDA. Nodes show random variables, lines show (possible) probabilistic dependence, rectangles show repetition, and shading shows observation.
Plate diagrams allow a reader to quickly see a “family resemblance” between related models, and once someone has become fully immersed in topic models, it is often possible to at a glance understand a model from its plate diagram. However, plate diagrams are imperfect; they lack some of the key information you need to understand the model. For instance, the exact probabilistic relationship between variables is underspecified.

1.5.4 What is so Great about Dirichlet?

Now that we have described what LDA is, we can return to its history. What is the innovation that separates LDA from PLSA, its predecessor? Naïvely, the difference is changing an “s” to a “d” (i.e., changing PLSA to LDA). The deeper story is about as consequential.

Instead of having a Dirichlet prior over $\theta$, PLSA assumes that $\theta$ is a discrete parameter. In practice, this means that documents are not encouraged to focus on a limited number of topics and often “spread out” to have small weights for many different topics. In theory, this means that there is not as sound a generative story for how a document came to be: you cannot run the generative process forward from scratch if you must have $\theta$ as a parameter to start with.

These differences are relatively minor. LDA has slightly easier inference—particularly when it comes to tweaking the model—which has caused it to become the more popular of the two models. Thus, we will focus on comparing models to LDA. This is not to diminish from PLSA and its unquestionable place in the literature, but it helps us present a more unified narrative for our reader.

1.5.5 Implementations

Hopefully the previous algorithm sketch has convinced you that implementing topic models is not a Herculean task; most skilled programmers can complete a reasonable implementation of topic models in less than a day. However, we would suggest not trying to implement basic LDA if you just want the output of a topic model, many solid implementations can help users get to useful results more quickly, particularly as topic models often require extensive preprocessing.
1.6 The Rest of this Survey

Mallet is fast and is a widely used implementation in Java \cite{McCallum2002}. This is where you should probably start, in our biased opinion. It runs in Java, uses highly-optimized Gibbs sampling implementations, and can work from a variety of text inputs. It is well documented, mature, and runs well on a multi-core machine, allowing it to process up to millions of documents. Variational inference is the other major option \cite{Blei2003,Langford2007}, but often requires a little more effort for new users to get a first result.

However, not all users are comfortable with Java; many implementations are available on other platforms and in many programming languages. Many of these implementations are well-built, but check whether they have all of the features of mature implementations like Mallet so that you know what (if anything) you’re missing.

However, if your corpus is truly large, consider techniques that can be parallelized over large computer clusters. These techniques can be based on variational inference \cite{Narayanamurthy2011,Zhai2012} or on sampling \cite{Newman2008}.

While these implementations allow you to run specific topic models, other frameworks allow you to specify arbitrary generative models. This enables quick prototyping of topic models and integrating topic models with other probabilistic frameworks like regression or collaborative filtering. Examples of these general frameworks include Stan \cite{StanDevelopmentTeam2014}, Theano \cite{TheanoDevelopmentTeam2016}, and Infer.net \cite{Minka2014}.

If you cannot find the specific model that you want among these existing software packages, the flexibility and simplicity of topic models and inference makes it relatively simple to adapt topic models to model specific phenomena (as we describe in following chapters).

1.6 The Rest of this Survey

In each of the following chapters, we focus on an application of topic models, gradually increasing the complexity of the underlying models.

\footnote{So many that change so quickly; thus, we are reluctant endorse specific ones here.}
The chapters do occasionally refer to each other, but a reader should be able to read each of the chapters independently.

The next chapter returns to the distinction between high level overviews and finding a needle in a haystack. We show how a high level overview can help users and algorithms find documents of interest. We show how a high level overview can help algorithms (Chapter 2) and users (Chapter 3) find documents of interest.

These tools help enable new applications of topic models: how understanding newspapers (Chapter 4) reveals the march of history, how the corpus of writers of fiction (Chapter 6) illuminates societal norms, how the writings of science reveal innovation (Chapter 5), or how politicians’ speeches (Chapter 7) reveal schisms in political organizations.

Finally, the survey closes with thoughts about how interested researchers can start building their own topic models (Chapter 9) and how topic models may change in the future (Chapter 10).
References


References


References


References


Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. Personalized search result diversification via structured learning. In Knowledge Discovery and Data Mining, 2014.


References


Franco Moretti. Operationalizing, or the function of measurement in literary theory. *New Left Review*, 84, Nov/Dec 2013b.


References


Daniel Ramage, Christopher D. Manning, and Daniel A. Mcfarland. Which universities lead and lag? toward university rankings based on scholarly output. In NIPS Workshop on Computational Social Science and the Wisdom of the Crowds, 2010b.


References


Yizhou Sun, Jiawei Han, Jing Gao, and Yintao Yu. iTopicModel: Information network-integrated topic modeling. In International Conference on Data Mining, 2009.


References


Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas Huang. Geographical topic discovery and comparison. In Proceedings of World Wide Web Conference, 2011.


References


<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-means clustering</td>
<td>51</td>
</tr>
<tr>
<td>JSTOR</td>
<td>56</td>
</tr>
<tr>
<td>American Revolution</td>
<td>51</td>
</tr>
<tr>
<td>approval rating</td>
<td>82</td>
</tr>
<tr>
<td>aspect model</td>
<td>87</td>
</tr>
<tr>
<td>Bilingual topical admixture</td>
<td>111</td>
</tr>
<tr>
<td>Chinese language</td>
<td>54, 96</td>
</tr>
<tr>
<td>close reading</td>
<td>69</td>
</tr>
<tr>
<td>comparable corpora</td>
<td>95</td>
</tr>
<tr>
<td>continuous time dynamic topic model</td>
<td>65</td>
</tr>
<tr>
<td>copycat model</td>
<td>66</td>
</tr>
<tr>
<td>decoding (machine translation)</td>
<td>100</td>
</tr>
<tr>
<td>deep learning</td>
<td>117, 128</td>
</tr>
<tr>
<td>diary</td>
<td>56</td>
</tr>
<tr>
<td>Dickens</td>
<td>73</td>
</tr>
<tr>
<td>dictionary</td>
<td>96, 98</td>
</tr>
<tr>
<td>Dirichlet distribution</td>
<td>10</td>
</tr>
<tr>
<td>alternatives</td>
<td>65</td>
</tr>
<tr>
<td>conjugacy</td>
<td>119</td>
</tr>
<tr>
<td>parameter</td>
<td>11</td>
</tr>
<tr>
<td>role in LDA</td>
<td>18</td>
</tr>
<tr>
<td>smoothing in language models</td>
<td>25</td>
</tr>
<tr>
<td>tree</td>
<td>97</td>
</tr>
<tr>
<td>disambiguation</td>
<td>105</td>
</tr>
<tr>
<td>discrete distribution</td>
<td>9, 65, 116</td>
</tr>
<tr>
<td>distant reading</td>
<td>70</td>
</tr>
<tr>
<td>distributed representation</td>
<td>128</td>
</tr>
<tr>
<td>document language modeling</td>
<td>23</td>
</tr>
<tr>
<td>dynamic influence model</td>
<td>67</td>
</tr>
<tr>
<td>dynamic topic model</td>
<td>64</td>
</tr>
<tr>
<td>expectation maximization</td>
<td>118</td>
</tr>
<tr>
<td>Facebook</td>
<td>84</td>
</tr>
<tr>
<td>fully-factorized distribution</td>
<td>119</td>
</tr>
<tr>
<td>Gaussian distribution</td>
<td>8, 64, 86, 116</td>
</tr>
<tr>
<td>German language</td>
<td>54, 56, 95</td>
</tr>
<tr>
<td>Gibbs sampling</td>
<td>14, 122</td>
</tr>
</tbody>
</table>
Index

for LDA, 15
grounded theory, 84
hidden topic Markov model, 109
Infer.Net, 118
influenza, 82
interactive TOpic Model and MEtadata (TOME), 44
interactivity
  information retrieval, 33
topic models, 46
interpretability, 45, 83, 117, 125
Japanese language, 54
labeled LDA, 42
language model
  machine translation, 106
  query expansion, 28
latent Dirichlet allocation, 7, 11
  document language model, 26
  generative process, 13
  implementations, 18
latent semantic analysis, 7, 26, 51, 108
likelihood evaluation, 45, 125
link latent Dirichlet allocation, 90
Mallet, 18
Martha Ballard, 55
mixed-membership block model, 89
Modern Language Association, 58
names of fictional characters, 72
National Institutes of Health, 62
nested Dirichlet process, 88
newspaper, 50
novels, 70
online latent Dirichlet allocation, 127
PageRank, 41
parallel corpus, 98
phrase (machine translation), 102
plate diagram, 17
  personalized retrieval, 35
poetry, 77
pollution in China, 82
polylingual latent Dirichlet allocation, 94
polylingual tree-based Latent Dirichlet allocation, 98
posterior predictive checks, 113
prediction vs. interpretation, 82
probabilistic latent semantic analysis, 7, 26, 51
query expansion, 27, 95
relevance model, 29
reordering (machine translation), 110
search personalization, 34
sentiment analysis, 84
smoothing, 24
spectral learning, 128
Stan, 118
statistical machine translation, 92
domain adaptation, 99
stochastic block model, 89
stylometry, 78

Full text available at: http://dx.doi.org/10.1561/1500000030
supervised latent Dirichlet allocation, 86 124
survey, 69 81
synthetic data, 123

Termite, 44
Theano, 118
topic coherence, 46
topic detection and tracking, 7
topic labeling, 40
Topic Model Visualization Engine, 43
topical guide, 43
Torch, 118
tree-based latent Dirichlet allocation, 96
Twitter, 91

upstream vs. downstream models, 86

variational inference, 118

Wikipedia, 41 93
WordNet, 96